15-251
Great Theoretical Ideas in Computer Science

Cryptography

March 31, 2015

What is cryptography?

kryptós, "hidden, secret"

graphein, "writing"

Study of secure communication in the presence of third parties

What is cryptography?

Adversary
Eavesdropper

"I will cut your throat"

"I will cut your throat"

What is cryptography?

“loru23n8uladjkfb!#@”

“I will cut your throat”

encryption

“loru23n8uladjkfb!#@”

“loru23n8uladjkfb!#@”

decryption

“I will cut your throat”

What is cryptography?

Study of protocols that avoid the bad affects of adversaries.

- Can we have secure online voting schemes?

- Digital signatures: demonstrating authenticity of digital doc.

- Can we do computation on encrypted data?

- Can I convince you that I have proved P=NP without giving you any information about the proof?

Reasons to like cryptography

Can do pretty cool and unexpected things.

Has many important applications.

Is fundamentally related to computational complexity.

In fact, computational complexity revolutionized crypto.

Application of computationally hard problems.

There is good math (e.g. number theory)

The plan

First, we have to review modular arithmetic.

Then we'll talk about private/secret key cryptography.

Finally, we'll talk about public key cryptography.

Review of Modular Arithmetic

Review of modular arithmetic

One way to view it:

The universe is $\mathbb{Z}$

For $a, b \in \mathbb{Z}$, $a \equiv_m b$ if m divides $a - b$

+ is the addition in $\mathbb{Z}$

$5 + 8 \equiv_6 1 + 6$

$\cdot$ is the multiplication in $\mathbb{Z}$

$3 \cdot 3 \equiv_6 5 \cdot 3$

Review of modular arithmetic

Another way to view it:

The universe is $\mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$

Well defined + within $\mathbb{Z}_m$

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Every element in $\mathbb{Z}$ corresponds to some element in $\mathbb{Z}_4$.

$5 \equiv_4 9$

5 and 9 correspond to the same element in $\mathbb{Z}_4$.

$7 + 2 \equiv_4 1 + 12$

$3 + 2 = 1 + 0$ in $\mathbb{Z}_4$

every row and column is a permutation of the elements

Review of modular arithmetic

Second way to view it:

The universe is $\mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$

Well defined + within $\mathbb{Z}_m$

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

We can also do subtraction:

$1 - 3$ is really $1 + (-3)$

What is -3 ? additive inverse of 3

It is the element x such that

$3 + x = 0$ 0 is additive identity

$$1 - 3 = 1 + (-3) = 1 + 1 = 2$$

every element has a unique inverse

Review of modular arithmetic

x	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Can also do multiplication

$$2 \cdot 3 = 2$$

How about division?

$$1 \div 2 \; = 1 \cdot \boxed{\frac{1}{2}}$$

multiplicative inverse of 2

The x such that $x \cdot 2 = 1$ 1 is multiplicative identity

No such x exists!

Not all elements have a multiplicative inverse.

$\mathbb{Z}_4$ is not a good universe with respect to multiplication.

Which elements have a multiplicative inverse in $\mathbb{Z}_m$?

Review of modular arithmetic

$x \in \mathbb{Z}_m$ has a multiplicative inverse iff $\mathrm{GCD}(x, m) = 1.$

x	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

x	1	3
1	1	3
3	3	1

Define $\mathbb{Z}_m^* = \{x \in \mathbb{Z}_m : \mathrm{GCD}(x, m) = 1.\}$

$\mathbb{Z}_m^*$ behaves nicely with respect to multiplication.

Review of modular arithmetic

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$\mathbb{Z}_4$

behaves nicely with respect to addition

x	1	3	5	7
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\mathbb{Z}_8^*$

behaves nicely with respect to multiplication

Review of modular arithmetic

When m is a prime, easy to see what $\mathbb{Z}_m^*$ is.

x	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Definition: $\varphi(m) = |\mathbb{Z}_m^*|$

So for p prime: $\varphi(p) = p - 1$

For p and q distinct primes: $\varphi(pq) = (p-1)(q-1)$

Review of modular arithmetic

Euler's Theorem:

For any $a \in \mathbb{Z}_m^*$: $a^{\varphi(m)} = 1$

Equivalently: for any a and m such that $\mathrm{GCD}(a, m) = 1$,

$$a^{\varphi(m)} \equiv_m 1$$

When m is a prime, this is known as:

