Turing and the Growth of Cryptography

Ronald L. Rivest

Viterbi Professor of EECS
MIT, Cambridge, MA

BU Turing 100 Celebration
November 11, 2012

Outline

Early context

Turing and crypto

70’s – PK Crypto

Crypto theory

Crypto business

Crypto policy

Attacks

More New Directions

Conclusions

Outline

Early context

Turing and crypto

70’s – PK Crypto

Crypto theory

Crypto business

Crypto policy

Attacks

More New Directions

Conclusions

Pierre de Fermat (1601-1665)
Leonhard Euler (1707–1783)

Fermat's Little Theorem (1640):

For any prime p and any a, $1 \leq a < p$:

$$a^{p-1} = 1 \pmod{p}$$

Pierre de Fermat (1601-1665)
Leonhard Euler (1707–1783)

Fermat's Little Theorem (1640):
For any prime p and any a, $1 \le a < p$:

$$a^{p-1} = 1 \pmod{p}$$

Euler's Theorem (1736):
If $\gcd(a, n) = 1$, then

$$a^{\phi(n)} = 1 \pmod{n} \ ,$$

where $\phi(n) = \#$ of $x < n$ such that $\gcd(x, n) = 1$.

Carl Friedrich Gauss (1777-1855)

Published *Disquisitiones Aritmeticae* at age 21

Carl Friedrich Gauss (1777-1855)

Published *Disquisitiones Aritmeticae* at age 21

“The problem of *distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors* is known to be one of the most important and useful in arithmetic. . . . the dignity of the science itself seems to require solution of a problem so elegant and so celebrated.”

William Stanley Jevons (1835–1882)

Published *The Principles of Science* (1874)

William Stanley Jevons (1835–1882)

Published *The Principles of Science* (1874)

Gave world's first *factoring challenge*:

> *"What two numbers multiplied together will produce 8616460799 ? I think it unlikely that anyone but myself will ever know."*

William Stanley Jevons (1835–1882)

Published *The Principles of Science* (1874)

Gave world's first *factoring challenge*:

> *"What two numbers multiplied together will produce 8616460799 ? I think it unlikely that anyone but myself will ever know."*

Factored by Derrick Lehmer in 1903. (89681 $*$ 96079)

World War I – Radio

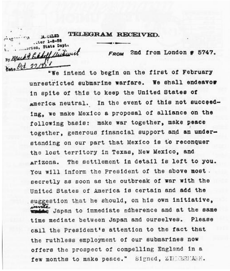
- A marvelous new communication technology—*radio* (Marconi, 1895)—enabled instantaneous communication with remote ships and forces, but also gave all transmitted messages to the enemy.

World War I – Radio

- A marvelous new communication technology—*radio* (Marconi, 1895)—enabled instantaneous communication with remote ships and forces, but also gave all transmitted messages to the enemy.
- Use of cryptography soars.

World War I – Radio

- A marvelous new communication technology—*radio* (Marconi, 1895)—enabled instantaneous communication with remote ships and forces, but also gave all transmitted messages to the enemy.
- Use of cryptography soars.



[illegible] CELED
[illegible] ctor 1-8-58
[illegible] State Dept.
By Mark A. Eckhoff [illegible]
Date Oct. 22,1957

TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

Decipherment of *Zimmermann Telegram* by British made American involvement in World War I inevitable.

Outline

- Early context
- Turing and crypto
- 70's – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

Alan Turing (1912–1954)

Developed foundations of theory of computability (1936).

Church-Turing Thesis (model of computation doesn't matter).

World War II – Enigma, Purple, JN25, Naval Enigma

- Cryptography performed by (typically, rotor) *machines*.

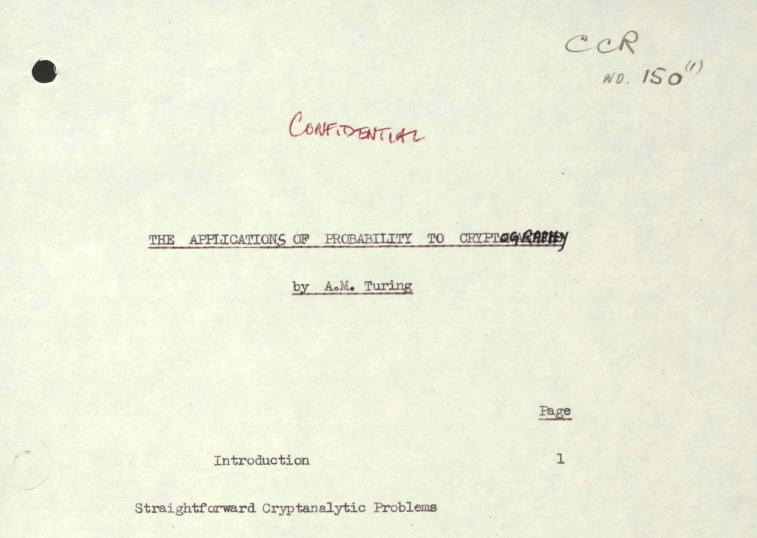
World War II – Enigma, Purple, JN25, Naval Enigma

- Cryptography performed by (typically, rotor) *machines*.
- Work of Alan Turing and others at Bletchley Park, and William Friedman and others in the USA, on breaking of Axis ciphers had great success and immense impact.

World War II – Enigma, Purple, JN25, Naval Enigma

- Cryptography performed by (typically, rotor) *machines*.
- Work of Alan Turing and others at Bletchley Park, and William Friedman and others in the USA, on breaking of Axis ciphers had great success and immense impact.
- Cryptanalytic effort involved development and use of early computers (Colossus).

