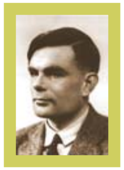
The Early Days of RSA -- History and Lessons

Ronald L. Rivest

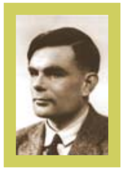
MIT Lab for Computer Science

ACM Turing Award Lecture



Lessons Learned

- Try to solve “real-world” problems
- … using computer science theory
- … and number theory.
- Be optimistic: do the “impossible”.
- Invention of RSA.
- Moore’s Law matters.
- Do cryptography in public.
- Crypto theory matters.
- Organizations matter: ACM, IACR, RSA

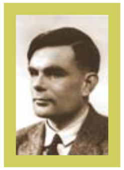


Try to solve real-world problems

- Diffie and Hellman published "New Directions in Cryptography" Nov '76:

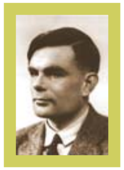
  > "We stand today at the brink of a revolution in cryptography."

- Proposed "*Public-Key Cryptosystem*". (This remarkable idea developed jointly with Merkle.)
- Introduced even more remarkable notion of *digital signatures.*
- *Good cryptography is motivated by applications. (e-commerce, mental poker, voting, auctions, …)*



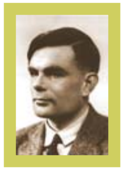
… using computer science theory

- In 1976 “complexity theory” and “algorithms” were just beginning…
- Cryptography is a “theory consumer”: it needs
  - *easy problems* (such as multiplication or prime-finding, for the “good guys”) and
  - *hard problems* (such as factorization, to defeat an adversary).



…and number theory

- Diffie/Hellman used number theory for “key agreement” (two parties agree on a secret key, using exponentiation modulo a prime number).
- Some algebraic structure seemed essential for a PKC; we kept returning to number theory and modular arithmetic…
- Difficulty of factoring not well studied then, but seemed hard…



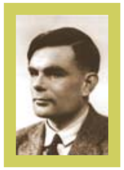
Be optimistic: do the “impossible“

- Diffie and Hellman left open the problem of realizing a PKC:

  $$D(E(M)) = E(D(M)) = M$$

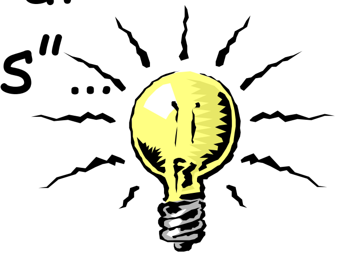
  where E is public, D is private.
- At times, we thought it impossible…
- Since then, we have learned “Meta-theorem of Cryptography“:

  Any apparently contradictory set of requirements can be met using right mathematical approach…

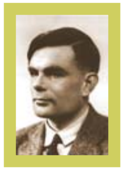


Invention of RSA

- Tried and discarded many approaches, including some “knapsack-based” ones. (Len was great at killing off bad ideas.)
- “Group of unknown size” seemed useful idea… as did “permutation polynomials”…
- After a “seder” at a student’s…
- “RSA” uses $n = pq$ product of primes:



$C = M^e \pmod{n}$ [public key (e,n)]
$M = C^d \pmod{n}$ [private key (d,n)]

