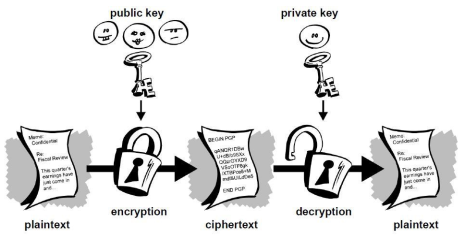
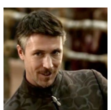
15-251
Great Ideas in Theoretical Computer Science

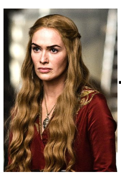
Lecture 27: Cryptography

April 26th, 2018



What is cryptography about?





"loru23n8uladjkfb!#@"

"I will cut your throat"
↓ encryption
"loru23n8uladjkfb!#@"

"loru23n8uladjkfb!#@"
decryption ↓
"I will cut your throat"

What is cryptography about?

Study of protocols that avoid the bad affects of adversaries.

The plan

Recall important things from **modular arithmetic**.

Private (secret) key cryptography.

Secret key sharing.

Public key cryptography.

Important Things to Remember from Last Time

$\mathbb{Z}_4$

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

$\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$

behaves nicely with respect to *addition*

$\mathbb{Z}_8^*$

•	1	3	5	7
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

$\mathbb{Z}_N^* = \{A \in \mathbb{Z}_N : \gcd(A, N) = 1\}$

behaves nicely with respect to *multiplication*

$\varphi(N) = |\mathbb{Z}_N^*|$

if P prime, $\varphi(P) = P - 1$

if P, Q distinct primes, $\varphi(PQ) = (P-1)(Q-1)$

$\mathbb{Z}_5^*$

$\bullet$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

$\varphi(5) = 4$

1^0	1^1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1	1
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	2	4	3	1	2	4	3	1
3^0	3^1	3^2	3^3	3^4	3^5	3^6	3^7	3^8
1	3	4	2	1	3	4	2	1
4^0	4^1	4^2	4^3	4^4	4^5	4^6	4^7	4^8
1	4	1	4	1	4	1	4	1

2 and 3 are called generators.

$\mathbb{Z}_5^*$

$\bullet$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

$\varphi(5) = 4$

1^0	1^1	1^2	1^3	1^4	1^5	1^6	1^7	1^8
1	1	1	1	1	1	1	1	1
2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
1	2	4	3	1	2	4	3	1
3^0	3^1	3^2	3^3	3^4	3^5	3^6	3^7	3^8
1	3	4	2	1	3	4	2	1
4^0	4^1	4^2	4^3	4^4	4^5	4^6	4^7	4^8
1	4	1	4	1	4	1	4	1

$$\forall A, \quad A^4 = 1 \qquad \Longrightarrow \; A^{4k} = (A^4)^k = 1$$

Euler's Theorem:

For any $A \in \mathbb{Z}_N^*$, $\quad A^{\varphi(N)} = 1$.

$$\begin{array}{cccc}
1 & & & \\
\parallel & & & \\
A^0 & A^1 & A^2 & \cdots \\
\parallel & \parallel & \parallel & \\
A^{\varphi(N)} & A^{\varphi(N)+1} & A^{\varphi(N)+2} & \cdots \\
\parallel & \parallel & \parallel & \\
A^{2\varphi(N)} & A^{2\varphi(N)+1} & A^{2\varphi(N)+2} & \cdots
\end{array}$$

IMPORTANT!!!

Complexity of Arithmetic Operations

- **>** addition $A +_N B$
  Do regular addition. Then take mod *N*.
- **>** subtraction $A -_N B$
  -*B* = *N*-*B*. Then do addition.
- **>** multiplication $A \cdot_N B$
  Do regular multiplication. Then take mod *N*.
- **>** division $A/_N B$
  Find B^{-1}. Then do multiplication.
- **>** exponentiation $A^B \bmod N$
  Fast modular exponentiation: repeatedly square and mod.
- **>** taking roots
- **>** logarithm

