The Cryptographic Lens: visions

Shafi Goldwasser
MIT
Weizmann
1983

FOREWORD

The 15th Annual
Symposium on
April 25-27,
Computability

on January 5,
papers. The pa-
anticipated that
consideration,
sponsoring organ-
contributed to

David Harel
Richard M. Karp
Nancy Lynch
Christos H. Papadimitriou
Ronald L. Rivest
Walter L. Ruzzo
Joel Seiferas
1983-2013: A Remarkable Journey

• Theoretical Computer Science
  Interaction, Randomization, Locality

• Impact on Technology and Science

The Computational Lens
The Cryptographic Lens

On Theoretical Computer Science

On Science and Technology
Historically

Shannon
“A Mathematical Theory of Communication” (1948)
“A Communication Theory of Secrecy Systems” (1945)

Turing
Inventor of the Universal computing machine
Theory and Practice: Breaking the enigma

War Time Research
Modern Cryptography
is not (just) about fighting the bad guys

- **Enabler** of “Surprising Abilities” which often seem paradoxical in the physical world
- **Catalyst** notions and techniques led to a series of “intellectual” leaps in TOC
- **Future** enable taking advantage of enormous data availability and global connectivity while keeping “civil liberties” and “economic stability” in check.
“Paradoxical” Abilities 1983-

• Exchanging Secret Messages without Ever Meeting

• Simultaneous Contract Signing Over the Phone

• Generating exponentially long pseudo random strings indistinguishable from random

• Proving a theorem without revealing the proof

• Playing any digital game without referees

• Private Information Retrieval

• Arbitrary Computations on Encrypted Data
Unifying Theme:
The Presence of the Adversary

- Integral Part of the Definition of the Problem
- Determines the Quality of Acceptable Solutions
- The Key to Analysis of Complex Systems
The Power of the Adversary

• Make no assumptions on the Adversary strategy

• Worst Case: Do not assume Adversary is Random

• But will assume **Computationally Bounded**
  
  – Realistic
  
  – Great power: Enlarges the range of Application
“Axiom 1”: Computationally Indistinguishability

If the “Adversary” cannot tell apart two different probability distributions then they are the “same”.

Any Poly Time Algorithm

Encryption, Pseudo Randomness, Simultaneity, Correctness
Computationally Indistinguishable Encryption

Probability distributions = encryptions of messages.

Any Poly-time Eavesdropper

Encryption Hiding All Partial Information is Possible [GM82]
Computationally Indistinguishable Randomness

Probability Distributions = exponentially long strings which adversary can randomly access

Any Poly Time Statistical Test

Pseudo Randomness Generation is Possible [BM82,Y82,GGM84]
“Axiom 2”: If you can simulate, might as well stay at home

The “insiders view” gives adversary zero knowledge if he can generate computationally indistinguishable “simulated view”
Catalytic Developments 1983-

- Zero Knowledge Proofs
  - Probabilistic Proof Systems
  - Hardness of Approximation
  - Delegating Computation to the Cloud
  - Quantum Interactive Experiments

- Pseudo Random Generators & Functions
  - De-randomization of randomized complexity classes
  - Simple Concepts are un learnable
  - Impossibility of Lower Bounds by Natural Proofs

- Hard Core Bit Proofs
  - List Decoding of Hadamard Codes
  - Efficient List Decoding for Reed-Solomon codes
  - Explicit Codes which achieve the list decoding bound

- Oblivious Transfer
  - Private Information Retrieval
  - Locally Decodable Codes
  - Linear rate codes with sub-linear decoding

- Techniques for Average Case Hardness
  - Random Self Reducability
  - Algorithms for Self/Testing
  - Property Testing
Classical Proofs

\[ a^2 + b^2 \]

Axiom 1
Axiom 2
Axiom 1 \( \Rightarrow \) A
A \( \Rightarrow \) B
QED

Prime-Number Thm
Ex: Efficiently \( Q(\text{Verifiable}) \) is Provable

Prover

Claim

Verifier

Solution \( x_1, \ldots, x_n \)

Hard Working

Polynomial Time in claim size

After interaction, Verifier knows:

1) Equation is solvable
2) A particular solution

Checks proof
Accept if satisfiable

Is there any other way?
I will not give you the solution, but I will prove to you that I could if I felt like it.

