ASYMMETRIC ENCRYPTION

Recommended Book

Steven Levy. *Crypto*. Penguin books. 2001.

A non-technical account of the history of public-key cryptography and the colorful characters involved.

Recall Symmetric Cryptography

- Before Alice and Bob can communicate securely, they need to have a common secret key K_{AB}.
- If Alice wishes to also communicate with Charlie then she and Charlie must also have another common secret key K_{AC}.
- If Alice generates K_{AB}, K_{AC}, they must be communicated to her partners over private and authenticated channels.

Public Key Encryption

- Alice has a secret key that is shared with nobody, and an associated public key that is known to everybody.
- Anyone (Bob, Charlie, ...) can use Alice's public key to send her an encrypted message which only she can decrypt.

Think of the public key like a phone number that you can look up in a database

Public Key Encryption

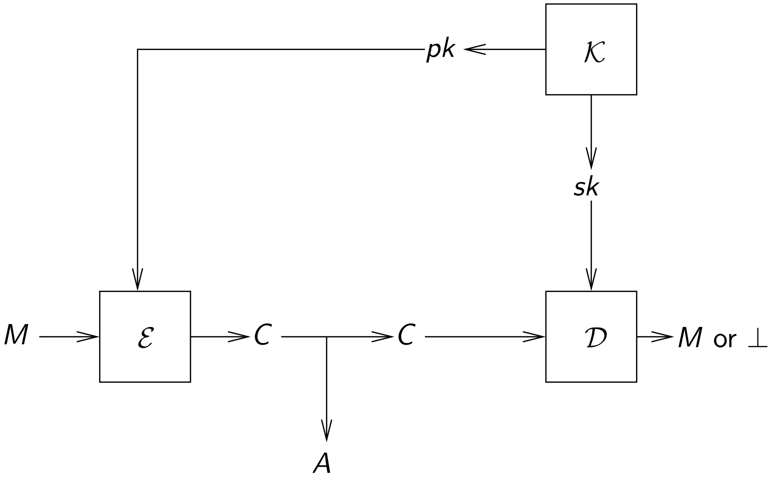
- Alice has a secret key that is shared with nobody, and an associated public key that is known to everybody.
- Anyone (Bob, Charlie, ...) can use Alice's public key to send her an encrypted message which only she can decrypt.

Think of the public key like a phone number that you can look up in a database

- Senders don't need secrets
- There are no shared secrets

Syntax of PKE

A public-key (or asymmetric) encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of three algorithms, where



How it Works

Step 1: Key generation
Alice locally computers $(pk, sk) \xleftarrow{\$} \mathcal{K}$ and stores sk.

Step 2: Alice enables any prospective sender to get pk.

Step 3: The sender encrypts under pk and Alice decrypts under sk.

We don't require privacy of pk but we do require authenticity: the sender should be assured pk is really Alice's key and not someone else's. One could

- Put public keys in a trusted but public "phone book", say a cryptographic DNS.
- Use certificates as we will see later.

Security of PKE Schemes: Issues

The issues are the same as for symmetric encryption:

- Want general purpose schemes
- Security should not rely on assumptions about usage setting
- Want to prevent leakage of partial information about plaintexts

Security requirements

Suppose sender computes

$$C_1 \xleftarrow{\$} \mathcal{E}_{pk}(M_1)\,;\ \cdots\ ;\ C_q \xleftarrow{\$} \mathcal{E}_{pk}(M_q)$$

Adversary A has $C_1, \ldots, C_q$

What if A	
Retrieves sk	Bad!
Retrieves M_1	Bad!

But also ...

Security requirements

We want to hide all partial information about the data stream.

Examples of partial information:

- Does $M_1 = M_2$?
- What is first bit of M_1?
- What is XOR of first bits of M_1, M_2?

Security requirements

We want to hide all partial information about the data stream.

Examples of partial information:

- Does $M_1 = M_2$?
- What is first bit of M_1?
- What is XOR of first bits of M_1, M_2?

Something we won't hide: the length of the message

New Issue

The adversary needs to be given the public key.