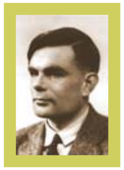


$100 RSA SciAm Challenge

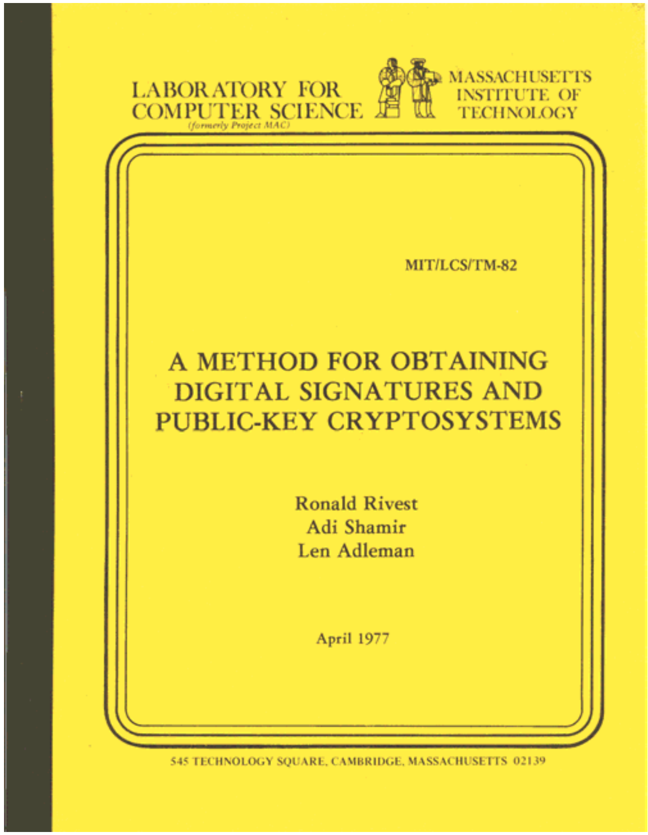
- Martin Gardner publishes *Scientific American* column about RSA in August '77, including our $100 challenge (129 digit n) and our infamous "40 quadrillion years" estimate required to factor

  RSA-129 =
  114,381,625,757,888,867,669,235,779,976,146,612,010,218,296,721,242,362,562,561,842,935,706,935,245,733,897,830,597,123,563,958,705,058,989,075,147,599,290,026,879,543,541
  (129 digits)

  or to decode encrypted message.



TM-82 4/77; CACM 2/78



LABORATORY FOR COMPUTER SCIENCE
(formerly Project MAC)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MIT/LCS/TM-82

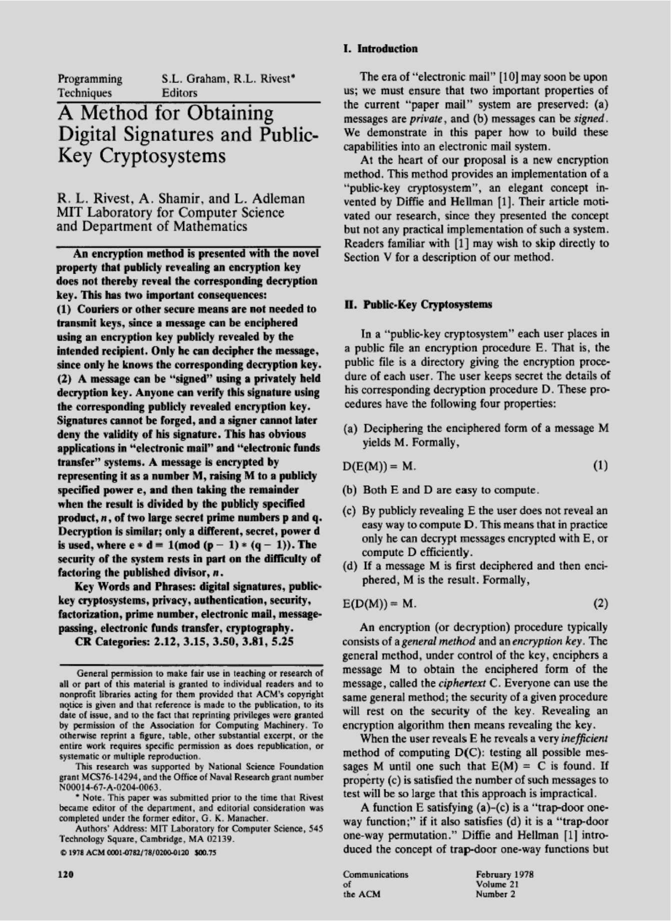
A METHOD FOR OBTAINING DIGITAL SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS

Ronald Rivest
Adi Shamir
Len Adleman

April 1977

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

(4000 mailed)



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 $e * d = 1 (\bmod (p-1) * (q-1))$. 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

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,

$$D(E(M)) = M. \tag{1}$$

(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,

$$E(D(M)) = M. \tag{2}$$

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

General permission to make fair use in teaching or research of all or part of this material is granted to individual readers and to nonprofit libraries acting for them provided that ACM's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Association for Computing Machinery. To otherwise reprint a figure, table, other substantial excerpt, or the entire work requires specific permission as does republication, or systematic or multiple reproduction.