Fermat's Little Theorem:

Let p be prime. For any $a \in \mathbb{Z}_p^*$: $a^{p-1} = 1$

Equivalently: for any a that is not divisible by p

$$a^{p-1} \equiv_p 1$$

Exponentiation in the modular world

x	1	3	5	7
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\mathbb{Z}_8^*$

1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1
3	3^2	3^3	3^4	3^5	3^6	3^7	3^8
3	1	3	1	3	1	3	1
5	5^2	5^3	5^4	5^5	5^6	5^7	5^8
5	1	5	1	5	1	5	1
7	7^2	7^3	7^4	7^5	7^6	7^7	7^8
7	1	7	1	7	1	7	1

Exponentiation in the modular world

x	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

$\mathbb{Z}_5^*$

1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1
2	2^2	2^3	2^4	2^5	2^6	2^7	2^8
2	4	3	1	2	4	3	1
3	3^2	3^3	3^4	3^5	3^6	3^7	3^8
3	4	2	1	3	4	2	1
4	4^2	4^3	4^4	4^5	4^6	4^7	4^8
4	1	4	1	4	1	4	1

We know $a^{\varphi(m)} = 1.$

So can reduce the exponent mod $\varphi(m)$. $a^n = a^{n \bmod \varphi(m)}$

Exponentiation in the modular world

We know $a^{\varphi(m)} = 1.$

So can reduce the exponent mod $\varphi(m)$. $a^n = a^{n \bmod \varphi(m)}$

(Note: this is in $\mathbb{Z}_m^*$. Doesn't always work for $a \in \mathbb{Z}_m - \mathbb{Z}_m^*$)

When exponentiating elements $a \in \mathbb{Z}_m^*$

can think of the exponent as living in $\mathbb{Z}_{\varphi(m)}$ (or $\mathbb{Z}^*_{\varphi(m)}$)

Example:

$m = 5 \qquad \varphi(m) = 4$

$4 \in \mathbb{Z}_5^* \qquad 4^{1+3} = 4^0 = 1$

$4^{3 \cdot 3} = 4^1 = 4$

Exponentiation in the modular world

x	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

$\mathbb{Z}_5^*$

1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1

2	2^2	2^3	2^4	2^5	2^6	2^7	2^8
2	4	3	1	2	4	3	1

3	3^2	3^3	3^4	3^5	3^6	3^7	3^8
3	4	2	1	3	4	2	1

4	4^2	4^3	4^4	4^5	4^6	4^7	4^8
4	1	4	1	4	1	4	1

2 and 3 are called generators.

One-way functions in the modular world

In crypto, one is interested in one-way functions:

a function that is easy to compute,
but hard to invert.

e.g. Multiplication

Given two primes p and q easy to compute pq.

Given pq, seems hard to recover p and q.

It is reasonable to try to encode your message using a one-way function.

In the modular world, some simple operations seem to be one-way.

One-way functions in the modular world

Consider the exponentiation function (over $\mathbb{N}$)

$$\exp(b, e) = b^e$$

$$\exp_e(b) = b^e$$

$$\exp_b(e) = b^e$$

Inverse of $\exp_e(b) = b^e$:

Given some number of the form b^e, compute b

i.e., compute the e th root of b^e

Inverse of $\exp_b(e) = b^e$:

Given some number of the form b^e, compute e

i.e., compute the log (base b) of b^e.

One-way functions in the modular world

<u>Example1</u>: RSA function

This will be like $\exp_e(b) = b^e$

So the inverse will be like taking the e'th root.

The universe is $\mathbb{Z}_m^*$, where m is a composite number.

(why composite?)

Fix an exponent $e \in \mathbb{Z}_{\varphi(m)}^*$.

For $b \in \mathbb{Z}_m^*$, $\mathrm{RSA}_{m,e}(b) = b^e \in \mathbb{Z}_m^*$ easy

For $b^e \in \mathbb{Z}_m^*$, $\mathrm{RSA}_{m,e}^{-1}(b^e) = b \in \mathbb{Z}_m^*$ seems hard

(note: you know m but not $\varphi(m)$)
(if you knew $\varphi(m)$, you could do the inverse efficiently)

One-way functions in the modular world

Example2:

This will be like $\exp_b(e) = b^e$

So the inverse will be like taking log base *b*.

