Providing Sound Foundations for Cryptography

On the work of Shafi Goldwasser and Silvio Micali

Oded Goldreich
Editor

ASSOCIATION FOR COMPUTING MACHINERY

A.M. TURING AWARD

Providing Sound Foundations for Cryptography

On the work of Shafi Goldwasser and Silvio Micali

Oded Goldreich, editor

Weizmann Institute of Science

ACM Books #30

Providing Sound Foundations for Cryptography: On the work of Shafi Goldwasser and Silvio Micali

Oded Goldreich, editor

books.acm.org
http://books.acm.org

ISBN: 978-1-4503-7266-4 hardcover
ISBN: 978-1-4503-7267-1 paperback
ISBN: 978-1-4503-7268-8 EPUB
ISBN: 978-1-4503-7269-5 eBook

Series ISSN: 2374-6769 print 2374-6777 electronic

DOIs:

10.1145/3335741 Book
10.1145/3335741.3335742 Preface
10.1145/3335741.3335743 Chapter 1
10.1145/3335741.3335744 Chapter 2
10.1145/3335741.3335745 Chapter 3
10.1145/3335741.3335746 Chapter 4
10.1145/3335741.3335747 Chapter 5
10.1145/3335741.3335748 Chapter 6
10.1145/3335741.3335749 Chapter 7
10.1145/3335741.3335750 Chapter 8
10.1145/3335741.3335751 Chapter 9
10.1145/3335741.3335752 Chapter 10
10.1145/3335741.3335753 Chapter 11
10.1145/3335741.3335754 Chapter 12
10.1145/3335741.3335755 Chapter 13
10.1145/3335741.3335756 Chapter 14
10.1145/3335741.3335757 Chapter 15
10.1145/3335741.3335758 Chapter 16
10.1145/3335741.3335759 Chapter 17
10.1145/3335741.3335760 Chapter 18
10.1145/3335741.3335761 Chapter 19
10.1145/3335741.3335762 Chapter 20
10.1145/3335741.3335763 Chapter 21
10.1145/3335741.3335764 Chapter 22
10.1145/3335741.3335765 Chapter 23
10.1145/3335741.3335766 Chapter 24
10.1145/3335741.3335767 Chapter 25
10.1145/3335741.3335768 Chapter 26
10.1145/3335741.3335769 Index/Bios

A publication in the ACM Books series, #30
Editor in Chief: M. Tamer Özsu, *University of Waterloo*

This book was typeset in Arnhem Pro 10/14 and Flama using ZzTEX.

First Edition

10 9 8 7 6 5 4 3 2 1

Contents

Preface

There are no privileges without duties.

—Advocate Klara Goldreich-Ingwer (1912–2004)

Cryptography is concerned with the construction of schemes that withstand any abuse: A cryptographic scheme is constructed so as to maintain a desired functionality, even under malicious attempts aimed at making it deviate from its prescribed behavior. The design of cryptographic systems must be based on firm foundations, whereas ad hoc approaches and heuristics are a very dangerous way to go. These foundations were developed mostly in the 1980s, in works that are all co-authored by Shafi Goldwasser and/or Silvio Micali. These works have transformed cryptography from an engineering discipline, lacking sound theoretical foundations, into a scientific field possessing a well-founded theory, which influences practice as well as contributes to other areas of theoretical computer science. The current book celebrates these works, which were the basis for bestowing the 2012 Turing Award upon Shafi Goldwasser and Silvio Micali.

◇ ◇ ◇

Cryptography as we know it today is based entirely on concepts, definitions, techniques, and feasibility results put forward and developed in the works of Goldwasser and/or Micali. A significant portion of this book reproduces some of these works, whose contents is briefly outlined next.

"Probabilistic Encryption" (Chapter 7). The pivot of the aforementioned body of work is the pioneering work "Probabilistic Encryption," whose title reflects the realization that a robust notion of secure encryption requires the use of randomization in the process of encrypting each message (and not only in the process of generating cryptographic keys). This work of Goldwasser and Micali defined the mind-set of the

field by establishing conceptual frameworks and demonstrating their usefulness. In particular:

- This work suggested viewing computationally indistinguishable objects as equivalent. This revolutionary suggestion has played a key role in all standard cryptographic definitions and has served as the pivot of the acclaimed theory of pseudorandomness (to be briefly reviewed below).
- This work suggested interpreting security as the ability to emulate an ideal setting. This suggestion, further clarified by Goldwasser and Micali in early versions of "The Knowledge Complexity of Interactive Proof Systems" (briefly reviewed below), has been adopted as the basic approach to defining security in almost all cryptographic settings. This approach, known as the ***simulation paradigm***, *resolves the Gordian knot that has frustrated previous attempts to define security by trying to enumerate all desired properties*. The simulation paradigm bypasses this enumeration by asserting that security means that anything that can be efficiently obtained by an attack on the cryptographic system can be essentially obtained (as efficiently) without attacking the system. Thus, any gain that an attacker claims is actually not due to the use of the cryptographic system.
- This work demonstrated the fruitfulness of the aforementioned paradigm shift by providing robust definitions for the most basic cryptographic primitive (i.e., encryption schemes) and by constructing a secure encryption scheme based on a standard complexity assumption. In addition to demonstrating the viability of the new-at-the-time approach, this paper set the standard for the two-step process to be followed by all subsequent works:
  - First, a robust definition is developed, based on the aforementioned approach.
  - Next, schemes satisfying this definition are proven to exist (and actually explicitly constructed) based on much better understood assumptions.

  For example, once defined, it was not a priori clear whether zero-knowledge proofs exist at all, and thus relating this question to well-known conjectures demonstrated the viability of zero-knowledge.
- This work also introduced important techniques, one being later termed *the hybrid argument*, which found numerous applications in cryptography and

in the theory of pseudorandomness. Notably, this work also heralded worst-case to average-case reductions (also known as random self-reducibility).

"The Knowledge Complexity of Interactive Proof Systems" (Chapter 8). The second most influential work of Goldwasser and Micali is their joint work on zero-knowledge, which after not being understood by most researchers for three years, and being revised several times, appeared in the "formal verification" session of STOC '85 (indicating that it was misunderstood even by the program committee that accepted it for presentation). I can testify to the fact that the lack of understanding has not been due to a poor presentation of the ideas, but rather to their revolutionary nature. (By the way, their earlier work "Probabilistic Encryption" also faced lack of understanding for a couple of years.)

Nowadays, it is well-understood that this work introduced two fascinating and highly influential concepts: the concept of interactive proofs and the concept of zero-knowledge. The concept of interactive proofs had a vast impact on complexity theory, to be briefly reviewed below. The concept of zero-knowledge, on top of being very intriguing (once one stops being confused by it), became a central tool in cryptography and led to fundamental discoveries regarding general secure multiparty computation. Initial indications to the vast potential impact of these concepts were provided by the results and discussions in the conference version of this work (reproduced in Chapter 8).

"How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits" (Chapter 9). This work defined pseudorandom generators as producing a sequence of unpredictable bits. This definition was later shown to be equivalent to being computationally indistinguishable from the uniform distribution over bit-strings of adequate length. The notion of computational indistinguishably used here is the same as the notion introduced in "Probabilistic Encryption," but subsequent works introduced a variety of alternative definitions yielding a host of notions of pseudorandom generators. This work also defined the notion of a hard-core predicate of a one-way function, and established its existence for the modular exponentiation function.

"How to Construct Random Functions" (Chapter 10). This work extended the theory of pseudorandomness to functions, and showed how to construct pseudorandom functions based on any pseudorandom generator. The notion of a pseudorandom

function found numerous applications in cryptography, starting from the construction of message authentication codes and private-key encryption schemes that withstand chosen ciphertext attacks.

"A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks" (Chapter 11). The result proved by this paper was considered impossible or at least "paradoxical" at the time, because it was (falsely) believed that a "constructive proof of unforgeability" (under passive attacks) implies a successful chosen-message attack.

"Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems" (Chapter 12). This work demonstrated the generality and wide applicability of zero-knowledge proofs. In particular, assuming the existence of secure commitment schemes, it showed how to construct zero-knowledge interactive proof systems for any set in NP, yielding a powerful tool for the design of various cryptographic schemes. Loosely speaking, zero-knowledge proofs offer a way for a party to prove that it has behaved according to a predetermined protocol, *without revelaing its own secrets*, and so they can be used to force parties to behave in "honest-but-curious" manner.

"How to Play any Mental Game—A Completeness Theorem for Protocols with Honest Majority" (Chapter 13). This work presented constructions of secure protocols for any multi-party computation problem. In other words, it shows how a trusted party can be emulated by a set of mutually distrustful parties. This result combines the construction of "privacy-preserving" protocols for the "honest-but-curious" model with a method (presented in Chapter 12) of forcing parties to behave in an honest-but-curious manner. The privacy-preserving protocols rely on the existence of a public-key encryption scheme and an Oblivious Transfer protocol, which can both be based on the existence of trapdoor permutations.

"Non-Interactive Zero-Knowledge (NIZK) Proof Systems" (Chapter 14). The model of noninteractive proof systems introduced in this work includes a common random string provided from the outside and available to both the prover and the verifier. The work showed how to provide zero-knowledge (noninteractive) proofs for any NP-assertion. Such NIZKs have been used as a building blocks in many subsequent works (e.g., in constructing public-key encryption schemes that withstand chosen-ciphertext attacks).

"Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation" (Chapter 15). This work obtained general results similar to those of the work presented in Chapter 13, except that it uses no intractability assumptions. Instead,

this work presumes the existence of private channels between each pair of parties (and a larger percentage of honest parties).

"Multi-Prover Interactive Proofs: How to Remove Intractability Assumptions" (Chapter 16). Motivated by the desire to construct zero-knowledge proof systems without relying on intractability assumptions, this work presented a model of multi-prover interactive proofs in which the provers cannot interact with one another during their interaction with the verifier. This model, denoted MIP, turned out to be closely related to the PCP model, which was introduced later and is briefly reviewed below.

Part II of this book reproduces the conference versions of the ten foregoing works (while using the titles of their journal versions, which are different in a few of the cases). These conference versions are extended abstracts that lack many of the details that support the claims made in them, but they best portray the spirit of innovation, boldness, and freshness that is characteristic of Shafi Goldwasser and Silvio Micali.

Part III of this book presents scientific surveys of the works of Shafi Goldwasser and Silvio Micali and of works that were directly inspired by their work. This part starts with a survey of the foundations of cryptography.

On the Foundations of Cryptography. Before spelling out what these foundations are, let us briefly reflect on the significance of such theoretical foundations to cryptographic practice. While the following argument is widely accepted nowadays, it required a convincing advocation in the 1980s. Needless to say, Shafi Goldwasser and Silvio Micali provided such advocation when presenting their pioneering work.

Surely, providing sound theoretical foundations is of great importance for any discipline, but more so for cryptography, since cryptography is concerned with the construction of schemes that should be robust against malicious attempts to make these schemes deviate from their prescribed functionality. A heuristic may make sense when the designer has a very good idea about the environment in which a scheme is to operate, yet a cryptographic scheme has to operate in a maliciously selected environment that typically transcends the designer's view. In fact, the adversary is likely to take the very actions that were dismissed or ignored by the designer. Thus, the design of cryptographic systems has to be based on *firm foundations*, as provided by the research project lead by Goldwasser and Micali in the 1980s.

The foundations of cryptography are the main paradigms, approaches and techniques used to conceptualize, define and provide solutions to natural cryptographic problems. These foundations will be reviewed in Chapter 17, starting with a presentation of some of the central tools used in cryptography; that is, computational difficulty (in the form of one-way functions), pseudorandomness, and zero-knowledge proofs. Based on these tools, the survey treats basic cryptographic applications such as encryption and signature schemes as well as the design of general secure cryptographic protocols. It is striking to note that the entire exposition is rooted directly or indirecting in works of Goldwasser and Micali. Indeed, the history of laying the foundations of cryptography is the story of the works of Goldwasser and Micali.

Impact on Complexity Theory. The revolutionary evolution of cryptography in the 1980s had a great impact on other areas of computer science, most notably on complexity theory. Some of this impact will be reviewed in Chapter 18. Among the direct contributions of the cryptographic evolution to Computer Science, I wish to highlight the theory of pseudorandomness and the study of probabilistic proof systems. Notably, Goldwasser and Micali played a key role also in the development of these specific areas.

A fresh view at the "question of randomness" was taken in the theory of computing: It has been postulated that a distribution is *pseudorandom* if it cannot be told apart from the uniform distribution by any efficient procedure. This paradigm, which was introduced in cryptography where efficient procedures were associated with polynomial-time algorithms that may be stronger than the (purported pseudorandom) generator, has been applied also with respect to a variety of limited classes of such distinguishing procedures, including polynomial-size circuits that are smaller than the running time of the generator, constant-depth circuits, space-bounded machines, local tests (cf., limited independence generators), linear tests (cf., small bias generators), nondeterministic polynomial-time machines, and more. Indeed, this paradigm has been the basis of a vast body of intriguing research concerned with the role of randomness in computation. Also worth noting are the application of pseudorandom functions (e.g., to hardness of *PAC learning* and to "Natural Proofs").

Various types of *probabilistic* proof systems have played a central role in the development of computer science in the last decades. Such nontraditional formulations of proof systems, which allow for a bounded probability of error and view the proof as a dynamic process rather than as a static object, have many advantages over the classical formulation of proof systems (which underlies NP). These

advantages are demonstrated by the known results regarding interactive proofs, zero-knowledge proofs, and probabilistically checkable proofs (PCP). The fruitful connection between PCPs and the complexity of natural approximation problems was also discovered in such a work. This connection has provided a breakthrough in the study of approximation algorithms, which has been almost literally stuck for two decades.

On Some Other Works of Goldwasser and Micali. Although the main topic of this book is the contributions of Goldwasser and Micali to the foundations of cryptography, it would be inappropriate not to mention their direct contributions to other areas within the theory of computation. Some of these contributions are surveyed in Chapter 18, where the perspective is of the impact of cryptography on complexity theory. In addition, Chapter 19 surveys a few other contributions, without mentioning the relations of some of them to cryptography. The selection of titles includes:

- "An $O(\sqrt{|V|} \cdot |E|)$-Time Algorithm for Finding Maximum Matching in General Graphs," which still holds the record for the fastest algorithm for this central computational problem.
- "Certifying Almost All Primes Using Elliptic Curves," which presented a randomized polynomial-time algorithm that produces (absolute) certificates of primality for almost all primes.
- "Private Coins versus Public Coins in Interactive Proof Systems," which provided a transformation of general interactive proof systems into ones in which the verifier only poses totally random challenges.
- "An Optimal Randomized Protocol for Synchronous Byzantine Agreement," which provided a constant-round protocol for this central problem.
- "PCPs and the Hardness of Approximating the Size of Maximum Cliques," which provided a PCP system of almost logarithmic randomness and query complexity for NP, and linked such systems to the complexity of a central approximation problem.
- "Computationally Sound Proofs," which presented natural notions of computationally-sound proof systems.
- "Property Testing and Its Connection to Learning and Approximation," which initiated a general study of approximate decision problems that can be solved in sublinear time, while focusing on testing properties of (dense) graphs.

- "Pseudo-Deterministic Algorithms," which initiated the study of probabilistic algorithms for solving search problems in a consistent manner (i.e., almost always return the same canonical solution).

For each of these selected works, the original abstract is reproduced, and a few additional comments about the work are made. It should be stressed that although Chapters 17–19 review many of the most influential works of Goldwasser and Micali, they are far from exhausting this list, as illustrated by Chapters 21, 24 and 26.

Scientific Vignettes by Some of Their Former Students. A few of Goldwasser's and Micali's former students were asked to write chapters about topics of their choice. Most of them agreed, and some of them delivered. Certainly, Shafi and Silvio do not educate their students to be timely. In their defense, one may say that they don't preach what they don't practice.

Zvika Brakerski's survey (Chapter 20), "Fundamentals of Fully Homomorphic Encryption," reviews a topic that was not pioneered by Goldwasser and Micali. In fact, the partial homomorphic property of the Goldwasser–Micali encryption scheme was considered more as a bug than as a feature, which led them to suggest using it only for the establishing of a key for a symmetric encryption scheme (see their "Why and How to Establish a Private Code on a Public Network," with Po Tong in FOCS 1982). Nevertheless, perspectives have changed, and the potential benefits of fully homomorphic encryption, envisioned by Rivest et al. (in 1978), have been materialized by the surprising discovery of fully homomorphic encryption schemes whose security are based on computational problems regarding lattices.

Computational problems regarding lattices are also the pivot of Daniele Micciancio's survey (Chapter 21), "Interactive Proofs for Lattice Problems." The starting point of this survey is a work of Goldreich and Goldwasser that presented perfect zero-knowledge interactive proof systems for central problems regarding lattices (in order to demonstrate that they are unlikely to be NP-hard). The survey provides the basic background for the computational aspects of lattices, and focuses on several interactive proof systems for various claims regarding lattices, while exposing their underlying ideas.

Johan Håstad's survey (Chapter 22), "Following a Tangent of Proofs," also starts with interactive proof systems, but its actual focus is on the non-approximability results that can be derived from *probabilistically checkable proofs* (PCPs), which in turn arised from multi-prover interactive proof systems. Håstad confesses that, at the time, he considered the multi-prover model to be "artificial" and doubted the justification of introducing an esoteric complexity class that corresponds to it. His past reaction was reminiscent of the reactions that other notions introduced

previously by Goldwasser and Micali have received (e.g., probabilistic encryption and zero-knowledge). Needless to say, in all cases, these skeptic reactions were proved wrong.

Rafael Pass's "Tutorial on Concurrent Zero-Knowledge" (Chapter 23) addresses the issue of preserving the zero-knowledge feature under "concurrent composition." The point is that the original definition of zero-knowledge refers to a stand-alone execution, and the preservation of security under sequential, parallel, and even concurrent executions is far from clear. While augmenting the original definition with auxiliary inputs suffices for sequential composition, preservation of security under parallel and concurrent executions requires some work. Dealing with concurrent executions is most challenging, and the tutorial presents the simplest known solution, which did not appear is isolation before.

Guy Rothblum's survey (Chapter 24), "Doubly-Efficient Interactive Proofs," revisits the notion of interactive proof systems with a focus on more strict complexity requirements. In particular, the (honest) prover strategy is required to run in polynomial time, and the verifier strategy is required to run in almost linear time. Such interactive proof systems, later termed *doubly efficient*, were first defined and constructed by Goldwasser, Kalai, and Rothblum. Interestingly, this notion was considered by Shafi, Silvio, and myself in the mid-1980s, but we failed to find any appealing example (i.e., one in which interaction speeds up verification).

The starting point of Salil Vadhan's survey (Chapter 25), "Computational Entropy," is the notion of computational indistinguishability, put forward by Goldwasser and Micali (see Chapter 7), as applied in the theory of pseudorandomness. This starting point leads to the introduction of computational analogues of other statistical notions such as entropy, min-entropy, KL-divergence, and more. These notions play a major role in the constructions of pseudorandom generators and statistically hiding commitment schemes, which are surveyed in this chapter.

Deviating for the framework that underlies all the foregoing, Yael Tauman Kalai and Leonid Reyzin's "Survey of Leakage-Resilient Cryptography" (Chapter 26) considers cases in which the computing devices used by the honest parties may leak partial information about the their computation or storage. That is, whereas the foregoing views algorithms and strategies as functions (which, once feed with inputs, return adequate outputs), the leakage models attempt to account for the fact that computation is taking place on a physical device that may be subject to various physical measurements, and leakage-resilient schemes attempt to protect against corresponding physical attacks. As noted in the survey, Goldwasser and Micali have contributed significantly also to this research direction.

♠ ♠ ♠

In contrast to this preface, which started with a review of the works of Goldwasser and Micali, the book starts with their lives and voices. Specifically, Part I contains a brief personal biography of each of them, an interview with each of them, which touches on both the personal and the professional, and revised transcripts of their Turing Award lectures.

Brief Biographies. Given the timidness of the theory of computation community, writing personal biographies of its pioneers seems quite challenging. On top of this, I was quite curious to see how a professional writer, who has no background in computer science, will view and portray Shafi and Silvio. I feel that both challenges were well addressed by Michelle Waitzman. It is quite remarkable that Michelle was able to identify key features of their personalities and link these features to characteristics of their scientific research. Her success is well reflected in the titles she choose for the personal biographies: "A Story Behind Every Problem: A Brief Biography of Shafi Goldwasser" and "One Obsession at a Time: A Brief Biography of Silvio Micali."

Interviews. Given that both Shafi and Silvio are very interactive personalities, interviewing them must have been a pleasure. The pleasure was shared among Alon Rosen, who interviewed Shafi Goldwasser, while building on his expertise in cryptography, and Stephen Ibaraki, who interviewed Silvio Micali (as part of an interview series with outstanding computer professionals). The interviews refer both to the personal life and professional work of Goldwasser and Micali, and the former aspects have some overlap with the biographies, where a common theme is indeed the relation of the personal and the professional. Lightly edited extracts from the two interviews are included in this volume.

The Turing Lectures. Finally, this volume includes lightly edited versions of the Turing lectures given by Shafi Goldwasser and Silvio Micali during the *46th Annual Symposium on the Theory of Computing*, which took place in New York, in June 2014. Shafi's lecture focused on the influence of cryptographic research on the rest of computer science, whereas Silvio's lecture focused on the evolution of the notion of proofs.

♡ ♡ ♡

I believe that the work of Shafi Goldwasser and Silvio Micali is of historical dimension. Its impact on the development of cryptography and related areas in

complexity theory has the flavor of a scientific revolution (in Kuhn's sense). Hence, whoever performs research in these areas is living in a world created and shaped by their work. In light of the above, it is our professional and personal duty to acknowledge our debt to these works. This assertion definitely holds about myself, having had also the privilege of benefiting from numerous interactions with Shafi and Silvio.

Oded Goldreich
Tel-Aviv, July 2019

Postscript: The ACM production of this book included re-typing the original papers (for Part II), rather than using facsimiles of these papers, and changing various aspects of the texts of Part III (e.g., the bibliographic conventions and the numbering of theorem-like environments). These production decisions were forced upon the editor, who strongly objected them both per merits and due to the likelihood of errors caused by implementing them.

Acknowledgments

The original papers reproduced in Chapters 8–16 were co-authored also by researchers other than Shafi Goldwasser and Silvio Micali. The list, in chronological order, includes Manuel Blum (Chapter 9), Charles Rackoff (Chapter 8), myself (Chapters 10 and 12–13), Ronald Rivest (Chapter 11), Avi Wigderson (Chapters 12–13 and 15–16), Paul Feldman (Chapter 14), Michael Ben-Or (Chapters 15–16), and Joe Kilian (Chapter 16). Indeed, it is fitting to start the acknowledgments by thanking these researchers. Needless to say, thanks are also due to the many researchers who made contributions that served as a basis and starting point for these works.

Next, I would like to thank the colleagues who have contributed scientific chapters to this book. The list includes Zvika Brakerski (Chapter 20), Daniele Micciancio (Chapter 21), Johan Håstad (Chapter 22), Rafael Pass (Chapter 23), Guy Rothblum (Chapter 24), Salil Vadhan (Chapter 25), and Yael Tauman Kalai and Leonid Reyzin (Chapter 26).

Special thanks to Michelle Waitzman, who wrote the two biographies (appearing as Chapters 1–2), and to Alon Rosen and Stephen Ibaraki for conducting the two interviews (appearing as Chapters 3 and 4, respectively).

Last, I wish to thank Tamer Özsu, the editor-in-chief of ACM Books, for monitoring and assisting the writing and production of this volume. I also thank Paul Anagnostopoulos of Windfall Software for supervising the production while trying to accommodate my various requests as much as allowed by the ACM.

Photo and Text Credits

Photos

Page viii Jason Dorfman MIT/CSAIL

Page xxix Jason Dorfman MIT/CSAIL

Page 1 Photo Courtesy of Silvio Micali and Shafi Goldwasser

Page 2 Photo Courtesy of Silvio Micali and Shafi Goldwasser

Page 158 Photo Courtesy of Silvio Micali and Shafi Goldwasser

Page 173 Photo Courtesy of Silvio Micali and Shafi Goldwasser

Page 411 Photo Courtesy of Silvio Micali and Shafi Goldwasser

Page 412 Photo Courtesy of Silvio Micali and Shafi Goldwasser

Page 795 Photo Courtesy of Oded Goldreich

Page 796 Photo Courtesy of Zvika Brakerski

Page 796 Photo Courtesy of Johan Håstad

Page 797 Photo Courtesy of Yael Tauman Kalai

Page 797 Photo Courtesy of Daniele Micciancio

Page 798 Photo Courtesy of Rafael Pass

Page 799 Photo Courtesy of Leonid Reyzin

Page 799 Photo Courtesy of Guy Rothblum

Page 800 Photo Courtesy of Salil Vadhan

Text

Page 157 © 2014 Silvio Micali

Page 176 Shafi Goldwasser and Silvio Micali. 1982. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing (STOC '82).* ACM, New York, NY, USA, 365–377. DOI: http://dx.doi.org/10.1145/800070.802212

Page 204 S. Goldwasser, S. Micali, and C. Rackoff. 1985. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing (STOC '85).* ACM, New York, NY, USA, 291–304. DOI: http://dx.doi.org/10.1145/22145.22178

Page 228 M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo random bits," in *23rd Annual Symposium on Foundations of Computer Science,* Chicago, IL, USA, 1982 pp. 112–117. DOI: http://dx.doi.org/10.1109/SFCS.1982.72

Page 242 Oded Goldreich, Shafi Goldwasser and Silvio Micali. "How to Construct Random Functions (Extended Abstract)," 25th Annual Symposium on Foundations of Computer Science, 1984, Singer Island, FL, USA, 1984, pp. 464–479. © 1984 IEEE. DOI: http://dx.doi.org/10.1145/6490.6503

Page 266 S. Goldwasser, S. Micali and R. L. Rivest, "A 'Paradoxical' Solution To The Signature Problem," 25th Annual Symposium on Foundations of Computer Science, 1984, Singer Island, FL, USA, 1984, pp. 441–448. DOI: http://dx.doi.org/10.1109/SFCS.1984.715946

Page 285 O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity and a methodology of cryptographic protocol design," 27th Annual Symposium on Foundations of Computer Science, 1986, Singer Island, FL, USA, 1986, pp. 174–187. DOI: http://dx.doi.org/10.1109/SFCS.1986.47

Page 308 O. Goldreich, S. Micali, and A. Wigderson. 1987. How to play ANY mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing (STOC '87),* Alfred V. Aho (Ed.). ACM, New York, NY, USA, 218–229. DOI: http://dx.doi.org/10.1145/28395.28420

Page 330 Manuel Blum, Paul Feldman, and Silvio Micali. 1988. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC '88).* ACM, New York, NY, USA, 103–112. DOI: https://doi.org/10.1145/62212.62222

Page 352 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. 1988. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC '88).* ACM, New York, NY, USA, 1–10. DOI: https://doi.org/10.1145/62212.62213

Page 374 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. 1988. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC '88).* ACM, New York, NY, USA, 113–131. DOI: https://doi.org/10.1145/62212.62223

Page 413 "On the foundations of cryptgraphy," by Oded Goldreich. Copyright © 2019 by Oded Goldreich. Reprinted by permission of Oded Goldreich

Page 497 "On the impact of crytography on complexity theory," by Oded Goldreich. Copyright © 2019 by Oded Goldreich. Reprinted by permission of Oded Goldreich

Page 527 "On some non-cryptographic works of Goldwasser and Micali," by Oded Goldreich. Copyright © 2019 by Oded Goldreich. Reprinted by permission of Oded Goldreich

Page 543 "Fundamentals of Fully Homomorphic Encryption—A Survey," by Zvika Brakerski. Copyright © 2018 Zvika Brakerski. First published in *Electronic Colloquium on Computational Complexity (ECCC),* July 7, 2018; https://eccc.weizmann.ac.il/report/2018/125/. Reprinted by permission of Zvika Brakerski.

Page 565 "Interactive Proofs for Lattice Problems," by Daniele Micciancio. Copyright © 2019 by Daniele Micciancio. Reprinted with permission of Daniele Micciancio.

Page 599 "Following a tangent of proofs," by Johan Håstad. Copyright © 2019 by Johan Håstad. Reprinted by permission of Johan Hastad.

Page 623 "A Tutorial on Concurrent Zero Knowledge," by Rafael Pass. Copyright © 2019 by Rafael Pass. Reprinted by permission of Rafael Pass.

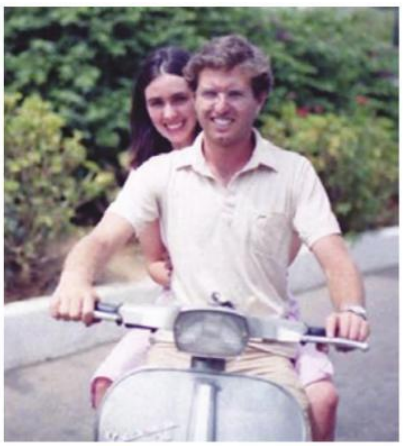
Page 649 "Doubly-Efficeint Interactive Proofs," by Guy Rothblum. Copyright © 2019 by Guy Rothblum. Reprinted by permission of Guy Rothblum.

Page 693 "Computational Entrophy," by Salil Vadhan. Copyright © 2019 by Salil Vadhan. Reprinted with permission of Salil Vadhan.

Page 727 "A Survey of leakage-Resilient Cryptography," by Yael Tauman Kalai and Leonid Reyzin. Copyright © 2019 by Yael Tauman Kalai and Leonid Reyzin. Reprinted with permission of Yael Tauman Kalai and Leonid Reyzin.

PART I

BIOGRAPHIES, INTERVIEWS, AND AWARD LECTURES



1 A Story Behind Every Problem: A Brief Biography of Shafi Goldwasser

Shafi Goldwasser has always loved a good story. As a little girl, she couldn't get enough of them and sometimes returned to the library several times a day to exchange one book for another. Early on, she expected that this would lead her to become a writer, but life had other plans for her.

Discovering that mathematics was even more exciting than literature, Shafi's path changed direction and she became a leading theoretical computer scientist. Her way of seeing the world—connecting ideas that may seem unconnected to most people—has led to a career full of accomplishments, awards, and admiration. But the storyteller in Shafi never left. Instead, it has given her a creative approach to working on mathematical problems. Shafi sees the "story" behind each problem that she researches. Where some might see a verifier checking a proof, she can picture a detective questioning a suspect.

Shafi says, "I usually find a problem interesting if there's a story associated with it. If I can think of the story of why a problem is interesting, not necessarily an application—something I could grab onto: a model, a story. I think my love of stories is kind of the way I think of these models."

One of Shafi's former students, Guy Rothblum, summarizes her talents this way: "Shafi is both incredibly brilliant and creative as a researcher. She makes things that used to be impossible—or that you would think were impossible—possible. She makes these incredible leaps between fields and finds these connections and you think, 'How in the world did she come up with this?"'

Fellow researcher Oded Goldreich agrees that Shafi's ideas can seem so unusual at first that it's tempting to dismiss them as impossible. "When Shafi suggests anything, one should resist the immediate reaction of saying 'this cannot work' and examine the core idea carefully," he says. "Carefully think about the core of what she said, rather than dismissing it on the spot as being 'odd."'

Shafi recognizes that she sees things differently than some of her peers, as she explained in an interview with technology journalist Stephen Ibaraki. "I don't think in narrow ways, I try to see things from a larger perspective. I try to think of a problem from different perspectives and I see connections to problems I've thought about in the past, or maybe something that other people are working on."

When Shafi was invited to present an AMS/MRSI congressional briefing on the topic of data protection in 2017, she explained how theoretical research like hers can have surprising applications many years later. After all, her early work on cryptography took place long before we lived in a world of online commerce and big data. She told the briefing attendees, "A problem that seems unsolvable actually often has technical solutions that are based on some basic research that was done 30 or 40 years ago by people who didn't know about the problem—or care about it."

In a field where the value of creativity is often not recognized, Shafi is a living example of what can be accomplished when a precise mathematical mind and a creative outlook are combined in one dedicated researcher.

1.1 Beaches and Books: An International Childhood

Shafi's parents grew up in very different circumstances. Her father, Zvi, was a young man studying law in Poland when the Second World War interrupted his plans. He fled to Russia, and eventually returned to Poland to help drive the German army out. Not knowing whether any of his relatives had survived the war, Zvi moved to Israel to get a fresh start. He would eventually reconnect with his mother and sister and bring them to Israel, too.

Rachael, Shafi's mother, was born in Israel and raised in an agricultural community. She was a student during the late 1940s, and returned home from her studies to find Zvi renting a room in her parents' house. She began teaching him Hebrew, and soon he asked her to be his wife. They married in 1948.

Zvi never completed his legal studies, but instead found work with the new Israeli health service. The young country lacked everything: doctors, hospitals, and of course funds. Zvi and his family were sent to New York City to try to entice Jewish Americans to help. Young doctors were encouraged to move to Israel, and wealthy

members of the community were asked to donate to the health service. The couple already had a young son, Nathan, when they moved overseas. In 1958, while they were living in New York, they had a baby girl and named her Shafrira, but she would be called Shafi by everyone she knew.

From her first day, Shafi had two nationalities. She was American-born to Israeli parents. She spent her earliest years in a coastal community just outside New York City, called Seagate. The community is surrounded by beaches and very close to the famous Coney Island beach and boardwalk. Shafi has fond memories of daily trips to the beach with her mother. She attended a local kindergarten in New York. In true urban fashion, the school's playground was on the roof of the building.

When Shafi was about six years old, her father was transferred back to Israel, and the family moved to Tel Aviv. They settled in an area known as the "Old North" of the city, and brought with them a Dodge Dart that they had bought in New York. It was a common car in the United States at the time, but a real novelty in Israel. "At that time in Israel they didn't exist," says Shafi. "In the United States we had this car, and it was just a small car. Then it arrived on the ship to Israel and it was huge—it was like the biggest car ever in the streets."

She had moved across the world, but Shafi still continued to have a strong connection to the sea. "My parents used to go to the beach every day in Tel Aviv, because my father loved swimming, and so did my mother since she grew up in Kfar Vitkin, which was near the beach. And we used to go every day—six o'clock in the morning," Shafi recalls. Her parents' love of swimming rubbed off on Shafi, who cherished their daily swims throughout her childhood.

She also had the opportunity to learn about rural life by going to visit her grandparents. Growing up in the urban surroundings of New York and Tel Aviv, this rural experience was a precious opportunity. "My grandparents from my mother's side, they lived in Kfar Vitkin. They had an agricultural farm or unit—they had cows and chickens—and I had cousins there. Every weekend we would go there to spend time with them, have lunch, go to the beach. Sometimes I would spend weeks there in the summer. So this connection with this farming place, or *moshav*, is very strong in my mind. That is really childhood, that and the beach."

Her family had arrived in Tel Aviv well after the beginning of the school year, so Shafi had to quickly adapt to a new type of schooling in a new language. She was a novelty to her fellow students, who were not used to people immigrating from the United States. "I think that for the rest of my duration at school, which was eight years—and even today, they remember me as the girl who came from America. Which shows you how Israel was at that time, that that was such a rare occurrence. And because I didn't know Hebrew for the first few weeks, I think they

sort of remembered me as someone who didn't know how to speak Hebrew in the beginning," she says.

The school was far from her home and she had to take a bus there, but it was recommended to Shafi's parents because it was affiliated with the ruling political party in Israel at the time, and it had long hours and provided a good education with strong fundamentals. However, not everything there was good, according to Shafi. "I kind of hated the food. I remember that one day they wanted me to stay and eat everything, because you're supposed to clean your plate, especially at that time in Israel. My mother happened to pass by and she told the teacher that there's no need to force me to eat anything. That's a very strong, protective memory of my mother, that I knew that I could really do whatever I want, which was always true."

It was during these school days that Shafi developed her love of reading and stories. She read novels, historical dramas, short stories—anything that would feed her imagination. Her mother was also a great lover of literature and history. Although writing stories of her own seemed like a natural progression for her, she was not prolific. "I liked to write essays and short stories, but I didn't write that much. With all my fantasies about being a writer, I don't have a bunch of manuscripts hidden in my drawers," she says.

She did team up with one of her friends to create a newspaper. Shafi wrote the articles and her friend was the business manager. But sales of the first issue were disappointing (she thinks they sold one or two copies) and the venture didn't last.

As for those short stories, Shafi was a shy author and didn't share her creative endeavors. "I think they were so full of my own desires and fantasies for the future that I would have considered them extremely personal and I don't think I would have shown them to anyone," she says.