In $\mathbb{Z}$

$$(B, E) \longrightarrow \boxed{\mathrm{EXP}} \longrightarrow B^E$$

Two inverse functions:

$$(B^E, E) \longrightarrow \boxed{\mathrm{ROOT}_E} \longrightarrow B$$

$$(B^E, B) \longrightarrow \boxed{\mathrm{LOG}_B} \longrightarrow E$$

In $\mathbb{Z}_N^*$

$$(B, E, N) \longrightarrow \boxed{\text{EXP}} \longrightarrow B^E \mod N$$

Two inverse functions:

$$(B^E, E, N) \longrightarrow \boxed{\text{ROOT}_E} \longrightarrow B$$

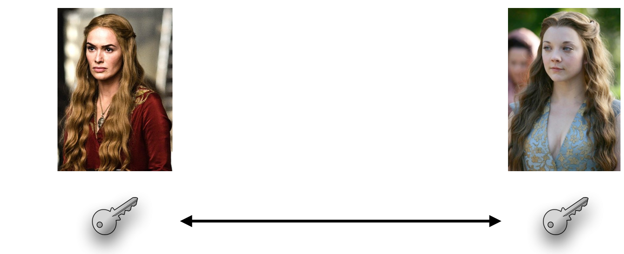
$$(B^E, B, N) \longrightarrow \boxed{\text{LOG}_B} \longrightarrow E$$

One-way function:

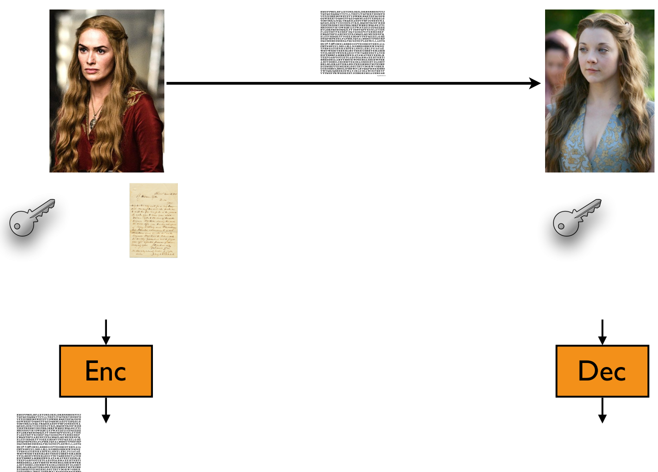
Private Key Cryptography
(Cryptography Before WW2)

Private key cryptography

Private key cryptography



Private key cryptography



A note about security

Better to consider worst-case conditions.

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

Completely sees ciphertext C.

Completely knows the algorithms Enc and Dec.

Caesar shift

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

```
abcdefghijklmnopqrstuvwxyz
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
defghijklmnopqrstuvwxyzabc
```

(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

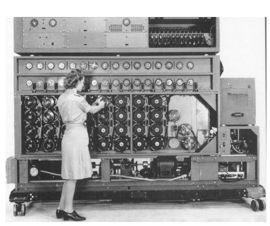
Substitution cipher

```
abcdefghijklmnopqrstuvwxyz
↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓↓
jkbdelmcfgnoxyrsvwzatupqhi
```

 : permutation of the alphabet

Enigma

A much more complex cipher.



One-time pad

M = message   K = key   C = encrypted message

(everything in binary)

Encryption:

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

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

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

One-time pad

M = message   K = key   C = encrypted message

(everything in binary)

Decryption:

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

Encryption: $C = M \oplus K$

Decryption:

One-time pad

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

One-time pad is perfectly secure:

One-time pad

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

Could we reuse the key?

<u>One-time only:</u>

Suppose you encrypt two messages M_1 and M_2 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:

Great Idea

A whole new world of possibilities

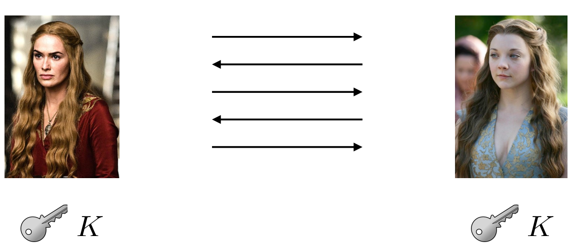
We can find a way to share a random secret key.
(over an insecure channel)

We can get rid of the secret key sharing part.
(public key cryptography)

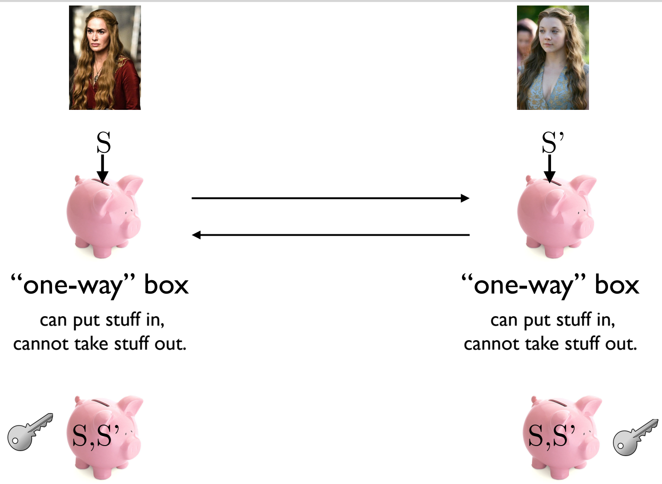
And do much more!!!