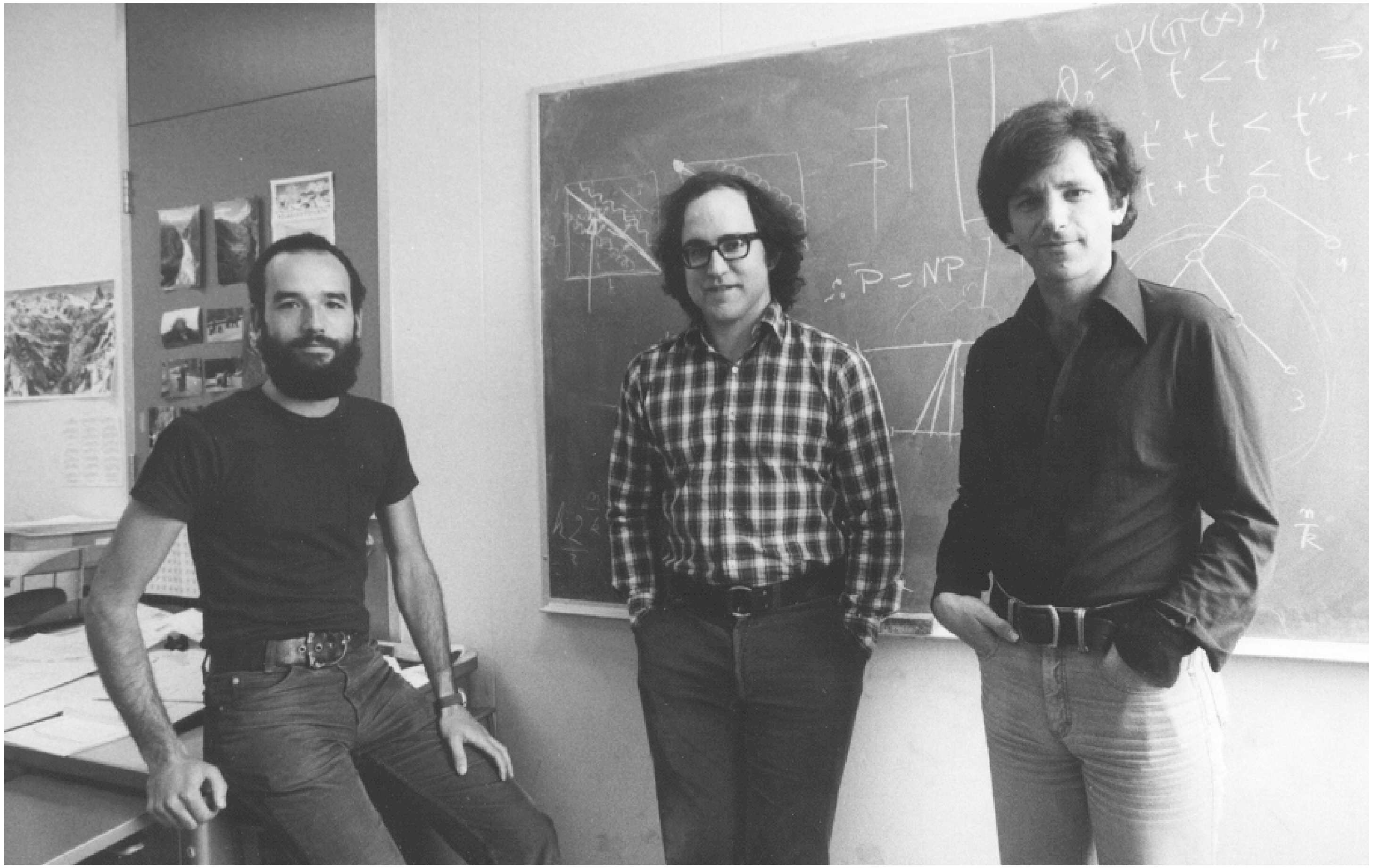
This research was supported by National Science Foundation grant MCS76-14294, and the Office of Naval Research grant number N00014-67-A-0204-0063.