Still learning about Turing's contributions



CCR

NO. 150(1)

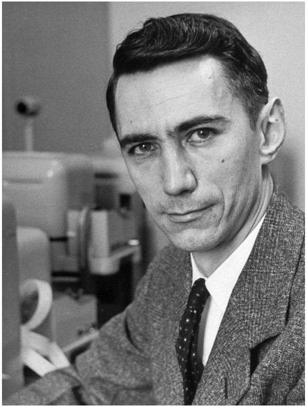
CONFIDENTIAL

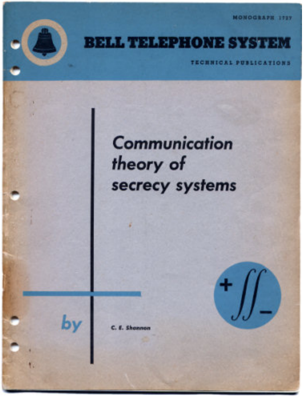
THE APPLICATIONS OF PROBABILITY TO CRYPTOGRAPHY

by A.M. Turing

(Declassified May 2012.)

Claude Shannon (1916–2001)





MONOGRAPH 1727

BELL TELEPHONE SYSTEM

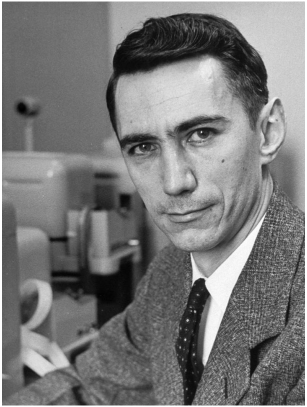
TECHNICAL PUBLICATIONS

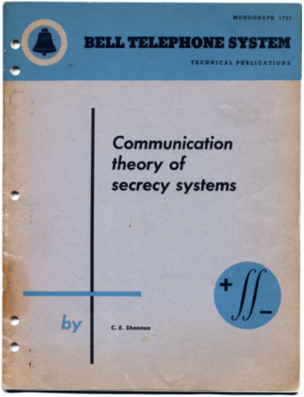
Communication
theory of
secrecy systems

by C. E. Shannon

- “Communication Theory of Secrecy Systems” Sept 1945 (Bell Labs memo, classified).

Claude Shannon (1916–2001)





MONOGRAPH 1727

BELL TELEPHONE SYSTEM

TECHNICAL PUBLICATIONS

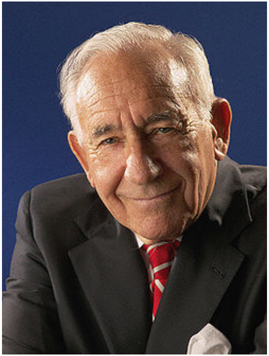
Communication
theory of
secrecy systems

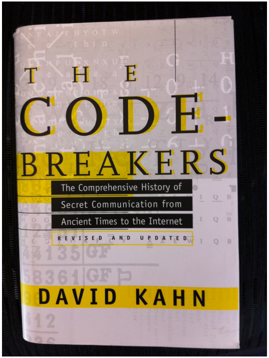
by

C. E. Shannon

- “Communication Theory of Secrecy Systems” Sept 1945 (Bell Labs memo, classified).
- Information-theoretic in character—proves unbreakability of one-time pad. (Published 1949).

Kahn – The Codebreakers





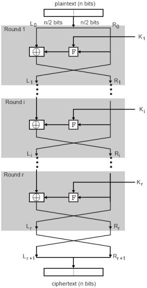
THE CODE-BREAKERS
The Comprehensive History of Secret Communication from Ancient Times to the Internet
REVISED AND UPDATED
DAVID KAHN

In 1967 David Kahn published
The Codebreakers—The Story of Secret Writing.
A monumental history of cryptography.
NSA attempted to suppress its publication.

Outline

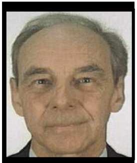
- Early context
- Turing and crypto
- 70’s – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

DES – U.S. Data Encryption Standard (1976)



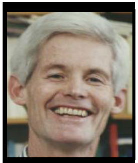
DES Designed at IBM; Horst Feistel supplied key elements of design, such as ladder structure. NSA helped, in return for keeping key size at 56 bits.(?)

Computational Complexity


Hartmanis

Stearns

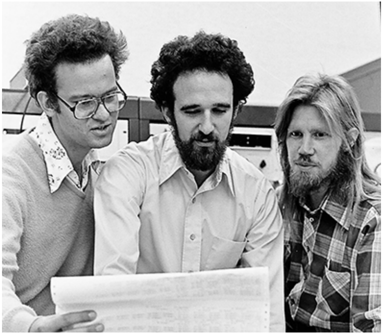
Blum


Cook

Karp

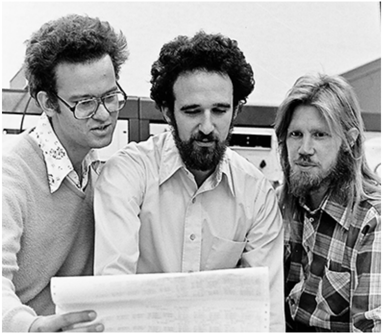
- Theory of Computational Complexity started in 1965 by Hartmanis and Stearns; expanded on by Blum, Cook, and Karp.
- Key notions: polynomial-time reductions; NP-completeness.