Secret Key Sharing

Secret Key Sharing



Secret Key Sharing



DH key exchange

In $\mathbb{Z}_N^*$

$(B, E, N) \longrightarrow \boxed{\mathrm{EXP}} \longrightarrow B^E \mod N$ **easy**

$(B^E, B, N) \longrightarrow \boxed{\mathrm{LOG}_B} \longrightarrow E$ **seems hard**

Careful.

We don't want
$$\begin{array}{cccccc} B^0 & B^1 & B^2 & B^3 & B^4 & \dots \\ \shortparallel & \shortparallel & \shortparallel & \shortparallel & \shortparallel & \\ 1 & B & 1 & B & 1 & \dots \end{array}$$

Much better to have a *generator* B.

DH key exchange

In $\mathbb{Z}_N^*$

$(B, E, N) \longrightarrow \boxed{\mathrm{EXP}} \longrightarrow B^E \mod N$ **easy**

$(B^E, B, N) \longrightarrow \boxed{\mathrm{LOG}_B} \longrightarrow E$ **seems hard**

We'll pick $N = P$ a prime number.

(This ensures there is a generator in $\mathbb{Z}_P^*$.)

We'll pick $B \in \mathbb{Z}_P^*$ so that it is a *generator*.

$$\{B^0,\ B^1,\ B^2,\ B^3,\ \cdots,\ B^{P-2}\} = \mathbb{Z}_P^*$$

DH key exchange

Secure?

Adversary sees: P, B, B^{E_1}, B^{E_2}

Hopefully he can't compute E_1 from B^{E_1}.
(our hope that LOG_B is hard)

Good news: No one knows how to compute LOG_B efficiently.

Bad news: Proving that it cannot be computed efficiently is at least as hard as the **P** vs **NP** problem.

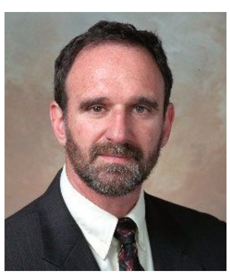
DH assumption:

Decisional DH assumption:

Diffie-Hellman key exchange

1976

Whitfield Diffie



Martin Hellman

To send a private message, one can use:

Diffie-Hellman
(to share a secret key)

+

One-time Pad

Note
This is only as secure as its weakest link, i.e. Diffie-Hellman.

Answers

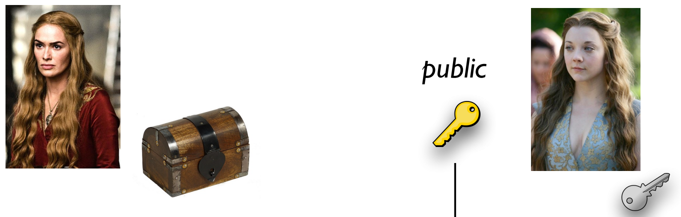
We can find a way to share a random secret key.
(over an insecure channel)

We can get rid of the secret key sharing part.
(public key cryptography)

And do much more!!!

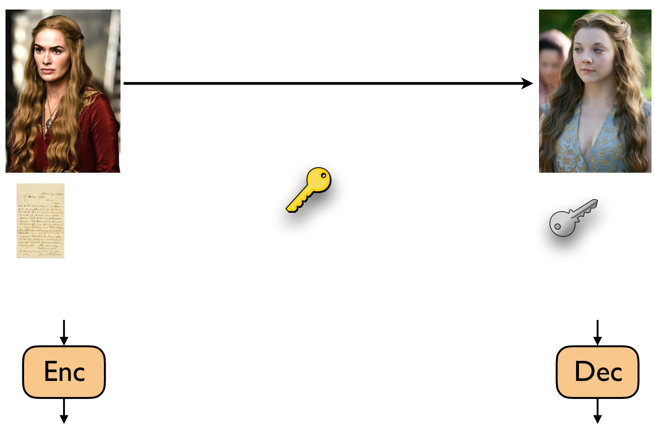
Public Key Cryptography
(Cryptography After WW2)

Public Key Cryptography



Can be used to lock.
But **can't** be used to unlock.