The universe is $\mathbb{Z}_p^*$, where p is a prime number.

Fix generator $b \in \mathbb{Z}_p^*$.

For $e \in \mathbb{Z}_{\varphi(p)}$, $\mathrm{EXP}_{p,b}(e) = b^e \in \mathbb{Z}_p^*$ easy

For $b^e \in \mathbb{Z}_p^*$, $\mathrm{EXP}_{p,b}^{-1}(b^e) = e \in \mathbb{Z}_{\varphi(p)}$ seems hard

Private Key Cryptography

Private key cryptography

Parties must agree on a key pair beforehand.

Private key cryptography

there must be a secure way of exchanging the key

Private key cryptography

C

 K_A  M

 K_B

(M, K_A)

↓

Enc

↓

C

Enc should be "one-way".

Try to ensure it using the secrecy of the key.

(C, K_B)

↓

Dec

↓

M

Security

Better to consider worst-case conditions.

Assume the adversary knows everything
except the key(s) and the message:

Completely sees C.

Completely knows the algorithms Enc and Dec.

Could also assume adversary knows some of the
message and the corresponding encoded text.

Caesar shift

<u>Example</u>: shift by 3

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
d e f g h i j k l m n o p q r s t u v w x y z a b c
```

(similarly for capital letters)

"Dear Math, please grow up and solve your own problems."

↓

"Ghdu Pdwk, sohdvh jurz xs dqg vroyh brxu rzq sureohpv."

 : the shift number

Easy to break.

Substitution cipher

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
j k b d e l m c f g n o x y r s v w z a t u p q h i
```

 : permutation of the alphabet

Easy to break by looking at letter frequencies.

Enigma

A much more complex cipher.

One-time pad

M = message   K = key   C = encrypted message

Encryption:

$$
\begin{array}{rl}
M = & 01011010111010100000111 \\
\oplus \quad K = & 11001100010101111000101 \\
\hline
C = & 10010110101111011000010
\end{array}
$$

$C = M \oplus K$ (bit-wise XOR)

For all i: $C[i] = M[i] + K[i] \pmod 2$

if K[i] is 1, change/flip C[i]

if K[i] is 0, don't change C[i]

One-time pad

M = message   K = key   C = encrypted message

Decryption:

$$
\begin{array}{rl}
C = & 10010110101111011000010 \\
\oplus \quad K = & 11001100010101111000101 \\
\hline
M = & 01011010111010100000111
\end{array}
$$

Encryption: $C = M \oplus K$

Decryption: $C \oplus K = M \oplus K \oplus K = M$

(because $K \oplus K = 0$)

One-time pad

$$
\begin{array}{rl}
M = & 01011010111010100000111 \\
\oplus \quad K = & 11001100010101111000101 \\
\hline
C = & 10010110101111011000010
\end{array}
$$

One-time pad is perfectly secure:

For any M, if K is chosen uniformly at random, then C is uniformly at random.

So you learn nothing about M by seeing C.

But you need to share a key that is as long as the message.

Could we reuse the key?

One-time pad

$$
\begin{array}{rl}
M = & 01011010111010100000111 \\
\oplus \quad K = & 11001100010101111000101 \\
\hline
C = & 10010110101111011000010
\end{array}
$$

Could we reuse the key?

One-time only:

Suppose you encrypt two messages with K.

$C_1 = M_1 \oplus K$

$C_2 = M_2 \oplus K$

Then $C_1 \oplus C_2 = M_1 \oplus M_2$

Shannon's Theorem

Is it possible to have a secure system like one-time pad with a smaller key size?

Shannon proved "no".

If K is shorter than M:

- An adversary with unlimited computational power could learn some information about M.

Questions

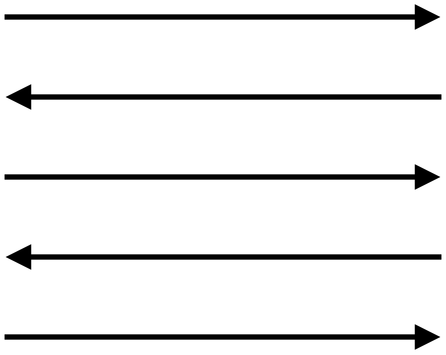
Can we relax the assumption that the adversary is computationally unbounded?

