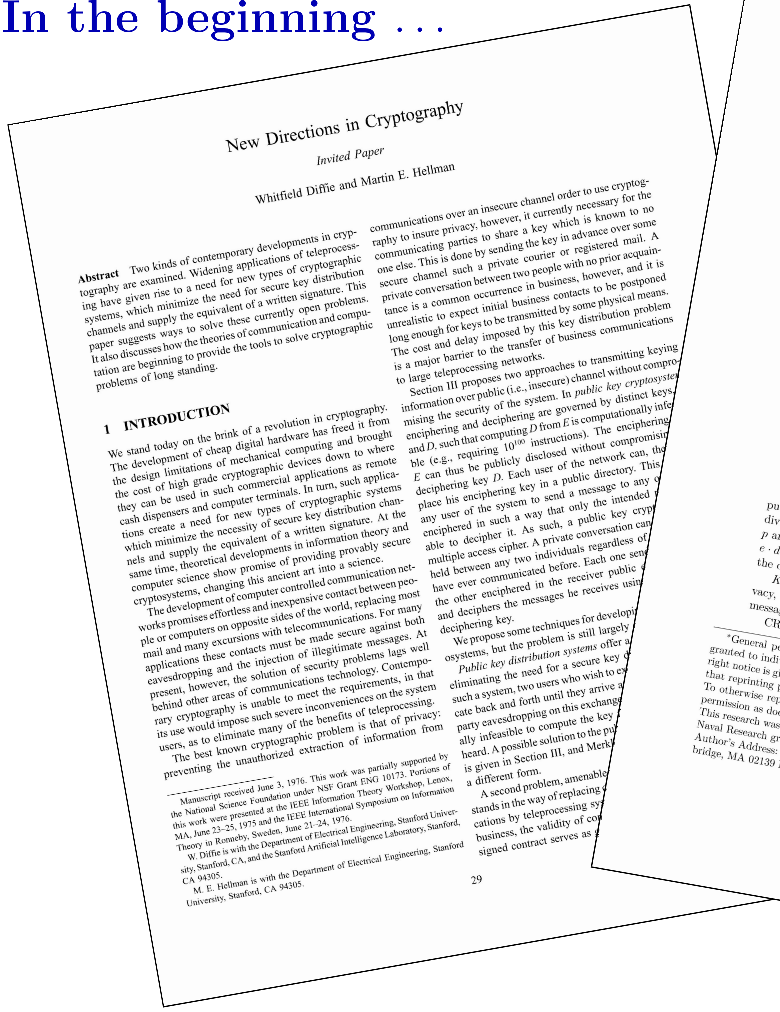
RSA Cryptography in the Textbook and in the Field

Gregory Quenell

In the beginning . . .



New Directions in Cryptography

Invited Paper

Whitfield Diffie and Martin E. Hellman

Abstract Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

1 INTRODUCTION

We stand today on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The development of computer controlled communication networks promises effortless and inexpensive contact between people or computers on opposite sides of the world, replacing most mail and many excursions with telecommunications. For many applications these contacts must be made secure against both eavesdropping and the injection of illegitimate messages. At present, however, the solution of security problems lags well behind other areas of communications technology. Contemporary cryptography is unable to meet the requirements, in that its use would impose such severe inconveniences on the system users, as to eliminate many of the benefits of teleprocessing.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel order to use cryptography to insure privacy, however, it currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such a private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channel without compromising the security of the system. In *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D, such that computing D from E is computationally infe[…] ble (e.g., requiring 10^{100} instructions). The enciphering [...] E can thus be publicly disclosed without compromisin[g …] deciphering key D. Each user of the network can, the[n …] place his enciphering key in a public directory. This [...] any user of the system to send a message to any o[…] enciphered in such a way that only the intended [...] able to decipher it. As such, a public key cryp[…] multiple access cipher. A private conversation can [...] held between any two individuals regardless of [...] have ever communicated before. Each one sen[…] the other enciphered in the receiver public [...] and deciphers the messages he receives usin[g …] deciphering key.

We propose some techniques for developi[…] osystems, but the problem is still largely [...]