Invention of Public Key Cryptography



- Ralph Merkle, and independently Marty Hellman and Whit Diffie, invented the notion of *public-key cryptography*.

Invention of Public Key Cryptography



- Ralph Merkle, and independently Marty Hellman and Whit Diffie, invented the notion of *public-key cryptography*.
- In November 1976, Diffie and Hellman published *New Directions in Cryptography*, proclaiming

  > *"We are at the brink of a revolution in cryptography."*

Public-key encryption (as proposed by Diffie/Hellman)

- Each party A has a *public key* PK_A others can use to encrypt messages to A:

$$C = PK_A(M)$$

Public-key encryption (as proposed by Diffie/Hellman)

- Each party A has a *public key* PK_A others can use to encrypt messages to A:

$$C = PK_A(M)$$

- Each party A also has a *secret key* SK_A for decrypting a received ciphertext C:

$$M = SK_A(C)$$

Public-key encryption (as proposed by Diffie/Hellman)

- Each party A has a *public key* PK_A others can use to encrypt messages to A:

$$C = PK_A(M)$$

- Each party A also has a *secret key* SK_A for decrypting a received ciphertext C:

$$M = SK_A(C)$$

- It is easy to compute matching public/secret key pairs.

Public-key encryption (as proposed by Diffie/Hellman)

- Each party A has a *public key* PK_A others can use to encrypt messages to A:

$$C = PK_A(M)$$

- Each party A also has a *secret key* SK_A for decrypting a received ciphertext C:

$$M = SK_A(C)$$

- It is easy to compute matching public/secret key pairs.
- Publishing PK_A does not compromise SK_A! It is *computationally infeasible* to obtain SK_A from PK_A. Each public key can thus be safely listed in a public directory with the owner's name.

Digital Signatures (as proposed by Diffie/Hellman)

- Idea: sign with SK_A; verify signature with PK_A.

Digital Signatures (as proposed by Diffie/Hellman)

- Idea: sign with SK_A; verify signature with PK_A.
- A produces signature σ for message M

$$\sigma = SK_A(M)$$

Digital Signatures (as proposed by Diffie/Hellman)

- Idea: sign with SK_A; verify signature with PK_A.
- A produces signature σ for message M

$$\sigma = SK_A(M)$$

- Given PK_A, M, and σ, anyone can verify validity of signature σ by checking:

$$M \stackrel{?}{=} PK_A(\sigma)$$

Digital Signatures (as proposed by Diffie/Hellman)

- Idea: sign with SK_A; verify signature with PK_A.
- A produces signature σ for message M

$$\sigma = SK_A(M)$$

- Given PK_A, M, and σ, anyone can verify validity of signature σ by checking:

$$M \stackrel{?}{=} PK_A(\sigma)$$

- Amazing ideas!

Digital Signatures (as proposed by Diffie/Hellman)

- Idea: sign with SK_A; verify signature with PK_A.
- A produces signature σ for message M

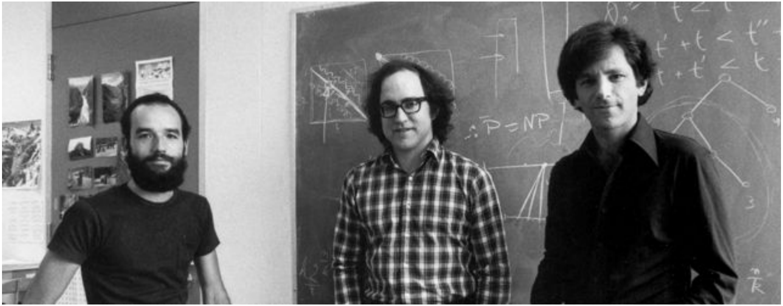
$$\sigma = SK_A(M)$$

- Given PK_A, M, and σ, anyone can verify validity of signature σ by checking:

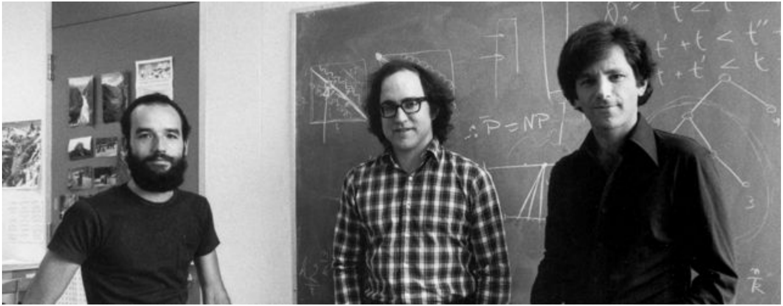
$$M \stackrel{?}{=} PK_A(\sigma)$$

- Amazing ideas!
- But they couldn't see how to implement them...