Shafi's childhood took another turn when at age 11 she became a big sister to a new baby girl, Ricky. "I remember when she was born. I was in the sixth grade or something like that. In fact, I remember that I'd made a deal with my parents. I really wanted a dog, and they said that the dog, I won't get. So then I said, 'Okay, so either a dog or a sister,' and we wrote this contract. And I have it actually. I found it a few years ago, when I was cleaning my parents' apartment. In any case, I got a sister."

Given her reluctance to share her writing with the world, it is perhaps fortunate that she found a new area of interest in high school: mathematics and science. It became clear to Shafi that she was good at these subjects, and in Israel at that time they were considered to be very important areas of study. "And that's still

true in Israel," she says. "Those in math and science, those are the people who are respected. I think it was the combination of finding it interesting, being good at it, and realizing that this is what was expected of me."

"I think that growing up in Israel during those times—and it's maybe not so much growing up in Israel as growing up as children to a generation of parents who either came to Israel after the (Second World) War, or grew up during the war for independence in Israel—there's a great deal of pragmatism in the education system and what they teach you, and in terms of your approach to life. There's always a goal; you always have the future in mind. There's a goal you're working toward."

The 1960s and early '70s was a difficult time to grow up in Israel. The country was at war at times during Shafi's childhood, and the threat of war was never far away. During the Six-Day War, which took place in June 1967, Shafi was in fourth grade and remembers hiding in a bunker when sirens sounded. Years later, her brother Nathan was doing his compulsory military service when the Yom Kippur War began in October 1973, and Shafi recalls that he came home briefly before heading off to war. "He told us that there's going to be a war. And my father said to him, 'What are you speaking nonsense for?' Because for Jews that came from the (Second World) War, the whole idea of talking about death and war—it was something that you just don't talk about because it's just bad luck or you just don't say things like that. Then he was called and he left, because he had to go back to the army, and we didn't see him for a few weeks."

Shafi's parents were worried about their son's safety. Zvi had lived through the horrors of the Second World War, and both parents were in Israel during the country's war of independence. That war lasted from 1947 to 1949, and a large number of the young soldiers lost their lives.

The Yom Kippur War also took its toll on Nathan. "I remember when he came back home the first time, he had a lot less hair. It was amazing that this kind of traumatic experience can do that. So he went with a full head of hair, and it receded."

After the war, Nathan had planned to study mathematics at Hebrew University in Israel. But Shafi's parents feared that he might be involved in another war and wanted him to leave Israel. "My father just wanted him out of Israel as fast as possible. He was so afraid for his safety that he wanted him to go to school in the States. And he got accepted to Carnegie Mellon and he left. That affected me because that started some sort of chain reaction in the family," says Shafi. In fact, all three siblings eventually ended up living in the United States as adults.

Shafi would not join her brother at Carnegie Mellon for several more years. In the meantime, she was exploring her interest in math and science. At Shafi's high school, students specialized for their final two years. "When we went into the specializations, there was a class that specialized in math and physics. There were a few girls, not too many. But they were very strong, the ones who were there were very strong." Despite being in the minority, Shafi never felt that her teachers treated her differently because she was a girl. "I was a good student always. And I felt that the teachers respected me."

"I enjoyed math, because there's always a right answer. At least in high school it seems like there's always a right answer. But I loved physics even more." Shafi was particularly attracted to physics because it gave her the tools to find solutions. "The understanding from axiomatic or first principles, how you get to a conclusion." She also felt that the problems associated with physics had stories associated with them. Physics was not just about manipulating numbers, it was about understanding how things in the real world affect one another.

Even at this early age, Shafi was looking at problems differently from her peers. Not many teenagers would describe the derivation from principles as "beautiful," but that's how Shafi felt about it. Her approach to problems was not tied to rote learning. On exams, she would consider the problems in more creative ways and come up with answers that took her teachers by surprise. Shafi's high school physics teacher may have been the first person to get a glimpse of the talent for making unexpected connections that has been the hallmark of her research.

Shafi credits her high school math and science teachers for encouraging her love of these subjects and igniting her curiosity. They sparked an interest in her that continued to build throughout her studies. What began as a general curiosity about these subjects in high school had opened her eyes to the excitement of studying and the pursuit of knowledge, which would lead her to explore fundamental questions throughout her career. She feels that she was lucky to have had those teachers early on who fed her curiosity and sent her down that path of investigating problems that excited her.

Shafi's parents also had a big influence on her decisions. Throughout her childhood, Shafi was encouraged to pursue great things in her life. Her father did not let traditional gender roles alter his expectations. "You know, there was no difference between men and women here, and he thought we could do anything. That was very unusual, and that was true all along. This whole idea that women should behave a certain way, they should get married, they should have families, that was completely beside the point for him. And he was very vocal about that. And he thought I was big-

ger than life. That was a good thing, to grow up having that image of yourself—that empowerment."

Although Shafi's mother was a homemaker, she also encouraged her daughters to be strong and not depend on anyone else to take care of them. "She would say to us, both me and my sister throughout growing up, that a woman has to take care of herself and she has to be independent, it's extremely important. And I think that probably was because she wasn't. My father was the one who was the breadwinner. And I think in her mind, anything that was a step toward accomplishing that was a good step."

At the time, Shafi's thoughts had not yet turned to the next stage in her education. In Israel, compulsory military service normally takes place between graduating high school and going to university. Students will take their exams at the end of high school, but applying to universities still seems a long time off. However, Shafi found herself in an unusual situation. Having her birthday late in the year meant that she was younger than most of her classmates. So even though she had completed high school, it would be almost a year before she was due to report for her military service. She had a substantial amount time on her hands.

"My father wanted me to go to the U.S. to study so that I didn't waste any time. This idea of wasting time is something very problematic, or was very problematic when I was growing up. Now it seems like everybody's just taking trips around the world as soon as they finish the army, or before the army, and wasting time is not called 'wasting time' any more but 'gaining life experience.' In any case, my father wanted me to go to the States, and as usual I did what he recommended."

1.2 The Mind-Blowing World of Computer Science

Sending their 17-year-old daughter overseas did not seem to worry Shafi's parents. Not only was she a bright, hard-working student, she was going to be studying at Carnegie Mellon University, where her older brother Nathan was a graduate student, and he would be nearby if she needed him.

Her flight landed in New York City and Shafi had her first opportunity to revisit the home of her early childhood. "I wasn't there for 11 years, but when I took the taxi from the airport to the city it seemed extremely familiar." Her brother met her in New York and traveled with her to Pittsburgh, where she would move into the student dorms at Carnegie Mellon and wait for the new academic year to begin.

Her brother knew the math professors at the university, and told them that his sister was spending a year there and that she was good at math. Based on his

word alone, Shafi was able to start classes as an undergraduate student in the math program without ever officially applying to the program.

Shafi's plan was to spend the year studying and then return to Israel in time for her military service. However, things didn't go according to plan. During that first year, Shafi decided that she'd rather not interrupt her studies, and she applied to the Israeli government to defer her military service. That deferral took several months to secure, and in time it became a permanent deferral and Shafi was able to focus on her education with no interruptions.

Shafi's first lecture was difficult for her to understand, and she thought perhaps she'd made a mistake and wasn't ready for these university-level math courses. But after telling her brother about her struggles, he realized that her problem was not with understanding the math, it was simply that she hadn't learned the mathematical terminology in English. After she got a rundown of how the Hebrew words translated into English, everything made much more sense, and Shafi had no more problems understanding the lectures. But the courses were still a big change from what she'd been used to at high school.

"When you get to math in college after high school, it's very abstract. There's this gap between the beautiful abstractions and this field (of computer science) that seems to capture things about life." Although Shafi felt that these abstract concepts were interesting, she was not sure that this was the field she wanted to pursue. She thought that this approach to mathematics was going to take too long to come to fruition, and that perhaps she should try studying computer science instead. The undergraduate mathematics program had a computer science specialization that students could select, and that was what Shafi chose to do.

That turned out to be a life-changing decision for Shafi. Soon, she had her first computer science classes and her first experience with computers. She recalls, "I was fascinated by their potential; I was fascinated by the first courses I took on computer programming, which had a lot of algorithm design. You design a program to resolve an algorithmic problem and there are many ways to do it and there are efficiency constraints and technical constraints and then—the program just did it! Now it's taken for granted, but that idea that you can use a computer to solve a mathematical task was sort of mind-blowing to me."

She had decided to study mathematics at Carnegie Mellon simply because her brother was there, but that decision was one that likely affected the rest of her career. She arrived in 1976, when there were few computer science departments at universities anywhere in the world. Carnegie Mellon was one of the pioneering institutions in the field and had attracted some of the top academics of the time. In fact, computer science was already well established at the university by the time

Shafi arrived. They had introduced their first computer science course in 1958, the year of Shafi's birth. By 1961, they had added a Ph.D. in the field, and in 1965 they established a computer science department.

"I think being at Carnegie Mellon was a godsend," she told Stephen Ibaraki. "It was a very exciting time. There were all these greats in the field: I took artificial intelligence with Raj Reddy, and I took my first algorithms course with Jon Bentley, and I took a course in software engineering from Anita Jones. All these people were tremendous lecturers and they taught me a tremendous amount. That made me realize how exciting computer science was."

Turing Award–winner Raj Reddy was a leading artificial intelligence researcher and one of the first academics to explore speech recognition. He sparked a strong interest in AI for Shafi. "I loved the idea of doing artificial intelligence. I thought that's maybe what I would do—understand the brain, understand how people think and how machines can mimic our thought process."

Shafi drew on her love of literature to help create a program that could generate poetry using artificial intelligence. "Compared to what they do today, it's probably totally childish," she said in an interview with the Heidelberg Laureate Forum. "But at the time, the whole possibility of writing down a sort of linguistic map of how language can be derived was fascinating."

The computer programming professors at Carnegie Mellon included Anita Jones and Mary Shaw. Jones would later become Director of Defense Research and Engineering for the U.S. Department of Defense and Vice-Chair of the National Science Board, which advises the President on science, engineering, and education. Shaw has been a faculty member at Carnegie Mellon since completing her Ph.D. there in 1972. She is considered one of the founders of the field of software architecture. Studying in a faculty with female professors who were "figures of importance" in computer science was empowering for Shafi. Women were far outnumbered by men in her classes, but Shafi had no trouble demonstrating to her professors that she was a very capable student. Professor Jones was working at the time with the university's 50-processor computer on a project called Cm*. She brought Shafi onto the project, making her one of the first people to work with a multiprocessor computer.

Shafi was excited about her newly discovered love of computer science, but she was still a teenager away from her friends and family for the first time. Most of her high school friends were doing their military service, and Shafi wrote letters back and forth with some of them, but they eventually drifted apart. At the same time, her fellow students in Pittsburgh were very welcoming. "The people I met were very curious about the world, and they were curious about me."

When she'd moved from the United States to Israel at the age of six, she'd grown up as the girl from America. Now, her American peers thought of her as the girl from Israel. It seemed as though Shafi was destined to be "exotic" no matter where she went. She found that her fellow students were more adventurous than she was. "I'd lived a very sheltered life, I think. At that time kids (in the United States) were much more adventurous in high school than I was. So that was surprising, to be in this place where everybody's exploring different aspects of life."

During her first year in the United States, Shafi's only contact with her parents was through writing letters. "I think in their minds I was capable of this journey. But really, internally, I was just a kid. I missed my parents very much. At that time, in Israel, somehow the idea of a phone call to the U.S.—it was like an impossibility. It wasn't really an impossibility, but it seemed *so* expensive, nobody called."

Her family did come to the United States for a visit during summer break, reuniting with Shafi and her brother. Her father would later return to attend her graduation. However, it would be several years before Shafi would have an opportunity to travel back to Israel. She spent her summers at Carnegie Mellon taking courses that didn't fit into her regular studies. Computer science was her passion, but she hadn't abandoned her love of a great story. "I found the literature courses that I took in the summer incredible—of course I was exposed to literature in school, but all these wonderful English-language plays and writers—I loved it, it was fabulous."

1.3 Blue Skies and Green Hills

After completing her bachelor's degree, Shafi had to decide whether to get a job in the industry, return to Israel, or further her education in the United States. She decided the third option was the most appealing and that she should apply to graduate school. Despite having gone through her undergraduate education entirely in the United States, however, nobody had informed Shafi about the required entrance exam for U.S. graduate schools. "I decided to apply to grad school and then I found out you're supposed to take this exam, the GRE. It's like the day before. I never opened a book. I'm not going to say what I got." Despite her lack of preparation and the feeling that she hadn't done well on the test, Shafi was accepted into the engineering program at Carnegie Mellon and the computer science program at the University of California, Berkeley.

She decided that she would continue her studies at Carnegie Mellon, but first Shafi needed to earn some tuition money. Her AI professor, Raj Reddy, recommended her for a job at RAND Corporation and she was offered a position for the

summer. Spending the summer in Santa Monica, California, was appealing, but the job itself opened up Shafi's eyes to the corporate world.

"Being at RAND and seeing that all these Ph.D.s were really running the show, it became very clear to me that I wanted be able to be in charge and define my own projects, rather than do what I'm told. I never really had any interest in business or being on the corporate side," she says.

During that summer, she met up with a friend who was also working in Southern California, and the pair took a road trip up the coast to Berkeley. Shafi was struck by the beauty of the California coastline, and her arrival at Berkeley for the first time quickly changed her plans for the future. The clear blue skies, the rolling green hills, and the charming buildings of the Berkeley campus were irresistible. Shafi decided to attend the university for her graduate studies.

"The computer science department—I loved the building where it was. There were big windows with a view of the campanile. I thought I was going back to Carnegie Mellon, but I had a look [at Berkeley] and I fell in love," she says.

This decision would change more than the scenery and weather for Shafi. She'd had every intention of pursuing artificial intelligence on her return to Carnegie Mellon. Once she was at Berkeley, however, she found herself moving in a different direction.

Shafi told the Heidelberg Laureate Forum, "On Mondays they'd have these seminars and three professors would get up and talk about their research. And based on that I decided to go into a master's with Dave Patterson on the RISC (restricted instruction set computer) project. The project was to collect statistics for the Pascal programming language—how often different commands were being used—because he was trying to optimize the instruction set, and those instructions would be included in the hardware. So I wrote this very large system and that's my master's."

It was her first big project and she threw herself into it. The idea of being so engaged in a project that she would continue working on it day and night was new to Shafi. As an undergraduate she had done a lot of reading and studying for exams, but this was different. She was in charge of her own project and deadlines, although expected to accomplish things and deliver results to her supervisor. It was her first taste of life as a researcher.

After completing her master's, Shafi finally returned to Israel for the summer to visit her family. Her family had come to the United States to see her, but Shafi had not been back to the country she had called home for most of her life since her departure after high school. Her sister was growing up quickly, and it had also been several years since she'd seen her mother. But the reunion was temporary since Shafi had already decided to return to Berkeley to get her Ph.D.

1.4 Theory and the Cryptography Revolution

Although her master's project had been programming based, once she returned for her Ph.D. Shafi was won over by the theory students and professors. "I meet all these theory students. And they're telling me, this Manuel Blum, he's great, and Dick Karp, and I should really go and talk to them. So I talked to Manuel Blum, and we hit it off and he says, 'Yeah, if you want to work with me, you can work with me.' The next year he teaches a course on computational number theory and I love it. It's clear to me that I've found something that I really like. Somehow it's extremely appealing to me," she told the Heidelberg Laureate Forum. "And then in the last few lectures he talks about RSA (Rivest–Shamir–Adleman) and cryptosystems and it's fabulous!" Shafi was fascinated by the combination of algorithms and number theory, the use of randomization in algorithmic design, and the connection to cryptography.

It was during that course in computational number theory taught by Manuel Blum that Shafi and Silvio Micali started down their shared path toward revolutionizing the field of cryptography—the work that would eventually earn them the Turing Award. Shafi was inspired by Blum's discussion of a theoretical problem in one of his lectures. "He presents this problem at the end," she says, and describes the story of a couple who are fighting over custody of their dog. One of them lives in San Francisco and the other in Los Angeles. They decide that flipping a coin would be a fair way to decide, but they can't do it in person, and neither one trusts the other to do it fairly. Can they do it over a distance and be sure of the result? Shafi was intrigued, as she told the Heidelberg Laureate Forum. "How would they do it using computational number theory ideas? Silvio Micali's also taking this class and I'm telling him this is really *the* problem. We should work on this. It's clear—I want to work on this."

The question was fascinating mathematically, and there was an added appeal for Shafi because Manuel Blum had presented the problem as a story, with characters who needed to resolve their situation. The story behind the problem was easy to see here, and the same could be said when Shafi and Silvio went on to work on the problem of playing "mental poker." The problem could easily be pictured by imagining someone shuffling nonexistent decks of cards, and having to encrypt 52 different potential cards without letting their opponents know anything about which cards they've been dealt.

When it came to those early research projects at Berkeley, Shafi claims that she and her peers simply "followed our excitement." It was not important to find real-world problems that needed to be solved, or to see commercial applications for their work. Intellectual curiosity and a challenging problem were enough to inspire

their work. She has always felt that it is important for researchers to ignore the current trends or popular problems that other researchers are working on. Only by getting past the dogma of the time is it possible to do truly innovative work.

Shafi found her place in the areas of cryptography and complexity theory, and also made some wonderful new friends. The theory students at Berkeley would become constant companions for the next several years. They would work together, eat together, relax together, and spend vast amounts of time talking to each other about theoretical computer science with endless enthusiasm.

This close group of friends included Silvio Micali, who would co-write Shafi's first paper with her and remain her close friend throughout their careers. She also became good friends with Vijay Vazirani, Faith Fich, Joan Plumstead, Mike Luby, Eric Bach, and Jeff Shallit. The students were also friendly with their professors and spent a lot of time with them. "The professors like Dick Karp and Manuel Blum and Eugene Lawler—all three of them were such open personalities and so perceptive and so wise, and they would go to a coffee shop with a group of graduate students and we would ask questions and talk about research. It was such a marvelous intellectual and dynamic and inspiring place," Shafi says. "Their enthusiasm for what they were doing and their clarity of thinking were priceless for me."

While Shafi might have seemed exotic to her American peers, she was equally fascinated by the other members of her multicultural group of friends, each of whom was a wonderful new source of stories. Silvio could share his stories about Italy, and Vijay had stories about India. Her new friends were worldly and colorful, and they helped to expand her experience of the world. Most were a bit older than Shafi, and she enjoyed talking about life and work with them when they hung out on campus and in local restaurants. Shafi also had good friends at Carnegie Mellon, but at undergraduate school there was a different atmosphere. People came to graduate school at Berkeley from all around the world and they had very different backgrounds and stories, which was an ideal atmosphere for Shafi.

Shafi and Silvio submitted their paper on playing mental poker to a conference held by STOC (Symposium on the Theory of Computing), which was attended by members of the theoretical computer science community. Their paper was accepted, and it was Shafi's first opportunity to talk about her research to her peers outside of Berkeley. STOC is one of the two main conferences for theoretical computer science. In those days there were no parallel sessions; only one presentation was given at a time, so all of the attendees could see every presentation. Other students might have been intimidated in this situation, but Shafi remembers feeling good about it. "Somehow I had confidence as a presenter, maybe unjustified to begin with."

According to former MIT graduate student Guy Rothblum, her confidence was not misplaced. "Both Silvio and Shafi have this sort of magnetic field where, when you're listening to what they're thinking about, it's clear that it's the most interesting thing in the world."

For Shafi, it was an eye-opening experience as a graduate student to attend and give presentations at conferences during her studies. "I realized that there was a community of theoretical computer science at large that was very excited about research we were doing. And I realized that there was this whole world out there of people who were really intensely dedicated to this, and that I was part of it as a graduate student and as a researcher; being able to present at these things, being able to be respected and listened to, to realize that my work was important."

This enlightening experience came just two years into Shafi's Ph.D. studies. She would continue down this research path until she completed her thesis, "Probabilistic Encryption: Theory and Applications," in 1983.

1.5 A Mecca for Cryptography

With her studies complete, Shafi headed to the east coast to take up a postdoctoral position at the Massachusetts Institute of Technology (MIT). She felt that there as no better place for a cryptography researcher, describing it as a "Mecca for cryptography," in particular because pioneering cryptography researcher Ron Rivest was on the MIT faculty, and because the RSA cryptographic system, invented by Rivest, Adi Shamir, and Leonard Adleman, was associated with MIT, since all three were members of the MIT community at the time. But MIT's commitment to the field of cryptography would soon become even more apparent. Within months of arriving as a postdoc, Shafi would be offered a staff position in the computer science department, and within a year, they would add Silvio Micali to their staff. In addition, Adi Shamir and Michael Ben-Or were visiting professors for a year, and Oded Goldreich started his postdoc there and eventually stayed for three years. In a brief period, MIT built one of the leading cryptography groups in the world. "When I came to MIT from Berkeley, it was just an explosion of research and research freedom. There was a very active group of researchers who I collaborated with and who made everything very exciting," she told Stephen Ibaraki.

This atmosphere enabled Shafi and Silvio to enjoy more of the productive collaboration that had begun between them back at Berkeley, and they continued to work on papers together, sometimes with other collaborators. Silvio reflected on what makes Shafi such an interesting research partner. "I sometimes joke that she has multiple personalities! So it's great to interact with her because it's like in-

teracting with more than one person. For example, she will advocate A, then the opposite of A, then B, then C. I find her unpredictable. I think unpredictability is a good thing in research," he says. "With some people, you talk to them and you get what you're going to get from them, but with Shafi you keep on going because she changes, and that's crucial."

Shafi agrees that one of the reasons she and Silvio are such productive collaborators is the differences in how they approach problems. "We don't have the same kind of mind. I think that collaboration between people with the same kind of mind is sort of useless. Silvio is very 'extrematic,' he's very abstract, and I'm much more intuitive."

Other colleagues at MIT can confirm Silvio's description of Shafi's unpredictable nature, and it's one of the things they enjoy most about spending time with her. According to Ronitt Rubinfeld, a professor in MIT's computer science department since 2004, Shafi's spontaneity extends beyond the research realm into her social life. Nights out with Shafi are such an adventure that her friends never turn down the chance to see what will happen next.

"They drop anything to be with her," Ronitt says, "knowing that if she suggested to go to a movie, once arriving at the theatre, the plan may change to going for a walk, but as soon as the walk starts, it changes to going to a cafe, and after five minutes at the cafe, who knows what would be next. But they don't really care what exactly they are doing when they are with her, they just care about being in her presence, because there is something about being with her that makes life exciting."

Shafi's tendency to act on her intuition can also be seen in the seemingly random ways that she finds problems she'd like to work on. Oded Goldreich has seen this in practice. "I believe that in most cases, she hears an idea (mostly in a talk) and takes it to a totally different place, which would make little sense to the person who communicated the idea but makes sense to her," he explains. "I think one should think of the ideas she hears and processes as raw material for her spontaneous imaginative processes." Oded also feels that Shafi's creative nature plays a huge role in her abilities as a researcher. "What is stunning with Shafi is her intuitive creativity—her spontaneous nature. She just sees things that nobody does. Her insights are totally out of the box."

Shafi and her peers continued to produce research that would change the study of cryptography going forward. They hadn't necessarily set out on a mission to revolutionize the field, but their approach to solving theoretical problems led to just such a revolution and laid the groundwork for many eventual applications. She told the Heidelberg Laureate Forum, "Nobody aims for revolutionary impact. I

believe that basic research is the only way that this type of impact will come about. I don't believe that there's anything that can be done that will have such fundamental implications if you already know the applications, because everybody else can do it. It can be interesting and good, but it's not revolutionary."

In addition to her research, Shafi was now teaching courses at MIT. During her first experiences, she had the support of more experienced professors. "Nobody really teaches you how to teach. At least in my time they didn't. The lucky part was that when they were teaching big undergraduate courses there were other people teaching with me."

Her teaching evolved over time, and she has developed her own courses over the years that cover her research topics. "I used to really go through the process of how you get to a result, especially if I was talking about my own research."

As a former graduate student who was supervised by Shafi, Guy Rothblum found Shafi's lectures on her own work very compelling. "She has a grasp of the big picture and she always knows how to explain what's revolutionary about the work. She's good at explaining the conceptual aspects of the work, and not just the technical part. In her talks, what she really homes in on and what she really gets across are what the big, new, important ideas are: what's exciting about this problem," he recalls. "It's unbelievably exciting to be talking with someone like that, who's sort of fearless, not only in terms of the kinds of problems she approaches, but who has also shown the right way, or the right direction to take in order to make progress on these sorts of very basic, big problems."

Creativity has continued to be a big part of Shafi's motivation, whether applied to her own work or to the students she is supervising. "The best part about being at a university is you meet new students and people are so talented and you never know where their talents lie," she says. "This creativity or this ability, it never ceases to amaze you. And that's one thing that I love about mentoring graduate students. Some people are very creative mathematically. And some people are creative in terms of finding problems. And some people are creative in seeing connections between different kinds of mathematics. And some people understand the connections between mathematics and other fields of science."

Because she puts so much value on creativity and individual talents, Shafi's students don't all follow in her footsteps or work closely with her on her current area of research. She has collaborated with several of her students on research projects, but with others she has provided more hands-off guidance. Guy Rothblum observed the range of research that has been produced by her students. "You look at her students and every student has done something different. She's an extraordinary mentor in that way—she teaches you a lot about how to think. And

she's fearless about what kinds of problems to approach. It's a type of environment where any problem is fair game if you're intellectually curious. Shafi was really good at guiding, but also letting students determine how to follow their own taste, their own curiosity."

Guy says that Shafi has a natural talent for seeking out the right problems to work on. "Shafi has this thing that can't really be taught—but it seems to rub off on some of her students—which is having this intuition or taste for problems. Just intuitively knowing what's a good problem to think about and being able to make a connection between two different areas."

The field of cryptography has developed beyond what Shafi might have imagined in the early 1980s, and she is now considering questions that were not on anyone's mind, perhaps even ten years ago. For example, how can a society balance the power of big data to create solutions that improve people's lives with the threats to personal privacy that can come from the use of that data? And should governments or law enforcement be able to override encryption protections in name of law and order?

The ethical questions may remain for a long time to come, but according to Shafi some technical solutions in the area of privacy already exist, they just need to become more widely available. She told the BBVA Foundation that she feels it is vital that people learn to value their personal data, and stop giving them away for free. She believes that using the cryptographic tools that are available today, privacy and security are compatible concepts. "We have effective cryptographic methods that are still not being used," she says, encouraging IT firms to "do more to build systems to make use of the beautiful ideas we have come up with in the cryptographic field that have never been implemented."

1.6 The Traveling Professor

While Shafi was very happy to be part of the faculty at MIT, she still had a strong attachment to Israel and her parents were still living there. In 1987 she became a visiting professor at Hebrew University. It was an opportunity to spend time back in the country of her childhood. Her stay there would also have a profound influence on the rest of her life because it was during this visit that she met the man who would become her husband: fellow computer scientist Nir Shavit. From that point on, Shafi would have a foot in two worlds.

The added complexity of her renewed attachment to Israel didn't slow down Shafi's progress as a respected researcher. She received the NSF (National Scientific Foundation) Presidential Young Investigator Award from 1987 to 1996, and the NSF

Award for Women in Science from 1991 to 1996. In 1993, she received her first Gödel Prize for outstanding papers in the area of theoretical computer science (which was the first one ever awarded) along with Silvio Micali and their collaborators in the field of interactive proof systems. She would win the prize again in 2001 with a group of researchers who worked on the PCP theorem in the area of complexity theory. The Gödel Prize is presented jointly by the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT) and the European Association for Theoretical Computer Science (EATCS).

When Shafi and Nir decided to get married, Shafi sought out a position at a university in Israel in order to put down more enduring roots there. She received several offers, and decided to join the prestigious Weizmann Institute in 1993. "Shimon Ullman was there and Adi Shamir and David Harel," she recalls. "Every single person there was very interesting—the top of their field—and that's still true about them. They hire the best and the brightest."

As a professor at two universities, Shafi split her time between Cambridge and Tel Aviv. Her family, which would eventually include her two children, Yonadav and Lior, moved from one place to the other every few years. Raising children is always a big adjustment, and raising them in two countries added to the challenge. "Being a parent is different than being a scientist. It's all-consuming and the well-being of these children is everything. It becomes everything. But having children and having these sort of dual homes academically meant that we lived our life in a certain way where we all spend a few years in Israel, then a few years in Boston, and then in Israel, and then in Boston. The good side was that we always went to the same place and the kids had the same schools. But the fact that it was kind of a predestined departure gave an alternative structure to our life which I think is unique. But you know what, they came out pretty good!"

It's not a lifestyle that all academics would seek out, but Shafi believes that working in two respected institutions gave her the best of both worlds in terms of her career. "Both at Weizmann and at MIT there's a very strong group for cryptography and complexity theory. There's probably more focus on complexity theory at Weizmann and maybe more focus on applications at MIT. But both places are among the best in the world," Shafi told the Heidelberg Laureate Forum.

The two universities are quite different when it comes to the overall environment. The Weizmann Institute is a more intimate campus with only graduate-level students, and a smaller number of them than MIT. (Weizmann currently has around 1000 graduate students enrolled, while MIT has almost 7000, plus an undergraduate program.) Most of the students at the Weizmann Institute are Israeli, although they do attract some students from overseas. MIT, on the other hand, is

a large, bustling campus. The students there, especially those pursuing postgraduate degrees, come from all over the world. It's a very stimulating and energetic environment, but also one that can be distracting for a researcher.

"MIT is very intense. There's a lot of people, a lot of graduate students. And there's continuous seminars and meetings, and you really feel like you're in the midst of it," Shafi explains. "Weizmann has fantastic faculty members, very, very good graduate students, but fewer. So there's more time to think, but there's less intensity, and I think that they're very different that way. So I think that after a few years in Weizmann, I was very eager to go back to MIT, and after a few years at MIT, I was very eager to rest a little bit and just kind of be able to think [at Weizmann]."

As her career has progressed, Shafi's contribution to computer science has been recognized in several ways. In 1996, she won the ACM Grace Murray Hopper Award, which is awarded to the outstanding young computer professional of the year, for her early work relating computation, randomness, knowledge, and proofs.

In 2006, she was named a distinguished alumna at the University of California, Berkeley. Shafi's position as a woman in a leading computer science role was recognized by ACM's Committee on Women in Computing with their Athena Lecturer Award in 2008.

In 2010, Shafi received the Benjamin Franklin Medal from the Franklin Institute. The Franklin Institute's Awards date back to 1824 and provide public recognition and encouragement of excellence in science and technology. As one of their honorees, she is in the company of some of the biggest names in science, including Nikola Tesla, Pierre and Marie Curie, Albert Einstein, Jane Goodall, and Stephen Hawking.

Shafi was one of the final recipients of the IEEE (Institute of Electrical and Electronics Engineers) Emanuel R. Piore Award for outstanding contributions in the field of information processing in relation to computer science. The award was established in 1977 and discontinued in 2012; Shafi received the award in 2011.

Shafi and her longtime friend and collaborator Silvio Micali won the ACM Turing Award in 2012 for their work together. She joins a very exclusive list: Only three women have received the Turing Award in its history of more than 50 years. The award is considered to be the pinnacle of achievement in the field of computer science, and she told Stephen Ibaraki that her peers were very supportive when they found out about it. "The reaction from my colleagues was really overwhelming. As soon as it was announced I heard from people who were graduate students with me at the time—many, many years ago—my own ex-students, my colleagues around the world, my friends, and everyone was extremely well-wishing. It seemed like they were truly happy for us. It's a wonderful feeling." This recognition also confirmed

to Shafi that her wide-ranging approach to research can pay off. "I certainly will feel more confident giving my students the kind of advice I've always given them: that they should work on risky projects, they should work on more general problems rather than trying to solve a specific problem that was posed by other people. Because in a sense that's what the award is for—that's the kind of work we did that has been awarded."

Her awards are not just recognition of her past achievements; they also help Shafi to establish her authority as a researcher. "I'm not sure that they have a direct impact on how I do research. I assume that they will affect my influence on the directions of computer science. Maybe in some sense it does allow one to work on more open-ended research."

All of this recognition puts Shafi in the position of many famous scientists who came before her: as someone who will inspire future generations of researchers. However, Shafi did not draw inspiration from these kinds of legendary scientists and mathematicians for her own career. She told Stephen Ibaraki, "I was inspired by my mentors, my colleagues, and my students. If you're looking for historical figures—that speaks less to me. Those who inspire me are people I was in contact with, not people I read about in books."

1.7 New Perspectives

Despite her many years of commitment to her research and her students, Shafi still sees the value in removing herself from her day-to-day routines to work in a new environment and get a fresh perspective. During the 2017–18 academic year, she was a fellow at the Radcliffe Institute for Advanced Study, which is part of Harvard University. The goal of the institute is to create an interdisciplinary, international community of 50 fellows each year across the arts, humanities, sciences, and social sciences. During her Radcliffe fellowship, Shafi turned her attention to applying encrypted computation methods to the analysis of social science data. She also took advantage of the time to work on a couple of projects that allowed her to flex her creative muscles in a completely different way: a book on pasta and protocols and a photography series on the women of New England.

Even when she's working in more familiar surroundings, Shafi finds unique approaches to expanding her way of looking at the world and to discovering new connections to explore. She explains, "A few years ago I was on sabbatical at Weizmann and I took this course about the connection between dance and science. It sounds like an unlikely connection, but there was a dance group, and a bunch of

scientists who loved them, who were meeting once a week. Each scientist would describe their research and a dance was choreographed about it."

Shafi's love of creative expression attracted her to this unconventional method of bringing scientific research to life. She even found inspiration in it for her own research. "There was a scientist whose research was about clocks and biological processes; how every cell has a clock. It intrigued me, this issue about clocks and individual cells. And after that I went on to start this research on how to model biological cells as computer cells, maybe having a little memory, maybe having an internal clock. How would they be communicating with each other?"

While biology and computer science are fields that have not had much interaction in the past, this is beginning to change. It's thanks to researchers like Shafi, who can make these creative connections between the two fields, that new areas of research are able to gain momentum.

In 2018, Shafi took her career in a new direction when she became the director of the Simons Institute for the Theory of Computing at Berkeley, returning to her alma mater (and the beautiful California campus) after more than three decades away.

The Simons Institute was founded in 2012 as a venue for collaborative research in theoretical computer science. Its founding director was Richard Karp, one of Shafi's former professors from her graduate school days at Berkeley. She felt compelled to move into this new leadership role that would put her wide experience in the field to use by making her a guiding force in the current and future directions of theoretical computer science research.

"I want to have impact, and the kind of impact that I'm talking about now is impact as the director of the Simons Institute or someone who directs—someone who has some influence about where the field is going in the sense of what's important and what's not important. I feel I have an intuition to serve me, and also a lot of experience."

Shafi believes that theoretical computer science is a field of fundamental importance in human society at this point, on equal footing with chemistry, physics, and biology. When Shafi's appointment to the Simons Institute was announced in 2017 she told the *Berkeley News*, "Algorithms govern our computing-based world in the same way that the laws of nature govern the physical one. Their mathematical underpinnings are thus as important to modern society as the periodic table, relativity or the genome."

As she told Stephen Ibaraki, the fact that she has already accomplished so much in her field does not mean that she is planning to slow down. "I've achieved some of my life goals already, but this is not going to change my passion for science and

the kind of problems I've been working on. I still hope that I have some important work to do."

As computers become a part of more and more human endeavors and interactions, it is crucial to have researchers like Shafi involved in the ongoing evolution of computer science. Seeing the connections between computer science and other important fields of knowledge may take society in unexpected directions, or prevent potential disasters. Her ability to identify ideas that others dismiss as impossible, and to see how to make them possible, is a unique gift with the potential to contribute to the advancement of society for years, and possibly generations, to come.

Shafi's sphere of influence continues to grow, and there is no doubt that she is creating a valuable legacy in the fields of cryptography and complexity theory, and perhaps many others.

2 One Obsession at a Time: A Brief Biography of Silvio Micali

Obsession can be a debilitating problem for some people, but for Silvio Micali it's his modus operandi. "I'm a monomaniac," he explains. "I pursue one thing at a time, for a long time." In Silvio's world, a long time generally equates to about five years. That's how long he tends to spend investigating a subject and working on a problem. Of course, some problems are solved more quickly than others, but for the most part a field of research will hold his attention for about five years. Rather than delving deeper and deeper into the same topic, or finding related problems that still need resolving, Silvio prefers to walk away and find something new to obsess over. "I'm leaving behind beautiful problems that ought to be solved. And they will be solved—they are being solved—but not by me."