Public key cryptography



RSA crypto system

In $\mathbb{Z}_N^*$

$(B, E, N) \rightarrow \boxed{\mathrm{EXP}} \rightarrow B^E \mod N$ **easy**

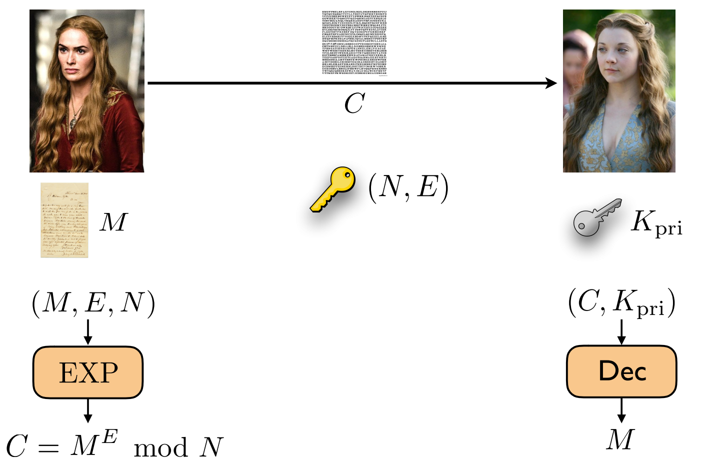
$(B^E, E, N) \rightarrow \boxed{\mathrm{ROOT}_E} \rightarrow B$ **seems hard**

What if we encode using EXP ? $(M = B)$

Public key can be (E, N) .

$\rightarrow \boxed{\text{Enc}} \rightarrow$

RSA crypto system



RSA crypto system

(M, E, N) $\quad M \in \mathbb{Z}_N^*$

$\downarrow$ $\quad E \in \mathbb{Z}_{\varphi(N)}$

EXP

$\downarrow$

$C = M^E \mod N$

(C, K_{pri})

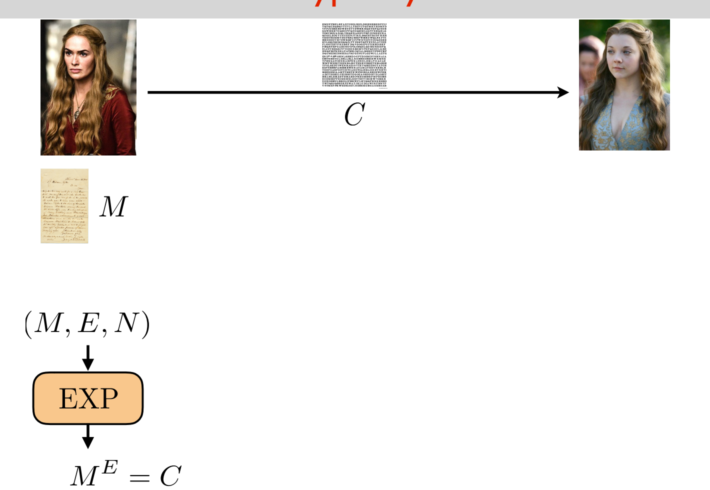
$\downarrow$

Dec???

$\downarrow$

M

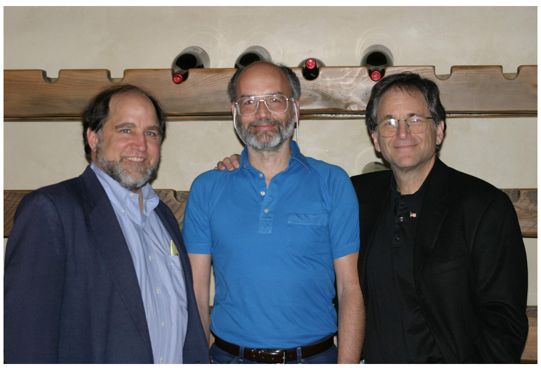
RSA crypto system



Secure?

RSA crypto system

1977



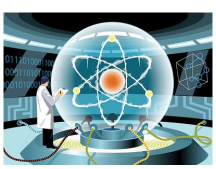
Ron Rivest Adi Shamir Leonard Adleman

Concluding remarks

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 $\varphi(N)$.



Quantum computers can factor efficiently.

Is this the only way to crack RSA?

We don't know!

So we are really <u>hoping</u> it is secure.