Intuition for definition of IND

Consider encrypting one of two possible message streams, either

$$M_0^1, \ldots, M_0^q$$

or

$$M_1^1, \ldots, M_1^q$$

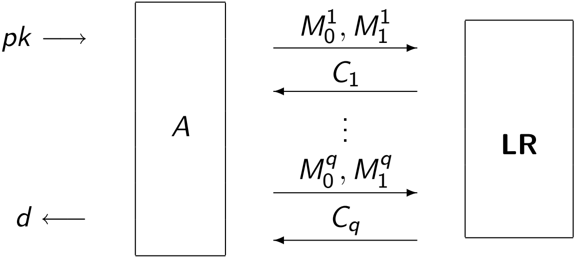
Adversary, given ciphertexts and both data streams, has to figure out which of the two streams was encrypted.

ind-cpa-adversaries

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an public-key encryption scheme

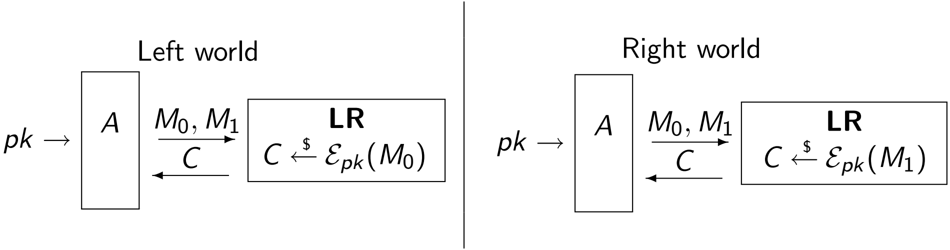
An ind-cpa adversary A has input pk and an oracle **LR**

- It can make a query M_0, M_1 consisting of any two equal-length messages
- It can do this many times
- Each time it gets back a ciphertext
- It eventually outputs a bit



ind-cpa-adversaries

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme



A's output d	Intended meaning: I think I am in the
1	Right world
0	Left world

The harder it is for A to guess world it is in, the more "secure" $\mathcal{AE}$ is as an encryption scheme.

The games

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme

Game $\mathrm{Left}_{\mathcal{AE}}$	Game $\mathrm{Right}_{\mathcal{AE}}$
procedure Initialize	**procedure Initialize**
$(pk, sk) \xleftarrow{\$} \mathcal{K}$; return pk	$(pk, sk) \xleftarrow{\$} \mathcal{K}$; return pk
procedure LR(M_0, M_1)	**procedure LR**(M_0, M_1)
Return $C \xleftarrow{\$} \mathcal{E}_{pk}(M_0)$	Return $C \xleftarrow{\$} \mathcal{E}_{pk}(M_1)$

Associated to $\mathcal{AE}, A$ are the probabilities

$$\Pr\left[\mathrm{Left}_{\mathcal{AE}}^{A} \Rightarrow 1\right] \qquad \Big| \qquad \Pr\left[\mathrm{Right}_{\mathcal{AE}}^{A} \Rightarrow 1\right]$$

that A outputs 1 in each world. The ind-cpa advantage of A is

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{ind-cpa}}(A) = \Pr\left[\mathrm{Right}_{\mathcal{AE}}^{A} \Rightarrow 1\right] - \Pr\left[\mathrm{Left}_{\mathcal{AE}}^{A} \Rightarrow 1\right]$$

Alternative formulation

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme and A an adversary.

Game IND-CPA$_{\mathcal{AE}}$

procedure Initialize
$b \stackrel{\$}{\leftarrow} \{0,1\}$
$(pk, sk) \stackrel{\$}{\leftarrow} \mathcal{K}$
return pk

procedure Finalize(b')
return $(b = b')$

procedure LR(M_0, M_1)
$C \stackrel{\$}{\leftarrow} \mathcal{E}_{pk}(M_b)$
return C

Then the ind-cpa advantage of A is

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(A) = 2 \cdot \Pr\left[\text{INDCPA}^{A}_{\mathcal{AE}} \Rightarrow \text{true}\right] - 1$$

Chosen-ciphertext attacks

Adversary has access to a decryption oracle

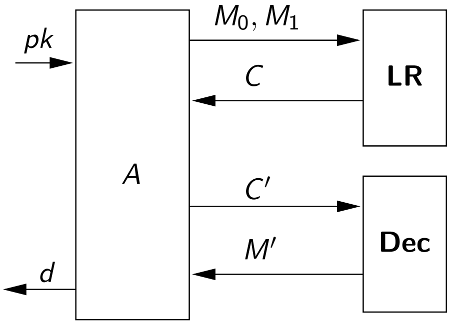
$$\begin{array}{l} C \longrightarrow \\ M \longleftarrow \end{array} \quad \boxed{\textbf{Dec}}$$

Adversary's goal is to learn partial information about un-decrypted messages from their ciphertexts.

ind-cca adversaries

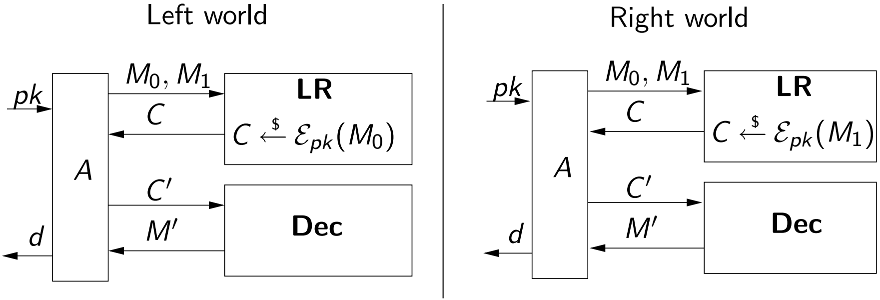
Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme. An ind-cca adversary A

- Has input public key pk
- Has access to a **LR** oracle and a decryption oracle **Dec**
- Outputs a bit



IND-CCA

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme and A an ind-cca adversary.