Can we find a way to share a random secret key? (over an insecure channel)

Can we get rid of the secret key sharing part?

Secret Key Sharing

Secret Key Sharing



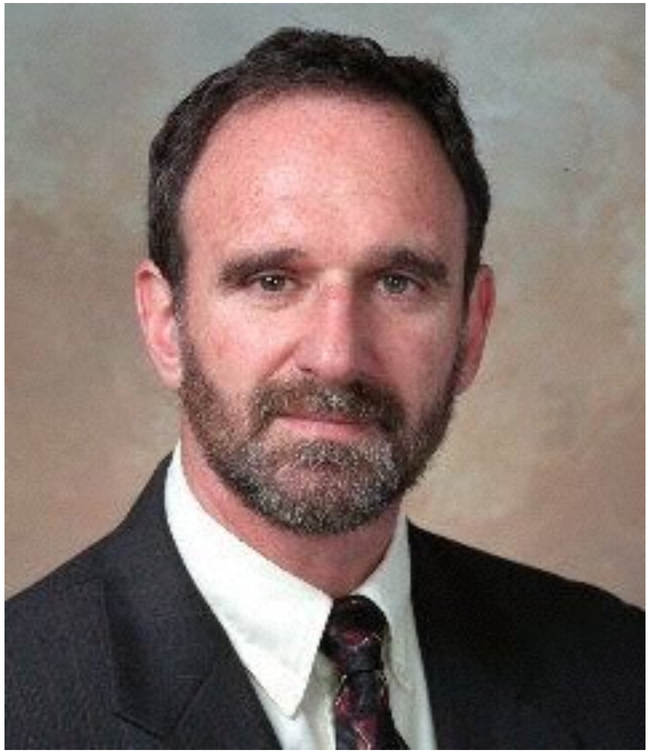
 K

 K

Diffie-Hellman key exchange

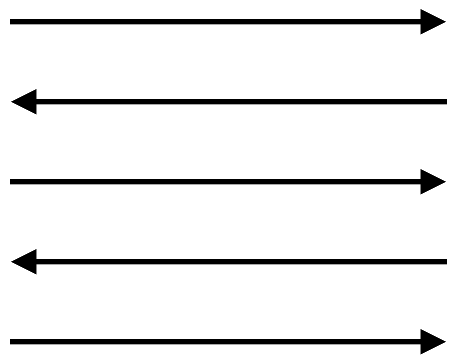
1976

Whitfield Diffie



Martin Hellman

Diffie-Hellman key exchange



 K

 K

Seems reasonable to try encode messages using a one-way function.

Let p be a prime, and let $b \in \mathbb{Z}_p^*$ be a generator.

For $e \in \mathbb{Z}_{\varphi(p)}$, $\quad \mathrm{EXP}_{p,b}(e) = b^e \in \mathbb{Z}_p^*$

Diffie-Hellman key exchange

Pick prime p

Pick generator $b \in \mathbb{Z}_p^*$

Pick random $r_1 \in \mathbb{Z}_{\varphi(p)}$

$$\xrightarrow{p,\, b,\, b^{r_1}}$$

Pick random $r_2 \in \mathbb{Z}_{\varphi(p)}$

$$\xleftarrow{b^{r_2}}$$

Compute

$$(b^{r_2})^{r_1} = b^{r_1 r_2}$$

Compute

$$(b^{r_1})^{r_2} = b^{r_1 r_2}$$

Diffie-Hellman key exchange

Efficient?

Pick prime p

Pick generator $b \in \mathbb{Z}_p^*$

Pick random $r_1 \in \mathbb{Z}_{\varphi(p)}$

$$\xrightarrow{p,\, b,\, b^{r_1}}$$

Pick random $r_2 \in \mathbb{Z}_{\varphi(p)}$

$$\xleftarrow{b^{r_2}}$$

Compute

$$(b^{r_2})^{r_1} = b^{r_1 r_2}$$

Compute

$$(b^{r_1})^{r_2} = b^{r_1 r_2}$$

Secure?

Adversary sees: p, b, b^{r_1}, b^{r_2}

Hopefully s/he can't compute r_1 from b^{r_1} .