Randomness

Interaction
Claim: $y = x^2 \mod N$ is solvable

Consider the two equations

(1) $z=r^2 \mod n$

(2) $zy=r^2 y \mod n$

If I solved both for you, you would be 100% certain that the claim is true since $\frac{\sqrt{zy}}{\sqrt{z}} = \sqrt{y}$

So, I will only give you a solution to one of the equations.

You choose which!

Accepts claim if he gets the right solutions

$\text{Prob}_{\text{coins}}(\text{Verifier catches mistake}) \geq 1-(1/2)^k$
Zero Knowledge Interactive Proofs (ZK-IP)

[GMR85]

"x in L?"

Prover

q1

a1

q2

Verifier

Randomized
Polynomial time

Accepts or Rejects
claim

COMPLETENESS: if $x \in L$, Bob will always accept.

This is what a proof ultimately is!

ZERO KNOWLEDGE
Many Uses of Zero Knowledge

Lots of Applications to cryptography..
Due to generality

**Theorem**[GoldreichMicaliWigderson86]:
If One Way Functions exist,
Any NP statement has a ZK interactive proof

Zero Knowledge and Nuclear Disarmament
[BarakGlasserGoldstone11]
Catalyst

Decoupled “Correctness” from “Knowledge of the proof”

Ask new questions about nature of proof

Questions have been asked and answered in last 25+ years leading up to current research on cloud computing
Classically: Can Efficiently Verify

<table>
<thead>
<tr>
<th>Class</th>
<th>Can Efficiently Verify</th>
<th>EQ(x₁,…,xₙ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>✓</td>
<td>∃ solution</td>
</tr>
<tr>
<td>Co-NP</td>
<td>?</td>
<td>0 solutions</td>
</tr>
<tr>
<td>#P</td>
<td>?</td>
<td>2¹⁰⁰ - 13 solutions</td>
</tr>
<tr>
<td>PSPACE</td>
<td>?</td>
<td>∀ ∃ ∀ ... ∈</td>
</tr>
</tbody>
</table>

Can you prove more via interactive proofs?
Interactively Provable = IP


NP
Co-NP
#P
PSPACE = IP

Other Ways to define probabilistic proof systems?
The Arrival of the Second Prover (MIP)

[BenorGoldwasserKilianWigderson88]

NP ✔
Co-NP ✔
#P ✔
PSPACE ✔

Why would two be better than one?
MIP has Zero Knowledge Proofs for NP unconditionally
Accept/Reject

Can check consistency, provers get caught if deviate
The Power of the Second Prover (MIP)

NP ✔
Co-NP ✔
#P ✔
PSPACE ✔
NEXPTIME ✔

[BabaiFortnow
Lund90]

Claim: ∃ solution for > 99% of the equations

\[
\begin{align*}
    x_1 + x_2 + x_3 &= 1 & x_1 + x_4 + x_7 &= 0 \\
    x_4 + x_5 + x_6 &= 1 & x_1 + x_4 + x_7 &= 0 \\
    x_7 + x_8 + x_9 &= 1 & x_3 + x_6 + x_9 &= 0
\end{align*}
\]

Led to PCP theorem: NP statements can be verified by reading a constant number of bits.

Requests from P1: Solution to equation, i.e. \( x_1, x_4, x_7 \)

Requests from P2: Value of variable in equation

Far Reaching Consequences to showing hardness of approximation.