Silvio's tendency to become obsessed with a problem can be traced back to his childhood anxieties. The first huge, theoretical problem he tried to tackle was whether the world in which he lives exists at all, or whether it is just a construct of his mind. This question plagued him, at times making it difficult to carry on with the everyday activities and interact with people who suddenly were possibly just figments of his imagination. It's actually a condition known commonly as solipsism syndrome, which calls into question whether reality is objective or subjective. Clearly, staring into the unknown and looking for provable answers is something that has intrigued Silvio throughout his life. This was the starting point for a career spent asking, and attempting to answer, some of the biggest questions in cryptography and beyond.

Natural curiosity is a big part of any great researcher's personality. Looking back, it may seem obvious that Silvio was born to be a researcher. But pursuing a career in theoretical computer science was far from a given for a young Sicilian man who

would not even encounter his first computer until graduate school. His journey has involved a mixture of purposeful focus and serendipity. He has had the good fortune to sometimes be in the right place at the right time, and the wherewithal to realize it.

Silvio Micali the researcher is well known in the computer science community. Less well known is Silvio Micali the devoted son and big brother, the husband and father, the mentor and teacher to many, and the colleague and treasured friend to a fortunate few.

2.1 A Childhood Among the Ruins

Sicily is a large island that lies just to the west of Italy's southern "toe." It is rich in history and culture, having been alternately occupied by the Phoenicians, Greeks, Carthaginians, Romans, Byzantines, Arabs, Normans, Germans, Spaniards, French and others before uniting with Italy in the 19th century. In the 1950s, however, Sicily was economically poor and underdeveloped. In October 1954, Silvio was born in the island's largest city, Palermo. It was his father's hometown, while his mother hailed from a nearby area.

His family soon relocated to Agrigento, a town perched on a hilltop near the island's southern coast. Silvio's father, Giovanni, was a judge, following in the footsteps of his own father, who'd been a lawyer and a judge as well. Silvio's mother, Franca, was a homemaker who looked after Silvio and his sister, Aurea, who was born one-and-a-half years after Silvio.

Agrigento is best known for its historical importance as home to the "valley of the temples"—a collection of ancient Greek temple ruins. In the 1950s, there was no real industry in the town apart from some agriculture and a small but important tourist trade. The town's tourists gave Silvio a strong sense of place, as he watched people from around the world who came to his town to see the temples. "You cannot appreciate history if you're Sicilian," he says, "because you are smack in the middle of it with so many cultures all around you." The present cannot be separated from the past there.

To this day, Sicily's population reflects its diverse history. Its people are descended from the large number of different ethnicities that each dominated the island at times. This was certainly true in Agrigento, where remnants of the past could be seen everywhere. This immersion in his town's multifaceted culture helped Silvio to forge a strong identity as a Sicilian, which has remained with him even after he adopted additional identities as an Italian and eventually as an American. Agrigento's much-admired ruins also instilled in him the idea that if you create

something beautiful, it will be recognized and appreciated for a long time. He began to understand the concept of leaving a legacy.

Silvio spent his early years attending the local primary school, and fighting with his sister the way only siblings can fight. "We're very close now, but we fought like cats and dogs until we were eighteen," he says.

His education was taken very seriously from the beginning. The teachers in Agrigento were very dedicated, and with little industry in the town, education was a high priority. "It's hard for people to understand, because we now live in a society which is much more diversified in achievements. But I lived in a society where [education] was the only possible level of achievement—perhaps to study the past, or to do something cultural or scientific—there were no other venues available," Silvio says. Although the town was economically depressed, its residents had an appreciation for education and culture that he has not experienced anywhere else—not even in Rome, where he spent the later part of his youth.

A good education was considered to be not just the first step to a better life, but the most noble pursuit. Education for the sake of education, and culture for the sake of culture, were valued in Silvio's upbringing. This idea, that the pursuit of knowledge for its own sake was a noble endeavor, would certainly influence his decisions later in life. "Education is the only thing that you can always keep with you," he says. "It's completely portable. I had a sense of its intrinsic value. Your own understanding of the world, your appreciation for research, nobody can take that away from you." In a town where there was little to do, discussion and debate were favorite pastimes, and great training for anyone going into groundbreaking fields of research where new ideas and theories must be fiercely defended.

Teachers were Silvio's early role models for his future career in academia. In fact, Silvio considers some of his middle school teachers to be among the most influential mentors in his life. They instilled in him an appreciation of the past and tradition, while preparing him for the road ahead. From an early age, he was attracted to mathematics and science, and also to the idea of having a job where he could discuss scholarly, important things. He felt that would be the best job in the world! At an age where many young people are still hoping to become sports stars or superheroes, Silvio was already dreaming of a career as a researcher.

Silvio's father also played a large role in his education. Silvio describes him as a "force of nature"—an influential man who liked to philosophize and debate. Silvio, however, grew tired of philosophy and felt he should focus on something else. Following the family tradition and becoming a lawyer was not something that appealed to him, and his father did not push him in that direction. It was only when Silvio was ready to enter university that his father urged him to consider

law as a more practical choice than becoming a mathematician, fearing that his son would find himself without a job. Having no experience with academic careers and research himself, Giovanni Micali was understandably nervous about Silvio's prospects. By that time, however, Silvio has fallen in love with mathematics and the legal profession was simply not an option for him.

While his mother was not as outwardly forceful as his father, she was at least as influential in Silvio's upbringing. Perhaps embodying a stereotype of Italian mothers, she held "heroic" expectations of her children. So while his father set a high bar in terms of educational achievement, his mother set a far higher bar, "in another dimension" according to Silvio, in terms of what her children should achieve in their lives. Since no child wants to disappoint his mother, Silvio took these expectations to heart, realistic or not, and felt compelled to do something great with his life.

It wasn't all work and no play during Silvio's childhood. During the summer breaks, he would spend his time playing sports and doing other outdoor activities. But during the school year his attention was on his schoolwork, and so he didn't really mix his summer activities into the rest of the year. Work time and play time were kept separate from an early age, perhaps indicating that Silvio was already developing the intense focus that would eventually make him such a dedicated researcher.

2.2 Rome: The World as a Museum

At the age of twelve, Silvio was thrown into an entirely new environment. His father was transferred to Rome to work for the *Tribunale di Rome*. He would be moved up later to Rome's Court of Appeals before eventually being offered a position at the *Corte Suprema di Cassazione* (Supreme Court of Cassation), Italy's highest court.

For a young boy from a Sicilian town with a population of around 50,000, Rome was another world—a large, cosmopolitan city where school was no longer just down the road and the sheer size of the city and the number of people were almost incomprehensible to him. The family lived in an area called Nuovo Salario, north of the city center.

Once he was in Rome, Silvio began to appreciate the legacy of the Romans in a way he hadn't in Sicily. Agrigento's ruins are mainly Greek, so although the Romans had ruled Sicily for a time, they hadn't left their mark on Silvio's childhood the way the Greeks had. In addition to adapting to a new, larger city, he was absorbing the new culture and history that it represented.

Silvio began attending high school in Rome, which involved traveling into the city center. High schools in Italy at that time took a classical approach to education. Students did not specialize according to their interests or abilities; they all focused on the same basic areas of study to give them the fundamentals of learning. "If you wanted to become a scientist, that's fine," Silvio says, "but you started by studying Latin, Greek, philosophy, history . . . and a *little* bit of Euclidean geometry." Despite the lack of focus on science and mathematics, that bit of continued exposure was enough to keep the flame lit under Silvio in his pursuit of a career in math and science.

Learning about the classics wasn't limited to the classroom. Living in Rome, the classics were all around him on the streets—in the architecture, the ruins, the fountains. "You're walking down the street and you see the place where Galileo's trial was held," he recalls. "It was really a fabulous time and it had a very, very big impact on me."

Silvio's father was also a big influence when it came to appreciating the historical treasures of Rome. "My father was an absolute maniac when it came to museums," he says. "I was fourteen or fifteen and we were living in Rome—all of a sudden it's like you're a kid in a candy store. There are museums everywhere!"

Every Sunday the pair would wake up early in order to be the first visitors to arrive at their museum of choice. They loved having the museum to themselves for a little while, when most people lingered in bed. They would stay all day, forgoing lunch (since museum cafes and snack shops were not common at the time), and clutching a book with descriptions of each piece in the collection. "Painting, painting, painting—stone, stone, stone—statue, statue, statue, until your legs would crumble under you," he recalls. At the end of the day they would leave exhausted.

This activity continued even beyond the city's many museums as Silvio's seemingly insatiable appetite for art and history drew him into Rome's churches to see famous works on display in their original settings. Masterpieces that appeared in his art history books could be seen up close. He loved the idea of touring the city on foot and enjoying art where it was meant to be displayed.

At the same time, Silvio was cultivating his mental capacity by trying to come up with theories. The fact that he did not have the necessary data or skills to prove any of his theories at the time was not an issue for him. Influenced by the study of classical philosophers, he would devote much of his time to an attempt to extract meaning from things and answer the most basic questions about life. "These basic questions never left me," he says.

When he was introduced to basic proofs for geometric concepts like the congruence of triangles in his classes at school, he would wonder how the proof was created. He was very intrigued by statements like "two triangles are equal." What does that mean? How can you deduce more by starting with less? It seemed almost magical to him. He felt like, as with the conservation of energy, you could not create something (a meaning) that wasn't already there. "Questions like 'What is a proof?' and these sorts of things fascinated me even when I was very young." It would be a while before Silvio had the mathematical and intellectual tools to start properly addressing these problems, but his awareness of them can be traced back to his teen years, when he began to question things that other students were willing to accept as axiomatic.

This questioning of the world around him could be quite challenging for Silvio's teachers. Now and then he would present them with some incomprehensible pages outlining his physical theory of the universe or other equally ambitious theories. His teachers were very patient and took his ambitions seriously, rather than dismissing them out of hand. It seems they had the foresight to understand that encouraging a curious young mind, even if he was attempting things far beyond his current abilities, was worthwhile and could build his confidence. If his early theories had been mocked or dismissed by his teachers, it is possible that Silvio would have become more cautious about attempting to answer those big questions he loves so much, and that have been such an important part of his career.

2.3 Preparing for a Nobel Prize . . . Or Not

When it came time to begin his college education, Silvio was already quite sure that his future would involve research. Silvio chose to attend La Sapienza University of Rome. It was the early 1970s, and at the time it was common for Italian students to attend a nearby university (if they lived in a major center) and continue living with their parents. The campus culture of the United States had not caught on in Europe, so the fully immersive student lifestyle that some of his future colleagues were already experiencing at that time was unknown to Silvio. Because he was still living in the family home, it was easy for Silvio to devote his time completely to study and not have to worry about the burdensome details of living on his own, like making his own food, doing laundry, or paying rent.

Despite his strong interest in mathematics, Silvio enrolled in the college's physics program. Why physics? He was swayed by the fact that there was a Nobel Prize for physics but none for mathematics. An ironic line of reasoning for a man who would eventually enter the field of computer science—in which it is also impossible

to win a Nobel Prize—and who would go on to win a number of prestigious prizes in his field. These would include the Turing Award, which has earned the nickname "the Nobel Prize of computer science."

Silvio experienced his first research-oriented courses at the age of nineteen. His classical high school education had perhaps left him behind some of his peers in terms of this type of study. He was starting from scratch on his understanding of advanced mathematics. But his enthusiasm for the subject made up for his lack of knowledge, and he felt strongly that this was what he wanted to do with his life. He wanted to become an academic and do theoretical research.

Although he was enrolled as a physics student, the Italian universities had discovered that the students emerging from the typical high schools of the time, where only the basics of mathematics were taught, were ill-equipped to understand advanced physics; they simply couldn't follow the calculations involved. So during his first semester at college, before his physics education began in earnest, Silvio spent his time learning about calculus and geometry. The following semester they began to learn physics. This was a departure from the normal course structure at the university, which generally involved full-year courses.

What was meant to be preparatory work for Silvio's study of physics ended up changing the direction of his education. "After six months of learning about mathematics I began to think, who cares about physics?" he says. He had decided that mathematics was a more interesting field and there was no looking back. Silvio changed his course of study and pursued a degree in mathematics, abandoning his dream of one day winning a Nobel Prize in physics.

Once he was learning math a higher level, Silvio was better equipped to explore some of the questions that had begun to dog him during high school. This first emerged when he began to study calculus during that fateful first semester of his degree. The rigorous reasoning and impressive information architecture left an impression on him.

Silvio considers his first calculus teacher, Luciano De Vito, a real gift and a big influence on his love of mathematics. He did all of his teaching using problems. He would come up with a sequence of problems that would push his students to reconstruct the definitions to use in the theorems, helping them to arrive at the definitions themselves rather than just presenting them to the students. This was much more work than simply learning the curriculum out of a textbook.

For some of his classmates, studying calculus was simply a means to an end, and they may not have valued a deeper understanding of it. But for someone with an inclination toward research, this type of learning was quite inspiring. It was a totally different approach to teaching than Silvio had experienced in the past,

and it gave him a much greater appreciation for the subject. It also helped him to see how having the right tools could help him to prove what he wanted to prove. Since Silvio already had a long history of exploring difficult questions, it was a revelation to understand that through his education he could obtain the tools to finally start discovering the answers. Silvio would later use a similar approach to teaching course material to his own graduate students.

During his mathematical studies, Silvio anticipated using the education he received in this area to undertake mathematical analysis. Professor De Vito was disappointed that Silvio had abandoned the idea of pursuing a career in physics, but felt that if was determined to pursue mathematics instead, the most interesting field was theoretical computer science. He thought that analysis was not a good pursuit for someone young and ambitious, despite the fact that he was a mathematical analyst himself. De Vito told Silvio about the work of Alan Turing and others, and as far as he was concerned, this was the only worthwhile area of research in mathematics.

At first, Silvio rejected this advice and pursued his interest in mathematical analysis. But in his final year of undergraduate study, he got his first glimpse into his future. He took a course in logic and one in lambda calculus, which functions as a kind of abstract programming, with Professor Corrado Böhm. Silvio enjoyed these courses more than many of the others he'd taken to date.

After those courses, he was convinced that he should pursue further studies in computer science. For his undergraduate thesis, he worked under Professor Böhm, who was one of the fathers of computer science in Italy and whom Silvio credits with "discovering" him and seeing his potential. He gave Silvio a lot of encouragement to pursue his interest in computer science, and they two wrote an article together, marking Silvio's first academic publication. Böhm advised Silvio to leave Italy for his graduate studies. In the 1970s there was no Ph.D. program in computer science available in Italy.

The Italian college system was quite different from the system he later experienced in the United States. Part of this contrast was due to the difference in campus culture. Many American students lived on campus in student residences or fraternities, while in Italy most students continued to live with their families. But the college itself took a different approach—less structured than the American system. Most courses were year-long courses, four per year, and exams could be taken whenever he felt he was ready for them. He didn't have to deal with the pressure of exam week or sitting in a lecture hall full of his classmates while everyone wrote the exam together. Instead, he could prepare at his own pace, and spread out his exams over several months if he wanted to. For someone who prefers to focus on one problem at a time, this was an ideal arrangement. Sometimes he would learn new concepts

very quickly and progress through the curriculum at a fast pace once he was "on the scent" of things. In other situations, he'd have the opportunity to kick ideas around for longer and to make sure he was confident about his understanding of a problem before moving on. For Silvio, the absence of time pressure was fundamental to absorbing material and to thinking deeply about a topic. Throughout his education and career, he has continued to consider unstructured time an absolute necessity.

Now that he has the benefit of his experience with students at MIT, he can see that the Italian system suited him very well. The American classes involve assigning problem sets one after the other, each of which is graded, and culminating in a final exam. For students who fall behind at the beginning, it can be very difficult to catch up. The more self-paced Italian system allowed him to study at his own natural rhythm. The notion of learning on a schedule was quite foreign to him when he eventually arrived in the United States, and quite terrifying as well.

Silvio completed his undergraduate program in 1978. Despite his enthusiasm to continue his studies, Silvio's timing was off. He missed the application deadline for the following year's academic intake and found himself forced to wait.

To keep himself busy and productive, he took a course in computer science over the summer. During this course, Silvio studied under another wonderful mentor who introduced him to the use of algorithms. Silvio had never seen an algorithm before (or a computer, for that matter). The teacher was Shimon Even, who hailed from Israel. Silvio became fascinated with algorithms, which cemented his decision to do a Ph.D. in theoretical computer science. Shimon Even would later refer to Silvio as "the brightest student I ever met."

If he'd wanted to remain in Italy and become an academic there, Silvio would have entered a system with three levels: He would have started as a researcher, then become an assistant professor, and eventually reached the level of full professor. By this point he had a couple of publications on lambda calculus under his belt and was considered a bit of an expert in this area. He did, in fact, take up a researcher position for a while after completing his undergraduate degree. But this did not seem like the right path for him, and he ended up resigning his position before long in order to become a student again. He would leave behind lambda calculus and forge ahead with algorithms.

Silvio applied to the University of California at Berkeley, and was accepted conditionally. His English was not at an acceptable level, and he needed to raise his score on the TOEFL (test of English as a foreign language) before he could be admitted to the Ph.D. program. The test was only administered once every six months, so if he failed again it would lead to another long delay in his plans. This

was a challenge quite different from those Silvio was used to. He is not a natural when it comes to languages. He studied French in middle school, and was able to make himself understood in that language years later when he presented one of his lambda calculus papers at a conference in France. However, French is closely related to Italian. English is a different story.

Silvio was not starting from scratch when it came to learning English. His father had had the foresight to decide that his children should learn English, long before it was considered the "universal language." They attended an English school in Rome to learn the language. It was far from where they lived and required a one-hour bus trip in each direction to attend the one-hour lesson. But half a day of sacrifice seemed worthwhile to Mr. Micali, who wanted to give his children the opportunity to embrace the world beyond Italy. However, it was not enough to prepare Silvio for graduate studies in the United States, and he had to work hard on this English in order to improve his test scores.

2.4 California, Here I Come!

Moving to America was a big decision, but it was one that Silvio was ready for. He'd always planned on leaving Italy at some point. Not because he disliked Italy, or wanted to leave it behind—in fact he still considers himself to be a "quintessential Italian" who loves his home country and his culture, and he has always visited regularly. He felt strongly, however, that leaving behind what he knew was essential to being innovative. He had to shake off the past in order to move forward.

He felt that this was especially important for someone who grew up in Italy, which is so steeped in history, tradition, and culture. The responsibility for preserving the past, this great history of Western civilization, weighs heavily on the Italian population—especially in Rome. Silvio likens it to living in a house full of expensive and delicate artwork. You aren't allowed to run around because you might break something. You have to show great respect for what is around you, and so your freedom is limited. It's easy to fall in with a uniform way of thinking in such an environment, which makes it difficult to do something truly new. "You cannot be disruptive and respectful at the same time," he says. Great research, Silvio believes, is disruptive. He confesses that his literary hero was Ulysses. He explores the world, taking years to return to his family. Silvio needed to find his own path, and to go on his own heroic journey of sorts, so that he could move beyond his roots.

Silvio arrived in Berkeley in March 1979. Despite his improved English scores on the TOEFL, he soon discovered that his language skills were going to make life in America difficult for him. Landing at the airport in San Francisco, he tried

asking people how to find the shuttle that would take him to Berkeley. Nobody could understand what he was trying to say.

He felt completely isolated in his new home. The language barrier was a major setback, but that was only part of the problem. Compared with the American students, Silvio had no real background in computer science. His undergraduate work had been in mathematics, while his American peers had been learning about the basics of computer science. Silvio's experience was limited to the one summer course he'd taken, so he found that he did not have the prerequisites for the courses he wanted to take. In fact, he found himself enrolled in "CS1"—the introductory course in computer science. Silvio, at 24 years old, was surrounded by 18-year-old students, and some as young as 16, with whom he had little in common. With no friends and a limited ability to have conversations in English, Silvio had practically no social life outside of his lectures. Not long after embarking on his great adventure, Silvio felt ready to pack up and go home. He spent a lot of time coming up with reasons that he should leave, convincing himself that this wasn't what he was meant to be doing. He wanted to create a narrative that justified his decision without admitting that he was doing it for purely emotional reasons.

When it looked like Silvio was ready to give up on Berkeley, it took just one person to change his mind, and to redirect the remainder of his education and possibly his career. He met a graduate student named David Lichtenstein, who was almost finished his Ph.D., and who was much closer in age to Silvio than his classmates in CS1. His new friend started to give him the helpful advice he needed in order to break out of his downward spiral and take control of his situation. The two remained friends for many years after David completed his doctorate.

David's first recommendation to Silvio was to forget about the rules and enroll in advanced courses even though he didn't have the prerequisites. He figured it was better to beg forgiveness later than to ask for permission. This one piece of advice changed everything for Silvio. When he returned to Berkeley after the summer break, he took courses with his graduate-level peers and began to make friends who had similar interests. It was the turning point in what could otherwise have been a very short career in computer science.

During that challenging first year at Berkeley, Silvio was on a fellowship provided by the Italian National Council of Research. He was given in advance half the money for the year; the other half he would receive in travelers checks that he collected at the Italian consulate, the preferred method in the days before electronic money transfers.

It was Silvio's first time living away from home, and so it was the first time he'd had to pay his own way and budget for himself. It was also the first time he had

to eat on his own, without his mother's delicious cooking. He hated the food at the university cafeteria, calling it "cruel and unusual punishment," and decided he could not eat it. Instead he found a local restaurant that served very good food and proceeded to eat there practically every night.

Soon enough, he ran out of money, having spent far too much on food. Rather than return to the school cafeteria, Silvio decided to try cooking for himself. Although he had sometimes helped his mother out in the kitchen, he hadn't really absorbed what she was doing in detail. He could remember many of the dishes she'd made but was unsure how to cook them, so he was forced to experiment. He made occasional (expensive) long distance calls home, sometimes waking his mother in the middle of the night, to find out what ingredients went into his favorite recipes. Even with her help, cooking Sicilian specialties was not easy since many of the ingredients were simply not available in California in the late 1970s.

Necessity became the mother of invention as Silvio struggled to creatively substitute ingredients in the recipes from home. Even mozzarella cheese, a staple in today's American supermarkets, was not to be found. In the end, the recipes ended up being his own, since if you keep substituting one ingredient after another you eventually end up with a different dish altogether. His substitutions were hit and miss, and some of Silvio's friends were subject to failed experiments where the recipes didn't turn out as planned. But he developed a love of cooking, believing it to be a "great aggregator" to bring his friends together. Silvio still enjoys having people over to share a home-cooked meal. He also still believes that there's no reason to eat badly, food being one of the necessities of life. Later on, Silvio's parties would become a highlight for his friends and colleagues, and his cooking always featured prominently.

It was also at Berkeley that Silvio finally had his first interaction with a computer. Even then, he never actually saw the computer. After all, this was before the era of desktop computing. He was able to use a console with a keyboard and a monitor, but the computer itself was located in another part of the building, hardwired to his console and those of other users. The one computer had to be shared among the students and faculty at Berkeley.

2.5 The "Perfect Storm" of Cryptography

After spending the summer in Italy with his family, Silvio returned to Berkeley in the fall and began to take the research courses he'd been missing out on during his first year. He took an algorithm course under Professor Richard Karp, and along

with Vijay Vazirani, who was a classmate, he developed an algorithm for maximum matching. This would be presented at the 21st Foundations of Computer Science (FOCS) conference in 1980. And so, in less than a year, he had evolved from taking introductory computer science classes to doing research that he would present in front of his academic peers in the field.

Vijay also would go on to a highly respected career as a researcher in the design of algorithms, as well as computational complexity theory, cryptography, and algorithmic game theory. The two would spend a lot of time together during their studies at Berkeley, along with a tight-knit group of their peers that included Shafi Goldwasser and Mike Luby. He also got to know Michael Sipser, who would eventually become the Dean of Science at MIT. It was an exciting and inspiring time for Silvio as he realized that he had found his "tribe" and was not alone in his love of research and his fascination with mathematics. The group became friends and supported one another. It finally seemed like enrolling at Berkeley had been the right choice after all. Any doubts about his choice disappeared during his second year on campus.

Silvio and his friends were a diverse group who had come together through a love of problem solving and a keen interest in computer science. Silvio had come from Italy, of course, while Shafi was from Israel, Vijay from India, and Mike from the United States. They all had different experiences and perspectives to share. There was much to discuss about each other's backgrounds and views on the world, but in the end most conversations would eventually steer themselves toward computer science, such was their enthusiasm for the subject.

This led to an atmosphere of complete immersion in computer science for Silvio. Whether he was attending lectures, doing research, or just hanging out with his friends, his whole life revolved around computer science and the problems that fascinated this group of budding researchers. They would strategize about the direction their careers should take, what problems they ought to work on, and what fields would be the most rewarding. Like students in every field of study, they were anxious about the future and concerned about making the right choices. They felt, even thought they were young and still learning, that they had important things to say, and that they carried a big responsibility.

Perhaps the atmosphere at Berkeley exacerbated these feelings of responsibility. Their studies during the early 1980s took place not long after the tumultuous student protest movements that were triggered by the war in Vietnam, the civil rights movement, and the free speech movement. Berkeley had been at the center of American counterculture and social reform during the 1960s and 70s. Although less famous, there were protests at Berkeley in response to earlier political issues as

well. In 1950, students rallied in support of their professors, who were being forced to take a McCarthy-era anti-communist loyalty oath.

The students of Berkeley had always felt that they could make a difference in the world and should speak out about injustice. It was a legacy of youth empowerment that would have been palpable to Silvio and his classmates. Perhaps computer science was not a political hot topic at the time, but since the field was still young and establishing itself, the sense of being responsible for its future direction was not misplaced. But the excitement of the seemingly endless possibilities in the field outweighed any anxiety on Silvio's part. In fact, he believes that if you're not a bit anxious about your decisions, you're not pushing yourself hard enough. Beginning with his early research at Berkeley, Silvio has always done his best work outside his comfort zone.

One of the courses Silvio took at Berkeley was in computational number theory, taught by Professor Manuel Blum. The course included a few lectures on cryptography, since it was not yet offered as a course of its own. Silvio was fascinated with cryptography right away. For him, it created a "perfect storm" because he had discovered an emerging field where there were a lot of things still to be done, and at Berkeley nobody had really embraced this area yet. There was a need for notions, for definitions, for tools. It fed his desire to theorize about a field and to start things from scratch, rather than simply applying the work of others. At last he found the field of study he'd been searching for. Blum would become Silvio's thesis advisor, supervising his work on cryptographically strong pseudorandom generation.

Manuel Blum's lectures were also the inspiration for Silvio and Shafi Goldwasser to begin collaborating on research projects. They were both very interested in a problem that Blum had described to the class: How do you toss a coin over the phone? The two found themselves compelled to solve this problem—obsessed, as Silvio would say. They would eventually move on from flipping coins over the phone to playing mental poker.

Although it was the first problem they explored together, the coin problem would be far from the last. The pair have worked together on and off for decades, and since they are both professors at the same university, they have the opportunity to discuss their research with one another even when they are not collaborating. Silvio enjoys the fact that Shafi is unpredictable and can approach a problem from very different, perhaps even conflicting, points of view.

In those early days, when doing something completely new was a big risk, teamwork was essential to Silvio. "You need the companionship and, particularly if you want to do something unusual, somebody else must believe in it too. It was very important to have her on my side."

The pair's paper "Probabilistic Encryption" was presented at a STOC (Symposium on Theory of Computing) conference in 1982, just as their time at Berkeley was coming to an end. It would prove to be a landmark paper in the field of cryptography. The Association for Computing Machinery (ACM) describes it as "one of the most influential papers in the history of computer science. It set the foundations on which thousands of researchers base their work."

The friendship that was born at Berkeley has stood the test of time. In fact, when Silvio got the call that he and Shafi were to receive the 2012 ACM Turing award, the two already had plans for their families to spend the day together skiing.

"She is my best friend, and that's really a gift," he says. The fact that they have a shared interest in the same field of research has contributed to their friendship, because it can be difficult to find friends who can truly understand you, but Shafi has an in-depth understanding of Silvio's work as well as supporting him as a friend.

The field of computer science was exciting to Silvio during his studies because it was so nascent at the time, and there seemed to be so much fundamental work to be accomplished. To advance science, he felt, one needs a portfolio of different approaches and different people putting ideas forward. Silvio admits that it is very difficult to do great research in a field that looks like a desert, "with no structure and everything looking the same in every direction." It's hard to know where to go. And yet this was exactly the type of research landscape that appealed to him—he aspired to tame the desert and unearth the structure on which future researchers could build their innovations. Anything less would not keep his interest.

Although Silvio's time at Berkeley was very focused on computer science once he surrounded himself with his fellow graduate students, the summers were a time to completely disconnect from this intense focus and return to his family and his Italian home. This break was much needed in order to refresh his mind and allow him to go back and be innovative. In his three months of leisure time, he would not just relax, but also have a chance to mull over what to work on next.

Silvio spent every summer break in Italy, partly in Rome, and partly at a flat that his parents rented on a quiet beach the Agrigento region of Sicily, in a town called Siculiana Marina. There were miles of protected beach along the harbor, in pristine natural condition. During the winter, the tiny fishing village was home to only twenty people or so; in the summer it was a little busier, but far from crowded. It was a perfect place to get away from it all, and certainly a contrast to the hot, busy streets of Rome. His father had a small fishing boat, and for one month of the year this respected judge would transform into an avid fisherman. Silvio would wake up long before dawn to help him set the nets. Silvio remembers that when he was younger these duties would keep him from socializing with the other kids

his age, since he'd have to go to bed early just as the others were heading out to have fun. As a graduate student who lived overseas for most of the year, he was able to appreciate this precious time with his father. As his father grew older and Silvio was rarely around to help, Mr. Micali eventually gave up fishing because it was too dangerous to go out alone. Instead, he took up mushroom hunting in the Alps as a new obsession. But Silvio still visits Sicily once in a while to see a childhood friend who owns a farm near Palermo.

Siculiana Marina was certainly a world away from the academic world, and this unstructured time was indispensable to Silvio. Perhaps because of his Italian upbringing, he maintained this very "European" attitude toward vacations while his American peers and professors seemed to feel more pressure to put their summers to productive use. But Silvio did sometimes convince a friend or two to join him in Italy. Silvio maintained these three-month breaks for as long as he could, until the demands of his work, and his assimilation into the American schedule, gradually reduced the time he was able to get away. Nonetheless, Silvio continues to visit Italy regularly, twice per year if he can, to see his parents. Both of Silvio's parents are still enjoying life in their nineties, but they are no longer able to make the journey to the United States for visits, so it's up to Silvio and his family to make the trip to Italy.

2.6 I Have a Ph.D., Now What?

After completing his Ph.D. in 1982, Silvio found himself heading to another new country—Canada. He decided to do a post-doctoral fellowship at the University of Toronto. The university had a strong theory group at that time, and Silvio would find new mentors in Steve Cook, Charlie Rackoff, and Allan Borodin. Silvio had already met both Steve and Allan, and it was Allan who had invited Silvio to pay them a visit in Toronto and convinced him to join their research group. It turned out to be a momentous decision for Silvio, and one that would have a great influence on his career. He began working with Charlie Rackoff, "a first-class researcher—very creative and also very obsessed about definitions," Silvio recalls. The two would go on to work together on many research collaborations over the years.

Thanks to the encouragement he received from the members of this group, Silvio felt that the environment was perfect for someone like him who was just beginning his career, and therefore had reason to be a bit nervous about putting forward his theories. "If you really want to do something that is always at the point of failure, you need support all the time," he says. In Toronto, he found the kind

of kinship and support that enabled him to undertake risky and uncertain work that would attempt to break new ground. The more he pushed himself, the more important it became to have this kind of support. From Steve Cook especially, Silvio found intellectual support in addition to emotional support.

This intellectual support was invaluable in boosting Silvio's confidence to pursue his ideas. He had so much regard for Steve's judgment that if Steve thought he was onto something, he felt he could move forward with much less worry. The result of all this was Silvio's early work on zero-knowledge proofs, including his explorations of how to define a proof. Once again, it appeared that Silvio had found himself in the right place, surrounded by the right people. While he considered Manuel Blum to have been very influential when he undertook his first forays into encryption at Berkeley, when Silvio moved onto this next phase of his research, Steve Cook and Charlie Rackoff proved to be equally influential.

Intellectual support has been a key factor in Silvio's success, but it was not always a given. Any researcher looking to push the boundaries of his field will meet with resistance, and Silvio was no exception. He calls the rejection of his theories "devastating," but at the same time rejection can be the impetus to commit to a high standard of research and fight for what he believes to be true. In fact, he feels that if a theorem is worth proving, it should be difficult to convince people of its importance. Otherwise, you are dealing with a widely accepted concept already, not something truly innovative. Silvio's paper on zero-knowledge proofs, for example, was rejected several times. But in the end, this made it a more thorough paper. If you struggle because your peers don't agree with you when you first argue your theory, you need to have the stamina to keep yourself on target until you have convinced them.

According to Silvio, great research requires the conviction to keep on your path when everyone else seems to be heading in the opposite direction. When it comes to being a researcher, Silvio believes that being stubborn is a prerequisite, but if you are too stubborn you can end up committed to something that ends up being wrong—it's a delicate balance. There were times when Silvio feared that he had taken an incorrect path in his research, and he had to make contingency plans in case his theory turned out to be incorrect or he was unable to solve the problem he had decided to tackle. He has had several occasions where he was close to admitting defeat, but has been able (and stubborn enough) to keep trying until he worked through the problem that was holding him back.

Along with being stubborn, Silvio is extremely focused when he's involved in a research project. He's the sort of researcher who will work day and night on an

interesting problem unless someone is there to make him stop. And even when he is compelled to put his work on hold by family or friends, it remains at the back of his mind while he is doing other things. He finds it very hard to step back from a problem he's trying to solve.

This is a tendency that was easier to indulge in his early days as a graduate student, and as a young researcher at the beginning of his career at the Massachusetts Institute of Technology (MIT). However, it became harder and harder as his life filled with other priorities, like a wife and children.

And MIT was in fact Silvio's next stop. In 1983, after his year in Toronto, a position opened up at MIT, where his friend Shafi Goldwasser was already doing post-doctoral work. It was a bold strategic move on the part of MIT. They already employed cryptographer Ronald Rivest, and decided to hire on Shafi, who was also doing research in cryptography. Creating an additional position for Silvio may have seemed like a large commitment to what was a relatively minor field of computer science—at the time, there were almost no cryptography courses offered at any university other than MIT, and to some degree at Berkeley. But MIT was willing to bet on cryptography becoming an important field, and they were setting their university up to be the leader in this research area. Little did they know that it would be essential to the security of the internet one day, making possible many of the online activities that people now take for granted.

When Silvio was offered the position, he took a road trip with his mother to make his way to his new home. Mrs. Micali had made the trip from Italy to Toronto (Mr. Micali was not able to get away from his work at the time) and the two traveled through Quebec and Maine, eventually arriving in Cambridge. It was a rare opportunity to spend time together without the distractions of work or the other members of the family.

2.7 Professor Micali of MIT

On arrival at MIT in 1983, Silvio became part of their growing cryptography group. At that time, fellow cryptographer Oded Goldreich also arrived to do postdoctoral work, and he would remain there until 1986. Oded had already been introduced to Silvio and Shafi's work on probabilistic encryption through Richard Karp. He had immediately realized that the pair were redefining the field, and that their work would form the basis of all future work in cryptography. Oded was even more captivated once he had the opportunity to get to know Silvio personally at MIT. "He was extremely charming and outstandingly inspiring and empowering," Oded recalls. The pair have worked together many times, and they also remain good friends.

Thanks to this team of enthusiastic and dedicated researchers, the atmosphere in the cryptography group was friendly and lively. Silvio, Shafi, and Oded were all young and single, and they spent much of their time together even outside of work—dining together, going to movies, and chatting for hours. At work, Silvio and Shafi had neighboring offices, and the whole department was an "open-door" environment where discussions with a student or visitor would end up involving multiple people. Research discussions would also bounce around the offices, which made for an open and enthusiastic exchange of ideas.

In 1990, when he was well settled in his role at MIT and his life in Cambridge, Silvio's life took a new turn: Silvio met Daniela, the woman who would eventually become his wife. The two met at a party in Cambridge. She is also Italian born, but she spoke English so well that Silvio believed her to be British when they first met. In his first attempts to engage this interesting woman in conversation, he found himself tripping over his words and sounding less than impressive. Finally, after an hour of difficult conversation, she interrupted him to tell him that it was fine if he wanted to speak Italian. It was the beginning of a beautiful and enduring relationship.