RSA (Ron Rivest, Adi Shamir, Len Adleman, 1977)

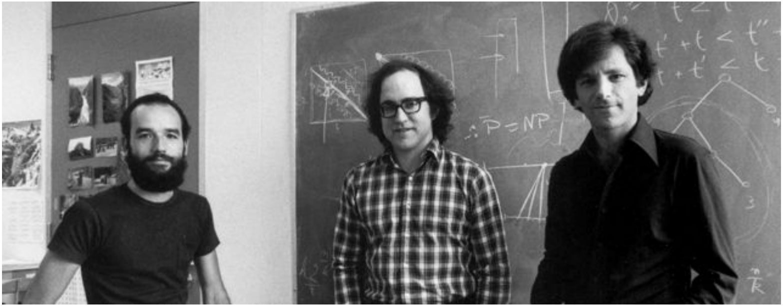


RSA (Ron Rivest, Adi Shamir, Len Adleman, 1977)



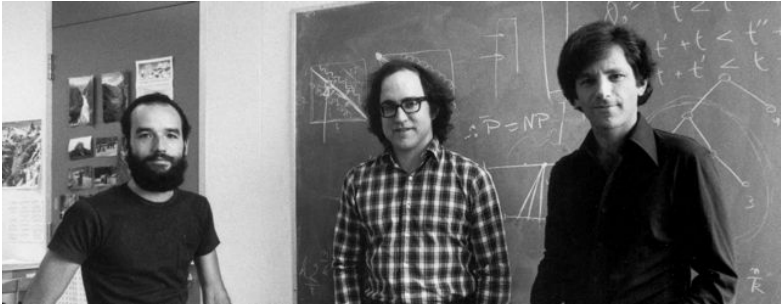
- Security relies (in part) on inability to factor product *n* of two large primes *p*, *q*.

RSA (Ron Rivest, Adi Shamir, Len Adleman, 1977)



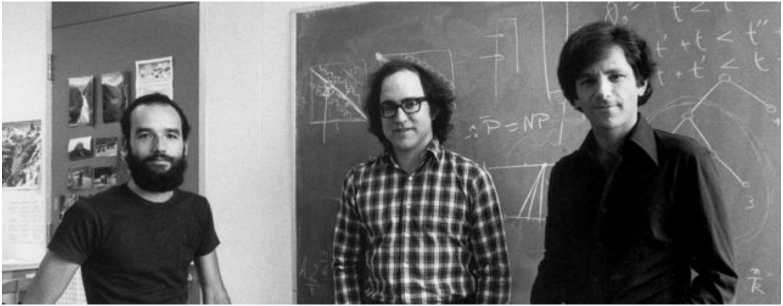
- Security relies (in part) on inability to factor product *n* of two large primes *p*, *q*.
- $PK = (n, e)$ where $n = pq$ and $\gcd(e, \phi(n)) = 1$

RSA (Ron Rivest, Adi Shamir, Len Adleman, 1977)



- Security relies (in part) on inability to factor product n of two large primes p, q.
- $PK = (n, e)$ where $n = pq$ and $\gcd(e, \phi(n)) = 1$
- $SK = d$ where $de = 1 \mod \phi(n)$

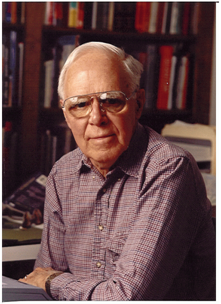
RSA (Ron Rivest, Adi Shamir, Len Adleman, 1977)

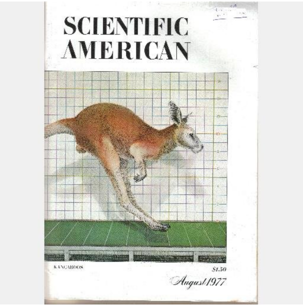


- Security relies (in part) on inability to factor product n of two large primes p, q.
- $PK = (n, e)$ where $n = pq$ and $\gcd(e, \phi(n)) = 1$
- $SK = d$ where $de = 1 \mod \phi(n)$
- Encryption/decryption (or signing/verify) are simple:

$$\begin{aligned} C &= PK(M) = M^e \mod n \\ M &= SK(C) = C^d \mod n \end{aligned}$$

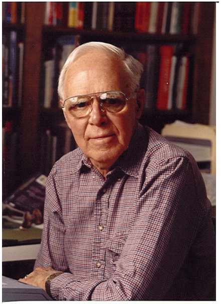
Martin Gardner column and RSA-129 challenge

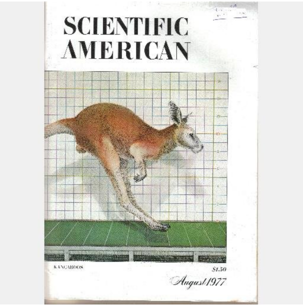




- Described public-key and RSA cryptosystem in his Scientific American column, *Mathematical Games*

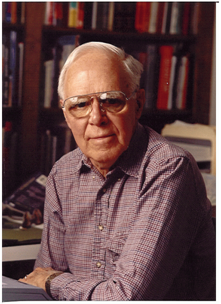
Martin Gardner column and RSA-129 challenge

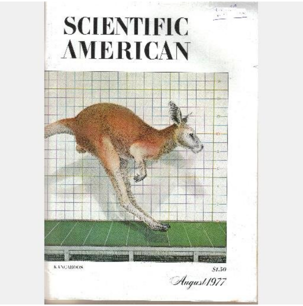




- Described public-key and RSA cryptosystem in his Scientific American column, *Mathematical Games*
- Offered copy of RSA technical memo.