(our hope that EXP is one-way)

<u>Good news</u>: No one knows how to invert EXP efficiently

<u>Bad news</u>: Proving this cannot be done is at least as hard as the P vs NP problem.

Diffie-Hellman assumption:

Computing $b^{r_1 r_2}$ from p, b, b^{r_1}, b^{r_2} is hard.

Decisional Diffie-Hellman assumption:

You actually learn no information about $b^{r_1 r_2}$.

Questions

Can we relax the assumption that the adversary is computationally unbounded?

Can we find a way to share a random secret key?

Can we get rid of the secret key sharing part?

Public Key Cryptography

Public Key Cryptography

public

private

Public Key Cryptography

public

private

Can be used to lock.
Can't be used to unlock.

Public key cryptography

C

M

K_{pub}

K_{pri}

(M, K_{pub})

↓

Enc

↓

C

Enc should be "one-way".

Try to ensure it using computational complexity.

(C, K_{pri})

↓

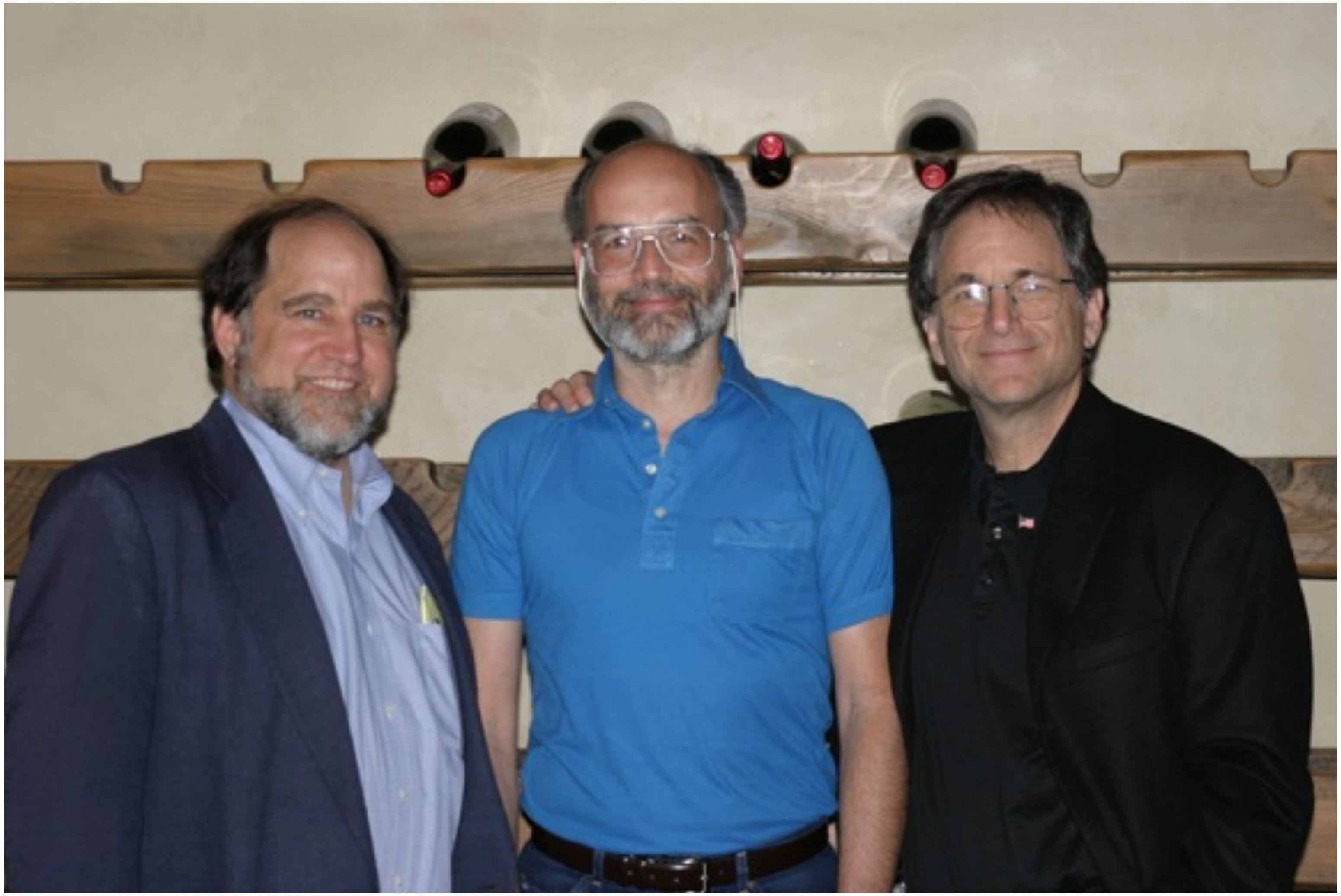
Dec

↓

M

RSA crypto system

1977



Ron Rivest Adi Shamir Leonard Adleman

RSA crypto system

Clifford Cocks

Discovered RSA system 3 years before them.

Remained secret until 1997. (was classified information)

RSA crypto system

Will use the RSA function as our "one-way" function.

Let N be composite. Fix $e \in \mathbb{Z}^*_{\varphi(N)}$.

For $b \in \mathbb{Z}^*_N$, $\mathrm{RSA}_{N,e}(b) = b^e \in \mathbb{Z}^*_N$

(You know N and e, you don't know $\varphi(N)$)

We want b to correspond to the message M.

We want (N, e) to be the public key.

RSA crypto system

Will use the RSA function as our "one-way" function.

Let N be composite. Fix $e \in \mathbb{Z}^*_{\varphi(N)}$.   why composite?

For $b \in \mathbb{Z}^*_N$, $\mathrm{RSA}_{N,e}(b) = b^e \in \mathbb{Z}^*_N$

(You know N and e, you don't know $\varphi(N)$)

The inverse function is taking the e'th root.