Daniela is a legal scholar and law professor at Boston University. When she and Silvio first met, she had completed her master's in law at Harvard and was attending university in Florence to complete her Ph.D. This made for a very long-distance relationship for the two at first. Nonetheless, after about a year and a half they married.

Having a law professor as a daughter-in-law was, of course, welcome news to Silvio's father, the respected judge who'd reluctantly accepted that his son would not be following in his footsteps. And to add to the irony, Daniela's parents were mathematicians, so they were equally pleased to find their daughter marrying someone with a love of mathematics that she did not have herself.

The couple now have two adult sons, Stefano and Enrico. Enrico is currently studying at MIT, with a keen interest in both biology and computer science. He's tackling computer science first, which is a brave decision considering the large shoes he may be expected to fill.

The children are well acquainted with their Italian roots. Until the age of five, they spoke only Italian at home. This has enabled them to have a stronger relationship with their grandparents back in Italy, who don't speak much English. In fact, the boys have a facility with languages that Silvio finds very impressive. Apart from fluent English and Italian, they can also speak French and Spanish.

Long before he had sons to think about, Silvio had to deal with other young minds—his students. Taking on an assistant professor position (which would lead to a full professorship in 1991) meant that Silvio was responsible for teaching and

supervising both undergraduate and graduate students, along with doing his own research. For someone who prefers to focus completely on one task at a time, this was rather inconvenient. He was expected to teach one course per semester, so at least he did not have to divide his attention between multiple courses. He typically teaches undergraduates during one semester, and graduate students during the other.

Silvio believes that his strength is teaching research courses because the material is all in his head and he simply has to share it with his students. His theory is that in order to understand a topic, one has to completely exhaust all of the possible ways to misunderstand it. He therefore examines all of the detours that can be taken, and that the students would perhaps be tempted to take. He has great empathy for the students who are exploring a new subject for the first time, because it's a journey he has already taken. Rather than just feeding the students his own results and conclusions, he invites them to experience the entire process that was required for him to reach those results. They see how he changed his mind at certain points and arrived at his conclusions, and he feels that it gives them a more complete understanding of the topic. As a side effect, it may also make them feel that their own doubts and struggles are not unusual, or a sign that failure is imminent, as they undertake their own original research. Surely if their eminent professor experiences these struggles, it should not be surprising that they are going through similar struggles themselves.

Leo Reyzin was a graduate student at MIT in the 1990s, and he took Silvio's course "Cryptography and Cryptoanalysis" during his first year at the university. He recalls Silvio's lectures with great admiration. "He's very inspiring. He treats every lecture as a performance. There's drama, there's tension—every lecture has to tell a story and draw the audience in. You don't give away the plot at first, you hold the audience in suspense," Leo says. "There's the bad way to do things and the right way to do things. And he deliberately misleads you and then says, 'Aha! That's what's wrong!' and you really have to stay on top of it to follow him. He does it to keep you thinking, to keep you on your toes."

Courses on basic topics outside his field of expertise are a different matter. These are the courses that Silvio is less confident about teaching. The textbooks explain how things should be done, but they don't reflect on the genesis of the ideas behind the lessons. The subjects he feels he teaches best are the ones he has struggled with himself, because then he understands how to explain them to students who may be struggling as well. To make up for this, he prepares more for the courses that are outside his expertise. He's also a terrible procrastinator when it comes to doing this

because he's usually involved in some research at the same time that is his main focus. Yet somehow it all comes together.

Because Silvio's path to understanding a topic involves an in-depth examination of the process that was taken to reach a particular conclusion, he likes to undertake this type of examination with his students so that the knowledge becomes an integral part of their psyche—they own it. To Silvio's dismay, the course curricula are not designed with this kind of detailed analysis in mind. So Silvio struggles to get through everything he is supposed to be teaching, and would much rather cover less material in greater depth. His students are sometimes expected to cover certain topics on their own using the course books, while they learn others in impressive detail during Silvio's lectures. In a system that relies on prerequisite courses as the students progress, covering everything in the course outline is taken for granted. Silvio must find a way to make that happen and to teach at a pace that does not always come naturally.

As a supervisor, Silvio sees his students not as young minds to be molded but as research peers. He will only take on a student who has taken a research course with him, so that he has a good idea of that student's understanding of his research and the field in general. This prevents students from coming to him with preconceived ideas about his research that aren't necessarily accurate. It also helps students to self-select as people who are fond of Silvio's style and his personality.

Leo Reyzin was one of those students who wanted to work under Silvio even before he arrived at MIT. He'd seen Silvio present at a seminar and a conference and was very interested in what Silvio had to say. "He seemed creative and energetic," Leo recalls.

Since the relationship will be a close one, and will last for several years, personal and professional compatibility are important. Together Silvio and his student will find a subject that they are really interested in and then jointly "obsess" over the research. He doesn't look for topics that are specifically "suitable" for a first major research project; he expects his students to take on the same kinds of big questions and innovative research that have always attracted him. Silvio won't hand off a piece of research to a student that he doesn't have the patience to look at himself, as a sort of outsourcing project. He takes an all-or-nothing approach, and the research takes as long as it takes until they solve the problem. Because he doesn't advocate lower-level research, there's a greater chance that his students may experience failure, but Silvio feels that the lessons learned from this process are valuable and will help his students to succeed going forward. Although they may have less experience, Silvio believes his students to be just as intelligent as any colleague he works with.

Because of this fully immersive approach to working with his students, Silvio generally only supervises one student at a time. He doesn't believe that he has the capacity to properly participate in the research of more than one student. "If you are obsessed about two different things," he says, "you are not really obsessed." They work together until the thesis is almost complete, and then Silvio will begin to look at another student to begin a new round of research with. This is another reason that it's so important that the pair find a topic they are both truly excited about. For the duration of the project, this will be Silvio's only research focus. He does not undertake his own individual research at the same time.

Leo Reyzin says, "He's very much goal and project oriented. It's not like there's a weekly hour-long meeting. If we're working on something then it's very intense. Silvio is all-consumed by things. When he's consumed by something he's really consumed by it."

During their time working together, both he and Silvio were at one point each expecting a child (Leo's first and Silvio's second). Knowing that fatherhood would soon be making more demands on their time, Silvio wanted to get as much work done as possible before that happened. "He said, 'I'm about to have a kid, you're about to have a kid, let's get to work *now*!"'

Leo says that during the times the two worked on separate projects, it was a challenge to get Silvio's attention, since he'd be focused on something else. "When he had time for you, he really did. He had hours and hours and hours. But when he didn't, he didn't. The only way to communicate with him at the time was to leave physical notes on his office door. He didn't do email; calling him was pointless."

Leo recalls the long hours spent working together on a research project. "We'd pace the halls and work on the whiteboard, and when the time came to write up the results we'd actually sit at the computer together and write, which is a very rare treat—to work with someone at one keyboard and just take over who's driving. We kind of completed each other's sentences."

This type of approach fits in well with Silvio's preference for collaborative research. Collaboration has been Silvio's preferred research method since his early work with Vijay Vazirani and Shafi Goldwasser at Berkeley. Of the more than 100 papers listed on Silvio's curriculum vitae, only a handful were written alone. Whether he was working with his fellow students during his graduate school days, with professional colleagues, or with his own graduate students as a supervisor, Silvio has almost always taken a team-based approach to research.

In fact, he traces his preference for working with others all the way back to his childhood anxiety about whether the outside world really existed, or whether he

was simply imagining it. Interacting with his parents and schoolmates helped him to overcome his doubts at the time. As a researcher, his collaborators play a similar role; they help to assure him that he's not making things up and that the research they're undertaking is rooted in reality.

Silvio also feels that when you are coming up with a new theory, you're going out on a limb and could be there for quite a long time as you work to confirm it. It can be a very uncomfortable place to remain alone, and it's easy to start doubting yourself. Talking things out is an important part of the process for Silvio, and he prefers to learn about new subjects through discussion rather than by reading about them. When working with a partner (or several), there is thoughtful discussion about the theory right from the beginning, and he believes that this makes the chances of heading down the wrong path much lower. It is no accident that Silvio's Turing Award honors his collaborative work with Shafi Goldwasser.

Although the collaborative process involves a lot of mutual support, that doesn't mean that it's all about agreement. "Argument is the essence of life!" Silvio proclaims. Perhaps this is another way in which his Italian upbringing comes through in his work. He grew up with a very forceful father, who was skilled in the art of argument through his legal training. Silvio describes their arguments as "incendiary," although there was great love and respect between them. His sister Aurea is also skilled at arguing, likely for the same reasons. As a result, Silvio grew up learning that he would have to be persuasive in order to get his way, and that argument is not antithetical to friendship or respect. Instead, he feels that opposing forces and clashes of opinions are required if you want to forge something new and great. When it comes to his research, he has at times felt that opinions were so divided that perhaps he and his partner should stop collaborating, but somehow they always end up on the same page in the end.

Oded Goldreich, who has collaborated extensively with Silvio, feels that "Silvio's collaborators are presented with such forceful and beautiful arguments that they do not feel bad when arguing with him. So tension does not arise, because one is compelled by his arguments and captured by his charm. Later, one may find a flaw in Silvio's arguments, but one finds it hard to be annoyed at him even then, since the charm stays and the beauty of the arguments stays too." Oded can also attest to Silvio's steadfastness in defending his point of view. "As to changing Silvio's mind or making him do anything he does not want to do—this is definitely impossible."

Oded offers support for the idea that Silvio's facility for argument has its roots in his earlier learning about philosophy. "I think that what Silvio talks about is not arguing, but rather the articulation of views. Indeed, the articulation of views is a key ingredient in interaction with him. Silvio does not just say 'let's do X,' but rather

articulates why it is a good idea to do *X*. Silvio's articulations are always grounded in philosophical considerations, and are richly framed in a wide context."

In addition to collaborating on research, Silvio mentors his graduate students in a variety of ways. One of his specialties is teaching his students how to present their work at conferences and seminars. Silvio is well known for his compelling "performances" when he lectures, and he tries to help his students to master "the Silvio method."

Leo Reyzin explains the process he went through with Silvio before his first big presentation. "He makes you prepare your slides, and you give the talk, and by slide five I can see he's just not there, he's tuning out. I say 'This is not working, is it?' and he says 'No, it's not, do you know why?"' The student then needs to go off and figure out what's wrong, fix it, and present to Silvio again and again until all of the problems are fixed and they are finally able to get through the whole presentation. If a student needs more specific feedback about what's not working, Silvio will provide it, but he prefers to let his students find the problems themselves. "I don't know how many times we rehearsed my first talk," says Leo. "It was over and over and over and every time we'd get a little farther into it. A lot of his former students are now faculty at various universities and I know them pretty well, and they're all good presenters—so it works."

Silvio also develops a strong personal relationship with his students. "He's a wonderful mentor," says Leo. "The number of conversations we had about life and career, and balancing what one wants out of an academic and nonacademic career, and how to balance having kids and your family obligations. He was so generous with his time and advice." He was also generous in other ways. "He never let me pay for our lunch while I was a student. Until you get your Ph.D. you can't pay for lunch. I guess I owe him a lot of lunches!"

Silvio's time at MIT is divided between teaching and research, making for a full schedule. Schedules and Silvio simply do not get along well. "If you don't get bored and spend time figuring out what to do, you cannot do original work," he says. During the month of January, he has no scheduled classes to teach. This gives Silvio time to think about things more deeply, with no distractions. He thinks that this time is crucial if you want to do something different; idleness and creativity go together. Nothing happens for a while and then something clicks. For Silvio, it is necessary to have unstructured time on your hands that you can shape any way you want.

In addition to his work at MIT, Silvio's career has taken him around the globe, presenting his research at conferences and universities. It's an inevitable part of being a researcher, and the more successful one is, the more requests are made

for this type of presentation. Silvio enjoys the opportunities to talk with his peers, although the fact that he describes these pleasant talks as "confrontations" is perhaps indicative of his argumentative conversational style. Rather than attending scheduled sessions at conferences, he prefers to sit down with people to have a discussion about a topic of interest.

While Silvio is never one to back away from an argument, his interactions with his peers in his own field and others are always gracious. According to Oded Goldreich, "Silvio is very generous. One may forget this when seeing him fight for some cause or interest of his; when he is doing anything, he does it full-heartedly. But when the fight is over, he is the most generous winner one can imagine. In the rare cases that he loses, he is also graceful about it."

When he is putting together a conference presentation about his research, Silvio tries to imagine himself in the audience, because he considers himself to be the worst kind of person to present to. When he attends another researcher's presentation, he often gets lost by the second slide. A complex illustration will draw his focus, causing him to stop listening to the presenter and get completely off track. He figures if he can understand a presentation, anyone can understand it. For his own talks, he tries to distill everything down to the simplest terms, which takes a lot of time and preparation. He uses his own hand-drawn cartoons to illustrate concepts because he finds them less distracting than more complex representations. He claims that he will use any trick in the book to make things easier to understand. In a field like theoretical computer science, as many scholars and interested laypersons can attest, this is no easy task.

2.8 Kudos and Companies

Another sign of his long and distinguished research career is the number of awards and honors that Silvio has received. In 1993, work on interactive proof systems that he did with Shafi Goldwasser and Charlie Rackoff was awarded the inaugural Gödel Prize. This prize is given jointly by European Association for Theoretical Computer Science (EATCS) and the Association for Computing Machinery for outstanding papers in the area of theoretical computer science.

In 2003, Silvio was elected to the American Academy of Arts and Science's Computer Science section. The Academy's members include more than 250 Nobel Prize laureates. Silvio was also elected to the National Academy of Sciences and the National Academy of Engineering in 2007. Both of these honors illustrate the high regard in which Silvio's peers hold him. These academies only bestow membership

on leaders in their fields and to be recognized by three such organizations shows an exceptional level of achievement.

Silvio was also the winner of the RSA Conference Award in Mathematics in 2003. His 2004 paper, co-written with his student Leo Reyzin, "Physically Observable Cryptography" won the inaugural TCC Test-of-Time Award in 2015. This award is presented at the Theory of Cryptography Conference (TCC) for a paper published at TCC at least eight years earlier that made a significant contribution to the theory of cryptography, preferably with influence in other areas of cryptography, theory, and beyond. Silvio was also named Berkeley Distinguished Alumnus of the Year in 2006 by the Electrical Engineering and Computer Science department of his alma mater.

Adding to this already impressive list of achievements, Silvio received the ACM Turing Award with Shafi Goldwasser in 2012, in recognition of their "transformative work that laid the complexity-theoretic foundations for the science of cryptography, and in the process pioneered new methods for efficient verification of mathematical proofs in complexity theory."

It is true that Silvio abandoned the hope of winning a Nobel Prize one day when he changed his undergraduate major from physics to mathematics, but perhaps his collection of other prestigious prizes has made up for that loss, at least in part. While recognition from his peers is always welcome, Silvio believes that winning major awards has additional benefits, both to himself and to computer science in general. He feels that the awards bring more attention to certain fields of research that might otherwise go unnoticed in the mainstream, and they invite outside observations on the work.

This type of judgment from outside one's field, although it might make some researchers uncomfortable, is necessary according to Silvio. He thinks that there is a danger to conducting research in a "bubble" where there is no outside judgment of what you are doing. Diverse opinions provide the necessary perspective to help researchers decide which paths to pursue.

Awards also help to introduce researchers in other fields to one's work. Certainly, Silvio was well known among the cryptography community long before he won the Turing Award, but afterward, researchers in other fields learned about him and his work. It has given him an opportunity to become a sort of ambassador in his areas of expertise, to answer questions, and to facilitate connections between scholars in different areas.

Another benefit he sees in winning awards is the permission it gives him to explore new areas of research that are not currently well recognized. These awards give him the credibility he needs in order to take on more risk, because there is

an implicit recognition that he is a top researcher and unlikely to be pursuing a frivolous idea. The idea of taking bigger risks is also easier from his own perspective, because the confidence boost of a major award makes him feel more energized about doing something new and innovative.

Groundbreaking research can lead to awards, certainly, but it can also find its way into the commercial realm. Silvio and Shafi's early work in cryptography has laid the groundwork for a number of online security features that are widely used today.

While you might expect that this in-depth knowledge of online security would make Silvio extremely cautious when it comes to his own online life, he does not actually practise what he preaches. "I'm extremely suspicious as a cryptographer when I'm doing research," he explains, "but then I don't even lock my door!" He doesn't use the best practices for creating passwords or lock his front door because he's more concerned about locking himself out than keeping his property safe. The work itself, however, he takes very seriously.

The practical applications of Silvio's work have continued throughout his career. He often sets out to solve a purely theoretical question—to change the way that an entire field is viewed or approached—but in the end the solutions usually have practical implications. Silvio believes that technology transfer is crucial, whether he is involved in the transfer himself or whether it is left to others who pick up where he has left off and develop applications for his research. "Knowledge has to be transferred to society," he says. "I really believe that this is important." Silvio owns dozens of patents covering several different areas of his research.

Silvio finds that actually achieving technology transfer is a challenge. The types of technologies he develops, such as digital signatures and simultaneous electronic transactions, require a shift in the way large numbers of people do business. The benefits of the technology must be so compelling that everybody is convinced to make the change in a short period of time. He uses fax machines as an example of this type of challenge. If you thought fax technology was interesting when it first emerged and purchased a fax machine, it was useless unless everyone you wanted to send faxes to also had one. The technology could only succeed if a certain level of market penetration took place. The timing can be as important as the technology itself. Introduce your technology to society before they're ready for it, and it will be passed by. Introduce it too late, and there will be a large number of competing technologies on the market. So while the chances of coming up with the next big thing are small, Silvio believes that in some cases, the benefit to society of a technological shift are so great that it's worth the risk.

Despite these challenges, Silvio has delved into the commercial side of research. He worked with a team in 2003 to develop CoreStreet, a credential validation technology, while on sabbatical from MIT. He also worked with Ronald Rivest on a micropayment system called Peppercoin. This type of endeavor is quite different from academic research, and Silvio is well aware of his limitations. He tries to bring in the best possible developers and engineers, plus those who can handle the fundraising and other business aspects of a startup.

Among other challenges, Silvio had to present the products to venture capitalists. Accustomed to audiences of computer scientists, Silvio needed to find nontechnical ways to explain the products—a true test of Silvio's dramatic presentation style. Both of these companies were later acquired, and Silvio returned to his position at MIT. Silvio recently took another sabbatical from MIT to focus on a business called Algorand, which has created a new type of distributed ledger. Although the field is already competitive, he thinks his product is superior and is optimistic about convincing others that it is the right choice. It seems that his powers of persuasion are strong, since in early 2018 he convinced venture capitalists to invest $4 million in the company.

2.9 The Road Ahead

Now in his sixties, Silvio has no plans to take it easy. When asked about retirement, he reacts as though it was the most absurd idea he's ever heard. "Retire from what?" he asks. "From life? It makes no sense to me." He believes that if there comes a time that he can no longer indulge his research obsessions, he will find something else that he can do well and he will become obsessed with that instead. He thinks that we all have an obligation to continue to contribute in our own way for as long as we are able, and to be fully engaged in life.

Although he is certain that many more projects lie in his future, Silvio has no idea what they might be. It's part of his obsessive, in-the-moment nature that he does not make long-term plans. Whatever comes up when his current project winds down, he'll decide on a direction at that time. Retrospectively, he can see that every five years or so he tends to switch to a new project, but it's not a timeline that he plans in advance, and the next project is never intentionally lined up and waiting in the wings.

What is clear is that Silvio will continue to commit himself completely to his undertakings, to enjoy time with his friends and family, to cook his mother's wonderful recipes, and to live his life one obsession at a time.

An Interview with Shafi Goldwasser

This is a partial transcript of an interview of Shafi Goldasser by Alon Rosen. The interview took place on November 23, 2017. The transcript was lightly edited for clarity.

Rosen: Hi. My name is Alon Rosen. I am a professor of computer science at the Herzliya Interdisciplinary Center in Israel. Today is the 23rd of November 2017 and I'm here in Rehovot at the Weizmann Institute of Science together with Shafi Goldwasser, who is being interviewed as part of the ACM Turing Award Winners project.

Hi, Shafi. We are here to conduct an interview about your life, about your achievements. Generally speaking, we will go chronologically and we will talk at two levels. The first level will be a general audience type of level and the second level will be more specific, more oriented towards people that specialize in the subject and are interested in the details. So let's begin with your high school experience.

Goldwasser: Okay. Well, first of all, thank you Alon for taking this opportunity to interview me.

High school. Right. Those years I remember quite vividly. The orientation changed a bit for me from sort of being interested in the sort of more humanity subjects to more the mathematical subjects. You know, mathematics, and the sciences. I remember I loved physics. I didn't really like life sciences, but physics and math I liked quite a bit. And I had a great math teacher from eleventh and twelfth grade. Somehow I did well and I think that was part of why I wanted to do it. I also had a great teacher for physics, and physics in my mind was just fantastic. You know, things made sense, you could derive things. I think early on that's what I wanted to study.

Rosen: What about mathematics?

Goldwasser: In mathematics, again I was good at it, but mathematics itself at that time was not described as mathematics with some sort of motivation. It was more the method, you know? So taking derivatives, integrals, and it was in trigonometry and all that. And I could perform it well, but it didn't have the stories associated with them that physics did.

Rosen: So it was more about the technique and less about . . . ?

Goldwasser: About technique rather than about motivation.

Rosen: And did you already then have the sense that you missed the concepts and the . . . ?

Goldwasser: I had no idea that there were concepts, you know? All I knew was that I liked the concepts in physics. The whole derivation from principles was beautiful in my eyes. And I remember questions on the exam and then you would have to think. And I have the impression of some memory where my [laughs] answer was different than others and he was surprised, the professor. But I cannot, for the life of me, remember what the question was or what the derivation was.

Rosen: So it sort of sounds like this professor, he had an encouraging influence on you.

Goldwasser: Yes, both of them. Yes.

Rosen: Okay. How significant do you think it is to have a good professor? To have someone who influences you that early?

Goldwasser: Extremely significant. I think if you're very lucky, there is someone early on—and that could be high school, it could be maybe college, but better in high school—that awakens something in you, a spark, an interest, so that maybe later you're not going to do exactly that but you know there's something about studying and about pursuing knowledge that is exciting. I think it's fundamental, and I don't think that it has to be more than one.

I had other good teachers there, you know. The literature, I remember the teacher. The history teacher. I remember learning Shakespeare in English class. But something about . . . there was some spark there in the science classes and in the math classes that I recall.

Rosen: So by then, your self-image was sort of that you were set towards studying scientifically oriented subject?

Goldwasser: No, not at all. [laughs] I loved to write, and I think that my inner image was that I was going to be a writer. But I guess—you're right—by the time we got to the eleventh and twelfth grade, my parents, or especially my father was very kind

of insistent that I should follow the realistic . . . this is what we call in Hebrew "realistic studies," or mathematics and physics studies. Because as people of his generation, and maybe people of the current generation in Israel as well, there was a real emphasis on pragmatism and the exact sciences, and that everything else is a bit less . . . It might be enjoyable, but it's not as real as what one must do in life.

Rosen: I'd be interested to hear now about your view on the global experience of Israel at the time.

Goldwasser: Sure. Yeah, I do have the tendency to talk about the personal stuff, but it's what I know best. But let me tell you a little bit about my memories about Israel. First of all, I lived here through a few wars, right? I remember the Six-Days War [of June 1967]. I think I was in fourth grade. I remember that. And I remember we went down to the bunker. I remember the sirens. And I remember right after the war, my family and I, we drove to Jerusalem. I remember still seeing the Wailing Wall before they kind of opened up the huge square. It was somewhat of a euphoria. Who knew that this would be a "tragedy forever."

But in any case, this is fourth grade. Then I remember Yom Kippur [October 1973] War. Yom Kippur War is a different story. Then I'm already in tenth grade I think and my brother was a soldier. I remember the first phone call that he made. My father asked him how was his commander, who was someone that my father felt that was going to protect him. And he said, "He is no longer." And I remember my father just burst out crying. He was just so worried about him. Then I remember when he came back home the first time . . . I don't know how long it was really, because he stayed in the army for about six months afterward. He was supposed to be released but he stayed longer because of the war. But I remember that he had a lot less hair. He had like those two sides of his forehead, his hair receded quite a bit. It was amazing that this kind of traumatic experience can do that.

Rosen: Do you think any of this had any effect on you in the long term, on your personality, outlook?

Goldwasser: I think it had an effect on my father. I think that when my brother came back from the army, he joined the Hebrew University, because he was going to go and study mathematics, and he went right away. They postponed the semester because of all these soldiers. They started a new semester in January, like a new school year. But my father just wanted him out of Israel as fast as possible. He was so afraid for his safety that he wanted him to go to school in the States. And within a year, like the second year he just sort . . . he somehow arranged . . . he kind of

made him apply abroad. And he got accepted to Carnegie Mellon and left. That affected me because that started some sort of chain reaction in the family.

Rosen: And okay, you said your brother wanted to study mathematics. What did he end up doing and how did it all affect you?

Goldwasser: He studied mathematics as his first degree, and then he went to business school at Carnegie Mellon. It's called GSIA, Graduate School of Industrial Administration. And then he went to work.

And I, when I arrived at Carnegie Mellon, I had like a year or so before my military service, since my father wanted me to go to the U.S. to study so that I don't waste any time. This idea of wasting time is something very problematic, or was very problematic when I was growing up. Now it seems like everybody in Israel is taking trips around the world as soon as they finished the army, or before the army, and wasting time is not called "wasting time" anymore but "gaining life experience." In any case, my father wanted me to go to the States, and as usual I did what he recommended and went to Carnegie Mellon, and I went to study mathematics.

So I arrive to the U.S., it's summer 1976. I land in the U.S. and my brother comes and picks me up in New York, and we spend a few days in New York. Then we took a bus to Pittsburgh. I knew nothing about Pittsburgh. I spent the summer in the dorms waiting for the school year to start. I actually never applied to the school. Just my brother told his professors that his sister is coming for a year and she's good at math. And since he was good at math and they knew that he was a talent, they said, "Does she want to come and study here?" and he said, "Yes," and they said, "Okay." And that was it. I became an undergraduate in mathematics, in applied mathematics.

But then it was applied mathematics and computer science. Now there's a undergraduate computer science program at Carnegie Mellon. At the time, there wasn't. And the truth is that I actually loved studying. This was a revelation. When you go to high school, you sort of do what you're told, right? But I found it really interesting. I found the math interesting, I found the computer science interesting. I took this introduction class in FORTRAN programming. In the beginning, I had no idea. There were these cards where you put an instruction on every card and it goes through a machine and then it executes each instruction. I've never seen a computer before, I haven't really heard about computers before, but it was fascinating. It was really marvelous.

Rosen: Okay, I have two questions now about the admissions, you said the admissions process was unorthodox in your case?

Goldwasser: I would say. [laughs]

Rosen: Now, I want to ask what would have happened today with admissions?

Goldwasser: Ah, today. Today, no, the whole college admissions in the U.S. is something bordering on insane. You know, there are standardized tests, there's grades, there's extracurricular activities, there are huge committees that sit and deal with every case. They accept legacy and people with talents that supplement whatever the needs of the school are, and who knows what else. And there's also a big mystery about this. All, in my opinion, geared toward making money on the admissions process. So, is the outcome any better? I believe serendipity is a big part of one's life trajectory, and maybe some of the serendipity is lost with this whole process that is very meticulous. But they're talking these days about having machine learning take over the admissions process, so we are in for a whole new era if that's going to be the case.

Rosen: Okay, so undergrad years?

Goldwasser: Right. Undergrad years I'm in Carnegie Mellon. I start in mathematics. There is even this program called Math Studies, which only a few kids go to, where there are these two professors who teach a handful of kids. It's supposed to go through all mathematics, you know, topology, geometry, algebra of course, logic, and everything in two years. And they spend essentially the first semester arguing with each other how to define each concept, definition, back and forth, back and forth. It's abstract beyond anything that I've ever seen because in Israeli high school, things are very method-oriented. They are teaching you how to perform, how to solve exercises. They don't really teach you . . . at least at that time, they didn't teach you about the concept of a limit or why are you taking derivatives and why you're integrating. Here, we are completely . . . it's all axiomatic.

So I go through this semester, maybe a year, and the whole thing is a two-year program, and after a year I quit. And I think to myself, "This is going to take too much time and I'm not the best at the class," and I decided I'm going to go and do computer science, sort of the computer science specialty within the math. So I take this class on—I think—combinatorics or data structures or algorithms, whatever, and it's trivial because my mind of course was so sharpened by this one year of dealing with abstractions and dealing with definitions that even if you don't think you're understanding them, you're completely in a different level. Then when you go back to something of a lower level, it's a triviality.

This is an interesting experience that I have seen time and again with myself, with my kids. You push yourself to a place which is much more abstract and much

more formal than maybe you care to be, and inevitably you start thinking more clearly, and you are able to sort of verbalize and conceptualize and define and understand. It's a fabulous discovery. Somebody has to prove a theorem about it explaining why is it that being able to verbalize, being able to define, and using precise concepts and precise thinking makes everything else simpler.

Rosen: So now you defend the very same thing that caused you to quit, like the abstraction?

Goldwasser: I know, I know. I mean in retrospect, maybe I should have stuck it out for another year, but that's what I did.

Rosen: Okay, so then you moved to computer science?

Goldwasser: I moved to computer science. I remember a lot of my professors at Carnegie Mellon. I remember Raj Reddy, who taught AI. He was the founder of real speech recognition. At the time, it was the Harpy project. And I remember Anita Jones. She taught software engineering. She was one of my recommenders to graduate school later. So was Raj Reddy. And I remember there was another professor, Nico Habermann, who taught us compilers and I had a compiler project that I did with a friend. I remember we wrote this compiler which never compiled. [laughs] I remember writing this program for generating poetry. Today, they talk in machine learning about GANs, these things that can generate let's say poetry in a way that's indistinguishable from let's say poetry of a particular poet. But at the time, the way these programs generating poetry would work is that you would have some sort of a notion of a verb and a noun and how a sentence is structured, then you would have a dictionary and you would form a poem. I loved that.

Rosen: How large were the classes back then?

Goldwasser: The classes were small. I would say there were like about twenty kids. Again, very few women. That I do remember, that I was one of two and the professor also treated us a little bit with, you know, half . . . I was going to say "forgiveness," but "forgiveness" might not be the right word. A little bit, you know, like we were silly, even though we weren't really. And that, after I start doing very well in the class, he realized that. But that was my feeling. It didn't matter to me much because I didn't think of myself that way, but I do remember that.

I remember coming from Israel, my command of English was not perfect to say the least, and on every program that I wrote there always were these comments where he says, "Indent, indent, indent." I didn't know what word "indent" meant until the end of the term, but then I realized that "indent" meant that I was

supposed to like, you know, indent the "for loops" and the different commands. So now I know what that means. But it was these silly things.

It was like. I remember the first lesson of calculus when you come from Israel to America, and I remember telling my brother, who was in school at that time, I said, "I can't do this. It's too difficult." So he sat down with me. This is the first class ever in calculus, and he said, "Okay, so what didn't you understand?" and then it turned out that I didn't know the words "multiply" and "divide" and "integrated" and "differentiate." Then he told me what they all meant in Hebrew and I said "Ah." That was it. Then it wasn't difficult.

Then I had to make a decision at the end of that year whether to go back to Israel to my army service or ask for a deferral. I asked for a deferral, because I actually kind of liked studying and I kind of wanted to continue.

Rosen: Happy moments?

Goldwasser: Oh, lots of happy moments. I made lots of new friends and also I became a young woman, so there's also like personal relationships that you develop which happen when you are a young woman, and that regardless of where you're at is very exciting, right? You're coming of age. And I came of age in Carnegie Mellon during those years, between the age of 17 and 20.

Rosen: Okay. Just to be a bit more specific about those years, any particular topics that you related to, specific ones, beyond the aspect of . . . ?

Goldwasser: Yeah. I was very interested in artificial intelligence at the time, I think because of the class that I took, because of this poetry generation, because of the whole concept of speech understanding and so forth, and also I think because this whole idea of understanding the brain and how we think and how we dream and why we dream, what we dream. That was fascinating to me.

So it was very clear to me when I finished that I would like to study this further. That's why I applied to graduate school. And I applied to graduate school at the same time that I applied for jobs, because I wasn't very clear about what I was going to do. There were sort of three options. In fact, this is the story of my life—there's always at least three options, sometimes four, but never one. And the options then were to go back to Israel or to go to graduate school or to get a job. The idea of going back to Israel was complex: I wanted to go back to Israel, but I was very afraid. Because at this point I was kind of distanced from it, and furthermore, I felt that I would like to go back to Israel, but at least I'd like to show something for all these years that I was away.

And I felt like I think a lot of people feel when they finish undergraduate school. At least I think they feel. That I knew nothing. Even though I studied for . . . I did my degree in three years in the States, although usually it's four. I studied during the year and I studied during the summers because I wanted to finish quickly so I could go back to Israel. At the end, you feel like, "What do I know more than anybody else? I want to own something. It'll be something that I'll understand better than anybody." It's not even so much the idea of understanding better than anybody, but actually going into some subject in depth. At that point, it could have been related to artificial intelligence or algorithms. I remember also an algorithms course that was taught by Jon Bentley, and it was fascinating. I loved that as well.

So I wanted to know, understand something really well. I was told that there is this thing called graduate school. You have to understand, I didn't come from an academic family, it wasn't something that was standard, but . . .

In any case, I was told that there was this thing called graduate school. I think that like a day or two before, they said that I'm supposed to take this exam called the GRE. I didn't prepare at all, but I signed up and I went to the GRE. I didn't even know you were supposed to prepare, you know? It seems ridiculous how naive I was. So I took the GRE. I don't think I did very well. But in any case, I applied to graduate school and I got accepted to Carnegie Mellon in engineering and Berkeley in computer science. First, I said to Carnegie Mellon that I'm going to go there, and I went for the summer to the RAND Corporation, where Raj Reddy actually recommended me as an intern. This was in Santa Monica, in California on the beach. And I remember this California. Wow. The beach. Fantastic, you know? I lived in Venice Beach and there's the roller skaters and the bikers and . . .

Rosen: Mellon . . . ?

Goldwasser: So I was admitted to Carnegie Mellon, which was the place I spent my undergraduate, and I was debating between the two, and I also had a bunch of job offers, but it was clear that I wasn't going to get a job. I was going to go to graduate school. And I decided I'll go to Carnegie Mellon. I mean I wasn't sure, but I decided I'll go to Carnegie Mellon, because I had friends there. You know, I had a boyfriend, whatever, you know the kind of things that people have, and friends.

But I had the summer job at RAND. And I remember that summer. I cannot tell you what I worked on, but I do remember that I was thinking to myself that the supervisors were all Ph.D.s, and they were telling me what to do. It was some sort of AI-related project. I remember thinking to myself, "Why should they tell me what to do? I should get a Ph.D. and I should tell somebody else what to do." [laughs]

In any case, so that summer was a fabulous summer. First of all, there was research and it was interesting, although I can't tell you what it was about because I really have no recollection whatsoever. And second of all, all of a sudden it was, you know, I had an apartment of my own on the beach, it was California as I said before. You know, there were the roller skaters and the bikes. And then one day me and [a friend of mine from CMU], we decided to take a drive up the coast, up the California coast and go and see Berkeley, and go visit somebody that she knew in Palo Alto. Anyway, we drove up the coast. And I remember driving into the Berkeley exit on University Avenue, and it was just blue skies that like you've never seen and the green hills in the background. I'm just sort of driving to campus. It's such a glorious image. I can't tell you . . . This is something you don't forget. And it was "Wow, California, Berkeley." Then I told CMU that I'm not coming and I told Berkeley that I'm coming, because it was just captivating.

Rosen: What year was that?

Goldwasser: This was 1979. So I arrived at Berkeley. I had to find an apartment, the usual things that graduate students do. I lived with a bunch of astronomer graduate students. In any case, I wanted to do artificial intelligence. At the time, there were few people at Berkeley doing artificial intelligence, but as I told you, serendipity is the name of the game. I was a TA, I had to support myself, so I had a teaching assistantship. Then I actually somehow got to work with Dave Patterson on the RISC project, reduced instruction set computer.

Rosen: Maybe you can tell a bit about that.

Goldwasser: About the RISC project? At the time, the RISC project was this idea of Patterson and other people at Intel at the time that the thing to do is to figure out which of the instructions are used most often, let's say programs in Pascal and C, and those are the instructions that should be put in hardware in order to speed up computation. My part of the project was to figure out which instructions in fact are being used most often in Pascal programs. So I was quite the programmer at the time. And I worked on this very large system, which I think adapted an existing Pascal compiler, a sort of thing that collects dynamic statistics, and I modified it sort of extensively to figure out which instructions should really be optimized or put in hardware. And that was my master's thesis, which I got at the end of that year.