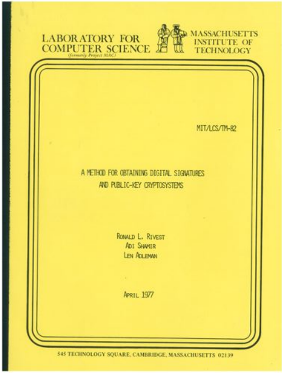
Martin Gardner column and RSA-129 challenge





- Described public-key and RSA cryptosystem in his Scientific American column, *Mathematical Games*
- Offered copy of RSA technical memo.
- Offered $100 to first person to break challenge ciphertext based on 129-digit product of primes. (Our) estimated time to solution: 40 quadrillion years

Publication of RSA memo and paper



LABORATORY FOR COMPUTER SCIENCE
(formerly Project MAC)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

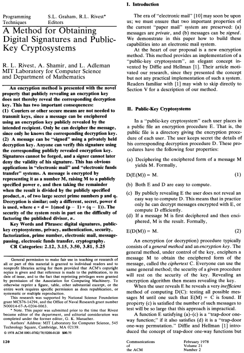
MIT/LCS/TM-82

A METHOD FOR OBTAINING DIGITAL SIGNATURES
AND PUBLIC-KEY CRYPTOSYSTEMS

RONALD L. RIVEST
ADI SHAMIR
LEN ADLEMAN

APRIL 1977

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139



Programming Techniques — S.L. Graham, R.L. Rivest* Editors

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R. L. Rivest, A. Shamir, and L. Adleman
MIT Laboratory for Computer Science
and Department of Mathematics

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. This has two important consequences:
(1) Couriers or other secure means are not needed to transmit keys, since a message can be enciphered using an encryption key publicly revealed by the intended recipient. Only he can decipher the message, since only he knows the corresponding decryption key.
(2) A message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems. A message is encrypted by representing it as a number M, raising M to a publicly specified power e, and then taking the remainder when the result is divided by the publicly specified product, *n*, of two large secret prime numbers p and q. Decryption is similar; only a different, secret, power d is used, where e * d = 1(mod (p − 1) * (q − 1)). The security of the system rests in part on the difficulty of factoring the published divisor, *n*.

Key Words and Phrases: digital signatures, public-key cryptosystems, privacy, authentication, security, factorization, prime number, electronic mail, message-passing, electronic funds transfer, cryptography.

CR Categories: 2.12, 3.15, 3.50, 3.81, 5.25

This research was supported by National Science Foundation grant MCS76-14294, and the Office of Naval Research grant number N00014-67-A-0204-0063.

* Note. This paper was submitted prior to the time that Rivest became editor of the department, and editorial consideration was completed under the former editor, G. K. Manacher.

Authors' Address: MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139.

120

I. Introduction

The era of "electronic mail" [10] may soon be upon us; we must ensure that two important properties of the current "paper mail" system are preserved: (a) messages are *private*, and (b) messages can be *signed*. We demonstrate in this paper how to build these capabilities into an electronic mail system.

At the heart of our proposal is a new encryption method. This method provides an implementation of a "public-key cryptosystem", an elegant concept invented by Diffie and Hellman [1]. Their article motivated our research, since they presented the concept but not any practical implementation of such a system. Readers familiar with [1] may wish to skip directly to Section V for a description of our method.

II. Public-Key Cryptosystems

In a "public-key cryptosystem" each user places in a public file an encryption procedure E. That is, the public file is a directory giving the encryption procedure of each user. The user keeps secret the details of his corresponding decryption procedure D. These procedures have the following four properties:

(a) Deciphering the enciphered form of a message M yields M. Formally,

$$D(E(M)) = M. \quad (1)$$

(b) Both E and D are easy to compute.

(c) By publicly revealing E the user does not reveal an easy way to compute D. This means that in practice only he can decrypt messages encrypted with E, or compute D efficiently.

(d) If a message M is first deciphered and then enciphered, M is the result. Formally,

$$E(D(M)) = M. \quad (2)$$

An encryption (or decryption) procedure typically consists of a *general method* and an *encryption key*. The general method, under control of the key, enciphers a message M to obtain the enciphered form of the message, called the *ciphertext* C. Everyone can use the same general method; the security of a given procedure will rest on the security of the key. Revealing an encryption algorithm then means revealing the key.

When the user reveals E he reveals a very *inefficient* method of computing D(C): testing all possible messages M until one such that E(M) = C is found. If property (c) is satisfied the number of such messages to test will be so large that this approach is impractical.

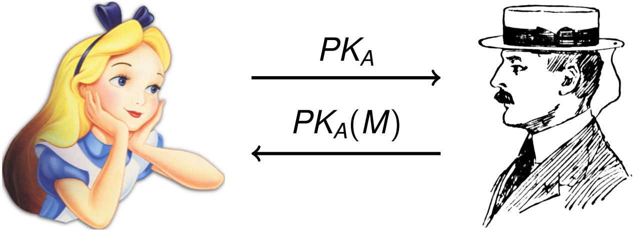
A function E satisfying (a)–(c) is a "trap-door one-way function;" if it also satisfies (d) it is a "trap-door one-way permutation." Diffie and Hellman [1] introduced the concept of trap-door one-way functions but

Communications of the ACM — February 1978, Volume 21, Number 2

LCS-82 Technical Memo (April 1977)
CACM article (Feb 1978)

Alice and Bob (1977, in RSA paper)

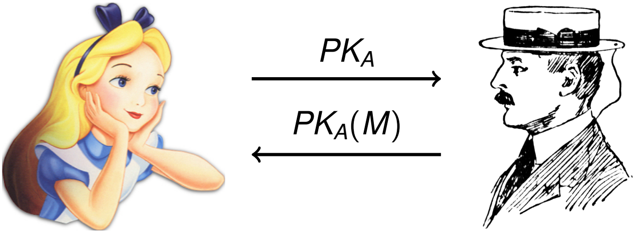
Alice and Bob (1977, in RSA paper)