Public key distribution systems offer a [...] eliminating the need for a secure key d[…] such a system, two users who wish to ex[…] cate back and forth until they arrive a[…] party eavesdropping on this exchange [...] ally infeasible to compute the key [...] heard. A possible solution to the pu[…] is given in Section III, and Merkl[…] a different form.

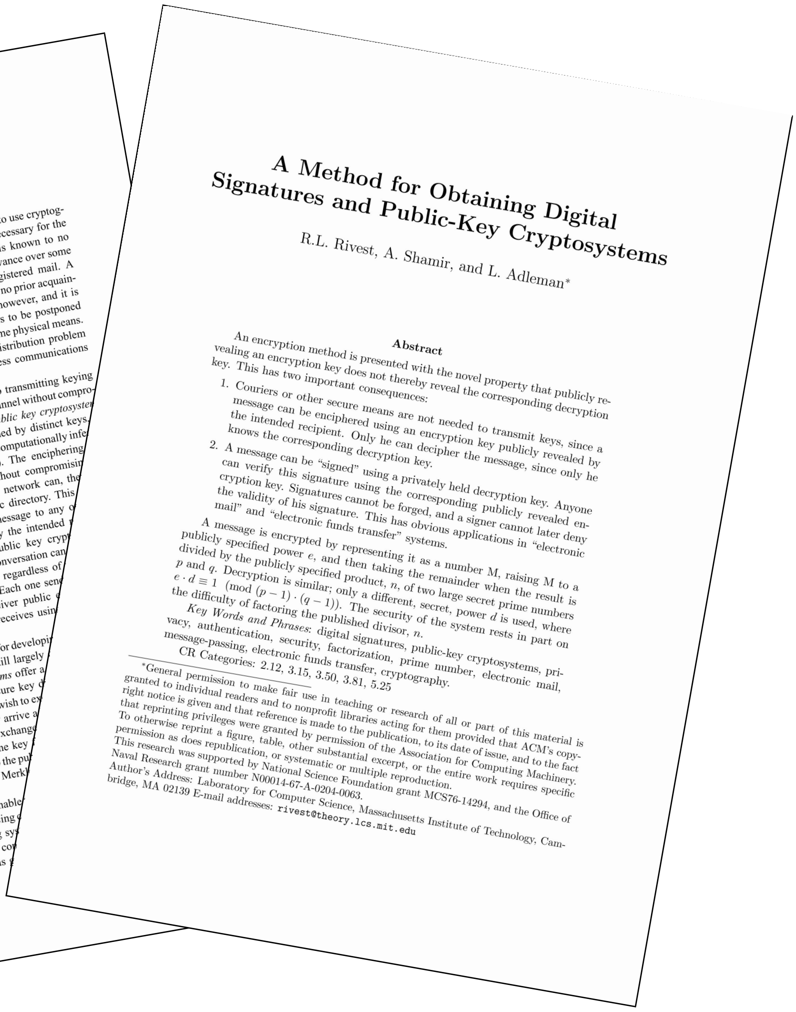
A second problem, amenable [...] stands in the way of replacing [...] cations by teleprocessing sy[…] business, the validity of co[…] signed contract serves as [...]

Manuscript received June 3, 1976. This work was partially supported by the National Science Foundation under NSF Grant ENG 10173. Portions of this work were presented at the IEEE Information Theory Workshop, Lenox, MA, June 23–25, 1975 and the IEEE International Symposium on Information Theory in Ronneby, Sweden, June 21–24, 1976.

W. Diffie is with the Department of Electrical Engineering, Stanford University, Stanford, CA, and the Stanford Artificial Intelligence Laboratory, Stanford, CA 94305.

M. E. Hellman is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

29



A Method for Obtaining Digital Signatures and Public-Key Cryptosystems

R.L. Rivest, A. Shamir, and L. Adleman*

Abstract

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 $e \cdot d \equiv 1 \pmod{(p-1)\cdot(q-1)}$. 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

*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.

Author's Address: Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 E-mail addresses: rivest@theory.lcs.mit.edu

In the beginning . . .