Much less communication!
Q: Can the correctness of a QBP computation be even checked by a classical verifier? [AharonovBenorEban10]

In a Parallel Universe

PSPACE = QIP

NEXPTIME = QMIP

[JJUN10, KMY03]

Theorem [ReichardtUngerVazirani13]: A Classical Verifier Can Verify the Computation of Two Entangled but Non-Communicating BQP Algorithms
The Evolution of Computing
The Evolution of Computing
A Migration of Data
A Migration of Computation
Brave New World... Enormous Potential in Globalization of Knowledge

Can we do it all
Without relinquish of control
Challenge 1:
Verify correctness of remote storage/computation

Why trust the server?
**Challenge:** to delegate P time computation so that Prover’s task **not much harder** than computing

1) Compute Program $f$
2) Prove Results

Interactive Proof for $L_f = \{(x,y) \text{ s.t. } f(x)=y\}$

$IP=PSPACE \Rightarrow$ any space $S$ algorithm, can be “delegated” to $2^{poly(S)}$ time prover and verified by $poly(S)$ time verifier
Active Research Area

<table>
<thead>
<tr>
<th>Model</th>
<th>Computation</th>
<th>Prover Time</th>
<th>Verifier Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive Proof [GoldwasserKalaiRothlum08]</td>
<td>CKT SIZE S, depth D</td>
<td>poly(S)</td>
<td>Quasi $</td>
</tr>
<tr>
<td>Computational Soundness. Assume FHE. [KalaiRazRothblum13]</td>
<td>TIME(T) Turing Machine</td>
<td>Poly(T,k)</td>
<td>Quasi $</td>
</tr>
<tr>
<td>Computational Soundness. Stronger Ass. [BitanksiCanettiCiessaTrom13]</td>
<td>NTIME(T) RAM</td>
<td>Poly(T,k)</td>
<td>Quasi $</td>
</tr>
<tr>
<td>Interactive Proof For $\varepsilon$-proximity [RothbVadanWigderson13]</td>
<td>CKT SIZE S, depth D</td>
<td>Poly($\varepsilon^{-1}, S$)</td>
<td>Sublinear $</td>
</tr>
</tbody>
</table>
Challenge 2: Compute on Encrypted Data

Program: $f$

Input Data: $x$

Privacy + Functionality?
Fully Homomorphic Encryption (FHE)

Hailed tool for computing on encrypted data
But, is it enough?

How FHE works:

\[ E[x_1] E[x_2] \ldots E[x_n] \]

FHE is not enough when the evaluator needs to decrypt the computation results.

When would we want to do that?
Example 1: Decrypt for Classification

Sender

\[ E[\text{email}] \rightarrow \text{Eval of spam filter} \rightarrow \text{Spam?} \rightarrow E[\text{email}] \rightarrow \text{Receiver} \]

Need to decrypt “spam filter” result but nothing else!
Example 2: Decrypt for Maintaining both our Civil Liberties & Safety

Snap

E(photo) → Suspect Data Base

Eval of comparison to suspect face

Yes, suspect is in photo

Law and Order

Need to know if suspect appears in the scene but nothing else!
Example 3: Conduct Medical study on Confidential Medical Information

- **Laboratory**: Evaluate algorithm checking for gene presence
- **Drug Company**: Tally positive
- **Make new gene therapy**

Need to know if result of the blood test are positive for X, not entire profile!
Filterable Decryption = Functional Encryption
[...BonehSahaiWaters11, O’Neill11]

Allow server to compute partial information \( f(x) \) from \( E(x) \) but nothing else:

Using \( sk_f \),
Server can compute \( f(x) \)

Security def: can simulate server’s view given \( f(x) \) even without seeing \( E[x] \).

Is this possible?

For inner product functions[\text{KSW’08, SSW09}];

More generally if you allow a ciphertext \( E[x] \) size which as large as f’s circuit size [\text{GVW12}]
Succinct Filterable Decryption
[GoldwasserKalaiPopaVinodZeldovich13]

Theorem:
Succinct Filterable Decryption that supports any polynomial time functions assuming the Sub-Exponential Hardness of Learning with Errors

Succinct:
F is circuit of depth d $\Rightarrow$ ciphertexts growing in d

Corollary: can address all of the aforementioned examples and ...much more
Corollary: Add function privacy & get ``obfuscation variant’’
The Cryptographic Lens

Our Physical world intuition should not constrain our expectation for what is possible for “Digital Privacy“

How can today’s Cryptographic methods and fine control of information affect complexity theory of tomorrow?