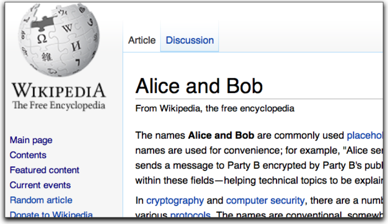
PK_A

$PK_A(M)$

Alice and Bob (1977, in RSA paper)



Alice and Bob now have a life of their own—they appear in hundreds of crypto papers, in `xkcd`, and even have their own Wikipedia page:



Alice and Bob

From Wikipedia, the free encyclopedia

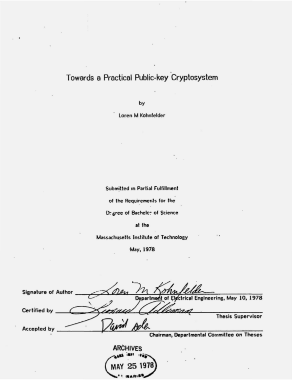
The names **Alice and Bob** are commonly used placehol
names are used for convenience; for example, "Alice sei
sends a message to Party B encrypted by Party B's publ
within these fields—helping technical topics to be explair

In cryptography and computer security, there are a numb
various protocols. The names are conventional, somewh

Independent Invention of Public-Key Revealed

In 1999 GCHQ announced that James Ellis, Clifford Cocks, and Malcolm Williamson had invented public-key cryptography, the “RSA” algorithm, and “Diffie-Hellman key exchange” in the 1970’s, before their invention outside.

Loren Kohnfelder – Invention of Digital Certificates



Towards a Practical Public-key Cryptosystem

by

Loren M Kohnfelder

Submitted in Partial Fulfillment
of the Requirements for the
Degree of Bachelor of Science
at the
Massachusetts Institute of Technology
May, 1978

Signature of Author Loren M Kohnfelder
Department of Electrical Engineering, May 10, 1978

Certified by Leonard Adleman
Thesis Supervisor

Accepted by David Adler
Chairman, Departmental Committee on Theses

- Loren Kohnfelder’s B.S. thesis (MIT 1978, supervised by Len Adleman), proposed notion of *digital certificate*—a digitally signed message attesting to another party’s public key.

Outline

- Early context
- Turing and crypto
- 70's – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

Theoretical Foundations of Security

- “Probabilistic Encryption” Shafi Goldwasser, Silvio Micali (1982) (Encryption should be *randomized*!)

Theoretical Foundations of Security

- “Probabilistic Encryption” Shafi Goldwasser, Silvio Micali (1982) (Encryption should be *randomized*!)
- “A Digital Signature Scheme Secure Against Adaptive Chosen Message Attacks” Goldwasser, Micali, Rivest (1988) (Uses well-defined *game* to define security objective.)

The Impact of “The Turing Test” on Cryptography

- Turing (1950) asked, “Can Machines Think?”

The Impact of “The Turing Test” on Cryptography

- Turing (1950) asked, “Can Machines Think?”
- Proposed “indistinguishability test” as the answer: If you can’t tell a machine from a human by conversing with it, then the machine can think.

The Impact of “The Turing Test” on Cryptography

- Turing (1950) asked, “Can Machines Think?”
- Proposed “indistinguishability test” as the answer: If you can’t tell a machine from a human by conversing with it, then the machine can think.
- This model has become the paradigm for many definitions of security in cryptography, usually under the name “compuational indistinguishability”.

The Impact of “The Turing Test” on Cryptography

- Turing (1950) asked, “Can Machines Think?”
- Proposed “indistinguishability test” as the answer: If you can’t tell a machine from a human by conversing with it, then the machine can think.
- This model has become the paradigm for many definitions of security in cryptography, usually under the name “compuational indistinguishability”.
- Goldwasser/Micali (1984): ciphertext indistinguishability.

The Impact of “The Turing Test” on Cryptography

- Turing (1950) asked, “Can Machines Think?”
- Proposed “indistinguishability test” as the answer: If you can’t tell a machine from a human by conversing with it, then the machine can think.
- This model has become the paradigm for many definitions of security in cryptography, usually under the name “compuational indistinguishability”.
- Goldwasser/Micali (1984): ciphertext indistinguishability.
- Blum/Micali (1982), Yao (1982): pseudorandom generators.

Outline

- Early context
- Turing and crypto
- 70’s – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

World Wide Web (Sir Tim Berners-Lee, 1990)

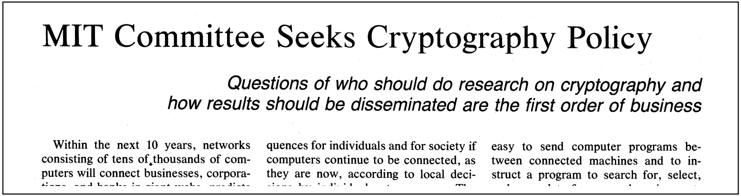
- Just as radio did, this new communication medium, the World-Wide Web, drove demand for cryptography to new heights.
- Cemented transition of cryptography from primarily military to primarily commercial.

Outline

- Early context
- Turing and crypto
- 70's – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

U.S. cryptography policy evolves

- U.S. government initially tried to control and limit public-sector research and use of cryptography
- Attempt to chill research via ITAR (1977)
- MIT “Changing Nature of Information” Committee (1981; Dertouzos, Low, Rosenblith, Deutch,Rivest,...)