* 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.

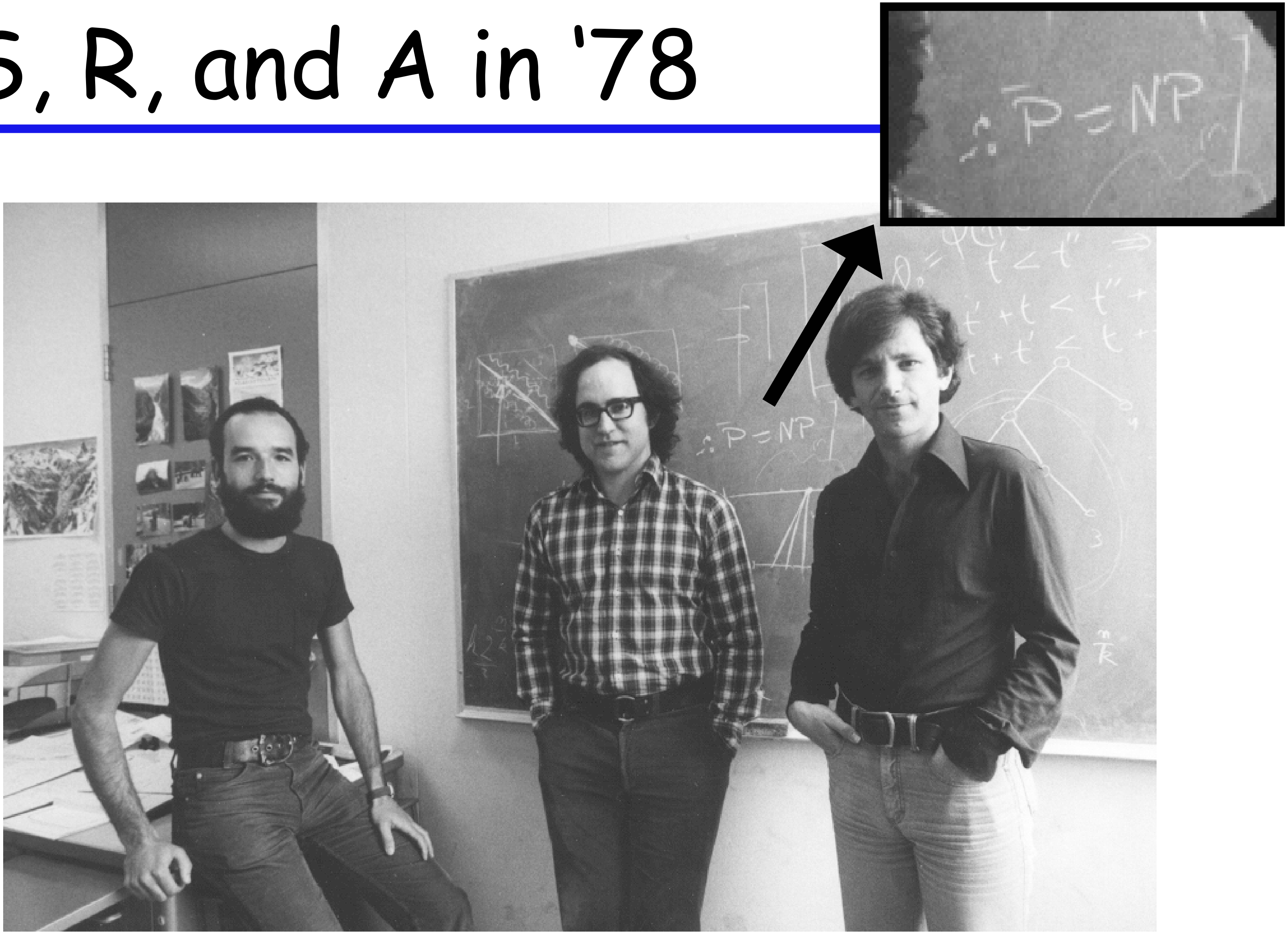
Authors' Address: MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139.