Rosen: Did you enjoy it?

Goldwasser: Actually, it was Professor Powell and Professor Patterson. Did I enjoy it? Yeah, it was interesting. You know, it was a lot of work. It was very intense. This whole idea of being incredibly focused on a project and being in the office from day

to night was born at that time. I mean as an undergraduate, you spend a lot of time in libraries and studying for exams, but this idea that you have your own project and you set your own deadlines, although you know the professors expect things of you, it really comes from that time.

But at that time also, all of a sudden I wanted to go back . . . after I had the master's, I wanted to go back to Israel. I wanted to see Israel again. It's been four years. And I went for the summer. That was one of the highest . . . After four years not being in Israel, just being around here and with my mother and my sister. My sister was already a big girl. I remember taking a bus to Yamit. This was a time when they were actually withdrawing from the Sinai Desert. So I was in Israel then for three weeks, and then I came back to Berkeley and I continued to my Ph.D.

Rosen: Is there something about the initial time in Berkeley that you recall that is worthy of mentioning?

Goldwasser: I remember the professors. There were the theory professors. There was Manuel Blum and Dick Karp and Gene Lawler. And I remember meeting theory students, the theory graduate students. There was Silvio, which later on became a very close friend and a close colleague of mine. There was Vijay Vazirani. There was Faith Fich. There was Joan Plumstead. There was Mike Luby. They were all contemporaries of mine and I liked them. You know, I liked some of them more than others [laughs] as things are, and they're interesting characters. I took a class I think from Gene Lawler on scheduling, and there was a TA there called Chip Martel. Anyway, and I did some projects on scheduling with Vijay and Silvio. I remember that.

Rosen: That was your first collaboration with Silvio?

Goldwasser: It was a project—right—in class. Yeah, that was the first collaboration. Then I met . . . I took a . . . I met Manuel Blum, and Manuel offered me to be his student. I spent the summer working with him, and that was fantastic because he was such an unusual thinker, and he wanted to work with me, or he suggested that I would be his graduate student. It was a huge compliment.

Rosen: You felt like it's a compliment at the time?

Goldwasser: Yeah, sure. It was a huge compliment.

Rosen: Who were his other graduate students at the time?

Goldwasser: I think that Vijay and Silvio were his graduate students. I think before that it was Mike Sipser and Dana Angluin, and we were sort of the new wave. There was the three of us, maybe Joan too, Plumstead.

Rosen: What was it about them that you liked at the time, do you remember?

Goldwasser: They were extremely intense. They really loved what they were doing. They would talk about this incessantly, but they were a lot of fun too. You know, Silvio was from Italy and Vijay was from India, and they were so colorful and they had fabulous sense of humor. And they went out to restaurants all the time and talked about work and told stories. It was really just somehow these were people of the world. So as much as I liked Carnegie Mellon and had a lot of good friends, this was like a different dimension of personalities. If you think about it, people come to graduate school from foreign countries. They have lived a different life, each of them. They're older, they're sort of more worldly, and I was taken by it.

Rosen: Any particular memories, events from that or before . . . ?

Goldwasser: Yeah. There is actually a memory or an event . . . I think it was after about maybe like six months in or almost close to a year in Berkeley, I'm like a graduate student, I had a down period. It was like it's too hard and I don't have any original ideas and I'm never going to get through this, and I'm lonely, I don't know anybody, because I didn't have friends yet, close friends. And who do I think I am? And I was torturing myself continuously. What do I think about going to graduate school? Who do I think I am that I can just do this?

You know, I decided to leave Carnegie Mellon where I had lots of friends and just kind of conquer this new place totally on my own. I remember going through this cycle again and again and again, and then I had this realization that okay, maybe it's all true. Maybe I will amount to nothing and maybe I know nothing, and maybe I'm a failure. But if I'm going to be against myself and I'm not going to be my own friend, then who else? I'm going to have to like myself whatever I am. I got to accept that. And some of that was like a very kind of deep, decisive moment, that from then on, everything became better.

Because I think it's very important to realize that, for graduate students especially, which have moments like this, I'm sure it's universal, where you go, you've decided on this big adventure, and then it's very unclear, right? Are you going to succeed? Are you not going to succeed? There's a lot of competition. Everybody seems better than you. And there's a—I think—tendency for self-beating, at least for some people, and it's very important to realize that it is what it is, you know you got to like yourself, because at the end of the day, this is what you've got.

Rosen: Okay. Grad school, research, Manuel Blum.

Goldwasser: Research, grad school, right. Manuel Blum. Okay. Manuel Blum took me as a student, but as things go, it takes time to find a research project. Then

Manuel taught this class on algorithmic number theory. In this class, he taught us about, first of all, the basic elements of number theory, primes and composite numbers and quadratic residues and quadratic nonresidues and generators and cyclic groups and all these things, and all from an algorithmic point of view. That is, how to test that a number is prime, how to generate a prime, how to find the quadratic residue, how to test that something is a quadratic residue, modular arithmetic, and so forth, and always from an algorithmic perspective and analyzing running times. I found it fascinating. I really loved it. You know, it's very basic. I like this stuff.

Rosen: I remember you teaching me this.

Goldwasser: That's right. So I really love this material. And at the end, he had a few lectures where he talked about cryptography. At that point, there was essentially [only] RSA encryption scheme, a public-key encryption scheme, which is a way to send messages between people who have never met before, secret messages. It all is based on the fact that it's hard to factor composite numbers which are a product of let's say of two primes, but it's easy to generate prime numbers. And that was nice. Then there was another lecture on another method by Merkle–Hellman which Adi Shamir broke. And he did some cryptanalysis. That was interesting as well.

And then he asked the question, which was I think really defining for the rest of my career. He said there is an Alice and Bob, and they are deciding to get a divorce. Alice is in Boston and Bob is in San Francisco, or vice versa, and they have to decide who gets the dog. And they want to be fair, so they decide to toss a coin, except they're not in the same place and they have to toss a coin over the phone, except neither one wants a dog. Or both want the dog, whichever is the case. And the idea that Alice just tosses the coin and then she says to Bob "It's heads" doesn't exactly work because they don't trust each other. So he asked, "How would you do that? Can you use number theory to do that?"

So what's the connection? You know, why number theory? And that was sort of fascinating. Can you use sort of number theory? The idea that let's say factoring numbers is a hard problem, is there a way to toss coins over the telephone?

And I start thinking about it, and I had an idea. The idea was . . . that there was this function, which is a modular exponentiation function, like g^x mod p. The idea was to essentially hide . . . for Alice to pick like a random x and send g^x mod prime p to Bob and have him guess what x is. This is a function which is hard to invert: From g^x mod p (and g) it's hard to find x. And Bob tries to guess x, or actually to be more precise, he tries to guess something about x, like whether x is odd or even or greater than p over 2 or smaller than p over 2. And he makes a guess, then

she tells him what x is and both can check if the guess is correct or not. If the guess is correct, it's like heads has been tossed, and if the guess is incorrect, it's like tails.

And Silvio and I talked about it. I told Silvio about this. Then you needed to prove something, right? You needed to prove that this is like a coin toss, that really it's impossible for Bob to guess better than 50–50 whether x was greater than p over 2 or smaller than p over 2. And we had some proof, but there was a bug in it. And that was sort of the beginning of a lot of cryptography.

Rosen: And I want to ask at this point how much context about cryptography did you have at the time beyond what Manuel [taught in the class]?

Goldwasser: Nothing. Zero. Uh . . . Zero.

Rosen: Did you know about Shannon's work?

Goldwasser: Nothing. That was not part of the class. The class was about number theory and applications of number theory. I think that's what interested Manuel.

Rosen: Yeah, so why did Manuel Blum teach that class at that time?

Goldwasser: Because we're talking about 1980. Was it 1980 or 1981? And the invention of public-key cryptography was 1976 and then the RSA . . .

Rosen: Maybe you can give some context to the general . . . ?

Goldwasser: Right. So 1976, there was this incredible paper by Diffie and Hellman which suggested this idea that we are having this possibility of digital communication, that eventually everybody's going to be communicating with everybody else over the digital network. This is the case today. It wasn't the case in '76, but the possibility was there. And they were asking, "How can we utilize this in order to kind of shift the world into this mode of electronic commerce?" I think they even talked about these things explicitly in this paper. And they brought up these two suggestions.

One is what they call public-key encryption, which is a way for let's say an Alice and a Bob who've never met before to communicate secretly. Somehow there would be a directory where Alice would publish something that they called a public key, and Bob could read Alice's public key and use that in order to send her coded messages that only she, who knew also a corresponding private key, would be able to read, but no one else could. This was one thing.

Another thing that they suggested is this idea of a digital signature, which is that people could sign documents so that everybody can verify that, say, Shafi signed it, but only Shafi could sign it. As you know, a handwritten signature, if I have a signature, it looks the same no matter which document I put it on. Here the

case was that you would take a document and you would do a transformation to a new document which is called a signed document, and the ability to perform the transformation would be something that each user in the system, Shafi or Alon, could do in a way unique to them, because they knew some information or some private key that enabled them to do so and yet there was a matching verification key that would be able to verify that this was signed by Shafi or alternatively something was signed by Alon. In any case, they proposed these two things. They didn't give ways to do it.

A year later, there was a paper by Rivest, Shamir, and Adleman where they showed how to do it using number theory. Around the same time, there was also a paper by Michael Rabin who showed yet a different way to do it also based on number theory.

And Manuel taught those three papers, because they were just mind-boggling. This whole idea, very tantalizing. Not only that; I think that Len Adleman was a student of Manuel's, so there was some affinity there as well. But one would have to ask Manuel why he taught that class. I think it was the first time he did teach that class, in any case. I think. You know what, maybe not. Maybe he has taught it before. Maybe, because there are these notes, these lecture notes on number theory by Dana Angluin. So he must have taught it before when Dana was a student, but I don't think he taught the public-key cryptography part of it.

Rosen: Who else was in the class besides you and Silvio?

Goldwasser: Me, Silvio, Vijay, Mike Luby. You know, the usual suspects. I mean all of the crowd at Berkeley was there. Jeff Shallit was another good friend at Berkeley, and Eric Bach.

Rosen: They went on to do computational number theory.

Goldwasser: That's right. You know, that's right. Eric has this very famous paper about how to generate primes in factored form, which is an important paper for generating generators for the multiplicative group mod a prime. Jeff Shallit also had very interesting work, and they later wrote a book together on computational number theory. And we were all colleagues, and friends. And we're still friends.

Rosen: Okay, so now it begins?

Goldwasser: Now it begins. Right, so okay. So Silvio and I decided to work on the following problem, and the problem was how to play mental poker. Because there was one other paper that Manuel mentioned, and that was a paper by Shamir, Rivest, and Adleman where they used their encryption scheme in order to show how to play mental poker.

What is mental poker? People probably know what poker is, although I didn't because my parents didn't play cards and the whole idea of card playing was supposed to be this thing that you did not do, somehow there was something improper about it. Anyway, so this mental poker protocol by Shamir, Rivest, and Adleman, the idea is again, we are two players, we don't have a physical deck, we want to play poker over the phone, over the computer line, and how are we going to do that? How are we going to deal cards in such a way that you're going to get a random hand, I'll get a random hand, and once we get the cards they're not in the deck anymore without knowing what each other's decks are? They had an ingenious idea where there was a way to deal cards in such a way . . . I mean it seemed like you don't know what my cards are, that I did choose random cards, and same for you.

But Lipton noticed that this protocol, there was a problem with it, that there was something about the implementation of this protocol that they proposed where it's true that you couldn't tell what my cards were, but you could possibly tell some information about my cards. For example, let's say that you could identify something was a high card versus a low card. So there was something about the encoding of the cards that did not hide all information about the card. Now for a card game, that's detrimental, right? If you know that I have a high versus a low card, then this changes your strategy completely.

So the problem we set out to solve was how are you going to play mental poker hiding all partial information about the cards? I remember that we're thinking about this problem and what do we need, and Silvio had this idea that we need to have some encryption scheme that . . . Not encryption scheme. We didn't talk about encryption. I think it was Silvio's idea that we needed a decision question, like a yes/no question, where it's hard to tell whether it's a "yes" or a "no" better than 50–50. But this was like an abstraction, right? And a little bit like the Diffie–Hellman.

Because I loved the number theory, I remember sitting in a seminar where some people were talking about something else . . . and in fact I must say that this repeats in my career over and over again. I get ideas while I sit in seminars when people talk about something else, which is probably a good reason to go to seminars. [laughs] And all of a sudden, I had this idea about quadratic residues. I said, "You know what . . . " I think to myself that the way to encode the zero and one, the decision question would be to decide whether the number is a quadratic residue or quadratic nonresidue modulo a composite number n, and this was a hard problem. I mean Manuel told us this was a hard problem, a hard problem in the sense that there were no efficient algorithms to solve it. And the reason why I thought it was a good idea is because it seemed to be a problem which is hard on

the average. In fact, not only that you cannot tell whether something is a quadratic residue or nonresidue, but you couldn't really do better than 50–50. And one would have to prove that, right?

But there was something about this problem, which is a notion . . . later on defined formally, which is called random self-reducibility. It was sort of a way of showing that if you had one number, if it was a quadratic residue you can generate lots of [random] quadratic residues, or if it was a quadratic nonresidue you could generate lots of [random] quadratic non-residues. And then that means that if you could sort of distinguish one sample from the other even a little bit, then you will be able to distinguish whether your original number was a quadratic residue or a quadratic nonresidue.

Rosen: How did you feel at that moment, or . . . ?

Goldwasser: That moment of thinking about the quadratic residuosity being the right problem and then telling Silvio? God, excitement. It's just incredible. Because pretty quickly, we could sort of come up with a proof.

And then, just to come back to the mental poker, the idea was that this would be a way to write down a card. Let's say the card is five of diamonds, okay. Then you write this down in binary, the five of diamonds—so that's in zero/ones—and now you want to encrypt the zero, encrypt the one, encrypt a zero, encrypt a one, each time encoding it by a different quadratic or nonquadratic residue. Quadratic residue for zeros let's say, nonresidues for one. You choose them at random. And now you have an encoding of the card, which is what we would call later probabilistic encryption.

Rosen: At the time, did you realize it's public-key encryption, or . . . ?

Goldwasser: We didn't even realize it was encryption. We had a card. We had a way to encode cards so that we could prove that there is no way you can distinguish one card from any other, because you couldn't distinguish zeros from ones better than 50–50.

Then, we went to Dick Karp, I think because Manuel was on leave at MIT for a semester, and we told him about this. He asked us, "What about other partial information, not just with a zero/one?" These questions professors ask you are incredibly significant, because you don't think this way, right? I mean now it's an immediate question, but at the time it was a very fundamental question. And then we went away and proved that if you could tell any partial information regarding (the sequence of bits that encodes) the card—and you had to define what partial information is—then you could actually reconstruct the individual bits of the card.

Which implied that you could tell whether a number was a quadratic residue versus a quadratic nonresidue, which was a hard problem.

Rosen: Can you tell something about the process of figuring out the right definition?

Goldwasser: The way I'm telling it to you, it's really derived from the goal. The goal was to play mental poker in such a way that it hides all partial information. In order to do that, it was clear that you had to encode every bit individually, and furthermore it was clear that you would have to encode them in a probabilistic manner, because otherwise you couldn't hide all partial information. Then there was that question of Karp's, so we arrived to the question "What is partial information?" It should be any function that kind of divides the world of cards into two parts. So any function that partitions the cards into sort of the left and the right, you know?

The process was just . . . it was like being in some kind of a mad state of creativity. And working with Silvio was just a very intense experience, as anybody who's worked with him knows. I mean there's no day and no night. And I think he's still that way. I'm not, but at the time I was. He was very intense, it was very exciting. And of course we didn't do it completely in isolation. There were these questions that Karp asked us, and then I think maybe it was him or maybe we understood already there was a way to encrypt here, that it doesn't have to do with card games. There's a way to encrypt the zero and encrypt the one.

That's something that was not known, because the public-key encryption of Rivest, Shamir, and Adleman or even the Diffie–Hellman concept, it really was intended for encrypting long messages which are unknown. And here zero and one, you know that everybody knows you're either encrypting a zero or a one, but they can't tell which is which. So this was a completely new way to encrypt information. We understood this is much bigger than our original goal, but . . .

And we went to consult people in number theory, you know, in the math department. There was Lehmer and he was the expert. We were supposed to talk to him and ask him, "Is it really the case that you cannot tell apart quadratic residues from nonresidues? Maybe not just perfectly, but better than 50–50?" And I remember this quote. He said . . . We told him the whole story and we asked him what would he do if he needed to distinguish whether a number was a square or a nonsquare mod n. He said that if it was less than n over 2, he would bet it was a square. We asked him why, and he said, "Because there's a lot of small perfect squares." But he said he's not a betting man. Then it turned out that this is okay because this doesn't give much of an advantage.

Rosen: When you came to him, did you feel the stakes are high?

Goldwasser: No. We came to him as two young graduate students and he was very accepting. A little bit maybe I thought he was a little humorous, because it's such a frivolous question, right? Playing cards, using quadratic residues. But I think the whole attitude of mathematicians to computer science has changed radically. Not to say that he wasn't helpful. He was extremely helpful. But in general I think, at the time, mathematics was this hard science and it was serious, right? And the whole computer scientists and the algorithm aspects and using it for cryptography was considered more of—I think—a toy activity. I think this is very, very different now. If I look at the mathematicians at MIT, and I'm sure it's true all over the world, they have respect because we are studying hard questions, we are studying important questions, we've made impact on the world. Cryptography certainly has made a lot of impact. It's making a lot of impact today. And only more so, as you well know as well.

Rosen: And I'm asking again about the stakes because I am curious to know, when did you realize how big your discovery is at the time?

Goldwasser: Right. So we realized that we have actually a scheme for encrypting single bits, something that was an open question that nobody addressed. And when you encrypt a single bit, obviously it's going to have to be a randomized method, because it is a public-key encryption, so everybody can encrypt a zero and a one. If all encryptions of zero were the same, when you see the encryption, you can just yourself try to encrypt zero or try to encrypt one, and if it's the same as what was sent, you know what was sent. So it has to be the case that there's lot of encryptions of zero and lots of encryptions of one, and an adversary shouldn't be able to distinguish whether we're encrypting zeros or ones. You cannot actually have any better than 50–50 plus negligible probability of success in guessing which random bit was encrypted.

Now, in the context of a protocol, if you think about this mental poker example, not only that you're encrypting the cards but there's a lot of other information going around. There's the dealing of the cards where many cards that are being encrypted. You could ask the question whether, having been part of this game, playing the cards, maybe you gain more and more knowledge as you go along so that now you are able to guess something about the unrevealed cards better than what can be inferred from the revealed cards. The definition, which we called semantic security, covers this too.

In order to prove semantic security, we came up with this idea of a proof by reduction, the idea being that you say . . . well, let's suppose that your goal in the world really, you have no interest in mental poker, but what you want to distinguish

is quadratic residues, quadratic nonresidues. Okay? And somebody tells you that there is this mental poker game that's built on encoding cards with quadratic residue and quadratic nonresidues, and they know how to cheat in this game. So what you say to yourself, "Okay, I'm going to show a reduction now. I'm going to show that if in fact there is this person"—or this adversary, which we usually call them—"who is able to cheat in the mental poker game, even by slightly better than he should, then there is a way to use this strategy and turn it into an algorithm that can distinguish quadratic residues from nonresidues."

Since you believe that quadratic residues and nonresidues cannot be distinguished in polynomial time, it means that such strategy does not exist. But how do you show such a reduction? In a sense you need to simulate everything, the entire view of the adversary—that is, the encoding of the cards and the dealing and everything that went on and was available to him to enable his cheating strategy. This is what's called proof by simulation, which later has become a big paradigm in cryptography, in how to actually give security proofs. You can prove security if you can sort of recreate the real world in which cryptography is used and its security is supposedly violated. And if you can simulate it although distinguishing quadratic residues from nonresidues is hard, then it means that this violation must have not been that useful, because you could have simulated this violation anyway.

Rosen: In hindsight, you can view Shannon's security as being the information-theoretic sort of analogue of semantic security. Did you see that at the time, or you came up . . . ?

Goldwasser: No, we didn't really know about Shannon's paper, because we were ignoramuses, [chuckles] which helped us actually. Shannon's information theory in fact, if you look at the definition, essentially says that the probability of two messages is the same given the ciphertext. That's one way to think of Shannon's security. An equivalent definition is the *a posteriori* and *a priori* probability of a message is the same, where the *a priori* is without given the ciphertext, and the *a posteriori* is given the ciphertext. In other words, the ciphertext gives no information about the message. Or, if you think about the first definition, given ciphertext for the bit zero or ciphertext for the bit one, there is no information in there that can tell you whether it was a zero or one.

If you think about semantic security, it's the computational analogue of it. That is, in principle, information theoretically you actually do have enough information to tell whether you're seeing an encryption of a zero or a one, because it's a public-key encryption scheme. But computationally within polynomial time, you don't,

if distinguishing quadratic residues from nonresidues is a hard problem, or if factoring integers is hard.

Of course, it could be that factoring integers is easy. We know that for quantum algorithms, factoring integers is easy. So if quantum computers can be built, then this whole tower of cards collapses. But this is only for the first probabilistic encryption scheme. Today we have a lot of other problems, not just quadratic residues versus quadratic nonresidues, not just the factoring problem, but also problems on integer lattices, which are problems essentially from geometry. Now, we can apply this idea of a decision question which is hard to solve in the sense that it is infeasible to decide better than 50–50, and encode zero by this decision question where the answer is yes and one by a decision question where the answer is no. And these lattice problems, I mention them because they are quantum-resilient. In other words, we don't know any quantum algorithms that can solve them efficiently. They are what we call post-quantum cryptographic candidates.

Rosen: Okay, so at the time, the idea of basing something on an unproven assumption, it was in the air, or was it kind of a bold move?

Goldwasser: Right. Well, if you think about RSA, they're also basing it on an unproven assumption. They are the first. They are assuming that factoring integers is a hard problem. We took another problem, which was distinguishing squares from nonsquares. But obviously that's an assumption, and you know mathematics prides itself by having proofs, and proofs are proofs and not conjectures. So there's an underlying conjecture here, and that is that there's a problem which we don't know how to solve efficiently. But if you think about it, all of complexity theory is predicated on the conjecture that the class P of polynomial-time problems and the class NP of problems which you can verify the correctness of the solution in polynomial time are different. So to give meat to the entire field, there is an underlying conjecture which is widely believed but not proven, and then one builds on that conjecture.

Rosen: And at the time, what was the atmosphere? Did you experience any resistance to this idea?

Goldwasser: To this probabilistic encryption? We submitted it to a conference and it got in the first time. This was a conference in San Francisco, in 1982. I think it was a STOC conference and I gave the paper, and the name of the paper was "How to Play Poker Hiding All Partial Information and Probabilistic Encryption." It was a long title. And I think that people were genuinely very positive, but speaking with

people afterwards, I think they had no idea what I was talking about. [laughs] But certainly in the cryptographic crowd, there was excitement.

Rosen: Was it your first talk in the conference?

Goldwasser: Yes.

Rosen: And how did you feel?

Goldwasser: I felt on the top of the world.

Rosen: How well attended was it, just . . . ?

Goldwasser: Oh. In that time, the conferences were very well attended. There were no parallel sessions and people came to the entire conference, and it was a fairly small community.

Rosen: Can you tell us a bit more about the atmosphere at the conferences back then?

Goldwasser: I think that, you know, very intimate, very informed people. They were already people who were working on different fields—you know, algorithms and complexity theory, here's a cryptography example, and distributed computing. People started talking then about Byzantine Agreement. A lot of these big ideas that are still around as sort of fundamental problems were being discovered at the time.

Rosen: Were you attending all talks?

Goldwasser: Yeah, I was. Everybody was.

Rosen: And was it accessible to everybody, to a wider audience than it is today? How do you compare?

Goldwasser: I think so. But it's natural. When a field is young and not overburdened by definitions and history and background, it's easier to understand. On the other hand, people give much better talks today. People have learned how to simplify their talks—PowerPoint has helped quite a bit—and people have more respect to distilling the essence rather than giving all details.

Rosen: And what happened next? How did things evolve?

Goldwasser: Then, I had been to Berkeley at that point for three and a half years, and I had a very strong urge to get a job and leave. Somehow, I think about it now, I don't know why it was so urgent to leave, but Berkeley seemed to me then like this small place and it's time to go. I applied for a postdoc and I got a postdoc with Ron Rivest at MIT. I was there for half a year actually. Then, they were looking for faculty members and I started interviewing for faculty positions all over the country

and also at MIT, and I got an offer for a faculty position and I started on the faculty in '83.

Rosen: After having published what results at that time?

Goldwasser: There was this probabilistic encryption paper. Then there was another paper which we start realizing that it's not just this particular quadratic residues versus nonresidue, but you can take actually any function which is what we call a one-way function. That is a function which is easy to compute but hard to invert. And in particular the RSA function. We asked what bit about it is well-hidden . . . The RSA function is you take an x and you take it to some power modulo a composite number n, like x^3 or x^5 mod n. The question is "What about x is really well-hidden?"—well-hidden in the sense that you can guess better than 50–50. So the paper was on that, looking at the bits of x and showing, proving that they are as hard to guess as it is to invert.

Rosen: And this was still at Berkeley, or . . . ?

Goldwasser: This was still at Berkeley, yeah.

Rosen: With who was the paper, do you remember?

Goldwasser: This was Silvio and Po Tong, who was another graduate student. I think that those were the two papers that I had, yeah.

[Editor: Actually, there were another couple of papers on signatures, both with Silvio Micali and Andy Yao. So, at that time, there were four conference publications altogether, and no journal publications at all.]

Rosen: Okay, so you start as faculty at MIT?

Goldwasser: I started as faculty at MIT and Silvio came a semester later. He was at University of Toronto and he also got a faculty position at MIT. It was like an incredibly intellectually exciting time. Oded Goldreich, who is now at Weizmann, came as a postdoc. There was Benny Chor, who was a graduate student there. Later also Yoram Moses came. I think Michael Ben-Or was there for some period of time. And all these people, they were young, they were brilliant, they were enthusiastic. We would work from day to night and then we would have dinners and talk about work and go to movies. And cryptography was starting to march along.

So I think that the next thing that I did was this paper on pseudorandom functions. There was an early paper by Manuel Blum and Silvio Micali on how to generate pseudorandom numbers in a way that you cannot distinguish these pseudorandom numbers from truly random. And the next question was how do you actually generate not just a polynomial-sized list of numbers but a very, very

long list of numbers, an exponentially long list of numbers, in a way that you could sort jump in the middle. Another way to think of it is a function. So . . .

Rosen: And what was the motivation for this specific question, given that you can generate a polynomially long?

Goldwasser: The motivation was that there are a lot of applications where you want to sort of random access. For example—I think this is one of the original motivations we had in the paper—is what we called an "identify friend or foe" system. We were saying, let's say that we are in a group, and we want to identify ourselves to each other, but there are some enemies that come along, and we don't want to use this password system where they ask, "What's the password?" I tell them what the password is, and now they know. Instead, I want them to ask me a random sort of question, which I can answer. And if we are from the same group, they can verify my answer is correct, but anybody else, really as far as they're concerned it's a random answer. So if you had what we call a pseudorandom function, there is a way for all of us who'll know the secret of this function—or what we call the seed of this function—to be able to compute this function f on any x, and then the random challenge would be x and I will tell you what f of x is. But being pseudorandom means that for anybody else, they can't tell it apart from a random function, so when they are asked x, to them f of x is like totally random. That's an application.

Rosen: So on that thread, I'm curious to hear how much of a role did practical motivation play in coming up with these notions?

Goldwasser: With these notions? That's a very good question, because it's not clear what you mean by practical. When you say "practical" today, you mean there's going to be a startup that's going to implement it. No such thing, no startups. Nobody implementing. So the level of practical that made any sense at that time was to say that there is a story, like identify friend-or-foe or people sending encrypted messages or people trying to authenticate themselves. And somehow I think those stories were important for narrative, because I've always liked stories, like the biblical stories. And in general I think people have an easier time to read, especially in a new field where there it isn't a mathematical problem that's been defined for many years and that people are interested in and they don't need any motivation. In a new field, you need to compel people, and stories are helpful.

But for us, it was really more of an intellectual story. The pseudorandom-number generator was just a polynomial sequence of numbers. Then the question about being able to kind of have an exponential sequence where you can sort of jump in the middle and just generate a polynomial number of them or this abstraction of a

pseudorandom function is what interested us. And once you had it, you could tell a story, many stories.

Rosen: So you didn't feel any pressure to practically motivate any of your . . . ?

Goldwasser: No, no. None.

Rosen: And what do you think about this versus the alternative? The need to find practical motivation.

Goldwasser: I think that every once in a while I have graduate students, and they come up with a question. For example, I have these two students now, they asked about pseudorandom functions, what happens if somebody knows the secret of how to generate these pseudorandom values? Does it still possess some cryptographic hardness? This is a very technical question. But some of the reactions they got is that "What is the application?" And they came to me and they asked me if they should work on it or stop, what's my opinion, is it interesting? I said, "It's very interesting." It's intellectually interesting. They had a beautiful sort of approach to it. They had a beautiful proof. And at the end, that's the nugget, right? It's sort of something that captivates you, you have to use some ingenuity to solve it, and you have insight. And if it's important, even for applications, it will emerge, but it's not necessarily obvious in the moment that you start. And sometimes if it is very obvious, first of all, lots of people work on it, and you know competition is good but only to a certain extent. If everybody's working on the same problem, there's some kind of . . . I don't know. I don't like to be in a space that's very crowded.

Rosen: How did it feel back then in the early MIT days in terms of competition?

Goldwasser: Right. As I said, we were a big, happy family, but [laughs] a big, happy family of a lot of people who wanted to do well. So we worked collaboratively, we've got a lot of joint papers, also with Benny on this thing called verifiable secret sharing and with Oded on pseudorandom function. But we each started, within a couple of years everybody started going in their own way as well, because you are in an academic system, they compare you, they promote you at different times, they tell you that you should kind of shine individually.

And I personally . . . You asked how I felt. Remember we talked about the crisis of becoming a graduate student. That was again a time which was extremely difficult, because you're trying to do something new, you're trying to do it on your own, you are always comparing yourself to the people around you who are always brilliant, and more brilliant than you are, and you don't know that they're all feeling the same thing. You know this imposter feeling? Apparently they're all feeling it.

Some admit it, some don't admit it. [laughs] But once you realize that this is the name of the game, I think again it's these moments of realization.

Rosen: So did you have such a moment?

Goldwasser: Yes, yes.

Rosen: When was that?

Goldwasser: I think I was talking to somebody . . . and I told them about how I feel and they told me about the imposter syndrome. Now everybody knows it, but then I never . . . I asked what it was and they explained, and it was like, "Ah, okay."

Rosen: That was a person external to the . . . ?

Goldwasser: Yeah. Like a friend, yeah.

Rosen: Okay. What about teaching? Do you have any memories?

Goldwasser: Yeah. Teaching we really started . . . I started and then Silvio also together teaching this class on cryptography. It was the course of Manuel Blum but with a lot more, because at this point the cryptography was a big part of it. There was the definition of bit security and the semantic security of an encryption and the mental poker, and the partial information, pseudorandom functions, pseudorandom number generator. It started being a field. And we haven't talked about zero-knowledge yet.

Rosen: That was before zero-knowledge?

Goldwasser: Around the same time. It was before it got in, but . . .

Rosen: Before we get to zero-knowledge, who were the students in this class that you remember?

Goldwasser: The students, yeah. There was Johan Håstad, there was Joe Kilian, there was Bill Aiello. I think in the early years there was Yishay Mansour, but I think he was a little bit later. Those are the students . . . there's Paul Feldman, who was a student of Silvio's. The others were student of mine. And they're all big names, fantastic researchers in their own right.

Rosen: How did the other MIT faculty treat the young field of cryptography? How did they perceive it?

Goldwasser: MIT is an incredible place. I think that they really have had the foresight of hiring people who were not necessarily in the mainstream of theoretical computing, but sort of doing something with the tools of theoretical computing which is a little bit on the fringes. Rivest was like that. Public-key cryptography after all was exciting, but it was unusual, right? And Silvio and I certainly, and

Charles Leiserson was doing also things which were, you know, with applications. At that time, I think it was data structures and stuff like that. Nancy Lynch was doing distributed computing and Byzantine Agreement and lower-bounds on Byzantine Agreement.

So I felt that they were incredibly proud of all achievements, and especially Ron Rivest, who was a major mentor. Because now that I think of it, he wasn't really much older than we were. Maybe 5, maybe 10 years, no more. And he was extremely supportive of us. We have a paper joined with him, digital signatures. But by and large, we each did our own thing, and I think Ron started working on computational learning fairly quickly, so he kind of left the cryptography field, except for its commercial aspects, for a few years.

Rosen: What other faculty do you remember from the time being supportive?

Goldwasser: Albert Meyer was very supportive. I think he was really a very significant mentor in his own way, sort of in the background. I mean Ron was in my field, so it was sort of more of a daily advice or monthly advice. But Albert was at the head of the theory group and he saw something in me and put me up for the Grace Murray Hopper Award, which made me feel good, made me be recognized.

Rosen: Okay. Is it time for zero-knowledge?

Goldwasser: Yeah, I think so. So zero-knowledge. Alright. So this whole idea of having a protocol where let's say two people are sending messages back and forth and there's a goal for the protocol usually. The goal might be to . . . In the context of going back to that mental poker, say you want to prove that the cards that you encoded were encoded properly, but you don't want to say what the encoding was. So there's a statement here, and that is that all 32 . . . sorry, all 52 cards have been encrypted and no two cards are the same, but you're not going to tell me which card is which. Then there is apparently a way to do it. Apparently. We showed a way to do this, which amounts to actually showing whether something is a quadratic residue or a quadratic nonresidue, so that I can prove to you that something is a quadratic residue or that something is an encryption of zero, or let's say the two things are encrypting different bits, in such a way that you will have learned nothing else.

Rosen: So you had a protocol?

Goldwasser: So we had a protocol. And now we had to have a definition. What does it mean, "prove so that you learn nothing else"? The definition went back to the simulation paradigm and it is called zero-knowledge. Let me explain what it means. So I'm a prover. I know something and I'm proving it to you. I'm proving you some mathematical statement without actually giving you the proof, which seems a bit

weird, so at the end you'll be convinced that the statement is correct. But what do I want? I want you not to be able to prove it to a third party. In fact, I want you to learn nothing from it. So how do you define it? The way you define it is that whatever you can compute after you interacted with me, that's no different than what you could have computed before you interact with me. And an equivalent definition to that is that you could essentially simulate the entire interaction between us. And if you could indeed do so, it means that interacting with me was useless to you, assuming the theorem statement is correct.

Rosen: And the name "simulator," when did it come about?

Goldwasser: Who remembers?

Rosen: At what stage? There's a story about multiple rejections?

Goldwasser: Ah, okay. Right. So this paper, we started. We didn't actually call it "simulation," I don't think. I think it had some other definition. They were many names for this paper. It started, it was "Participatory proofs . . . " "Interactive proofs such that they hide all partial information." There were many, many names until we got to the final name, which was "Interactive proofs and zero knowledge" or "The Knowledge Complexity of Interactive Proof." And the paper was rejected three times. God knows. But we were very persistent, you know?

Rosen: How did you feel with each rejection? What's the . . .

Goldwasser: Well, you know there were three of us. I mean in the beginning there were two of us actually on this paper, Silvio and I. And then Charlie Rackoff joined. He improved the paper, but it also got rejected. Because there were three of us, we could sort of build each other up. And how did we feel? We felt like everybody else was an idiot. [laughs]

Rosen: You had this confidence back then that you're onto something?

Goldwasser: But this concept was so interesting and we liked them, and it was clear that this is a great paper.

Rosen: And Charlie Rackoff was at the time where?

Goldwasser: He was in Toronto.

Rosen: In Toronto, so how did the interaction work back then?

Goldwasser: I think Silvio and Charlie interacted when Silvio was in Toronto. They had some paper on coin tossing or something. Then Silvio came to MIT and we continued working on the interactive proofs, but I think there must have been some interaction between them. I wasn't . . . It really wasn't a three-way interaction.

Rosen: But how was communication with people from other institutions working in general?

Goldwasser: Well, there was email, but there certainly wasn't the World Wide Web, or it wasn't immediate. There were phone calls, a lot of phone calls. There were visits.