MIT Committee Seeks Cryptography Policy

Questions of who should do research on cryptography and how results should be disseminated are the first order of business

Within the next 10 years, networks consisting of tens of thousands of computers will connect businesses, corporations, and banks in giant webs, predicts

quences for individuals and for society if computers continue to be connected, as they are now, according to local decisions by individual

easy to send computer programs between connected machines and to instruct a program to search for, select,

Science, 13 Mar 1981

U.S. cryptography policy evolves

- U.S. government tried to mandate availability of all encryption keys via “key escrow” and/or “Clipper Chip” (1993)

U.S. cryptography policy evolves

- U.S. government tried to mandate availability of all encryption keys via “key escrow” and/or “Clipper Chip” (1993)

U.S. cryptography policy evolves

- U.S. government tried to mandate availability of all encryption keys via “key escrow” and/or “Clipper Chip” (1993)

- Today, US policy leans toward strong cybersecurity, including strong cryptography, for all information systems as a matter of national security.

Outline

- Early context
- Turing and crypto
- 70’s – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

Factorization of RSA-129 (April 1994)

- RSA-129 =

```
11438162575788886766923577997614661201021829
67212423625625618429357069352457338978305971
23563958705058989075147599290026879543541
```

Factorization of RSA-129 (April 1994)

- RSA-129 =

```
114381625757888867669235779976146612010218296
721242362562561842935706935245733897830597 1
23563958705058989075147599290026879543541
```

- Derek Atkins, Michael Graff, Arjen Lenstra, Paul Leyland: RSA-129 =

```
3490529510847650949147849619903898133417764 6
38493387843990820577 x
3276913299326670954996198819083446141317764 2
967992942539798288533
```

Factorization of RSA-129 (April 1994)

- RSA-129 =

```
114381625757888867669235779976146612010218296
721242362562561842935706935245733897830597123
563958705058989075147599290026879543541
```

- Derek Atkins, Michael Graff, Arjen Lenstra,
  Paul Leyland: RSA-129 =

```
349052951084765094914784961990389813341776463
8493387843990820577 x
327691329932667095499619881908344614131776429
67992942539798288533
```

- 8 months work by about 600 volunteers from more than 20 countries; 5000 MIPS-years.

Factorization of RSA-129 (April 1994)

- RSA-129 =

  ```
  11438162575788886766923577997614661201021829
  67212423625625618429357069352457338978305971
  23563958705058989075147599290026879543541
  ```

- Derek Atkins, Michael Graff, Arjen Lenstra,
  Paul Leyland: RSA-129 =

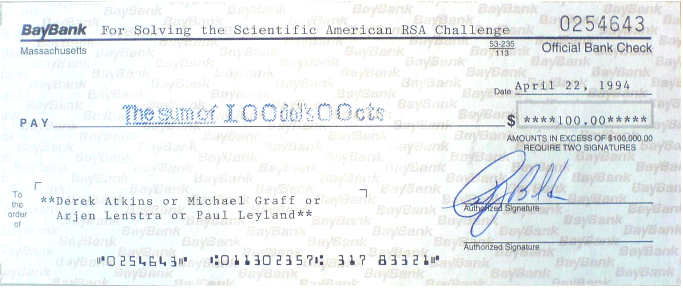
  ```
  34905295108476509491478496199038981334177646
  38493387843990820577 x
  32769132993266709549961988190834461413177642
  967992942539798288533
  ```

- 8 months work by about 600 volunteers from more than 20 countries; 5000 MIPS-years.

- secret message:

  `The Magic Words Are Squeamish Ossifrage`

THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE
THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE
THE MAGIC WORDS ARE
010, 218, 296, 72



BayBank Massachusetts

For Solving the Scientific American RSA Challenge

0254643

53-235
113

Official Bank Check

Date April 22, 1994

PAY The sum of 100 dols 00 cts $ ****100.00*****

AMOUNTS IN EXCESS OF $100,000.00 REQUIRE TWO SIGNATURES

To the order of

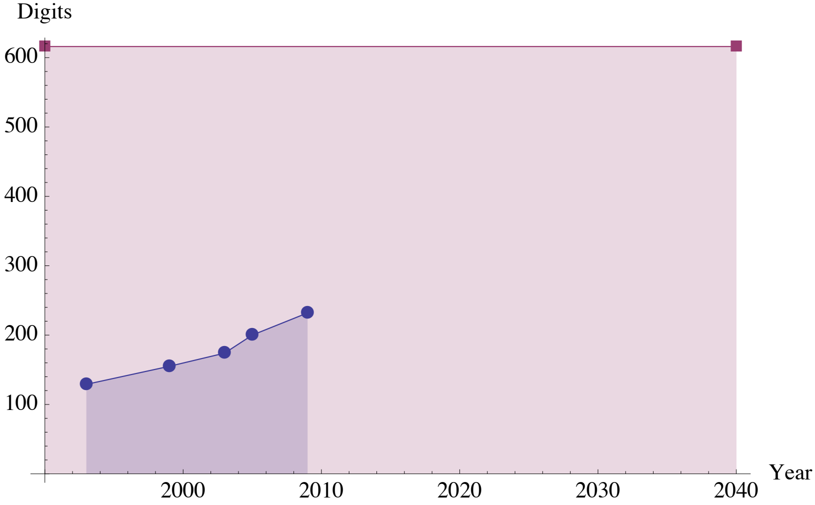
**Derek Atkins or Michael Graff or
Arjen Lenstra or Paul Leyland**

Authorized Signature

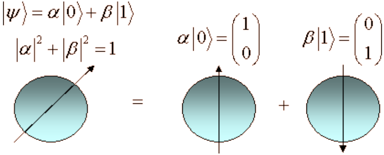
Authorized Signature

⑈0254643⑈ ⑆011302357⑆ 317 83321⑈

Factoring Records



Factoring on a Quantum Computer?