Diffie and Hellman 1976: A "one-way" function can be used to pass secret information over an insecure channel.

$$x \mapsto g^x (\bmod\ p) \text{ is easy; finding discrete logarithms is hard.}$$

Rivest, Shamir, and Adleman 1977: A "trap-door" function can be used to implement public-key cryptography.

$$x \mapsto x^e (\bmod\ N) \text{ is easy; the inverse mapping is hard}$$
$$\textit{unless} \text{ you know } \varphi(N).$$

Textbook RSA

- Choose two large primes p and q. Set $N = pq$.
- $\mathcal{M}$, the message space and $\mathcal{C}$, the ciphertext space, are both $\mathbb{Z}_N^*$.
- Choose an encryption exponent e that is relatively prime to $\varphi(N) = (p-1)(q-1)$.
- Use the Euclidean algorithm to find $d \equiv e^{-1} \pmod{\varphi(N)}$.

Encryption:

For $m \in \mathcal{M}$, define $c = \mathrm{Enc}(m) = m^e \bmod N$

Decryption:

For $c \in \mathcal{C}$, define $\mathrm{Dec}(c) = c^d \bmod N$.

Then it is easy to check that

$$\mathrm{Dec}(\mathrm{Enc}(m)) \equiv (m^e)^d \equiv m^{ed} \equiv m^{k\varphi(N)+1} \equiv m \pmod{N}.$$

Textbook RSA: Public-key encryption

- **Bob** selects p, q, and e. He computes N and d.
- **Bob** makes N and e public. This is the encryption key.

- **Alice** has a message m. She computes $c = m^e \bmod N$ and sends c to Bob.
- **Bob** knows the value of d, so he can compute $c^d \bmod N$, and thus recover the value of m.

- **Eve**, who traditionally listens in on all conversations between Bob and Alice, knows the values of

$$N \quad \text{and} \quad e,$$

and she sees the ciphertext c, (which is equal to m^e). Can she find m?

Textbook RSA: Textbook security

Eve can recover m from $c = m^e$ . . .

if . . . she can compute $d = e^{-1}$, which she can do

if . . . she knows the value of $\varphi(N)$, which she can find

if . . . she can factor N.

Textbook RSA: Textbook security

Eve can recover m from $c = m^e$ ...

if ... she can compute $d = e^{-1}$, which she can do

if ... she knows the value of $\varphi(N)$, which she can find

if ... she can factor N.

So security is related to the difficulty of factoring N ...

Textbook RSA: Textbook security

Eve can recover m from $c = m^e$ ...

if ($\Leftarrow$) she can compute $d = e^{-1}$, which she can do
if ($\Leftarrow$) she knows the value of $\varphi(N)$, which she can find
if ($\Leftarrow$) she can factor N.

So security is related to the difficulty of factoring N ...

... *but the arrows go the wrong way.*

If factoring N is easy, then Eve can easily break RSA.

If factoring N is *not* easy, then ??

This is what we believe.

Textbook RSA: Textbook security

Eve can recover m from $c = m^e$ ...

if $(\Leftarrow)$ she can compute $d = e^{-1}$, which she can do

if $(\Leftarrow)$ she knows the value of $\varphi(N)$, which she can find

$\Leftrightarrow$ she can factor N.

▷ Reversing arrow #3: If Eve knows N and $\varphi(N)$, can she factor N?

Yes: $N = pq$ and $N + 1 - \varphi(N) = p + q$.

Textbook RSA: Textbook security

Eve can recover m from $c = m^e$ ...

if $(\Leftarrow)$ she can compute $d = e^{-1}$, which she can do

$\Leftrightarrow$ she knows the value of $\varphi(N)$, which she can find

$\Leftrightarrow$ she can factor N.

- $\triangleright$ Reversing arrow #3: If Eve knows N and $\varphi(N)$, can she factor N?

  Yes: $N = pq$ and $N + 1 - \varphi(N) = p + q$.

- $\triangleright$ Reversing arrow #2: If Eve knows N and e and can find a number d such that $x^{ed} \equiv x \pmod{N}$ for all x, can she find $\varphi(N)$?

  Yes, though this is less obvious.