Rosen: Do you remember any notable visits, visitors and/or visits from the time or from . . . ?

Goldwasser: Adi Shamir used to come to work with Ron. Again, I told you that Oded Goldreich was around. And that's about it.

Rosen: Okay. So zero-knowledge was rejected and you said the manuscript improved over time with the rejection?

Goldwasser: It did improve over time. Sort of in the beginning, I think the simulation was under computational assumption, then it became without an assumption. Finally, it got in. We were mighty happy. And we went to the conference. I'm trying to remember who gave that talk, if it was me or Silvio. I don't remember.

[Editor: Silvio gave the talk.]

But in any case, at the same time, at the same conference there was another paper, which was called "Arthur–Merlin Games." This was a paper by Babai, who introduced this concept where there was a prover and a verifier like we had, except the prover's name was Merlin and the verifier's name was Arthur. And the difference between a verifier and Arthur was that Arthur was just tossing coins, he was very naïve, and Merlin then, based on Arthur's coins, he would kind of teach him things or prove to him things, such that if he was proving a correct statement, Arthur would believe it, which we call completeness, and if he was proving an incorrect statement, it doesn't matter what strategy Merlin would employ, Arthur would not believe it. That was the same as interactive proofs, except our verifier didn't just toss coins. He tossed coins and did computations, and based on these computations would send messages.

Rosen: And his motivation was totally . . .

Goldwasser: His motivation, there was some group-theoretic problems that he wanted to show were in NP, but he couldn't, so he allowed this extension . . . In NP, you also can think of it as a proof system where there is an all-powerful prover and he writes down a string which is a short proof that can be checked in polynomial time. An interactive proof, it can go back and forth, back and forth, so the prover can send the string, the verifier asks the question based on some coin tosses, the

prover sends another string, go back and forth, back and forth, and in the end the verifier says, "I'm convinced."

Rosen: So essentially in your paper, there are two main topics . . .

Goldwasser: Yes, there were the interactive proof systems and the zero-knowledge ones, which are an important special case.

But, just to finish the previous thought: As I was saying, what Babai was trying to show, some problem, some group-theoretic problem was in NP, but he couldn't, so what he did is he added this Arthur that was able to toss coins. And for an Arthur that could toss coins, there was a short interaction by which you could show some group membership problem.

Rosen: And when did you realize that it's a similar related concept? At the conference? Was it at the time of the conference?

Goldwasser: I think it was at the conference.

Rosen: And did you already realize back then, view it as a generalization of proof systems?

Goldwasser: Yeah, we did. I don't know if he did, because for him it was really a way to show a complexity bound, the complexity of certain problems. He defined a complexity class and showed that these problems are in this complexity class. For us, it was always a proof system, because we were coming from the cryptographic setting. So there were parties. There were these Alice and Bob, where Alice was the prover, say, and Bob was the verifier.

Rosen: To what extent did you understand the important open problems that emerged from this new concept at the time?

Goldwasser: Yeah, they were abundant. One question was whether this system of Babai and interactive proofs were the same. He had this system of Arthur–Merlin. We had this verifier–prover. Arthur could only toss coins, the verifier could actually toss coins and compute on them, and that seemed to be a very important feature that enabled you to prove things you couldn't do just with coin tossing. So that was a clear question. Then Mike Sipser and I, we proved that those two classes were the same.

Interestingly, it all started again from the quadratic residue question, which was a question that kind of followed my career, because it seemed like to prove that something was a quadratic nonresidue required, without sort of revealing information, required a verifier's power to hide the results of his coin tosses. And I was talking to Mike about this, and then he had this idea that we could look at

the set of all quadratic residues and the set of quadratic nonresidues, and talk about what are the union of those sets. Anyway, we talk about size of sets and relate that to the question of whether a number was a quadratic residue or quadratic nonresidue, which is related in turn to the question whether Arthur–Merlin games and interactive proofs are the same class or not.

Rosen: Did you have any applications in mind beyond the original mental poker application?

Goldwasser: Not really. It was again a concept. How do you prove a theorem in such a way that you will believe the statement but you will learn nothing else, with the definition that I gave you, and that you won't be able to prove the theorem to a third party?

But very quickly after, as soon as the paper came out, Adi Shamir pointed out the application for preventing identity theft. Here in this situation, you would think about me. What identifies me is the fact that I know how to prove some theorem and nobody else knows, because it's a difficult theorem to prove. But I have the proof. How do I have the proof? Maybe the proof is something like I know the factorization of some number. How do I know it? Because I took two primes and I multiplied them, so of course I know how to factor it. Now I want to prove to you that I know this factorization or something about this factorization that only I will know. That would identify Shafi: that there's this composite number and she knows how to factor it. He realized that this is an identification method, and he took actually a protocol that we have for proving that something is a quadratic residue and made it more efficient in terms of how many rounds you need to accomplish it, and it . . . This is the work of Fiat and Shamir, and this became an identification scheme.

But the interesting thing about zero-knowledge is that is really the tip of the iceberg. Really, "the tip of the iceberg" is the wrong analogy. In any case, that's just scratching the surface, because it turned out that even though we showed the applications of zero-knowledge in the sense of particular number-theoretic questions you could do in zero-knowledge, like whether something is a square or a nonsquare, it had a much wider applicability.

There's a follow-up paper by Silvio Micali, Oded Goldreich, and Avi Wigderson where they showed how a prover can prove to a verifier that a graph is three-colorable, and that's an NP-complete problem, and what follows from this is that you can actually show any NP statement in zero-knowledge. So I can prove to you any statement that has a short proof in such a way that at the end, you'll believe the statement but you will have no idea of the proof. In order to do that, they

used computational assumptions, so this was under the assumption that one-way functions exist.

What this means—okay, going a little bit into the field—is that essentially we can take any protocol, any protocol between let's say multiple people, not just two, where there's a program say that specifies what messages I'm supposed to send to Alon and what messages Alon has sent to a third party and so forth. The thing is that the messages that I'm supposed to send are based, let's say, on my passwords or some private information I have. The messages you have, you're supposed to send are based on what you have received from me and your private information . . . So I do my computation, I send the message. If we're all honest, everything's fine.

But suppose I'm a liar. I'm an adversary. We're in a cryptographic setting. We're all liars in some sense, or we have to protect ourselves in any case. How do you know I'm sending the right message? How do you know I did the computation correctly, based on my private information and all the messages I receive? Well, that's an NP statement, right? So there's a statement to prove, and that is that I am sending the correct message. If I can prove that in zero-knowledge, it means that I can actually transform all protocols that work when people behave properly to protocols that work when people behave improperly, because essentially every message I send is accompanied with the proof that it is the correct message, and it's a zero-knowledge proof so I'm not revealing anything about my secrets.

Rosen: What about other applications?

Goldwasser: Lots of other applications. The next application is something called multi-party computation, which is a little related to what I just said, but it's actually much more relevant to today. So let's talk about the fact that we are now living in this data-driven society and different parties, it might be different hospitals or different national agencies, and they have a lot of data. If you think about hospitals, it could be one hospital has my genomic information and another hospital has my blood type, my blood test over the years. Another hospital might know something about illnesses that I have experienced. And they would like to compute something based on this data, but they don't want to reveal to each other the data. Another example might be that I am the tax authorities and you are the immigration office and somebody else is, I don't know, another governmental agency. And because of regulations, they're not allowed to share their information. Still, they would like to compute some function that's based on all of the data together.

That's what we call multi-party computation. There's multiple parties, each one has data which is confidential, and they want to compute some function that depends on all the data without revealing it to each other. It turns out that it can be

done. And it can be done partially . . . there's a little bit of algebra involved, it's beautiful theory, but what does zero-knowledge have to do with it? If everybody's honest, it can be done. It's an interesting method of how. But what if somebody's not honest? Maybe they're not following the protocol. Well, you just tag on zero-knowledge proofs to each one of their messages, and then even if they are potentially dishonest, you will be guaranteed correctness because they will be caught if they deviate from the protocol.

Rosen: Did you foresee the generality of the method at the time?

Goldwasser: No, no. It's way . . . way ahead of its time.

Rosen: And again, what was the reaction back then?

Goldwasser: About multi-party computation?

Rosen: Yeah, to these new revolutionary ideas.

Goldwasser: First, there was a paper by Goldreich, Micali, and Wigderson, who did this multi-party computation based on the existence of Oblivious Transfer. That got in. I think it had strong reaction. I mean good reaction. But then there was a follow-up paper that is by myself, Miki Ben-Or, and Avi Wigderson which happened at a time that I was visiting Hebrew University on sabbatical, and that did not have computational assumptions.

So there was a sort of a partition, within theoretical computer science, maybe less so these days. Some of them are so intrigued by the concepts and they're willing to make assumptions like the existence of one-way functions or that it's hard to factor integers and so forth. Others, such assumptions discount results for them, so when you can prove an information-theoretic result without assumptions, they're happier. So I think that the fact that there were information-theoretic analogues was very helpful for this whole theory to be adopted.

Rosen: Okay. Before we move on, I'd like to ask more about applications.

Goldwasser: Actually, I want to say something more about zero-knowledge.

First, it was intellectual curiosity. Then Fiat and Shamir realized this is important for preventing identity theft. Next step was that this enabled a conversion of protocols from honest parties to potentially misbehaving parties. But then all of a sudden in recent years, it had some very unusual usages.

One of them was by some researchers in Princeton together with Boaz Barak where they talked about the use of zero-knowledge for nuclear disarmament. Now it sounds like, you know, out of nowhere. The idea there is that you want to be able . . . let's say the Russians and the Americans want to make sure that they are disarming

nuclear warheads, but they don't want to show each other the technology. How do you prove that a nuclear warhead is in fact a nuclear warhead without looking inside? It sounds like you want to prove a statement but give zero-knowledge. And it's not just by association. There's actually a concrete method that they use which uses a lot of underlying principles from the mathematics of zero-knowledge.

Another example, which Moni Naor from Weizmann came up with, is suppose you are a suspect in a crime and you want to prove that you did not commit it, so they are asking you to give some DNA so that they can compare it to the forensic. The point is you don't want to give it because maybe you are planning on doing a crime in the future or your children are. So how do you prove that you were not in the crime scene, or your DNA does not match without actually giving the DNA? Again, zero-knowledge is the answer.

So there's all these applications all over the place. The last application is the blockchains. Today, as you know, there's this whole idea of Bitcoin, blockchains, how do we put transactions out on a blockchain so that they are serialized in time? And some of the questions are, okay, so you want to put transactions, or transactions meaning things you've done, you want to have records that everybody can see. But sometimes you don't want everybody to know the details of the records. You might want to prove that two records are the same, or other properties of the records, and you want to do that in zero-knowledge. So it has actually become very well known to people in the trade these days and there are even companies that specialize in zero-knowledge.

Rosen: And also digital signatures?

Goldwasser: Yes, also digital signatures. Yes. So what are you asking about that?

Rosen: Fiat–Shamir, the standards digital signatures over the Web is based on ideas going back to zero-knowledge, the ones that started in the late '80s.

Goldwasser: So digital signatures were invented, as I said, in Diffie–Hellman's paper. Then RSA had implementation, but there was really no definition of security. So obviously . . . it shouldn't be forgeable. But what would that mean exactly? Let's say someone's a notary public, so they're able to sign. You want to make sure that even though I can go to the notary public and give him documents at will for them to sign, that I am not able to learn how they sign and be able to sign any other document in the future. This is what we call digital signature secure against chosen message attack. In other words, I can choose the documents that I feed the notary public to sign and yet, even though I see polynomial number of signatures, I'm not

able to produce yet one more document for which I sign it without the help of the notary public.

Rosen: And you came up with the first definition of what this means.

Goldwasser: Definition and construction, we had a way to do it.

[Editor: At the time, this notion of "existential forgery" was considered paradoxical and it was not clear if it could be achieved. Indeed, as in the case of encryption and zero-knowledge, the utmost robust notion of security was coupled by a proof of feasibility under better understood assumptions. That is, robust definitions were coupled with constructions that achieve them under widely believed assumptions such as the infeasibility of factoring.]

Rosen: And then eventually it became crucial to the development of electronic commerce over the Internet.

Goldwasser: Absolutely.

Rosen: Okay, so moving onto information-theoretic and unconditional results. Maybe first we talk about geographically, where are you located now, your area?

Goldwasser: Yeah, so this is 1986 and I . . . Actually, we should talk about primality then before.

Rosen: Right. So let's talk first about primality?

Goldwasser: Yeah. Okay, so as I told you, interactive proofs, or maybe I didn't mention it, but we were talking about the fact there's a prover and there's a verifier. The verifier is tossing coins. They go back and forth. The big distinction of interactive proofs from classical proofs is that there is a probability of error. I proved to you something and with very, very high probability you know it's correct. Or another way to say that, there's a very small probability that I managed to cheat and prove an incorrect statement. That's what enables zero-knowledge.

So, as I told you, I was always interested in number theory, and there was this problem around, which was how do you test numbers for being prime? And a beautiful old result by Solovay and Strassen and Rabin are algorithms for testing numbers whether they're primes or not, fast algorithms that have a probability of error. So at the end, you run this algorithm, you know with very good probability that your number is prime. In fact, what it is, is that if it's composite, you're likely to detect that it's composite, and if you don't detect that it's composite, you say, "It's probably prime." So an interesting question was can you have a primality test that doesn't have any probability of error? Can we test that a number is prime or composite and be 100% correct? And can you do that without actually factoring

the number? That was work that I really enjoyed tremendously and did with my graduate student Joe Kilian at the time.

Rosen: And do you want to tell us more a bit about it . . . the story?

Goldwasser: Yeah. I was in a conference again. As I told you, sitting in lectures really works well for me. I was in a conference, and René Schoof gave a talk about some algorithm he had for taking square roots mod p for small numbers. It had something to do with elliptic curves over finite fields, which was something I knew nothing about, but he described what an elliptic curve was and he had some algorithm for counting how many points are on a curve. And this whole elliptic curve was defined with respect to a prime. So there was some equation, you know, like y^2 is equal to x^3 plus ax plus b mod p, and you could count the number of solutions (y, x) in this defined group, and he was doing some operations on the group.

In any case, he had an algorithm. And when I was sitting in this lecture, I started thinking to myself, "What if you'd run this algorithm mod p, except you didn't know whether p was a prime or composite? How would the algorithm perform? Would it work? Would it not work?" And I asked him that question. I think it sounded like a really weird question and he was like, "Well, it probably would be garbage if you ran it mod p where p was composite."

So then I went back to Cambridge and I think I invited Schoof to come and give the talk at MIT. And he came and gave the talk again, so I understood a bit more. Then I start talking to Joe about the question of what if this prime was a composite, and we start talking about how to use these elliptic curves working mod a modulus which we're trying to tell whether it's a prime or composite, and then the rest is history. We had a primality test based on elliptic curves that was randomized but there was no error probability.

Rosen: That was in '86?

Goldwasser: That was in '86, yeah.

Rosen: Okay. And then what?

Goldwasser: Then what? So then just, you know, it was '86 or '87 and I haven't been in Israel for many years. I used to come visit, but I was really pining away in some sense to being in Israel for some extended period of time. And I had a sabbatical and I decided to spend it in Israel. And I came to the Hebrew University and there, there was Avi Wigderson and Nati Linial and Michael Ben-Or. I didn't know what I was going to work on. I was teaching a course about primality and elliptic curves, and they were very excited because elliptic curves were creatures that they didn't

use in computer science. They haven't been used that much either since, but in any case, I was teaching this class.

Then I remember that I was in Avi's office and he asked me this question. He says, "What else is there to do in cryptography? Because we've already done encryption and we had like good definitions and signatures and identification schemes and zero-knowledge, and what else is there?" So this is a question for some reason people ask many times, many years later. At that time sort of under the pressure of the moment, which was always very good for me to be asked questions under the pressure of the moment, [laughs] I answered like, "Well, you know, we make assumptions, and maybe we could make some sort of physical assumptions rather than computational assumptions like that factoring is hard, and we could prove results absolutely."

Somehow that conversation led to two different papers. One of them was, when I told you about interactive proofs, I told you that there was that result that said that you can actually prove any statement in zero-knowledge using an interactive proof if one-way functions exist. If you like, if factoring is hard. And that's a conditional result, right? So one question is, can you do it without any assumptions? Well, what we came up with at the time, and this was with Joe Kilian also, was this model where there wasn't a single prover and a single verifier, but there were two provers. Now that sounds weird. Like, why two? You know, anyway this prover is supposedly very powerful. Why does he need another powerful friend?

So there was this idea that these two provers, they are like committing a crime. What's the crime? The crime is that they are trying to convince you of an incorrect theorem. And just like the police, the police is like the verifier, it's interrogating these provers. In order to check that their alibi holds up, they put them in separate rooms. They ask some questions from one, you know, potential criminal, and then they go and they ask the other, and they compare the answers. Now, this defines a model. What's the model? We have two provers. We have one verifier. The verifier can ask questions from each one depending on the question he asked the other, and the restriction on the two of them is they can't speak to each other.

That's a new definition of a proof system. We still want there should be proofs for correct statements, and there shouldn't be proofs for incorrect statements no matter what these two guys. But now we have an assumption, except it's not that factoring is hard but that these two guys are isolated from each other. And of course I had some idea that it's not so bizarre, because we can think of an ID card, because I was thinking about Adi's motivation—that instead of having one ID card, you would have two of them and you put them into a bank machine. There were already bank machines at that point. Which might not sound interesting to you, but ATMs are also an invention that occurred during that time. [laughs]

Rosen: I'm not that young.

Goldwasser: You're not that young. [laughs] Okay. Neither one of us. In any case, so there are two cards, and you think about there's two cards, there's two provers, they're proving that they are Shafi. And the ATM is the verifier and it could make noise so they can't talk to each other, they can't see what questions are being asked. We had a patent on this.

Rosen: Okay, so I think maybe now maybe we can actually go down the line with this line of research and then I'll go back to the other area later.

Goldwasser: Right, right. In any case, we had this model, the two provers. Why did we invent this model? Because it turned out that you could prove that every theorem that has a short proof, can be proved in this model in zero-knowledge. That is, there is a two-prover interactive proof, where these two provers are in separate rooms, and they're going to convince the verifier of the correctness of the statement without giving him the proof in zero-knowledge, no assumptions. Okay, so there was a system. We did it for zero-knowledge in order to remove the assumptions like factoring is hard.

Then there was a paper by I think Fortnow, Rompel, and Sipser where they asked how many rounds you needed for this two-prover system. Then a whole bunch of results started to follow.

And then there was this incredible, incredible result by Noam Nisan, who was a postdoc at the time at MIT. What he showed was you can, with a two-prover system, prove the value of a permanent to a verifier. Now I don't want to get into the technical definition, but this is a very, very hard problem. It is extremely . . . It's beyond NP. And all of a sudden it seemed like. . . . And additionally it's a complete problem for counting sharp-P class and it seems like the two provers were extremely powerful. And what followed after that is that using the techniques that Noam used, within sort of a whirlwind of results it has been shown that this class of interactive proofs with a single prover was as powerful as polynomial space. And then again, within months or weeks, it was shown that this class of two-prover interactive proofs was as powerful as non-deterministic exponential time. All of a sudden, these weird creatures that we've introduced with provers and verifiers and interactions and people locked in different rooms were sort of grounded in the traditional complexity theory with classes like polynomial space and nondeterministic exponential time and equivalences were shown.

Rosen: And how did you feel at that time?

Goldwasser: I thought it was . . . first of all, the mathematics was fantastic. It was really new matter . . . It was arithmetization, expressing decision problems using polynomials. So the math was fascinating and I thought that . . .

Rosen: You were pleased?

Goldwasser: I was pleased. Yes, I was very pleased.

Rosen: Okay. Let's continue on that line and then we'll rewind back.

Goldwasser: Yeah, then I had a couple of years later, I think it was like 1990, I was in Princeton for a sabbatical and I think Joe Kilian gave a talk there about something about . . . I can't remember anymore. Some two-prover proof system in nondeterministic exponential time. And there was something about his talk that made me think that you could sort of simulate nondeterministic . . . you could do all nondeterministic exponential time in exponential time. Which like you would show collapse of these deterministic and nondeterministic classes. And I told Muli Safra about that, who was actually my postdoc at the time I think, and he was also in Princeton.

We started talking about it and then it turned out that that would be true if—now it seems like a rabbit out of a hat—if some graph-theoretic problem was easy to approximate. The graph-theoretic problem is called the clique problem. It's like you have a graph and you would like to find a subset of the graph where all vertices have edges between them. It turned out that if you could approximate the size of the largest clique in a graph, then you could have showed that nondeterministic exponential time was equal to exponential time. Turning this on its head, it says that it's hard to approximate the size of the largest clique in a graph if nondeterministic exponential time is not equal to exponential time. Then when you sort of downsize this, you get essentially a result that says that it's hard to approximate clique if P is different than NP. So there's an NP-hardness result hiding in there.

Rosen: So you sort of started with complexity, went to cryptography, and came back?

Goldwasser: And came back, yeah. And this whole idea of using multi-prover interactive proofs, something that then morphed to something called probabilistically checkable proofs, PCPs, started with that work, and how to use that in order to prove hardness of approximation started with that work. That's become a complete field, which I'm very proud of.

Rosen: Rightfully so. So, okay. So now you want to continue a bit on this thread or go back to the other paper with the . . . ?

Goldwasser: Let me just say a few more things about this. We've talked about interactive proofs, right? Single prover and verifier. We've talked about this multi-prover interactive proof. What is this probabilistically checkable proof? So far, everything was just very general, right? There are these two provers, there's a verifier, they exchange messages, at the end the verifier accepts the proof, doesn't accept the proof, there's some probability of error. But now we start quantifying things a bit. So you can talk about how much randomness is the verifier using? How many coins does it have to toss? You can talk about the length of the messages that are being sent. You can talk about how many questions are being asked and you can talk about the probability of error. And once you start quantifying this, I mean these are parameters, and if you change these parameters, they can be sort of very tightly coupled to the problems that you can either approximate or nonapproximate.

But let me say it in a different way. There's this third creature, which I mentioned, probabilistically checkable proof. What is that? There the idea is much easier to understand. In a sense, it doesn't require the stories of provers and verifiers and so forth, even though I love stories and I would never have got into any of this without stories. So probabilistically checkable proof, the idea is the following. Usually people think of proofs, mathematicians think a proof is a string that you can read in a book, right? It starts from statement one, statements follow, and then QED. Probabilistically checkable proof is a way to write a proof in such a way that you can actually . . . you don't have to read the entire proof. You can probe it at some locations, not in all of them, and you should think of it as if I'm choosing these locations at random, and make some check on those locations you've probed, some local checks, and if there is a mistake in the original proof, there's a very good chance you'll find a mistake in the local check.

So it's these proofs which are probabilistically checkable because you're sort of choosing the locations at random, and furthermore you have to read a lot less than reading the entire proof. Of course, you don't get certainty. You get probability of error. And now the kind of parameters that I talked about a minute ago come into play. How many places in the proof do you have to look at? What is the probability of error? What are the sizes of the questions and answers? And these are parameters that, in the original paper that I had with Muli, and then with Lovasz and Feige who joined . . . , we joined forces, these parameters were improved, and subsequently more by work by Arora and Safra and then by the well-known paper by Arora, Lund, Motwani, Sudan, and Szegedy to be sort of optimal, where you really need just log n randomness and look at constant number of bits of the proof and you will catch a mistake if it exists.

Rosen: Now let's rewind back to the late '80s to the second result you were alluding to.

Goldwasser: Right. That's a result with Ben-Or and Avi, and that's about how to do multi-party computation, the same problem I told you about with the different hospitals that want to compute some function of their data without sharing it. What we showed was how to turn this problem into an algebraic problem where the data that you have is represented as essentially shares of a polynomial. This is called secret sharing that was invented by Adi Shamir. It is a way to take a piece of data and share it among n people so that only looking at some of the shares you have no idea what the data is, but if you have sufficient number of shares you can reconstruct it.

But Adi's secret sharing was just a way to share data. What we were asking is how do you compute on data? So now we have these three hospitals. Let's say each one of them has shared their data, secret-shared among all three. But that's not enough. They want to do a computation on it, like they want to do maybe some linear regression or they want to find out how many patients are there whose DNA is of a specific type and it had infections in the past and their blood test is in a certain range. So they want to do maybe set intersection or something like that. You can write any such function as essentially a sequence of operations on the data, which essentially looks like summing and multiplying.

What we realized is how you can take these shares of secrets, which were essentially values of polynomials, and compute with them. How can we add them and multiply them where each of us only has their shares? I have the shares of your data, I have shares of everybody else's data, and using these shares I can essentially compute a share of the sum of the data, a share of the product of the data. I can keep doing this iteratively, so essentially any program that we want to run on this data can be run in such a way that at the end I will only have a share of the result and I will have learned nothing about the data except for that share of the result. And since all of us have shares of the result, now we can reconstruct the result. That means that I knew my input, I'm going to know the result, and I can tell whatever is implied by knowing my input and the result, but nothing else. And this is . . . It's important. [laughs] Yeah.

Rosen: Why is it important?

Goldwasser: Again, for lots and lots of applications these days. If you want to connect it, if we kind of zoom to 2017, you know all the rave now is machine learning, right? Everybody's talking about these neural nets and logistic regression and how it is going to change our lives, for medical, for actual medicine, precision medicine, for targeting consumers, for making decisions on who to set on bail and so forth.

But there is a question, and that is a lot of this is driven by the fact that we have tons and tons of data about people, and this data sometimes should not be shared. And it's held let's say by either individuals or by entities that even are bound by regulation not to share it. So how are you going to get them to use their data for running a machine learning algorithm without sharing it; that is, in a way that respects the privacy of individuals?

The technique of multi-party computation is essential for that, because you may think of coming up with a machine learning algorithm, let's say in the training phase, taking the data, training on it and figuring out a model that can do predictions as a protocol that has access to data toward the end of coming up with a prediction algorithm, but not for seeing the data explicitly. And multi-party computation because of its generality can be used.

Now there's a difference here between theory and practice. On paper all is good. That is, we wrote papers and we proved theorems. But in order to use it in practice in a way that's efficient enough, you need to do a lot of optimization, you need to improve, you need to implement. Only time will tell if these methods will be used as they are or they will be modified, and hopefully not modified to such an extent that they will be insecure.

Rosen: Well, they are already being deployed in a commercial context.

Goldwasser: Yes.

Rosen: Okay. Now I'd like to ask you about some retrospective about advising students throughout the years. You'd had many great students, well known, very successful, and in several ways, in several generations.

Goldwasser: Alright. First of all, I have had incredible students, and these students, I am thankful for that every day. Early in my career I worked with my colleagues. You know, I worked with Silvio and Oded and Avi and others, so I did not write papers with my students. But now I do. In any case, then the students were really more doing their own thing and I was advising them in the sense that they would tell me about their stuff, and sometimes questions came from me, sometimes questions came from them. Now it's more that I'm in an advisory role, that most of the questions come from me, but the students do a lot of the work. I think that my advising style must have changed because it became much more working together with the students than it was before.

I'm always in awe at the fact that there's a new student and there's a new talent and that they really make something out of nothing. Not in the sense that they are nothing. In the sense that they come up with new ideas and new questions,

and where does it come from? That's the incredible thing of working in university. There's this young generation one after the other, and they are so excited about what they do and they are remarkable. So that's really a gift of being able to be in university.

Rosen: Okay. Can you mention different styles of students, of researchers that you encountered? Different characters?

Goldwasser: Different characters. I've met lots of characters. [laughs] I remember Joe Kilian was really into limericks and a great sense of humor and a very creative, unusual researcher. Then there are people who are very like technically extremely sharp, right? Like Johan, but so was Joe too. I'm mentioning them in the beginning, because at my advanced age [laughs] it's easy to remember the past rather than the present. No, but I've had amazing students really all along. Some of my students are faculty members at Weizmann where we're sitting right now, like Zvika Brakerski and Guy Rothblum, who've both done amazing things. Then some of my students are faculty members at MIT, like Vinod Vaikuntanathan. Then there's Yael Kalai. And I have former students all over Israel, like Yishay Mansour and Adi Akavia, and many others all over the world.

Rosen: So now let's talk about the property testing and delegation?

Goldwasser: Sure. Okay, so property testing.

Rosen: How did it all start?

Goldwasser: How did it all start? I actually think that my first thoughts in the direction of property testing come again to a talk that I attended in Hebrew University, of Michael Kearns' actually, where he talked about learning. He had some model of statistical query learning. In any case, and then I drove back with him to Tel Aviv and we had some conversation in the car that made me start thinking about the question of not learning where you have examples and you're trying to predict a label of a future example, but more about being able to tell a property of whether the examples you are seeing belong to one distribution or another distribution.

Or another way to say about it . . . What do I mean by examples? Let's say that you have a function and you can't look at the function table. You actually don't have a description of the function, but you can query the function in different places. And what you would like to find out is a property of this function. So what could be an example of a function? An example could be . . . let's say there's a graph and I actually can't look at the whole graph because the graph might be extremely large, but what I could apply a function to two vertices and the function will say one if there's an edge between them and zero otherwise.

So that's a description of the graph. It's a function. So there's sort of an indirect description. Now I'd like to ask questions about this graph. Does this graph have a large clique? Is this graph connected? Can this graph be partitioned into two sets of vertices that there's only edges going between the sets and not between vertices within the two sets? It's called a bipartite graph. So that's a property. And obviously some of these questions you have to look at the entire graph. You have to sort of ask the function, the entire function table for every pair of vertices, what the edge is and then solve the problem.

Then property testing paradigm says, "You know what, let's relax the question, because we really cannot write down the whole graph, we cannot query the function in all places. We'd like to tell whether the graph that's being described by this function which I can sort of query is close to a graph that has that property." So if we think . . . Let's look at a specific graph property that's say bipartite. This graph that the function is describing, is it bipartite or is it far from being bipartite? But what do I mean by far from being bipartite? It means that if you look at the closest graph to it by removing edges or adding edges, let's say it's epsilon apart, you have to add epsilon or subtract epsilon fraction of the edges. So there's a fraction of edges that you have to insert or delete, and I'd like to tell which is the case. Is it a bipartite graph or is it far from any bipartite graph? And I'd like to do that by querying the function in very few places.

So for the layman, let's think of it this way. We are not living in the age of dinosaurs anymore, right? We find bones of dinosaurs. Can we just by looking at bones of dinosaurs tell whether the entire dinosaurs was a tyrannosaurus? Was it a meat-eater or herbivore? Apparently people make conjectures based on very little data. So the question here is if I can only look at very little places in the graph, either given or I can query the graph at places of my choice, can I tell something about the graph more globally, like being bipartite or being far from bipartite?

This is the way I like to describe property testing, and that's a field that was kind of started in a paper together with Oded Goldreich and Dana Ron. We wrote on testing properties of graphs and more generally testing properties of natural structures. You know, graphs as a natural structure or other functions are possible too, not just to describe graphs. And we would like to find out whether a function let's say is monotone and we can't write down the whole function table. We can just query the function in a few places. Can you tell if it's monotone or far from monotone? This is a direction that's become a whole field. I mean that paper, I think, was fairly influential.

And then you asked me about delegation?

Rosen: And lattices, if you want to mention some more about lattices.

Goldwasser: So time moves on and people start talking about different models of computation like cloud computing. And the idea of cloud computing is that there are these computers out there and I'm a client, and I'd like to use the computers and they will do all the computation for me and then give me the results. So the clear question is how do I know they are even computing it correctly? I am delegating my computation to an outside computer. I want to get some proof that the result has been correctly computed. We call this a delegation problem, and that's a problem that is a little bit similar to interactive proofs because this computer proves a statement to me. The statement is that it did the computation correctly. That's been a problem that I've been very interested in.

And the delegation paradigm isn't just delegating computation, but you can think about it in other contexts, like you want to delegate in the context of error-correcting codes. Let's say I want to code a message in such a way that even if there's noise on the line, you can detect it. Then there's the question of how much work you have to invest in order to encode and how much work do you have to invest to decode, and you can talk about delegating work of the encoder to the decoder or vice versa. So this whole delegation paradigm is something that I've been interested in in the last, I don't know, 15 years already. And that's been fascinating. This is work with my students Yael Kalai and Guy Rothblum. So that's something that I'm still interested in. I think that this delegation paradigm is very powerful in today's sort of modern computational world.

And you asked about lattices. As I mentioned, the theory of lattices has become a source of hard computational problems. Like if you define some sort of integer lattice via basis, find the short vector in the lattice . . . This theory and these hard problems have become the basis of what we call post-quantum cryptography. And implementing sort of essentially cryptographic primitive based on these type of problems is a fascinating field which I've been involved in.

Rosen: And you were very early on.

Goldwasser: Yeah. This was work with Oded Goldreich, where we sort of asked this question of interactive proofs to show that a shortest vector in a lattice is not so short and we introduced some new methods in this field.

Rosen: You actually, yeah, introduced a method to show that it's unlikely to be as hard to approximate as other approximations.

Goldwasser: Yeah. But in any case, the method is more important than actually the result, because the method is essentially what underlies a lot of proofs of

security in modern cryptographic systems that are the basis of this post-quantum cryptography.

And I want to mention actually one more student, Daniele Micciancio, who was one of my students, which I love very much. He started working on logic actually with Albert Meyer, this was his master thesis, then he came and worked with me about digital signatures. And for his exam . . . There are these exams at MIT which don't exist anymore where you're supposed to give a student a few papers and then they are supposed to read it and do some original contribution within three weeks. So I gave him some papers on lattices and he came up with some beautiful new result proving the hardness of approximation of shortest vector in a lattice, and that became his field of research. I feel privileged to have suggested the problem to him, or the papers to him. I think he's one of the sort of guiding lights in the field of lattice-based cryptography.

Rosen: Okay. You want to mention something more about students?

Goldwasser: I think that I have a new crop of students which are wonderful, and they're doing . . . Today it's actually interesting. A lot of the students are not only interested in sort of the science, but they're actually interested also in impact on society. So this is sort of a modern wave. I mean as you see people, you know there is this generation that's just interested in going to startups and the generation that's just interested in doing complexity theory and then doing cryptography. And the new generation that I have at least, they're very interested in the impact of the methods on today's world. And when I say impact, I don't mean just implementing systems that are run efficiently, but really questions of like how is this going to change the world from a society point of view?

Rosen: So for them, the application might be more of a guideline?

Goldwasser: The application might be more of a guideline but it's not an application that is necessarily only having to do with utility. It actually also has to do with doing good. I mean privacy anyway is doing good, in my book, but it's beyond that.

Rosen: And what's your take on privacy, whether it's doing good, whether it helps?

Goldwasser: Of course it's doing good. I mean, you know the line that I think they attribute to Judge Brandeis, but I think it was Brandeis and another lawyer that they were in a law firm together. This is after the original cameras were invented, the kind of cameras, portable cameras that you could take out of the camera shop. And they wrote this paper about "What about the right to be left alone?" You know, it's very nice that you can take photographs, but now I could have my pbotograph taken without my permission. Now imagine where we are at. Right? Everything we do on

our iPhone, every Google query we make, every email we send is being recorded by these giant companies and they are deriving conclusions from it, like giving us advertising for us. So the right to be left alone is something nobody imagines anymore, you know with all these sensors and the cameras. It really alters our reality and I think we need to think about it.

Rosen: And you don't think it's too late by now to do anything about it?

Goldwasser: You know, it's just like talking about the environment, right? So with the environment, we have a lot of pollution, but somehow it's self-regulating. Not as well as it should be, but there are climate agreements and people don't sell the kind of cars they used to. There's emission controls. So my feeling is that every revolution has at some point people realize that there are some things to fix. And I don't see why the lack of privacy is not going to be the same, because the methods exist. And we can develop more methods. But people have to be aware, people have to kind of pull back, people have to implement these methods on top of the existing ability to spy or to have sensors and . . .

Rosen: And what about the negative implications of the ability to encrypt data and hide it from others?

Goldwasser: I guess the negative implications is that we could go dark, right? This idea that now that the encryption methods are being developed and they're so strong and they're so well known, that we won't be able to pursue criminals, right? So being able to read messages, being able to wiretap, being able to listen to digital communication is a police tool. It is a national security tool. We all know there's more and more threats. So by enabling this encryption for the public, you are in a sense making it more difficult for law enforcement to behave. I buy it, but it's a very thin line, right?

On one hand, privacy has so many good outcomes. It's enabled electronic commerce. It's enabled a use of remote computers for delegating computation. It's going to enable doing machine learning on data while keeping it private. On the other hand, there are these criminals who should be caught and we should enable law enforcement to catch them.