We believe it is computationally hard.

Remark: If you knew $\varphi(N)$, you could do this efficiently.

(Can compute the inverse of e in $\mathbb{Z}^*_{\varphi(N)}$. exercise
$(b^e)^{e^{-1}} = b^{ee^{-1}} = b$)

This is great: Can be our advantage over an adversary.

RSA crypto system

Let N be composite. Fix $e \in \mathbb{Z}^*_{\varphi(N)}$.

For $b \in \mathbb{Z}^*_N$, $\mathrm{RSA}_{N,e}(b) = b^e \in \mathbb{Z}^*_N$

We want b to correspond to the message M.

We want (N, e) to be the public key.

Pick two random distinct primes p, q.

Let $N = p \cdot q$.

Compute $\varphi(N) = (p-1)(q-1)$.

Pick some $e \in \mathbb{Z}^*_{\varphi(N)}$.

Publish (N, e) as the public key.

RSA crypto system

Pick two random distinct primes p, q.

Let $N = p \cdot q$.

Compute $\varphi(N) = (p-1)(q-1)$.

Pick some $e \in \mathbb{Z}^*_{\varphi(N)}$.

Publish (N, e) as the public key. 

Let $b = M$. (what if $M \notin \mathbb{Z}^*_N$?)

Compute $C = \mathrm{RSA}_{N,e}(b) = b^e \in \mathbb{Z}^*_N$

Send C. 

RSA crypto system

Let $b = M.$

Compute $C = \mathrm{RSA}_{N,e}(b) = b^e \in \mathbb{Z}_N^*$

Send C. 

Get $C \; = b^e \in \mathbb{Z}_N^*$

Needs to recover b .

Can compute e^{-1} in $\mathbb{Z}_{\varphi(N)}^*$ 

exercise

Compute $(b^e)^{e^{-1}} = b^{ee^{-1}} = b$

RSA crypto system

C

$N = pq$

M

(N, e)

$e \in \mathbb{Z}^*_{\varphi(N)}$

e^{-1}

(M, N, e)

$\downarrow$

Enc

$\downarrow$

$C = M^e$ in $\mathbb{Z}_N^*$

Efficient?

(C, e^{-1})

$\downarrow$

Dec

$\downarrow$

$M = C^{e^{-1}}$ in $\mathbb{Z}_N^*$

Secure?

A variant of this is widely used in practice.

From N, if we can efficiently compute $\varphi(N)$, we can crack RSA.

If we can factor N, we can compute $\varphi(N)$.

Is this the only way to crack RSA?

We don't know!

So we are really hoping it is secure.