Textbook RSA: Textbook security

Eve can recover m from $c = m^e$ ...

if ($\Leftarrow$) she can compute $d = e^{-1}$, which she can do
$\Leftrightarrow$ she knows the value of $\varphi(N)$, which she can find
$\Leftrightarrow$ she can factor N.

- Reversing arrow #3: If Eve knows N and $\varphi(N)$, can she factor N?

  Yes: $N = pq$ and $N + 1 - \varphi(N) = p + q$.

- Reversing arrow #2: If Eve knows N and e and can find a number d such that $x^{ed} \equiv x \pmod{N}$ for all x, can she find $\varphi(N)$?

  Yes, though this is less obvious.

- Reversing arrow #1: If Eve knows how to find e^{th} roots in $\mathbb{Z}_N^*$, can she find the inverse of e modulo $\varphi(N)$?

  Unknown.

Textbook RSA: Textbook security

Result: For RSA security, we have to believe that

1. Factoring is hard *and*

2. Eve has no efficient way to extract e^{th} roots in $\mathbb{Z}_N^*$.

Textbook RSA: Textbook security

Result: For RSA security, we have to believe that

1. Factoring is hard *and*

2. Eve has no efficient way to extract e^{th} roots in $\mathbb{Z}_N^*$.

This is the RSA Assumption.

Cryptographic Security Definitions

These depend on the attacker's

Goals: Does she want to . . .

- read m?
- alter m?
- forge a new m'?
- gain partial information about m?

Capabilities: Can she . . .

- see just the ciphertext?
- intercept and alter the ciphertext?
- use the encryption machinery?
- use the decryption machinery (just temporarily)?

Cryptographic Security Definitions

These depend on the attacker's

Goals: Does she want to ...

- read m?
- alter m?
- forge a new m'?
- gain partial information about m?

Capabilities: Can she ...

- see just the ciphertext?
- intercept and alter the ciphertext?
- use the encryption machinery?
- use the decryption machinery (just temporarily)?

Most security definitions are presented as games.

- We give the attacker a goal and a set of powers.
- If the attacker can reach the goal in a reasonable amount of time, she wins, and the system is insecure.
- If *no* attacker can win the game, the system is secure.

Cryptographic Security Definitions

An appropriate security definition for RSA is

Semantic Security under a Chosen-Plaintext Attack

In a chosen-plaintext attack (CPA), the attacker gets free use of the encryption machinery in the first part of the game.

The attacker wins a semantic security (SS) game if she can learn anything about an encrypted message, apart from its length.

Cryptographic Security Definitions

An appropriate security definition for RSA is

Semantic Security under a Chosen-Plaintext Attack

The SS-CPA Game

0. The defender sets up the encryption machinery. (In this case, he chooses N and e).

1. The attacker submits a message m and receives its encryption c. She may repeat this as many times as she likes.

2. The attacker submits two messages, m_0 and m_1, of equal length. The defender flips a 0/1 coin to obtain a random bit b. He returns the encryption of m_b to the attacker.

3. The attacker tries to guess whether she has been given the encryption of m_0 or m_1. If she can guess correctly with probability significantly greater than $\frac{1}{2}$, she wins.

Cryptographic Security Definitions

Is textbook RSA semantically secure?

Cryptographic Security Definitions

Is textbook RSA semantically secure? **No!**

The attacker can win every time:

1. The attacker submits a single message m_0 and receives its encryption c_0.

2. The attacker submits m_0 and some $m_1 \neq m_0$ of the same length. The defender returns an encryption c.

3. If $c = c_0$, the attacker says $b = 0$; otherwise, she says $b = 1$. She is correct with probability 1.

Cryptographic Security Definitions

Is textbook RSA semantically secure? **No!**

The attacker can win every time:

1. The attacker submits a single message m_0 and receives its encryption c_0.
2. The attacker submits m_0 and some $m_1 \neq m_0$ of the same length. The defender returns an encryption c.
3. If $c = c_0$, the attacker says $b = 0$; otherwise, she says $b = 1$. She is correct with probability 1.

Reasonable Question: How can *any* system be CPA semantically secure?

Answer: As long as $\mathrm{Enc}(\cdot)$ is a function, it can't.