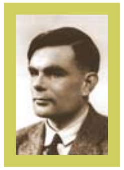
© 1978 ACM 0001-0782/78/0200-0120 $00.75

S, R, and *A* in '78



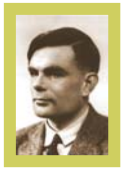
S, R, and *A* in '78





The wonderful Zn*

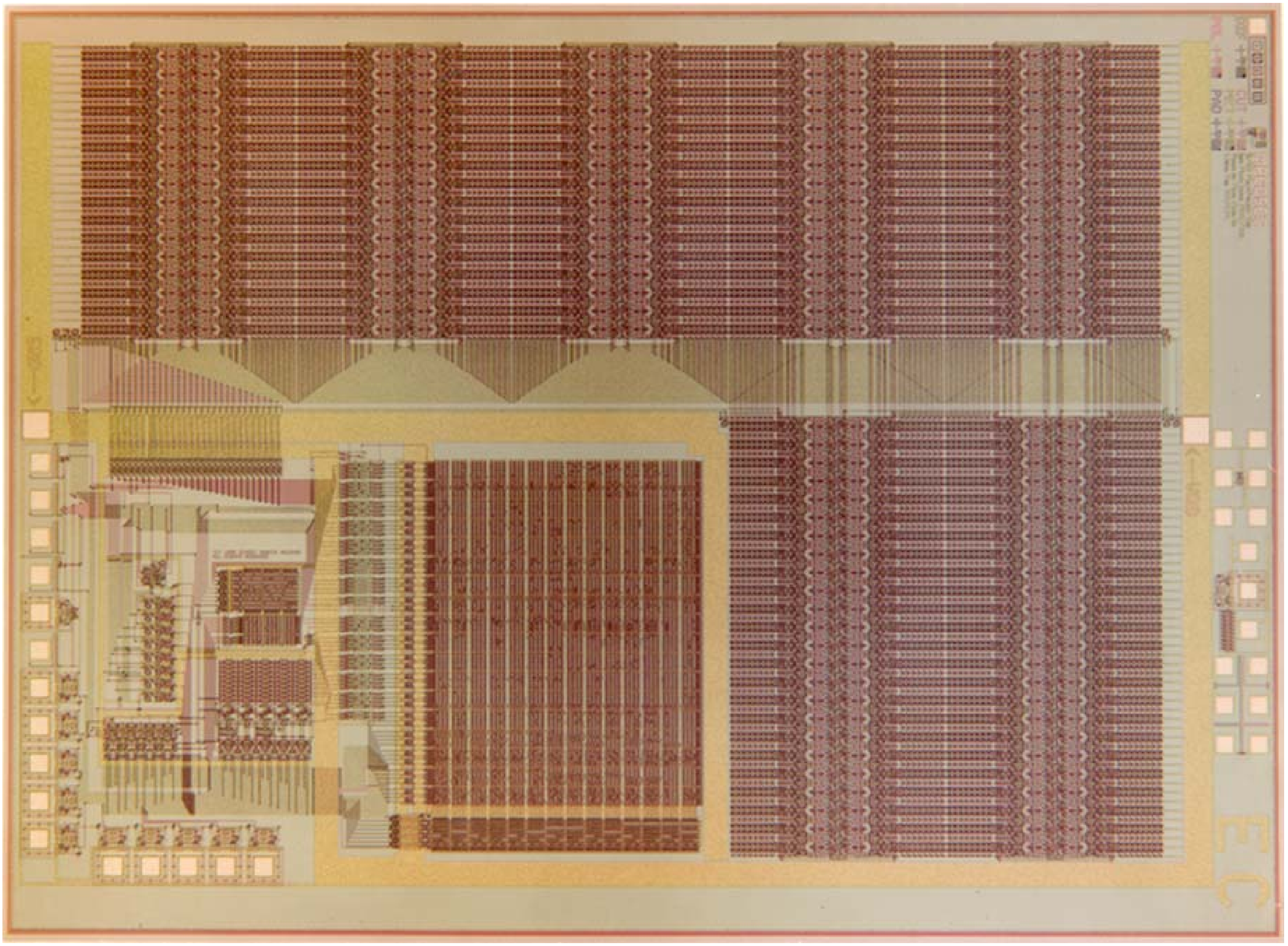
- Zn* = multiplicative group modulo n = pq
- Factoring makes it hard for adversary
  - to compute size of group
  - to compute discrete logs
- Taking e-th roots modulo n is hard ("RSA Assumption")
- Taking e-th roots is hard, where the adversary can pick e>1.
  ("Strong RSA Assumption")