How do you reconcile the two? One opinion is that you just say, "Well, tough. Let the law enforcement figure out other methods to catch criminals and don't give up on privacy." And another point of view, which is the other extreme, is let the law enforcement have all the keys to all the encryption algorithms out there. And maybe there's a third sort of economic model where you sort of think of cost–benefit analysis and you're able to trade it off, so you can sort of trade off privacy

in policing. I don't think people have looked at it, but just again, if we go back to the example of environmental science, there is sort of a cost–benefit analysis of putting regulations, and there are resources that are renewable, resources that are not renewable, and there's measures. So this is not really my expertise, but I can imagine a world where that kind of theory is developed also with respect to privacy.

Rosen: What about the future?

Goldwasser: That's the thing about the future, you don't know do you? As we say in Israel, "all will be well." [laughs] No, you're asking about the scientific future.

Rosen: Not necessarily.

Goldwasser: Not necessarily. The future is that I'd love to continue doing research. I love interacting with young people, with postdocs, with graduate students. I'm still inventing new questions. We haven't talked about them, but that might be in another interview. And I still get excited from new questions and new answers.

I'm looking at what has happened to cryptography. It's kind of amazing in terms of the number of people and the impact and the excitement, so this is sort of a future which is inevitable. There's no question that cryptography has a future. And personally I hope to do more. I hope the field will do more. I'm very optimistic.

Rosen: Where do you see yourself five years from now?

Goldwasser: You know what, I think that's the one question I can't answer. [laughs] I don't know.

Rosen: In terms of aspirations, just . . . ?

Goldwasser: I want to keep on working. I want to keep on creating. I want to have ideas. I want to have impact, and the kind of impact that I'm talking about now is also impact as let's say the director of the Simons Institute or someone who directs . . . someone who has some influence about where the field is going in the sense of what's important and what's not important. I think that I've had a good hunch and I feel I have an intuition to serve me and also a lot of experience. So if I have made impact in the next five years both in terms of research and in terms of leadership, if my kids do well and they're happy, then I will be very happy in five years.

Rosen: Okay. Thank you very much, Shafi.

Goldwasser: Okay. Thank you.

4 An Interview with Silvio Micali

This is a full transcript of an interview of Silvio Micali by Stephen Ibarkaki. The interview took place on October 15, 2013. The transcript was lightly edited for clarity.

Ibaraki: Welcome today to our interview series with outstanding professionals. I'm Stephen Ibaraki, and I'm conducting an exclusive interview with Professor Silvio Micali, ACM Turing Award recipient in 2012. The Turing Award is widely considered the Nobel Prize of computing. Professor Silvio Micali is also a world-renowned, distinguished researcher, and a professor at MIT.

Now, Silvio, you have a lifetime of outstanding research contribution with lasting significant global impact. Thank you for coming in today and sharing your considerable expertise, deep accumulated insights, and wisdom with our audience.

Micali: Thank you, Stephen. It is a pleasure talking to you and your audience.

Ibaraki: Now, Silvio, you have this extraordinary honor now. When did you hear about this, and how did you feel at the time? What was the reaction of your colleagues and your family?

Micali: Well, I heard about it on a Friday afternoon. We were planning to leave for a family ski trip with my colleague Shafi, my co-recipient of the Turing Award. And then the telephone rang . . . So it was quite a coincidence, you might say.

How did I feel about winning the Turing award? What can I say? I felt good. I felt good in particular to have won it with Shafi. You must know that we were graduate students together. We worked for many years and overcame many difficulties, even multiple rejections of our work, before we got an award. And so I was very happy to get the award together with her. Shafi and I had good interaction. You know, we were trying to develop a theory of interaction, it takes two to interact, and when you interacted with Shafi you were actually interacting with at least seven people,

(laughter) depending on which of her multiple personalities were in charge on that day. So that was how I personally felt.

About the feelings of my colleagues, I actually was very happy to see that there was a very large, positive reaction. You must know that we are a very interactive community. We collaborate a lot across institutions, so, what can I say? I put a premium on their opinion and I'm glad to see that it was positive. Some of my colleagues were actually so kind, almost happier than we were. Of course, some of them did not react at all. So, some may have disagreed on the importance of our results, or taken them for granted. Whatever the case, it's important to have dissenting opinions, right?

In sum, I felt that the overall response was very positive. And my family was ecstatic.

Ibaraki: Well, I can see how your family would be very pleased, because you're a legend, you're an icon in the industry, and, of course, you're part of the historical record forever. [laughter]

Micali: Well, maybe not forever. But it's good enough for us, right?

Ibaraki: Now, Silvio, how will the ACM Turing Award impact your work, your influence, and your thinking?

Micali: Oh well, to tell you the truth, on the one side we should strive for absolute truth and novelty. But on the other side, you know, we should strive, or at least I do strive, also for universal recognition. Somehow, the coexistence of these two goals is good, in my opinion. If the pursuit of absolute truth required disregarding social judgment, then we would have a lot of trouble on our hands. OK, greater recognition and strife for truth can be antagonistic. In the short term, somehow, if you choose universal recognition, then you have to work on problems everybody perceives to be important. In other words, that choice requires pursuing a more established and conservative line of research. So: What do I hope from the Turing Award? That, taking care of some of my desire for recognition, it leaves me free to go on a limb and take some more scientific risk, to go and explore new wildernesses, so to speak. This is the impact that, I hope, the Turing Award will have on my work.

As for my influence, let's see . . . First of all, you know, I have nothing against recognition or having some influence. After all, we work very hard to increase our reputation. This said, my peers [laughter] will continue to judge my work according to strict standards, as they should. However, I do see that the Turing Award can actually give me some additional influence on researchers outside my field. So, I

hope I can use this additional influence wisely when interacting with scientists from other disciplines.

Finally, if I may add another thing, awards tend to make us feel good. And if we feel good, we can do more things, have more energy. So I hope to put this extra energy to work in my thinking, my teaching, and everything else.

Ibaraki: Now, again, you have this amazing body of work, and you've got this significant achievement in the ACM Turing Award. From that, then, what are your life goals that you want to achieve, and how will you achieve them?

Micali: Oh, wow, life goals? . . . This is a hefty and difficult question, Stephen. In fact, it's so personal that if I answer it truthfully I will be a little bit enigmatic, OK?

My goals essentially are to understand the world and to be understood. And these, in my mind, actually are quite the same goal. So how to achieve understanding myself and understanding others? By really getting into the minds of others, and letting them into mine, if I can. And through a combination of supreme confidence and supreme doubt.

Ibaraki: OK, we're now going to talk about your work that led to the Turing Award. And the first question is: What led you to co-write one of the most influential papers in computing science as a graduate student in 1983?

Micali: All right, if you want me to outline [laughter] the story of that work, I'll tell you, it is a tale of fearlessness and shamelessness, luck and ignorance, everything combined, OK?

Let me start with luck. You know, I'm not ashamed to start with luck, because nothing substantial can be accomplished without it. My good luck was to be in Berkeley, in a wonderful atmosphere, with fantastic teachers and great fellow students. In particular, I was lucky to be in a course taught by Manuel Blum on computational number theory, whose last three lectures—maybe four, no more, actually—were on public-key cryptography. Cryptography at that point was not that developed, at least in academia. Manuel was an absolutely inspiring teacher, and cryptography was an incendiary material. So it was a match made in heaven. [Laughter] If you'll allow me the pun, the match lit.

So that was how we started. A problem mentioned in class was that of mental poker. In other words, can you and I play cards over the phone, or by email? There was an approach to this problem proposed in the past, but it did not quite work. So Shafi and I decided to solve it. That's where fearless and shameless come in, right? Because the problem was actually very hefty, and a satisfactory solution would've

taken years of further development and many more techniques than we had at the time. And we ultimately built those techniques, but at the time our youth and inability to properly size the problem were a big help in taking on this challenge. Simplifying things, we essentially thought first about encrypting the cards, and then about implementing the *dealing*, the random shuffling of the cards. The first step was actually challenging enough to make us understand that we needed a new encryption scheme and a new notion of security.

OK, without getting into too many details, encryption at that time was deterministic. This means that every encryption method used to have a single ciphertext corresponding to a given message. What made a ciphertext hard to understand was the "length" (technically speaking, I should say the "entropy," but never mind) of its corresponding message. In fact, you can imagine that it is hard to guess a long message in its entirety, right? Yet, with deterministic encryption, if you were lucky, if you guessed the message in its entirety, then, being encryption deterministic, you could actually verify the correctness of your guess. What makes mental poker really challenging is that the possible "messages" are only 52, because there are only 52 cards. So, in this application, it's easy to guess the intended message, because it is easy to cycle through all 52 of them, right? In other words, in this application, the message space, so to speak, is very, very sparse. And so we decided that if we wanted to encrypt such few messages, then we had to encrypt them probabilistically. That is, we had to flip coins to choose a ciphertext of a given message.

Think of it like this. I have not just one way to encrypt a message, but I have many, many, many, many, many ways, exponentially many ways (in the number of coins you toss), and I flip coins to choose which one to use and then send you the corresponding ciphertext. Now, a fundamental property should be that, even though every message can be ciphered in so many ways, from every single one of its ciphertexts, you can actually retrieve the original message that I send you. That, essentially, is the idea.

Actually, we decided to further generalize the problem at hand and considered a worse situation. How about having only two possible messages: say, 0 and 1? That is, if you want to encrypt a single, randomly selected, bit? What should we want from encryption in such a case? We should want to make sure that, from a ciphertext, one should not be able to guess the corresponding bit with probability better than 50–50. Mind you, that everybody can always get the bit correctly with probability 50%, right? Indeed, even if you don't know anything about encryption at all, when you see a ciphertext, you flip a coin and say, "If heads, I predict zero; if tails, I predict one." You flip the coin, and you'll be right with probability one-half. So to claim that you are "breaking" the encryption scheme, you must at least

do a tiny, teeny better—"epsilon better", as we say—than 50%. Perhaps, you must be able to correctly guess the bit with probability 51%, or 50.1%, or 50.001%, or something like this. A one-bit encryption scheme should be considered secure only when it is practically impossible to have even such small advantages over random guessing. (This essentially started our development of the notion of *computational indistinguishability*, as we called it later on.)

We then proved the following theorem: Namely, if we can encrypt a single bit in this way, then we can as securely encrypt arbitrarily many multi-bit messages. The underlying proof technique came to be known as *the hybrid argument*. Thanks to this theorem, to the hybrid argument, all that remained was finding a candidate scheme for encrypting a single bit. The ability to securely encrypt arbitrary message spaces would automatically follow.

Here is where ignorance actually came to the rescue. And not only ignorance, but luck again, of course, because knowing a lot of things is tantamount to having a haystack in your mind, right? And among so many, many, many, many pieces of straw, you look for a special one, "the needle." This means trouble because you might never find the needle among so many pieces of straw, or you may find it when it's too late. Shafi and I were lucky, because we wanted to construct a candidate one-bit cryptosystem based on computational number theory, and we didn't know much computational number theory. So, if some facts at all could be put together to construct our cryptosystem, we had to choose them from the very few facts we knew. We got lucky, because the needle was possible to find in our small stack. The needle we zeroed in was the *quadratic residuosity* problem.

Essentially the problem is distinguishing squares from nonsquares modulo N, where N is a large integer whose prime factorization you do not know. I will not bother you with the details, but you can easily disregard some numbers from being squares modulo N, but for another half of the numbers modulo N, when N is of a certain form, it is not at all clear how to distinguish squares from nonsquares. Thus, we thought that the difficulty of making such a distinction might be useful to encrypt a single bit. But: Was the quadratic residuosity problem really computationally difficult?

We started by asking our advisor, then we started asking our other authorities, and somehow nobody knew how to solve the quadratic residuosity problem. So we said, what the heck? Let's assume it is computationally hard and build on it our candidate cryptosystem. We took a risk. The danger was that, after publishing our system, somebody could come up the next day and say, "What are you talking about, quadratic residuosity? Here is how to solve it." But we took the risk. Again, we were young, so we didn't have a reputation to maintain yet, or perhaps we

disregarded our reputations, or whatever. So, with ignorance, luck, and risk taking, things worked out.

By the way, today we know much more, and if quadratic residuosity were to become easy tomorrow, it would not be a problem, because at this point we have enough candidates to base our cryptosystem on (in fact, we have a way to distill them). So, in some sense, timing was crucial, and timing is another form of luck, right? Again, I think I made it abundantly clear, I strongly believe that, never mind all our good deeds and whatever we do to deserve our successes, luck has a major part. I am Italian, right? My ancestors, the Romans—I mean, were very determined people. They conquered a lot of the then-known world, but at the end they really knew whom to thank, and they built a monumental temple to luck, to Fortune. If you go to Rome, take a trip to nearby Palestrina. There is an entire mountain transformed into a temple, the temple of the *Fortuna Primigenia*. In the end, luck matters.

But then, you know, you have to work for your luck. So, Shafi and I developed various techniques, in particular *random self-reducibility*, to help us prove that quadratic residuosity, the problem we selected, really had all the properties we wanted. We came up with the hybrid argument and with computational indistinguishability. These actually were techniques that we introduced in our work on probabilistic encryption for a particular context, but that also proved crucial in subsequent and harder contexts. So in some sense, we were wise, or lucky again, to use them in a simpler problem to begin with.

Ibaraki: Well, it's a particularly amazing piece of work. It reflects an inflection point in history, the work that you did. And when you talk about luck, I guess that's where preparation and opportunity meet, so . . . [laughter]

Micali: Absolutely. Luck favors the prepared, [laughter] but luck is needed anyway.

Ibaraki: Now, can you provide added details behind your approach, *the simulation paradigm*?

Micali: Sure. I actually find the simulation paradigm the most natural thing. Let me forget mathematics for a minute and put you in the right mood. It's a simple concept, really. It's a very human concept. So let me recall a personal episode, which I'm sure is actually common to all of us, and yet is very personal to all of us. Here we go.

I remember, when I was a kid, of somehow getting an acute attack of classic solipsism, which is a fancy way to say that I started being fearful that there was no outside reality, that it was all in my head, that I was alone, that the world was a product of my imagination, etc., etc. You know, it could very well have been a

power trip. I'm sure it was. But somehow, at the time, I recall the feeling to be one of loneliness and despair. So my mother got to work: You know, it lasted a few days . . . She sat next to me on my bed and said, "I listen to what you say, but I'm here. I do exist. Let me help you." And I said, "No! You're not here! I place you next to me on my bed, I'm letting you say these things," and so on and so forth. I eventually got out of it, but somehow I was able to positively turn all these feelings into science.

What impressed me at the time—I remember this distinctly—was how impossible it was to break that symmetry, I mean, to decide which was virtual and which was real. And, if I cannot distinguish the real from the virtual, then in what sense could they be considered different? Somehow that thought stuck with me. Fast-forward a few decades, and we have the simulation paradigm.

So what is the simulation paradigm? Essentially, it is the technique that ensures that no information, or not much information, is leaked in a cryptographic interaction. In cryptography, there is no you, Stephen, or me, Silvio. What distinguishes you from anybody else is a secret that only you have. It's called a secret *key* in cryptographic lingo. It is a secret number that only you know, and you use it to send your messages in a cryptographic transaction. Of course, nobody's going to be so dumb to send his own secret key along with his messages in a cryptographic protocol. But you use your secret key to generate the messages you send. In some sense, somebody who sees the messages you send essentially sees a kind of shadow of your secret, projected on an imaginary wall, a hypothetical wall. And perhaps, if an adversary sees enough shadows of your secret key—say, from many angles—then he could reconstruct it.

Indeed, I may not know the shape of an object, but after seeing its projection onto one wall, other projections onto other walls, I start getting the zest of it and become able to reconstruct the unknown shape. So, when you're taking part in a cryptographic protocol you are in a bind. If you never use your secret key, the secrecy of your key is guaranteed, but you are not doing anything that's cryptographically relevant either. On the other hand, if you use your secret key, which you must do to accomplish anything of interest, you actually reveal shadowed images of your secret key. So, will it remain secret at the end?

The solution of the riddle is to send messages using your secret key in a way that the adversary, without knowing your secret key, can simulate you, can reproduce what you say in essentially the exact same way in which you say things. So by watching you, the adversary watches your reality, but you ensure that he, without knowledge of your secret key, is able to generate a virtual reality that is actually identical to the one you generate for him. And if you succeed in acting in a simulatable way, then your secret is secure. Why? Because if the adversary could imitate

what you say without knowing your secret key, then what you say cannot inadvertently betray your secret key. So that's the whole idea of how to ensure that the amount of secret information you reveal is "contained." This containment is what the simulation paradigm gives you.

So if you go back now to solipsism, I could not decide whether [laughter] the world is real or I am making my own virtual reality, but at least I could put this impossibility to good use. Because the impossibility of distinguishing reality from a creation of our mind is our best way to guarantee the security of a cryptographic protocol.

May I abuse your patience a little bit more to give you a concrete example of how to apply the simulation paradigm? Consider public-key encryption. What do you do in this setting? Assume that you select a specific secret message to send me. You probabilistically encrypt it in my key, and then you send me the resulting ciphertext. Call it C. C is a good acronym for a ciphertext. Assume now that there is an adversary in between us. Then, what is his real view? His real view (besides my public encryption key) is this string, C, the ciphertext that you actually so produced, OK? However, nothing stops the adversary, without ever seeing C, from choosing a random message, a creature of his own mind without any objective reality; then, from encrypting it probabilistically using my public key, so as to obtain a virtual ciphertext D that nobody sent; and, finally, from looking at D. So now the adversary has actually two worlds: one, C, that you created by encrypting your specific message—M, call it; and another one, D, that the adversary himself created probabilistically by encrypting a random message. And if a cryptosystem guarantees that his real view—the ciphertext C that you sent—and his virtual view—the D that he himself created—are essentially indistinguishable, then the secrecy of your specific message is safe, right?

Simplifying a bit, this is what the simulation paradigm means in encryption, but the principle is the same across other applications. It may actually become a little bit harder to implement and to grasp in these other applications, but the idea is the same.

Ibaraki: Now, can you further describe your notions of encryption security—for example, *semantic security* and *indistinguishability*—and how these measures must be met for schemes to provide security across a wide range of cryptographic applications?

Micali: All right, so we are going from technical to more technical. OK, let me try.

Semantic security is essentially what you intuitively want from an encryption scheme. In some sense, it extends Shannon's notion of perfect secrecy, which was applicable only to a very constrained scenario; namely, when a sender and a receiver

share beforehand a string of random bits, and only need to encrypt messages whose total bit length does not exceed that of their shared random string.

When somebody is going to transmit a message, we have, from context, a probability distribution—we call it a message space—of what he is going to say, right? Consider all messages that are a thousand letters long. Then, some messages actually have probability zero—for instance, those that contain five consonants in a row, just because one cannot even pronounce them. Of the remaining messages, some have higher probability than others, depending again on the contexts we are in. In sum, there is an *a priori* probability distribution from which the message sender is going to choose his message. In this setting, you want to guarantee not only the secrecy of the chosen message in its entirety, but also that of partial information about the message.

So, what should this partial information be? You can think that it's a function from the message space to some other, perhaps smaller, space. For instance, you may be satisfied to figure out whether the sender's message is about attacking or retreating, or whether it expresses worry, and things like this. (Indeed, you would like to understand that your enemy is worried, even though you cannot quite understand what he is saying.) For simplicity, assume that this "partial information" function F you are interested in maps any message into a number between one and 1,000, say, OK? Even if you're not able to decrypt the message sent, you may be satisfied to learn the value of F on the message sent.

Now consider the following situation. Assume that somebody tells you that the sender has selected a message m from the message space, and has sent it by magic, by teleportation, to its destination. So, what is the value of $F(m)$? If you would like to win this game, what would you answer? You would say: "Well, if I try to be as right as I can be, what is the most popular value, the most probable value this F can take?" Since F maps every possible message to a number between one and 1000, and since you know from context what is the probability distribution over all possible message, you figure out that, say, maybe 727 is the most popular value of F, and it occurs with probability 2%. So, if you answer 727 you'll be automatically correct with probability 2%, right? You don't need any cryptoanalysis. You don't need to know anything. You just know what the message space is, what the distribution is, and you choose the most popular value for F, given this distribution.

OK, now consider a dramatically different situation. The sender not only has chosen the message m from the given probability distribution, but also encrypts it, transmits an encryption of it, and so you also see the encryption of this message. Not only do you know that the message m has been selected according to the given probability distribution, but, lo and behold, you have an encryption of m. Now, can you guess what $F(m)$ is better than before? Remember, before seeing the

encryption of m, without any cryptanalysis, you could be right 2% of the time. Now, by cryptanalyzing the encryption of m, can you improve your probability of correctly guessing $F(m)$? If you cannot improve it to more than 2%, that is, not 2.01%, not 2.001%, not 2 plus epsilon percent, then we call the encryption semantically secure. OK, that is the whole idea. Now it's computational complexity, rather than information theory à la Shannon, that is being used to drive the notion of semantic security.

Actually, we developed computational indistinguishability as a tool to prove semantic security, and we proved that if we had a system which was computationally indistinguishable, then it was also semantic secure. We actually proved that also the opposite was true, that is, that semantic security implies computational indistinguishability, and that other notions of security are all equivalent to each other. And this is the most reassuring thing there can be in science, when you try to approach a new object. You use one avenue, then another one, then a third one, and suddenly you realize that all these avenues are absolutely equivalent.

Going back to Turing, at the time in which the notion of computation was up for grabs, people were trying to figure it out. "OK, I understand poetry. I understand other human endeavors. But how should I define computation?" Turing defined it using Turing machines. Church used lambda calculus. Another definition was recursive functions. And then, at some point, it was figured out that all these definitions were provably equivalent to one another. So one did not have to pick and choose which definition was the right one, because they were one and the same. It is this identity of different looking notions that reassures us that the right notion has been achieved.

So the equivalence of semantic security and computation indistinguishability, and other notions as well, tell us that a robust notion of secure encryption has been reached. Being equivalent, you might prefer to use semantic security to best convey what secure encryption means. But you may want to stick to computational indistinguishability when you want to prove that a particular encryption scheme is secure, because proofs are simpler when you use computational indistinguishability.

Ibaraki: It's just so amazing, [laughter] the level of thinking. And I can see now the profound impact of your work. And speaking about that, how do you see your work revolutionizing the study of cryptography, and laying the foundation for the theory of cryptographic security?

Micali: Well, cryptography has existed since time immemorial. For thousands of years people wanted to encrypt their messages. But they did not design a cryptosystem so as to achieve a predefined rigorous goal of security. They simply designed a

cryptosystem which "achieved whatever it achieved." They tried the best they could. They tried to poke their system as best as they could. There were no notions of security, no proofs, only heuristics. They essentially considered a laundry list of possible attacks, and then checked that each attack that they knew of failed. There was no guarantee that a new, yet known, attack would fail too.

So, later, even when the encryption was based on a mathematical problem like in the RSA, there was only a loose connection between the human problem of decryption, that is, between breaking the system and the difficulty of the purely mathematical problem that was chosen as the basis of a cryptosystem. Solving the underlying purely mathematical problem is one thing, and may be very difficult. But decrypting messages exchanged in a cryptosystem loosely based on that mathematical problem is a totally different thing, because you are helped by grammar constraints, by logical constraints, by context, by a lot of other things. Right? So these two problems are not quite the same.

Let me give you an example. Assume that the problem you have chosen as the basis of your cryptosystem is factoring integers. This is actually a great problem. Some numbers are primes, like two, three, five, seven. It turns out that you can randomly pick two large primes—say, a thousand digits each. Then, you can easily multiply them—in fact, you can still do it by pen and paper. But then, if you give their product to someone else and say, "I multiplied two random primes to get to this number; which primes did I use?" then nobody knows how to factor your product and retrieve the two primes you started with. Gauss and plenty of other mathematicians have looked at this problem without being able to solve it. So factoring integers is a very difficult pure mathematical problem. But it has nothing to do with decrypting.

When building a cryptosystem loosely based on factoring, we built it so that, if you knew how to factor, then you knew how to decrypt. But this is not a very interesting direction, right? The interesting direction is the opposite one. What we really want is that nobody could decrypt our messages, or even gain partial information about them, without being able to factor, so that, if factoring is hard, then the system is absolutely unbreakable. And if somebody somehow decrypts what I encrypted because he's able to solve the factoring problem, thus succeeding where Gauss and company failed, you know what? He deserves to know what I was saying. [laughter] OK?

So, the main contribution of Shafi and me was building cryptosystems for which one could rigorously prove that the purely mathematical underlying problem is absolutely identical to the very human problem of decrypting or even getting partial information about encrypted messages. In a sense, we found a way to rigorously

reduce apples to oranges! By now, one routinely designs cryptosystems with this notion of security embedded and with this type of reduction. In sum, I believe that replacing heuristics with proofs, and introducing these sophisticated apples-to-oranges reductions was our contribution to the field.

Ibaraki: And what a contribution! Again, an historical inflection point, [laughter] which really marks a huge shift, in my opinion, so . . .

Micali: Thanks!

Ibaraki: Now, Silvio, can you talk more about your work with knowledge complexity and *zero-knowledge proofs*?

Micali: Yes. Proofs are supposed to convey knowledge, right? There is a theorem statement. You don't know if it is true at all, so you ask somebody to prove it to you. He or she provides you a proof, and at this point, at the end of the proof, if the proof checks, after you verify it, you know not only that the statement as claimed is true, but you also know a lot of other things. You know *why* the statement is true. You must get a lot of details to get convinced that the theorem statement is true.

Assume instead that we want to reduce to a minimum the amount of knowledge necessary to convince somebody that the theorem is true. What should this minimum be? Well, at the end of the day, the minimum should be that you learned that the statement is true, which you didn't know beforehand, right? That is the minimum I really need to reveal in any proof. Now, a zero-knowledge proof is a proof that reveals only that minimum: that the statement is true, without adding any other piece of knowledge.

But the question is how can you tell that no other knowledge has leaked from the proof? This is another application of a simulation paradigm, which we were discussing before. Essentially, you want to prove a theorem in a way that ensures that if somebody knew beforehand that the statement of a theorem was true, then he could reconstruct the proof you give to him in exactly the same way in which you provide it. In other words, how do I know that from this proof, from this big interaction, I don't learn much more than the statement of a theorem being true? Indeed, from this interaction you learn that the theorem is true, and I wanted to give you this. But if you could simulate my proof in its entirety if you knew beforehand that this theorem was true, then there is no other further information in my proof. This is what a zero-knowledge proof is.

Sometimes you may want to reveal a little bit more. For example, think of an election. There is no theorem here, but there may still be a "zero-knowledge interaction." Assume that you have a hundred people in a room, and they want to

carry out a very simple election, the simplest election: a referendum. OK? So what do we want to do? We want to tally our yeas and nays. Each one votes yea or nay, and we want to tally how many yeas there are. Assume there actually are 60 yeas and 40 nays. So you want to compute that there are 60 yeas and 40 nays, but you don't want to reveal who voted for what. You want to keep *private* the votes, but you want to compute the tally *correctly*. So you want to have correctness and privacy at the same time.

Now, correctness without privacy, that's not a problem, because I can just say, "OK, ladies and gentlemen, whoever votes yes raises their hands. I count 60 hands, so there are 60 yeas in this referendum." On the other hand, if I want to have privacy alone, without the correctness of the tally at all, I can say, "Everybody writes a yea or a nay on a piece of paper, and throws the paper to the fireplace." By so doing, total secrecy is easily guaranteed. But then, what is the tally? So what we want, instead, is that, without trusting anybody, we can compute the tally of 60 yeas in a way that we have no idea who voted yes and who voted no.

Of course, if we trust somebody, she can just say, "Oh, just whisper in my ear what your vote is and I promise not to tell anybody, and further, I promise to announce the correct tally." Sure! I mean, this is not going to fly with anybody, and with cryptographers in particular. So, the idea is that we replace this trust in some individual, in order to guarantee correctness and privacy simultaneously, by just talking to each other, and trust that the majority of us are honest. Essentially, the idea is a *blending* of correctness and privacy. And because correctness matters in all human enterprises, and privacy matters to all humans, I believe that this blending is a good building block for a theory of human interaction.

Ibaraki: That's very interesting. So what do you see as the implications of this work, and how does the work extend to other domains?

Micali: All right. The implications. First of all, you can imagine that in a general cryptographic protocol, or in an economic transaction, you want to have both correctness and secrecy. Let me give you an example. Assume that you go to a carpet store, right? And you see a carpet there. As it happens, in such stores, carpets are not tagged with their prices. So you say, "I'm interested in this carpet. How much does it cost?" And the other guy says, "Well, wait a second. How much are you willing to offer?" "No, no, no, you go first," right?

The situation is very complicated, and we could benefit from a new transaction, one that we didn't quite have available before, such as the following. We engage in a cryptographic protocol in which I, as the buyer, choose my input to the protocol to be the maximum buying price I am willing to pay, and you, as the seller, choose

as your input the minimum selling price you may consider. And now, through our protocol, without telling each other these two values, we just compare them. If there is no overlap, that is, if my maximum buying price is below your minimum selling price, we only learn, "Sorry, guys, the two of you cannot transact. No carpet sale today." On the other hand, if there is an overlap, we end up with a contract, digitally signed by both of us, stating that the carpet is sold and is now mine at the price that, say, sits in the middle between my minimum buying price and your maximum buying price, or whatever price formula we want to choose. So, this transaction is something that somehow enlarges the realm of the possibilities we have in our "paper world," our ordinary-world transactions.

But in my opinion the implications of this theory go beyond business transactions. Because enabling secure transactions enables more interaction. Let me give you another example. Assume now we have a dating game, OK? There are two individuals, and I go first and say, "Hey, on a scale of one to ten, I like you ten. How much do you like me?" And the answer comes back: "Two." [laughter] With such an answer, I know I will never interact with anybody in the near future, because I need to recover psychologically, right? But assume now that you can actually interact in another way, in which you can somehow compare these two numbers but only figure out whether both of you like each other ten, or whether both of you don't like each other ten. In such an interaction, I've much less to lose in self-image, and thus I can safely interact much more in this fashion. In other words, if I can control the amount of privacy I might lose, I can confidently enter into many more transactions than before. So this is another implication of correctness and privacy. It enables not only business transactions, but also personal transactions.

You ask about other domains. There are plenty of other domains. Because essentially, at this point, from just encryption, cryptography has become *the science of adversarial computing*. And adversaries are everywhere, [laughter] as everybody knows, not only cryptographers. In a proof, the adversary is whoever wants to convince you of false statement. In encryption, the adversary is somebody who wants to understand information about your messages. In pseudorandom number generation, the adversary is somebody who wants you to generate biased rather than unbiased coin flips, etc., etc. More generally, the best way to model a very complex system is to model it *adversarially*. Because the more complex a system is, the more it looks like there is really an evil guy there trying to wreck it apart, to make sure that nothing works.

So, essentially, this theory is becoming more and more hand in glove with fault-tolerant computing, where you really want to make sure that, you know, a network of computers continues to work properly together, even though some of them fail,

and fail in a way that is seemingly controlled by an adversary. In a different domain, this theory has by now encompassed all pseudorandom number generation. It has also provided bounds for what is learnable. Valiant and Kearns have somehow used Shafi's and my results together with Oded Goldreich on pseudorandom functions to figure out what cannot be efficiently learned. In sum, because adversarial computing is so pervasive, and allows us to model so many things, there are many, many domains to which this work may apply.

Ibaraki: I see. I mean, that's fascinating, I can see this now and in ten years' time a Nobel Prize.

Micali: [laughter] Thanks. I don't know about this, but thanks.

Ibaraki: Now, you've somewhat addressed this in all the different kinds of answers you've provided, and the dialogue we've had so far, but how does your work address important practical problems, such as the protection of data from being viewed or modified, and providing a secure means of communication and transactions over the Internet?

Micali: All right, yes, sure. You know, encryption is not the only thing you want to do on the Internet. Protection of data from being viewed, we have discussed, but from being modified we have not yet discussed, right? About protection against data modification, Shafi and Ron Rivest, my colleague at MIT and a prior Turing Award winner, and I developed a *digital signature scheme* that actually has set the standard for subsequent digital signatures. Can I describe briefly what this involves? Let me go on a limb and take another five minutes.

Essentially, what is a digital scheme? A digital signature scheme involves a pair of matching keys, a *secret key* that allows me to sign messages and a *public key* that enables everyone to verify the messages I sign using my secret key. The crucial property is that the public verification key does not betray the secret signing key. That is, knowledge of the verification key should not enable one to compute the signing key in any remotely feasible time, such as a few millions years, even with the fastest computer. So to prove that a given message, M, comes from me, I use my secret signing key to compute a short string S, my digital signature of M. Such digital signature S depends on M, because different M's would have different digital signatures from me. But then you can use my signature S and M and my public verification key to see whether S is indeed the correct signature of mine for the message M. If this is the case, you can rest assured that I consented to the message M, right?

Now, for this to work, it is necessary that these signatures are unforgeable by somebody else. OK, but what does this mean? In the past, it used to mean that an adversary could not come in, look at my public verification key, and forge my signature of his favorite message, such as, you know, "Silvio owes me a million dollars." But, we need more security than that. So what do we need? We also need that somebody cannot modify a prior signature of mine so as to forge my signature on a modified message. So, for instance, if I did sign, "I, Silvio, owe you, Stephen, $1000," somebody should not be able to change it "I, Silvio, owe Stephen (or somebody else) $2000," right?

Even more, you want that somebody cannot ask me to sign a few things, and then, assuming that I do agree and sign them, learn to sign other messages. Think of a notary public, who essentially is somebody who signs messages chosen by other people. And of course, he could use digital signatures to digitally sign messages. So you don't know how to forge the digital signatures of this notary public, but you can ask him to sign a given message, and he does. Then you say, "Ah, that's interesting. I just learned something that I didn't know before. I think I start getting the idea how the signatures of this notary public look like, but I'm not quite sure, so let me ask him for a second one. Could you please sign this second message?" And the notary public signs it again. You say, "Oh, gee, now I'm getting the gist of it." And so you go on with this process a bunch of times. You request signatures. The guy agrees and sends them back. So what one should really want is that, at the end, you cannot sign any new message at all. In other words, forging someone else's signatures should not only be hard from scratch, but also *unlearnable*.

When I arrived in this country, you know, English was a cryptosystem for me. More or less, I could not really be understood by anybody. But then I was able to ask questions, "How do you say this? How do you say that?" And slowly slowly, I learned enough to get by. So we don't want this to happen in a secure digital signature scheme. We want a more stringent notion of security. We want signatures that are unlearnable. I believe that this requirement is crucial if you really want to prevent data from being tampered with over the Internet. And signature schemes guaranteeing this stronger property have already been developed.

Ibaraki: Now, what is the impact of your work on computational complexity?

Micali: Well, *interactive proofs* were crucial to complexity theory, because they let us understand which class of problems have an efficient proof. Remember, proving a theorem is the most frustrating thing. Proofs are very frustrating to write down, and it is very frustrating to read them. Interactive proofs actually transform this

frustrating thing into a game between the *prover* and the *verifier*. Somehow, if the theorem is true, and I act as a prover, then I should win a very simple game between you and me. Say that the game has five moves: I move, you move, I move, you move, and one of us wins, and then we can determine who wins. If the theorem is true, then I should win all the time. If the theorem is false, I should win at most half of the time. So if we play this game, say, 100 times, and you see that I win 100 times in a row, you conclude, "Well, you know what? The best explanation is that the theorem is actually true."

Figuring out which theorems are easily provable is important in complexity theory. As for another impact in complexity theory, my work on *pseudorandomness*, with Manuel Blum first and with Shafi later, essentially has helped us understand which problems can be solved deterministically. Thanks to Solovay and Strassen, and Rabin, by now we know that there are plenty of problems that can be efficiently solved probabilistically. But then what happens if your computer cannot flip coins? Somehow the theory of pseudorandom number generation allows us to understand what problems can be solved efficiently and deterministically.

More generally, a lot of my work depends on a *one-way function*, OK? A one-way function F is a function that has two crucial ingredients, very antagonistic to each other. The first is that the function F is easy to evaluate, which means on input x, you can compute $F(x)$ very, very quickly. The second is that the function is hard to invert, meaning that given $F(x)$, you have no idea how to retrieve one such x. Essentially that is the mathematical analogue of the one-way phenomena that we so commonly experience in the real world.