Cryptographic Security Definitions

Is textbook RSA semantically secure? **No!**

The attacker can win every time:

1. The attacker submits a single message m_0 and receives its encryption c_0.
2. The attacker submits m_0 and some $m_1 \neq m_0$ of the same length. The defender returns an encryption c.
3. If $c = c_0$, the attacker says $b = 0$; otherwise, she says $b = 1$. She is correct with probability 1.

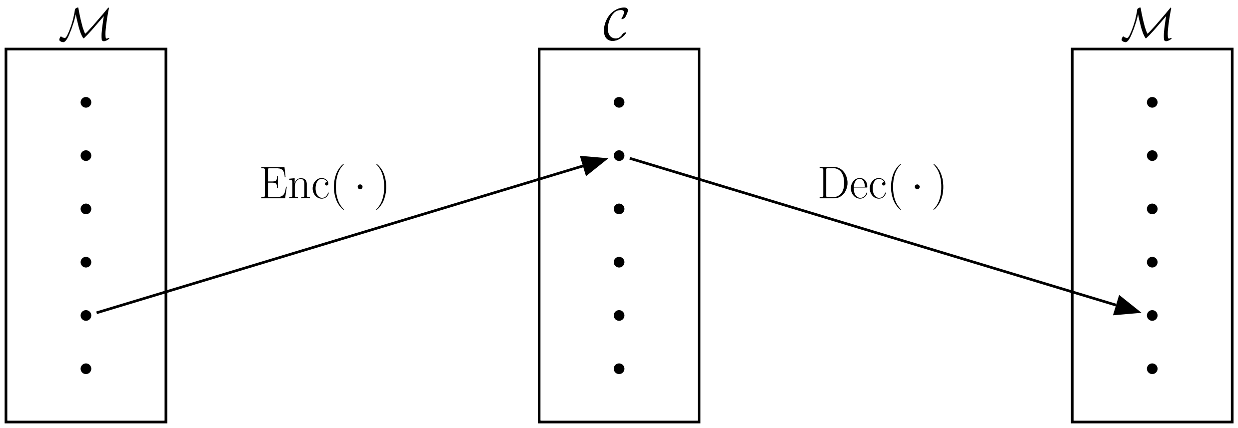
Reasonable Question: How can *any* system be CPA semantically secure?

deterministic

Answer: As long as Enc($\cdot$) is a function, it can't.

We need Enc($\cdot$) to be a *randomized function*.

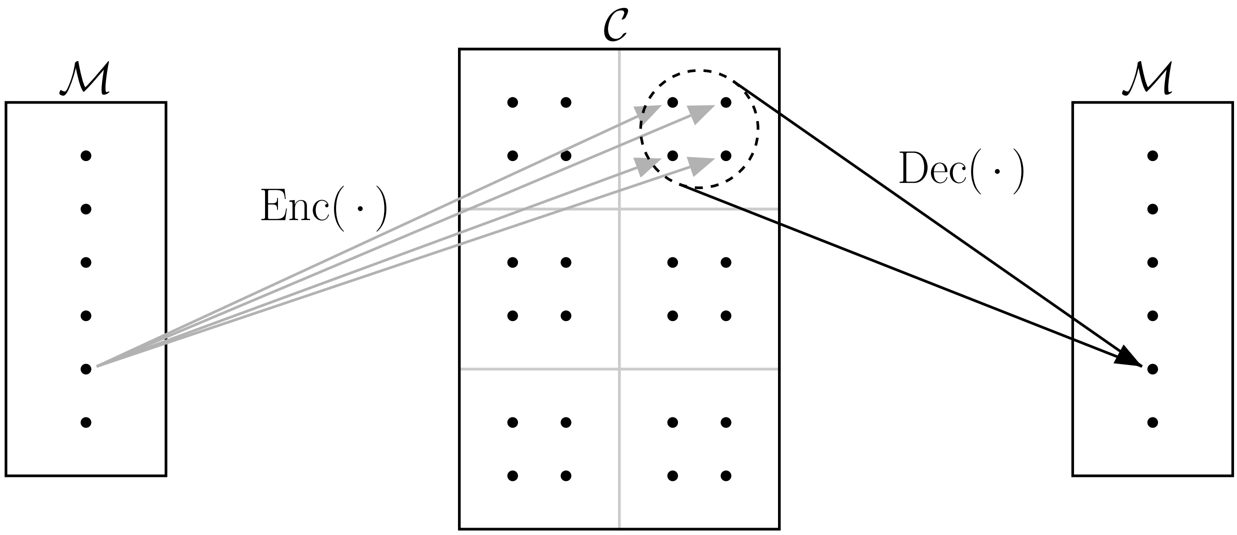
Randomized Encryption – What?