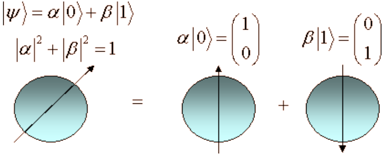
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\alpha|^2 + |\beta|^2 = 1$$

$$\alpha|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \beta|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In 1994, Peter Shor invented a fast factorization algorithm that runs on a (hypothetical) *quantum computer* and works by determining multiplicative period of elements mod *n*.

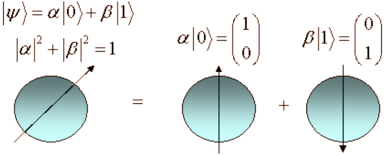
Factoring on a Quantum Computer?


$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
$|\alpha|^2 + |\beta|^2 = 1$
$\alpha|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$
$\beta|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$
=
+

In 1994, Peter Shor invented a fast factorization algorithm that runs on a (hypothetical) *quantum computer* and works by determining multiplicative period of elements mod *n*.

- In 2001, researchers at IBM used this algorithm on a (real) quantum computer to factor $15 = 3 \times 5$.

Factoring on a Quantum Computer?


$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
$|\alpha|^2 + |\beta|^2 = 1$
$\alpha|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
$\beta|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
=
+

In 1994, Peter Shor invented a fast factorization algorithm that runs on a (hypothetical) *quantum computer* and works by determining multiplicative period of elements mod *n*.

- In 2001, researchers at IBM used this algorithm on a (real) quantum computer to factor $15 = 3 \times 5$.
- Dark clouds on horizon for RSA?

Outline

- Early context
- Turing and crypto
- 70's – PK Crypto
- Crypto theory
- Crypto business
- Crypto policy
- Attacks
- More New Directions
- Conclusions

Many new research problems and directions

- secret-sharing
- anonymity
- commitments
- multi-party protocols
- elliptic curves
- crypto hardware
- key leakage
- proxy encryption
- crypto for smart cards
- password-based keys
- random oracles
- oblivious transfer
- ...

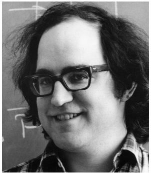
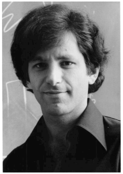
- zero-knowledge proofs
- payment systems
- voting systems
- homomorphic encryption
- lattice-based crypto
- private information retrieval
- public-key infrastructure
- concurrent protocols
- randomness extractors
- tweakable encryption
- differential cryptanalysis
- identity-based encryption
- ...

Many new research problems and directions

- secret-sharing
- anonymity
- commitments
- multi-party protocols
- elliptic curves
- crypto hardware
- key leakage
- proxy encryption
- crypto for smart cards
- password-based keys
- random oracles
- oblivious transfer
- ...

- zero-knowledge proofs
- payment systems
- voting systems
- homomorphic encryption
- lattice-based crypto
- private information retrieval
- public-key infrastructure
- concurrent protocols
- randomness extractors
- tweakable encryption
- differential cryptanalysis
- identity-based encryption
- ...

Fully Homomorphic Encryption

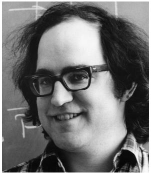
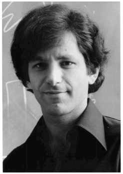



?

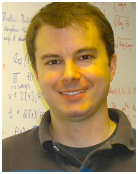
- In 1978, Rivest, Adleman, and Dertouzos asked,

  “Can one compute on encrypted data,
  while keeping it encrypted?”

Fully Homomorphic Encryption

?



!

- In 1978, Rivest, Adleman, and Dertouzos asked,

  “Can one compute on encrypted data,
  while keeping it encrypted?”

- In 2009, Craig Gentry (Stanford,IBM) gave solution based on use of lattices. If efficiency can be greatly improved, could be huge implications (e.g. for cloud computing).

Conclusions

- Cryptography is not the solution to all of our cybersecurity problems, but it is an essential component of any solution.

Conclusions

- Cryptography is not the solution to all of our cybersecurity problems, but it is an essential component of any solution.
- Research in cryptography is a fascinating blend of mathematics, statistics, theoretical computer science, electrical engineering, and psychology.

Conclusions

- Cryptography is not the solution to all of our cybersecurity problems, but it is an essential component of any solution.
- Research in cryptography is a fascinating blend of mathematics, statistics, theoretical computer science, electrical engineering, and psychology.
- While we have accomplished a lot in a few decades, much remains to be done.

Conclusions

- Cryptography is not the solution to all of our cybersecurity problems, but it is an essential component of any solution.
- Research in cryptography is a fascinating blend of mathematics, statistics, theoretical computer science, electrical engineering, and psychology.
- While we have accomplished a lot in a few decades, much remains to be done.
- Like Alice and Bob, cryptography is here to stay.

Conclusions

- Cryptography is not the solution to all of our cybersecurity problems, but it is an essential component of any solution.
- Research in cryptography is a fascinating blend of mathematics, statistics, theoretical computer science, electrical engineering, and psychology.
- While we have accomplished a lot in a few decades, much remains to be done.
- Like Alice and Bob, cryptography is here to stay.
- Turing's influence extends beyond the breaking of Enigma, to the proper formulation of adequate definitions of security.

Happy Birthday, Alan Turing!