For instance, if I take a glass, and I smash it on the floor, that is very easy, but to reconstitute the original glass from its pieces is much harder. So this a one-way phenomenon. As for another example, it is easy to scramble an egg, yes? But to unscramble it is a totally different (and in fact much harder) story. So a one-way function essentially incorporates in itself both easy and hard computation. Thus, it's not surprising that understanding one-way functions increases our understanding of complexity theory, which is the field devoted to figure out which problems are easy to solve and which ones are not.

Ibaraki: Yeah, that's fascinating. What are your thoughts about things like in quantum mechanics and the twin particle effect, and sort of the impact that's going to have perhaps on your field? Or do you see sort of the work of Judea Pearl in causality and counterfactuals and external validity and artificial intelligence—do you see some kind of connection between some of this research you've done and those areas, at all?

Micali: Well, certainly let me address the field that [laughter] is more dangerous to mine, quantum computing. We need hard problems to base cryptography on. As we said, we want to take a purely computational problem, a purely mathematical problem, and massage it around and transform it by magic into a very human problem, like proving "This message comes from Stephen," right? Of course what is easy computation and what is hard computation depends a lot on the available computational model. If you have an abacus, what is hard and what is easy is one thing. If you have now a modern computer, but still a classical computer, it's something else. The jury is still out on whether quantum effects can practically and dramatically speed up computation or not, but they might. In this case, first of all, we have to redefine what is easy and what is hard, and then define functions that are one-way for quantum computers, rather than for digital ones. So some specific candidates for one-way functions, such as factoring, may disappear, but that does not mean that we cannot generate other candidates, because we now have a more general theory of one-way computation.

Ibaraki: You know, amongst our listeners there's people who are not necessarily heavily involved in all the technical aspects, and in some ways they could be consumers, because they're in senior management now, and their technical years are long past. So what are the practical implications and applications of your work influencing all of our daily lives?

Micali: All right, the simple practical example is that of a password. I'm sure everybody has dealt with passwords, right? For thousands of years, a password has been some secret phrase, such as "Abracadabra," that I use to enter, say, a castle. If I'm a medieval knight, and I'm on the other side of a moat, and I see the bridge is drawn, and I want it to be lowered, I say to the guard upstairs, "Abracadabra," and recognizing the password, the guard lowers the bridge and I can come in. You can use your mother's maiden name as a password. I can use the name of my favorite uncle. Either way, it is a secret that we actually need to communicate. This password system, of course, has some drawbacks. Essentially, if in the moat of the castle, in the water there, there is somebody, he can hear that the knight whispers "Abracadabra" before getting into the castle. Thus, at a later time, he can impersonate the knight with no problems. He puts on helmet and armor, says "Abracadabra," and the bridge will be lowered for him too.

In addition, a classical password system has another drawback: The gatekeeper himself knows know the password, so if I use the same password for other systems, say, not only to enter the castle but also to log in at MIT and to log into my bank site and wire money out of it, I am in danger, because I actually am enabling any

verifier of one of these systems to impersonate me to any other system. So, what am I going to do?

I generate somehow a theorem whose proof only I know. For instance, I take two large random primes and I multiply them together to generate an integer N, and then I tell MIT, the castle, and my bank, "This number N here is Silvio's number. Anybody who proves to you that it is product of exactly two primes, let him enter my castle, let him wire money on my behalf, let him access my files at MIT. With my consent."

But how do I prove that N is the product of two primes? Do I send over the two primes I originally multiplied? Absolutely not: such a proof could be copied and used to impersonate me to another system. I use instead a zero-knowledge proof. That is, when, say, I want to log in at MIT, I engage its server with a zero-knowledge proof that N is the product of two primes. Such a proof can be verified by everybody, and thus by MIT's server. But it's a zero-knowledge proof, so nobody having verified that N is indeed the product of two primes is able to prove this to anybody else. Because, after a zero-knowledge proof, you don't learn how to prove the statement—you only learn that the statement is true.

So suddenly you essentially a have an ideal password system. It lets you safely use the same passwords with multiple systems, it is very efficient, and it is implementable via a smart card. It is the most practical application that I can think of.

Ibaraki: Silvio, you're this giant in industry and education and research and so on, and your work resonates throughout the world, and so I know our audience would be interested if you can additionally profile your extensive research history, its lasting impact, and some valuable lessons you wish to share from each of your top research areas that we haven't talked about yet.

Micali: [laughter] All right. First of all, let me just mention, without any details, that, in addition to whatever else we just discussed, I've been working on distributed computing, on private information retrieval, etc. But perhaps, you know, we should move from the technical work to the lessons learned.

The most valuable lesson that has worked for me (and many others) is to really generalize and simplify the concrete examples that motivate you. Concrete examples are wonderful. They really drive us. But they are also typically messy, right? They contain an abundance of details that may blind us. So my lesson would be just, you know, get rid of as many details possible. Generalize your problem as much as possible. Back up, and back further up until you see the whole picture in its simplicity. Generalize a problem until it becomes either impossible to solve or very simple to solve. Back up to get the full view and drive yourself to a corner. And once you have

no escape you may lose, but you may also find additional strength and win big. At the end of the day, who needs partial victories? So my lesson would be drive yourself to a corner and go from there.

Ibaraki: That's an interesting concept. So how many times have you done that?

Micali: Oh, I've done it as a graduate student. I've done it as an undergraduate student. I've done it as an assistant professor. [laughter] I've done it a few times. The amazing thing is that it often works. So I'm not advocating without practicing, let's put it this way.

Ibaraki: It seems to me that concept could be applied to so many other areas, perhaps friends and family and business deals, as well.

Micali: Why not? Never sit at a negotiating table if you cannot get up and leave at any time, and never shoot for "just friendship." You know, sometimes I think it's worth it to risk it all.

Ibaraki: I see, Silvio. So this could be a book beyond your research [laughter] that the general public will read.

Micali: [laughter] I'm sure I'm not alone, right? I'm sure many people would agree with me.

Ibaraki: Now, Silvio, you talked about your past research, and you also talked about some of the other areas that you have researched. Can you get into more detail about your current research interests?

Micali: Yes. Somehow, at a late age, unfortunately, I encountered a beautiful notion that was put forward some half a century ago by economists, *mechanism design*. Essentially, this is a way to choose an optimal outcome without data. Optimizing is never easy, even if you have the data, but if you don't have the data it is actually much harder. And so why don't you have that data? Because other people, the so-called *players*, have the data. You may say, "Why can't you just ask them?" Well, because they may have a stake in the outcome you choose, and therefore, when you ask them for the data, they may lie so as to manipulate in their favor the outcome you choose. And so you must engineer a game so that, when everybody plays it so as to maximize his own utility, you learn, as a side product, which outcome you should choose. It's a fascinating field, and that's what I'm currently working on, from my own special perspective, of course.

Ibaraki: And then what are the broad implications and applications of this work?

Micali: Well, in principle, any decision-maker, in particular any politician, would stand to benefit from mechanism design. If you really want to go one step farther,

mechanism design may be the best way to engineer a system, like the Internet, that is very decentralized, in which no one is in charge. And because no one is in charge, you can put all the rules and laws that you want, but unless you design the system so that everybody is incentivized to stick to the rules, the system will never quite work. So mechanism design may actually be used in engineering large decentralized systems. And finally, you know, I'll not be surprised if mechanism design were to provide us with key insights for understanding successful biological systems. Perhaps our complex organisms are not the visible product of some unlikely kind of equilibrium, a very fragile thing, but actually are the robust outcomes of properly and slowly designed mechanisms.

Ibaraki: Oh, fascinating. And again I mention, gee, maybe a Nobel Prize, as well.

Micali: [laughter] Ahi Ahi Ahi!

Ibaraki: What are your future research interests?

Micali: If you stress *future*, the answer is the brain. Yes, the brain might be my future interest, and not only mine. [laughter] In fact, other computer scientists before me—in particular, Les Valiant—started working on it. I think that I'm considering working on it.

Ibaraki: Oh, that's fascinating. In terms of that work, you're thinking of applying sort of a mathematical model to it, or getting more sort of into the engineering side, or getting into sort of the works like external validity or causation and some of that area? Sort of what's the approach?

Micali: Remember that I truthfully answered your question by stressing *future*. So, right now we don't know, at least I don't know which angle it's going to be, but certainly it's going to be a computational angle. At the end, I believe that a big part of the brain's function, and memory in particular, should be modeled as a computer, and you want to put things in memory, and retrieve them efficiently, and with some redundancy. And we know a lot about how to store, retrieve, and manipulate information when we have total liberty to decide the components. Here, the components are decided beforehand, but perhaps some of the lessons we learn from distributed computing may be applicable to the brain, too. More than this, I do not know. Right now I'm working on mechanism design, as I was saying.

Ibaraki: It's interesting, the whole concept of that kind of research, and I'm thinking of Daniel Dennett and *Consciousness Explained*, or Descartes and this sort of mind/body connection, or Penrose and some of the work that he's done in thinking about the brain, but from a model of a philosophical sense, or Kurzweil, and this idea of a singularity, which in some circles is controversial. Do you have any

feelings about that sort of idea about a soul and a brain, and is there something more that we don't understand?

Micali: We don't understand *a lot*. But [laughter], if you ask me, remember whatever I said about the simulation paradigm? Never mind the mind and the body. Really, the question is whether the whole universe can fit in the brain, right? [laughter] I mean, I'm a little bit of an extremist here. But again, there are tremendous possibilities, but I have not given them the rigorous thought that I've given to some other fields yet.

Ibaraki: Silvio, I could just see those roots of this kind of thinking going back to when you were a child, and talking to your mother. [laughter] Now, what are your most difficult challenges in research, and what valuable lessons do you wish to share?

Micali: Well, my challenges, if I can be frank, are inability to work alone and lack of knowledge. And so the lessons I wish to share are the same ones that I used to cope with my challenges: collaboration and imagination. So what if you cannot work alone? You can always collaborate, provided that you hold on to your own individual obsessions, no matter how extensively you collaborate. And again, who cares if your arsenal is quite small? Be imaginative, forge your own tool, and march ahead.

Ibaraki: Every time you get researchers together, or you get, I guess, any group or cohort together, you're going to get a lot of discussion. You're going to get debate. You're going to get some controversy. You're going to get different points of view. So what would you describe as additional areas of controversy in the areas that you research?

Micali: [laughter] Well, controversy is . . . Everything is controversial. Actually, I think that the main controversy, not only in my research area but in any area, is the very definition of an area. This is the most contentious item in research. To be clear, defining an area is both necessary and useful to focus the effort of future work, to flesh out the problems, to attract fresh minds, etc., etc. But it's also a constraint. It's a boundary, right? And boundaries may always incarcerate us. So we have to be very, very, very careful.

Our theoretical community is just amazing. I really love my community. It has invaded new territories with determination, ferocity, and cleverness, like a bunch of conquistadores, but fortunately [laughter] no physical bloodshed. But even we, a progressive and ready-to-abandon-all-boundaries society, risk to transform our-

selves into 'the guardians of the sacred fire'. And at a very great speed. We start fighting to protect the purity of our field against outside contamination. It is mind-boggling to me.

Suddenly, the game is not to find solutions to problems that do not yet exist, but to solve *older* problems. And the older the better, because you get more credit for solving, you know, a 50-year-old problem, than you get for solving a 10-year-old problem, etc., etc. Of course, you need both to pose and solve new problems and to solve old problems. But I don't understand the emphasis on old problems, right? That is really a disease.

If you chair a prestigious conference, or you are the editor in chief of a flagship journal, somehow you start feeling that you are expected to become a businessman, to satisfy the customers who put you there. So if you publish outlandish material, the number of subscribers may drop in droves. How would you look? Can you accept this damage to your reputation? Publishing such material may cost you further advancement. On the other hand, refusing to publish dangerous new material is hidden from the public eye, so you may actually harm the growth of your field, but no one will ever know. I'm actually saddened by the fact that journals and conferences publish a disproportionate amount of small—but declared big—advances on the status quo. I believe that the incentives are misplaced, and we can and must do better and never define in too strict a way any area.

Ibaraki: Hmm, that's quite fascinating what you just stated there. I mean, it kind of reminds me of this idea of disruptive innovation or research, and this concept of innovators being a platform where they sort of model what creates breakthrough innovation, what creates breakthrough disruptive innovation. They find sort of these five qualities, one of which is always actively questioning everybody, everything and everybody, always actively observing everything and everybody, always actively experimenting in diverse areas, even across areas that are outside of your domain, to get a different perspective. And the final two elements are associating, and that is synthesizing all the kind of different concepts in all the different areas, and integrating that information as you sort of proceed day to day. And then finally networking. Networking with others, but particularly with those who hold diverse views, and perhaps contradictory views, or even to the point where it could break the system, or close to sort of your collaborative team. It sounds like you're sort of speaking to that, not to get into this sort of groupthink idea.

Micali: Oh, absolutely. Of course there is the risk that, if everything is innovating so fast, then we cannot discern anything anymore. We need some rigidity, I don't

know, it's an old question whether geometry could have been invented if we were water animals. I mean, [laughter] we need some solid terrain, perhaps, to hypothesize a triangle, and so on and so forth. Or maybe not. But what I'm saying is that we must do better than just barring this. It would be nice if every journal or conference actually accepted, say, every ten articles, two oddball articles, if people actually expected two such articles. And I bet they would be read with interest. Even simple policies like this would go a long way to incentivize us to question ourselves and our own fields and to make progress.

Ibaraki: I see. So, actively embrace outliers. [laughter]

Micali: Yes, yes, bring them into the fold. We need outliers. But we also need, you know, to make progress on very established questions. My problem is that I perceive a disproportionate emphasis on traditional work. Of course innovation will break through once in a while, but not at the right rate. We can actually control and optimize the rate a bit. Actually, quite a bit.

Ibaraki: Now, Silvio, can you describe the types of research being created or updated that will drive our experiences in five or ten years, and what will these experiences be like? Can you paint a picture for our audience?

Micali: Well, frankly, my prediction for future research can only be based on what I know, so I expect more and better of the same. I don't know how interesting that may be. All expert predictions matter less than the developments we cannot predict. I mean, if our predictions were exact, our future would be doomed to boredom and missed opportunity. I personally look forward to major surprises, [laughter] and I must confess that those I cannot anticipate.

Ibaraki: Now, you have this remarkable background—your educational background, that is—at the University of Rome and in Berkeley. So what specific challenges in your education at these two famous institutions were catalysts to inflection points in your lifetime of contributions, and how and why did this happen?

Micali: Oh, wow! Thanks for asking. [laughter] I really would like to give credit to both great educational systems, in Rome and in Berkeley, and the actual people behind them, who really shaped me . . . So let me have a crack at explaining. First of all, both universities, and in particular the specific teachers I met, have been very *flexible*. This really shaped my attitude towards research.

In the United States, to tell you the truth, a course is run more tightly than in Italy. As a student, you are continuously monitored with problem sets, and the exam coincides, so to speak, with the last day of the course. There is not much room for negotiations [laughter] of alternative dates. In the Italian system, instead, you are

much more in charge of yourself. There are lectures, of course, there are sessions of exercises, but it's totally up to you to attend or not to attend. And the exam, you can actually take it when you feel ready: after a month, after a few months, after a year, even. That for me was really ideal, because I would have not functioned otherwise. Typically, I took four yearlong courses, where yearlong [laughter] meant from November to May. Then I took, say, one exam in June, another in July, one in September, and one in October. And then courses started again. It was crucial for me to be able to take an exam when I felt ready. I absorb things slowly, and that flexibility was extremely important to me.

People-wise, I really, really admire and I owe a lot to Professor Luciano DeVito. He taught us mathematical analysis. You must know that, in the typical Italian fashion of the time, I took a classical high school: lots of humanities, history, philosophy, and very little math. In fact, the only math that I was exposed to was Euclidian geometry, maybe because it was Greek. [laughter] Yet, I was fascinated by it enough to decide to enroll in physics, and thus I was exposed for the first time to mathematical analysis. A marvelous field. You started talking about infinity in rigorous terms. It was wonderful. But whatever made this course unique, as I realized later, was that this guy, DeVito, organized the entire course around problems. He never engaged in a classical definition-theorem-proof sequence. He would ask, "How might 'area' be defined?" And then a big debate started. Sometimes, he posed problems that we could not solve right away, but we solved them very much later. The problems were really center stage, and we were obliged, actually, to define things if we wanted to make progress. And somehow this necessity to define things became an ability, and helped me tremendously in my career.

So, in essence, his course was entirely devoted to research and that was the first course that I ever took, OK? I loved it so much to conclude, "Who cares about physics? Actually, what I care about is mathematics." I understood mathematics to be analysis. So I told him, "Professor DeVito, I really want to switch to mathematics." To my surprise, the guy says no: "You cannot switch." I said, "Why not?" Because, he says, he's proud to have been an analyst himself, but analysis was for older people like him, and a young person like me would be better off staying in physics. OK! [laughter]

I followed his advice, thinking that perhaps I could change his mind if I actually proved something. At some point, he mentioned the general axioms of measurability according to Lebegue and the existence of a set non-Lebegue-measurable. Somehow I decided to find such a set. But I was unprepared for the problem, and could not solve it, at least not right away. So I totally obsessed about it, to the point that I actually neglected to follow his lectures. I was behind in the course. In fact, I

dropped all the other courses as well. But eventually, I managed to solve the problem, and I presented a solution to him. He was very happy and gave me an A+ at the first exam opportunity, while I was intending to take the exam much later. For me, that was really a transformative experience. Somehow I got the notion that it was OK to carve a path on my own, and that somehow research has to be center stage. I really felt empowered. More importantly, he now gave me permission to switch to mathematics, but he added, "If you really want to do mathematics, then you should focus on . . . " —he didn't use the word, but he essentially described theoretical computer science. He told me about Gödel and Turing. But then he says [laughter] "You really watch out, because to do this stuff you need a big stomach."

I switched to mathematics, but I neglected his advice and followed instead courses in analysis. But, in the fourth year, I paid attention to what he said enough to follow informally as a listener two courses, one on lambda calculus with Corrado Böhm, which was and is the father of Italian computer science, and one in logic with Giuseppe Iacopini. Corrado has always been very enthusiastic. He sought me out. He convinced me to leave analysis, to actually formally enroll in his class, and also he gave me a challenge. He said, you know, "Why don't you take the class, and why don't you try to prove that?" That challenge then became my undergraduate thesis and our first paper.

So now at this point, to tell you the truth, I was convinced that I wanted to do computer science, but I was totally unprepared. At the time there was no CS graduate program in Italy. So Corrado quite unselfishly suggested that I pursue a doctoral degree in CS abroad. Before applying, I followed a one-month summer school in computer science. The idea was to choose four courses out of some eight. I chose my four, but then dropped two. Since no degrees were awarded, why not? The course I liked the most was on *graph algorithms*, and it was taught by Shimon Even. Shimon was a wonderful teacher from the Technion, in Israel. He really introduced me to algorithmic thinking, and he became a beloved friend and mentor. The other course was also on algorithms, but more general, and it was taught by Fabrizio Luccio, from Pisa. At the end, they both gave me the same advice as Corrado: "I think you have to go abroad."

Somehow all three of them suggested Berkeley as the more suitable place for me to study. So I applied, I actually was admitted, and eventually went. I must tell you that was another lesson for me: Receiving caring advice on how to complete my studies from people who were not my advisers and had no formal responsibilities towards me, somehow gave me the impression that research really was an enterprise without borders, that I was helped by people who owed me nothing and who

really encouraged me to go far, [laughter] in a way. I realized I was entering a community of the mind without borders, and that the people out there actually cared to advance Science with a capital S. It was an amazing message, right?

I was saying that I was studying history in high school, but my history books were de facto centered on the history of nation states. Or at least I read them that way. So somehow I didn't take notice of the fact that there was a parallel, in fact, a transversal universe. I mean, I knew, on paper, that science was a big enterprise. You know, Archimedes and Eratosthenes exchanged letters, Pythagoras traveled all over the Mediterranean, medieval scholars moved from Bologna, to Prague, Paris, etc., etc. Artists were born in one city, lived in another, worked in another, and died in another yet. Really, all this I knew, but somehow I never *registered* it. So, from that point on I must tell you that geopolitical boundaries faded in the background forever. And so that's it. Gotten this mythical view of what science ought to be, I decided that, yes, I would go to Berkeley.

Now, Berkeley: I was admitted, but not right away. My score in the test of English as a foreign language was actually abysmally low, so I had to take it again. Finally, I got a barely decent score, and I could begin at Berkeley in, I think, March 1979. And I was utterly miserable. First of all, I realized that my English was really poor, that I could not communicate with anyone, that I knew no one, and that I had no prerequisites in computer science, while Berkeley had a very tough prerequisite tree. So the only course I could take was CS1—the name says it all—an entry programming course attended by 18-year-old people, and even precocious younger people. I was 25, so there was very little mingling there. The other courses were equally elementary. So, bottom line, I decided I'd finish the trimester, I'd pack up, and I'd go back home. Accordingly, I also decided I might as well enjoy the city.

Just when I lowered my guard—perhaps because I was a bit more open—I actually met David Lichtenstein, who at that time was a Ph.D. student about to graduate. He took me under his wing, really, and was another marvelous example of the generous help that had been showered on me over the years. He showed me around San Francisco. He told me: "Forget about prerequisites. I think you need to do research. Why don't you pick up along the knowledge you need?" He told me that Professor Blum was actually finishing chairing the department of CS at Berkeley that summer, and said, "He's a great advisor, and you're lucky, because he has not taken new students during his chairmanship, so in the fall when he steps down he needs new students, so why don't you propose yourself?" I said, "I will try." He actually had [laughter] another reason in favor of Manuel: Being from Caracas, Manuel spoke Spanish. So, you know, "He can understand your Italian." [laughter] Because apparently my English wasn't good enough.

So I went to see Manuel. He was very kind, but he said that his hands were full, that we should reconsider everything in the fall, that we should wait. But David didn't give up. He was determined that I started doing research and stay motivated. And so he told me about a problem posed by another graduate student, who had just graduated with a superb thesis, Mike Sipser, now a leading complexity theorist, and the chair of MIT Math Department. In his thesis, Mike had left an open problem, and David suggested that I should try to solve it. The problem was in automata theory, which, of course, I knew nothing about it. So, he said, "No problem, so I'll give you a crash course in it." The "course" took two hours, or maybe three, at a coffee shop in Berkeley. We were sitting there, sipping cappuccinos, and he was telling me one definition, then giving a small exercise. He was patient and understanding, and so on and so forth. And at the end, with the last cappuccino—I call it the four-cappuccino course; that's how many cappuccinos [laughter] I was able to drink in one session—he says, "OK, now, here is the problem you should try to solve."

A few days later, I was actually able to solve it, and told David, and he says, "That's wonderful. Now you have to go back to Manuel and explain it to him too." I said, "But Manuel said his hands are full." He said, "Never mind, Manuel knows the problem. In fact, he was the advisor of Mike, so he would like to see the solution." So I went to Manuel, and David was right: Manuel wanted to hear the problem right away. He cleared his schedule, cleared the board, and let me explain the solution. And at that point, he agreed to pick me up as a student, and from that point on we only spoke of research. I mean, *I've never seen anybody so research-oriented as Manuel*. He really was wonderful.

So, at this point, you know, I decided to stay in Berkeley, and I showed up again in the fall. By then, David actually was no longer there—he already went off to his job—but that's when I met Shafi, and actually Vijay Vazirani, too, and Mike Luby, a group of extraordinary researchers and great people, as they turned out to be. We formed a gang of sorts. We dined out with modest finances, but still enjoyed the food, working together, and actually trying to solve the problem sets together. It was really wonderful. We took a course of Dick Karp, which was to test the flexibility of Berkeley [laughter] despite being a U.S. university.

Dick Karp is a fabulous teacher, too, and he ran a famous algorithm class. And he mentioned a problem, fortunately or unfortunately, kind of early on in the course. It was a problem in algorithmic graph theory, the same subject that Shimon made me enthusiastic about. The problem was extending the running time of the best-known algorithm for *matching* from bipartite graphs, which are special types of graphs, to general graphs. So Vijay and I decided to work together to try to solve

the problem. We spent hours and hours together. Vijay was capable of satisfying all the other courses and taking care of the other problem sets. As for me, I, again, dropped out from all courses, including the one with Dick Karp. I only spoke to Vijay. But at the end, by our good fortune, by the end of the course we found a solution, right? (That solution, by the way, continues to be to this very day the most efficient solution for general graphs.)

So now what do I do? Dick, I must say, to his honor, let me pass the course, which by any standard I should have failed, with a B-minus. I mean, Dick is a very generous and fair person. I'm sure I must have tested his patience, [laughter] but he had to give me a B-minus. And now I was a little bit in trouble, because surviving as a Ph.D. student with only one course with a B-minus on my transcript was no joke. And to make things worse, I had to pass a barrier to continue the Ph.D. program, the *prelim exam*, and having not taken any hardware classes, I failed the hardware portion of this exam. So Dick and Manuel had to personally testify in front of the relevant committee so that I could continue my doctorate. Somehow, they were very persuasive, so I was allowed to continue.

And then our gang attended Manuel's course on computational number theory, and we all had a great time, we cemented our friendship, and at that point Shafi and I joined forces on cryptography for many, many years. And she actually had a tremendous influence on me in many ways. In particular, she convinced me that, given that nothing came easy to me, I might as well focus on hard things only. I must say that her insights, personal and scientific, really spurred me on in decades of joint work. I was indeed very fortunate to join forces with such a scientist and a friend and a colleague. So at this point I had a course that I loved, taken from Manuel, and the companionship from great friends and researchers. The people were much more flexible than the system, and I really felt, you know, that I really was in the proper crowd.

Manuel, I don't know if you know him, but he's a permanent revolutionary. We already spoke about Dick Karp. And then there was also another faculty member at the time, Andrew Yao, who started also as a physicist but, unlike me, a real one, with a Ph.D., a post-doctorate, etc., but then he got fascinated by computation and switched to computer science, and was then a professor at Berkeley, too. And I'm glad he was, because Shafi and I and he actually had a marvelous and fruitful interaction.

All these guys were actually marvelous teachers, but in very different ways. I mean, I have the fondest memories. Such a high standard to live by. It's scary, really. Manuel, I don't know, he was a magician. He did not explain a theorem. He actually forced you, actually all of his students, to prove the theorem on the spot: the trials,

errors, anxiety, heartbeat accelerations, the whole shebang when you try to solve a problem. Dick was most clear, organized, a perfect sense of timing. I mean, I have a terrible memory, but whatever he explained I still remember. And Andy, he was not formally one of my professors, but I attended all of the lectures that he gave, and two of them really changed my life. You know, one was on *Shamir's secret sharing*, when I was a student, and another one when I was an assistant professor at MIT. The latter was a lecture on what came to be known as Yao's *garbled circuit*, which had also tremendous influence, not only on the field but on me in particular.

At Berkeley, I really think that the flexibility, the focus on research, and the ability to pardon cutting corners—provided that you actually strive forward at least in one direction—was really what made me what I really am. I really felt I was in a magical place. Remember, I had a very Europe-centered point of view, right? I thought of Berkeley as the far edge of a civilized universe, in front of the Pacific Ocean, that mythical barrier to mankind, right? I felt I was in heaven, but, keeping with the metaphor, [laughter] I also felt, "Who could live in such a small city, except, you know, monks?" I saw them as monks, Manuel and Dick and Andy, living in this remote hermitage at the confine of Earth. Really, it's hard to communicate such a personal experience, you know. I learned so much. And I learned what I really wanted to learn: that finally I was not alone. I learned that Science really had the power to understand anything, even things that seem to be impenetrable to quantification or rational analysis altogether. I learned the power of interaction. I've never forgotten it. And really, I learned that experiences that cannot be written down or repeated in any way, like these I just described, really are the most permanent and precious. Ever since, I became a very big fan of the oral tradition. We should go back to this oral tradition, the strongest and most effective tradition we ever had.

Ibaraki: Silvio, that's just an amazing history, in terms of the mentors that you've had, and continue to have, the collaboration with so many people, as you indicated, the flexibility that you were given, and I guess now that's generated [laughter] some questions in my mind. You had this very unique kind of program, both at University of Rome and at Berkeley, where people have given you some agility and some, as you indicated, some flexibility. Now, do you pass that on in terms of your interaction with your students, and so on, your graduate students? Has that influenced your interactions with potential researchers?

Micali: Oh, absolutely it has influenced. The extent to which I actually succeed at giving back what I received, that I don't know. But I certainly try. [laughter] I have my own rigidity to worry about, of course, but you bet I try to be as flexible to others as my teachers have been to me.

Ibaraki: You know, it's interesting: in this dialogue I can feel and sense your energy and your passion for the research that you do and the things that excite you, and I know that your collaborators, and those students that you influence, as well, would feel that same passion, so they'd be very excited [laughter] to work with you, I think.

Micali: Oh, these are very passionate people. You're right.

Ibaraki: Another question is that, you know, you worked with your colleague Shafi for some time, starting at Berkeley, and that collaboration has continued. You know, any time you collaborate with somebody, sometimes there's tension, and how do you manage that tension? Or, you know, let's say if you disagree on a point of view, how do you manage that?

Micali: Well, the best thing is not to manage. Somehow, tension gets resolved. Tension is good, right? Somehow you are pulled in two directions, but I think you generate energy. I think that as long as there is goodwill, this energy gets released in a positive direction. I've never tried to be, quote, "polite," in an interaction. It doesn't work for me. And other people have been very genuine when interacting with me. Sometimes, we start "polite," but then as we become more and more friendly with one another [laughter] and we become more and more direct, tension rises. And to tell you the truth, I think it's good. I don't think we should manage tension. If the tension becomes too high, and you have to say "Go to hell" for a day, and "I'll never work with you anymore," you can always restart on the next day. But if you try to keep everything at a quiet or moderate tension level, I'm not sure . . . It may work for others, by the way. I don't want to dampen it. But it just doesn't work with me, and with the people with whom I've had the pleasure or the honor of working.

Ibaraki: In the past your supervisors and collaborators, but also your mentors in the past, have given you a lot of flexibility, sort of allowing this sort of oral tradition in terms of you proving that you had the expertise or the knowledge, or you've done the required research in your problem solving. You know, there's this new idea that came out of Stanford—oh, I guess it's not new, but it sort of got more attention back in 2011. That's this idea of massive, open, online courses, you know, where they had the artificial intelligence course, 160,000 students enrolled from 190 countries, volunteers translating in 44 languages, and MIT and Harvard had started something called edX, and it's sort of in that same area, or Coursera, you know, is all about MOOCs. What's your opinion of MOOCs, and do you see that in conflict with sort of the traditional side of teaching, or do you see it sort of aligned

with how you were kind of mentored, and the kind of support that you've received in your life?

Micali: So, we have to distinguish here the personal, what is good for me, from what may be good for others. Tell you the truth, I really believe that enabling a very large audience to get educated is something extremely beautiful and extremely useful. Ideally, we'd like to do this one-on-one, but if we cannot, then these online courses are perhaps, you know, a very good alternative. For me, actually, personally, it does not work, but that does not mean that it's not good. Just I'll be a very poor e-teacher. Remember that I continue to struggle with the doubt of whether there is somebody "on the other side," right? So, I hate writing letters, because who knows if the other one will ever receive it. And if it is received, in what state or mood he or she is. So I prefer a phone call to a letter anytime. Actually, if it's something that is very important, I really insist on physical presence. And so the notion that I, personally, could go in front of a microphone and a camera and deliver an e-course gives me . . . [laughter] I shudder at the thought.

But, however, that does not mean that it's not good. Actually it may be a way. But I really believe that there has got to be room for an old-fashioned way, for, as you say, oral tradition, personal interaction within a small group of people. I believe that you can actually subliminally transmit so much more this way. It's just a way that does not scale. So I don't want that in order to guarantee scale we suppress this other mode, but we can certainly augment it with e-learning and remote learning. I think that it is, again, a beautiful project that I certainly applaud. I'm not sure that I'll be successful in this particular mode. But that's just me.

Ibaraki: Again illustrating your continuing leadership, one of the things you did was you cofounded the information and security group, and because you're one of the cofounders of this very important group, can you detail your objectives in both the short and long term?

Micali: Those are actually quite simple, really: to foster interest, education, and research in cryptography. Pure and simple. I think that's the goal of any research group that has been founded, and ours is no different. It just focuses on cryptography, that I still like [laughter] despite my recent adventures in mechanism design.

Ibaraki: Now, throughout this interview it's clear that you have a lot of energy that you put to different areas, and one of them is *Advances in Computing Research*, that five-volume textbook series. Why are you so supportive of that series? You know, what motivates you? What generates all of that passion?

Micali: All right. First of all, about this specific series, let me tell you right away that I'm very proud of that volume. I mean, the volume I edited was dedicated to randomness in computation, and I believe that the interplay of *randomness and computation* is crucial to our field, and I'm proud, actually, of the confidence bestowed upon me by the many contributors—who, by the way, are all great leaders in our field—and by the editor of entire series, Franco Preparata. So, I'm very proud of that volume. I liked it, and I still like it.

But let me generalize your question a little bit. I believe that this volume, like other volumes, in whatever form—because the form changes—are occasions of common and focused reflection on what we try to understand, and are very important. To advance a given field, we need original technical contributions. But, somehow, I also find that it's important that, from time to time, we take a little bit of time to record our coordinates in our journey, right? As the saying goes, *how* we got to know things is at least as important as what we know. And I could not agree more. Unfortunately, it's much more efficient to communicate only the sleek proof of whatever we found, ignoring the torturous path that usually leads to it. The path is forgotten, and that is a pity. And those with any experience of scientific discovery know only too well that such a path is very far [laughter] from sleek and linear. No one could exactly guess in advance the conceptual barriers that preceded a solution. "Where were we, conceptually, beforehand?" I find this to be a fascinating aspect of science, too. And it's one which is very hard to reconstruct afterwards.

Personally, I do not subscribe to the theory that history helps us avoid the mistakes of the past. If it does, it may do so only in part, in a very indirect way. But I believe in the history of ideas *for its own sake.* Period. I mean, knowing humanity's past journey may actually make us better men, and, if we are better men, then we can actually do better science. And all this may be true. But if it's not true, I don't care. I still want to know the *history* of our ideas. And this is because, at the end, I really believe that, we develop *one* reality, but I don't believe that there is a *single* reality for us to discover over time, that we just, you know, peel off *the* reality. I think science is a variegated process. We always choose what to discover, and, in that sense, we continually define our own scientific reality. Most people like stories. I think that scientific development is really a fascinating story. So I really believe that once in a while we should really find the time to document the stage of the path we are in. I think that's important. It may slow us down a little bit, but it may also motivate us, right?

Hopefully, it will not stifle us. Because if you start staring at your own navel, pretty much you don't look further up anymore. Nevertheless, I think it's a risk

worth taking. Ultimately, if we don't care about how we got here, we may also not consider it important to decide where we want to go.

Ibaraki: Silvio, when I look at your profile of all the things you've done in the past, it's just an incredible profile, just an inspiring list of contributions over so many years. And as a result, you've also won some awards and recognitions. So can you share some valuable experiences and lessons from your prior awards and recognitions?

Micali: Well, OK, valuable or not, you know, my experience seems to follow the following track: First I'm happy, then I'm depressed, then I meet other awardees and I feel better again. That's the trajectory. Somehow I meet these other awardees for the first time, like, let's say, in the induction to an academy. Sometimes I actually have first and very different discussions with these people I always wanted to know, and because these are very motivated people, they tell me about their own goals. These are their goals, not mine. But somehow I realize how worthy and clear their very personal and very different goals are. We live in an era of extreme specialization in science, right? Most of the time, you know, I don't even walk to another floor, and even less to another department. So these I find very special and very motivating moments.

Ibaraki: Silvio, you laid many of the foundational pillars in your pioneering work, and distilling from your experiences, what are the greater burning challenges and research problems for today's youth to solve, to inspire them to go into computing?

Micali: Get into computing! Because computation is everywhere. Perhaps computation is a mental construct that we superimpose to the world, but then we only experience the world via ourselves. So computation is everywhere, in one way or another. So the real questions that I'd like to know and try to induce others to solve is to what extent can we use computation to understand physical, biological, and social laws, and can we perhaps use computation to influence some of these laws? I think these are very big questions, and we need all the manpower we can get to answer them, or even to scratch at their answers.

Ibaraki: It's interesting, your answer kind of reminds me of this folded game where there's these sort of problems of how proteins fold into enzymes, and now they use computers, and just people. They crowdsource it. It's a solution, where they throw it out to math as a people, including middle-schoolers, and they solve problems in this area that couldn't be solved by supercomputers and experienced researchers. Or there's this other online game which they're using to model economic behavior. So it's kind of interesting, this idea that computing is everywhere, and how can it influence some of the other domains that are out there. Or perhaps it is very