In **deterministic encryption**,

$\mathrm{Enc}(\,\cdot\,)$ is a one-to-one function, and $\mathrm{Dec}(\,\cdot\,)$ is its inverse.

Randomized Encryption – What?



In **randomized encryption**,

Enc(m) chooses a random point in $\text{Dec}^{-1}(m)$.

Consequences of this:

Enc(·) isn't really a function.

Dec(·) is a many-to-one function.

The ciphertext space $\mathcal{C}$ is bigger than the message space $\mathcal{M}$.

Randomized Encryption – Why? (1)

Semantic Security in World War II

May 20, 1942: Cryptanalysts at Pearl Harbor partially decrypt a radio transmission from Admiral Yamamoto. It appears to be an order to attack location AF.

Prior intercepts suggest that AF is Midway Island, but Admiral Nimitz is unwilling to send defense forces to Midway without more evidence.

The Pearl Harbor cryptanalysts instruct the Allied garrison at Midway to broadcast, in the clear, a message saying that the Midway fresh-water distillation plant has broken down.

Two days later, in the intercepts of Japanese radio traffic, Allied Intelligence finds the message "Location AF is short of water."

Admiral Nimitz is satisfied, and orders defense forces to Midway.

Randomized Encryption – Why? (2)

Deterministic Encryption Leaks Information

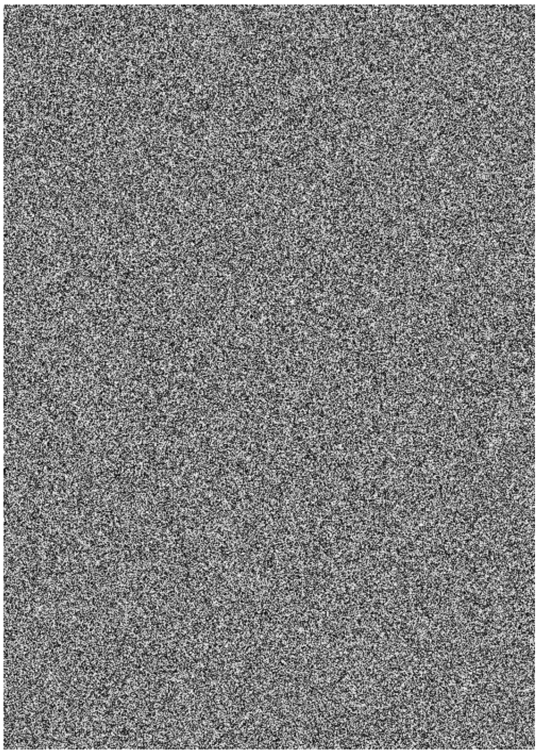
Original message

Deterministic encryption

Randomized Encryption – Why? (2)