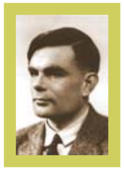


Moore's Law matters.

- Time to do RSA decryption on a 1 MIPS VAX was around 30 seconds (VERY SLOW…)
- IBM PC debuts in 1981
- Still, we worked on efficient special-purpose implementation (e.g. special circuit board, and then the "RSA chip", which did RSA in 0.4 seconds) to prove practicality of RSA.
- Moore's Law to the rescue---software now runs 2000x faster…
- Now software and the Web rule…

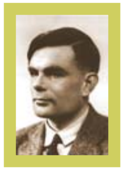
Photo of RSA chip





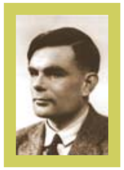
Do cryptography in public.

- Confidence in cryptographic schemes derives from intensive *public review.*
- *Public standards* (e.g. PKCS series)
- Vigorous public research effort results in many new cryptographic proposals, definitions, and attacks



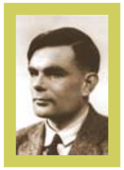
Other PKC proposals

- 1978: Merkle/Hellman (knapsack)
- 1979: Rabin/Williams (factoring)
- 1984: Goldwasser/Micali (QR)
- 1985: El Gamal (DLP)
- 1985: Miller/Koblitz (elliptic curves)
- 1998: Cramer/Shoup
- … many others, too



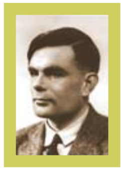
$100 RSA Challenge Met '94

- RSA-129 was factored in 1994, using thousands of computers on Internet. "The magic words are squeamish ossifrage."
- Cheapest purchase of computing time ever!
- Gives credibility to difficulty of factoring, and helps establish key sizes needed for security.



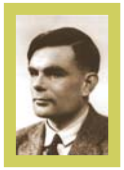
Factoring milestones

- '84: 69D (D = "digits")
  (Sandia; Time magazine)
- '91: 100D
  (Quadratic sieve)
- '94: 129D ($100 challenge number)
  (Distributed QS)
- '99: 155D
  (512-bits; Number field sieve)
- '01: 15 = 3 * 5
  (4 bits; IBM quantum computer!)



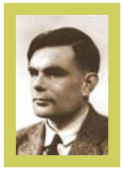
Other attacks on RSA

- Cycling attacks (?)
- Attacks based on “weak keys” (?)
- Attacks based on lack of randomization or improper “padding”
  (use e.g. Bellare/Rogaway’s OAEP ’94)
- Timing analysis, power analysis, fault attacks, …
- See Boneh’s “Twenty Years of Attacks on the RSA Cryptosystem”.



Crypto theory matters

- probabilistic encryption,
- chosen-ciphertext attacks
- GMR digital signatures,
- zero-knowledge protocols,
- concrete complexity of cryptographic reductions; practice-oriented provable security
- …



Organizations matter

- ACM
  - e.g. CACM published RSA paper
- IACR (David Chaum)
  - sponsors CRYPTO conferences
- RSA (Jim Bidzos)
  - sponsors RSA conferences
  - leader in many policy debates
  - helped to set crypto standards

(The End)