Deterministic Encryption Leaks Information

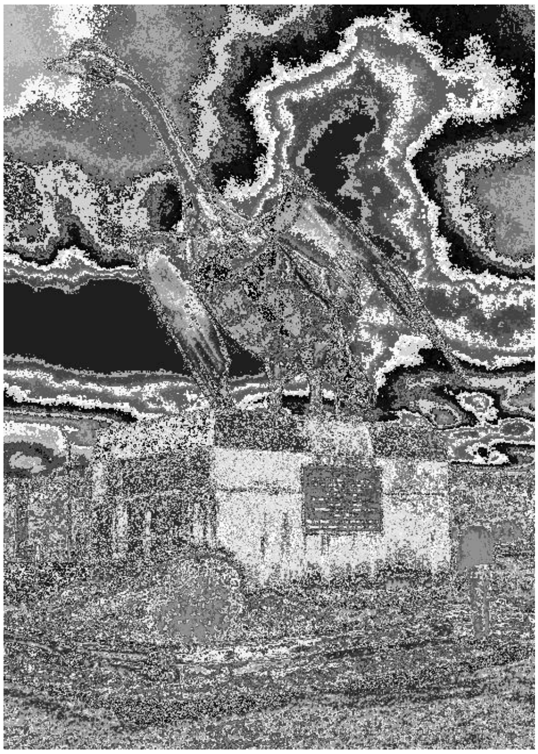
Original message



Randomized encryption

Randomized Encryption – Why? (2)

Deterministic Encryption Leaks Information



Where are we?

Randomized Encryption – Why? (3)

Deterministic Encryption Doesn't Work with Small Message Spaces

Bob sends Alice his Social-Security number m using deterministic RSA:

$$c = m^e \bmod N$$

Eve intercepts c. She knows e and N, so she can just compute $x^e \bmod N$ for all 10^9 values $x = \boxed{d_1}\boxed{d_2}\boxed{d_3}-\boxed{d_4}\boxed{d_5}-\boxed{d_6}\boxed{d_7}\boxed{d_8}\boxed{d_9}$.

When x^e matches c, Eve has found Bob's secret.

Even better (or worse), if Bob acquired his SSN before 2011 and Eve knows where he lived at the time, her search space is reduced to only 10^6 or 10^7 numbers.

Randomized encryption – How?

A document called **ISO/IEC 18033-2** contains a standard protocol for using RSA encryption in a semantically-secure way. The protocol requires:

A symmetric-key encryption scheme

For each k in a keyspace $\mathcal{K}$, we have

$$\mathrm{Enc}_k : \mathcal{M} \to \mathcal{C}; \qquad \mathrm{Dec}_k = \mathrm{Enc}_k^{-1}$$

Each function Enc_k should be indistinguishable from a random (invertible) function $\mathcal{M} \to \mathcal{C}$.

A hash function $H : \mathbb{Z}_N \to \mathcal{K}$

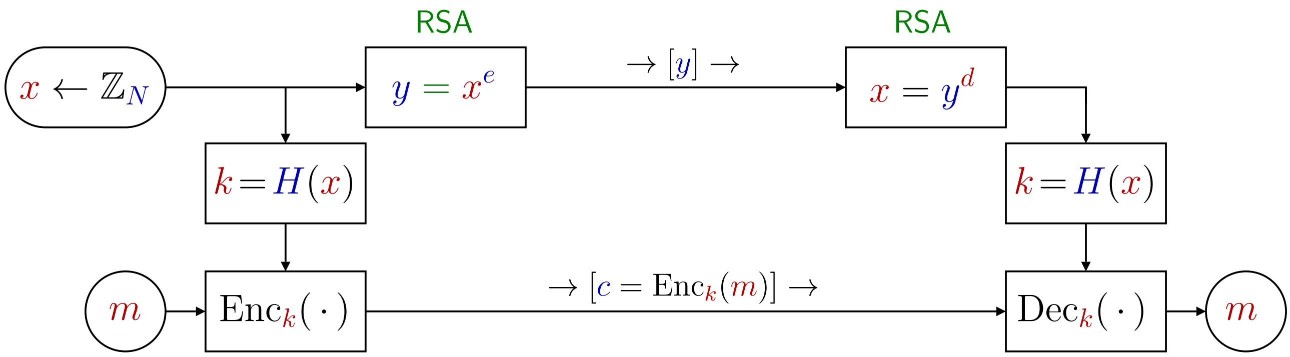
Anyone can query H as an oracle, but no one knows its inner workings. For any set $S \subset \mathbb{Z}_N$, knowing the values of $H(x)$ for all $x \in S$ should give no information about the value of $H(y)$ for any $y \notin S$.

RSA KEM/DEM (ISO/IEC)

Encryption: Bob generates a random "pre-key" $x \in \mathbb{Z}_N$. He feeds x to a (public) hash function to produce a symmetric key k, which he uses to encrypt the message m.

He sends Alice the (symmetric-key) encryption c of m, and the RSA (public-key) encryption y of x.

Decryption: Alice decrypts y to get the "pre-key" x. She then uses the public hash function to recover the symmetric key k, and decrypts c to recover m.

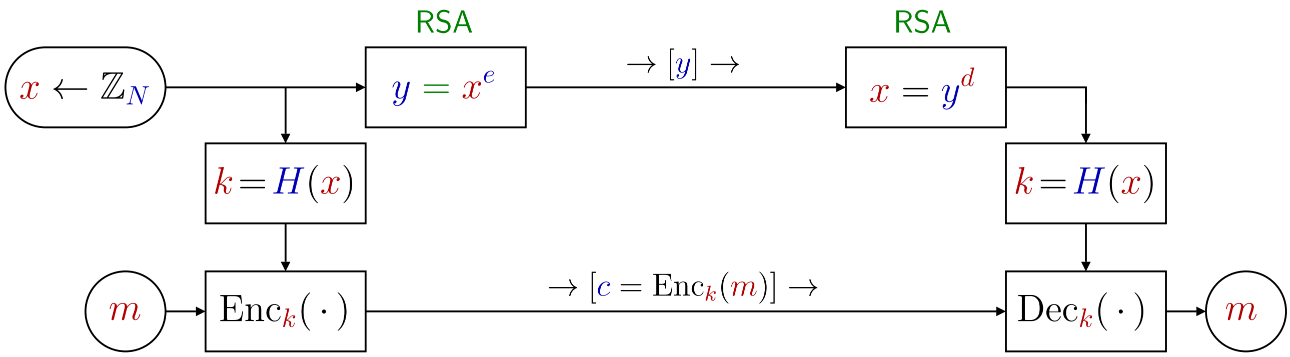


RSA KEM/DEM (ISO/IEC)

Eavesdropping: The message is protected by k. If H is a good hash function, then Eve cannot find k without first knowing x. By the RSA assumption, Eve cannot recover x from y.

This system is randomized, so it can be semantically secure.

Furthermore, since the message is protected by k and H, a partial break of the RSA branch will not give Eve any information about m.



RSA ISO/IEC KEM/DEM in SSL/TLS (OMG. . .)

Public-key encryption is much slower than private-key encryption, so it's typically used only at the beginning of a session to enchange the keys that will be used for encrypting the real stuff.

Client		Server
Hello? Server?	→	
	←	Hi! Here's my public encryption key: pk, and here's a note from my CA.
[If the CA says OK, then . . .]		
Here's a random pmk. I'll send it to you using the RSA KEM/DEM scheme from the previous slide.	→	
		Got it. Now we both know pmk.
[Calculates AES keys from pmk]		[Calculates AES keys from pmk]

RSA ISO/IEC KEM/DEM in SSL/TLS (OMG. . .)

Public-key encryption is much slower than private-key encryption, so it's typically used only at the beginning of a session to enchange the keys that will be used for encrypting the real stuff.

Client	Server
Hello? Server? →	
	Hi! Here's my public encryption key: pk, and here's a note from my CA.
[If the CA says OK, then . . .] ←	
Here's a random pmk. I'll send it to you using the RSA KEM/DEM scheme from the previous slide. →	
	Got it. Now we both know pmk.
[Calculates AES keys from pmk]	[Calculates AES keys from pmk]
AES high-speed machinery ⟷	AES high-speed machinery

References

Dan Boneh and Victor Shoup, *A Graduate Course in Applied Cryptography*, prepublication version 0.4, 2017.

Whitfield Diffie and Martin E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, IT-22, November 1976.

David Kahn, *The Codebreakers*, Scribner, 1967.

Jonathan Katz and Yehuda Lindell, *Introduction to Modern Cryptography*, second edition, CRC Press, 2015.

Neal Koblitz, *A Course in Number Theory and Cryptography*, second edition, Springer-Verlag, 2006.

Neal Stephenson, *REAMDE*, William Morrow, 2011.

Douglas R. Stinson, *Cryptography: Theory and Practice*, second edition, CRC Press, 2002.