A's output d	Intended meaning: I think I am in the
1	Right world
0	Left world

The harder it is for A to guess world it is in, the more "secure" $\mathcal{AE}$ is as an encryption scheme.

Avoiding a problem

Encryption can only hide information about un-decrypted messages!

An adversary could just decrypt ciphertext returned from **LR**.

We address this by making the following rule:

- An ind-cca adversary A is not allowed to query **Dec** on a ciphertext previously returned by **LR**

The games

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme and A be an adversary.

Game $\mathrm{Left}_{\mathcal{AE}}$	Game $\mathrm{Right}_{\mathcal{AE}}$
procedure Initialize $(pk, sk) \xleftarrow{\$} \mathcal{K}$; $S \leftarrow \emptyset$; return pk	**procedure Initialize** $(pk, sk) \xleftarrow{\$} \mathcal{K}$; $S \leftarrow \emptyset$; return pk
procedure LR(M_0, M_1) $C \xleftarrow{\$} \mathcal{E}_{pk}(M_0)$; $S \leftarrow S \cup \{C\}$ return C	**procedure LR**(M_0, M_1) $C \xleftarrow{\$} \mathcal{E}_{pk}(M_1)$; $S \leftarrow S \cup \{C\}$ return C
procedure Dec(C) if $C \in S$ then $M \leftarrow \perp$ else $M \leftarrow \mathcal{D}_{sk}(C)$ return M	**procedure Dec**(C) if $C \in S$ then $M \leftarrow \perp$ else $M \leftarrow \mathcal{D}_{sk}(C)$ return M

The ind-cca advantage of A is

$$\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{AE}}(A) = \Pr\left[\mathrm{Right}^{A}_{\mathcal{AE}} \Rightarrow 1\right] - \Pr\left[\mathrm{Left}^{A}_{\mathcal{AE}} \Rightarrow 1\right]$$

Let $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a PKE scheme and A an adversary.

Game $\mathrm{INDCCA}_{\mathcal{AE}}$ **procedure Initialize** $b \xleftarrow{\$} \{0,1\}$; $S \leftarrow \emptyset$ $(pk, sk) \xleftarrow{\$} \mathcal{K}$ return pk **procedure Finalize**(b') return $(b = b')$	**procedure LR**(M_0, M_1) $C \xleftarrow{\$} \mathcal{E}_{pk}(M_b)$ $S \leftarrow S \cup \{C\}$ return C **procedure Dec**(C) if $C \in S$ then $M \leftarrow \perp$ else $M \leftarrow \mathcal{D}_{sk}(C)$ return M

Then the ind-cca advantage of A is

$$\mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{AE}}(A) = 2 \cdot \Pr\left[\mathrm{INDCCA}^{A}_{\mathcal{AE}} \Rightarrow \mathsf{true}\right] - 1$$

Simplification

We may assume A makes only one **LR** query. The hybrid argument used in the symmetric case can be used here too to show that this can decrease its advantage by at most the number of **LR** queries.

Note that in the symmetric case we gave the 1-query adversary an encryption oracle, but that is not needed here since it has the public key which enables it to encrypt.

More Formally

Theorem: Let $\mathcal{AE}$ be a PKE scheme and A an $\mathrm{ind\text{-}cpa}$ adversary making q **LR** queries and having running time t. Then there is a $\mathrm{ind\text{-}cpa}$ adversary A_1 making 1 **LR** query such that

$$\mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{AE}}(A) \leq q \cdot \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{AE}}(A_1)$$

and the running time of A_1 is about t.

Theorem: Let $\mathcal{AE}$ be a PKE scheme and A an $\mathrm{ind\text{-}cca}$ adversary making q_e **LR** queries and q_d **Dec** queries and having running time t. Then there is a $\mathrm{ind\text{-}cca}$ adversary A_1 making 1 **LR** query and q_d **Dec** queries such that

$$\mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{AE}}(A) \leq q_e \cdot \mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{AE}}(A_1)$$

and the running time of A_1 is about t.

Building a PKE Scheme

We would like security to result from the hardness of computing discrete logarithms.

Let the receiver's public key be g where $G = \langle g \rangle$ is a cyclic group. Let's let the encryption of x be g^x. Then

$$\underbrace{g^x}_{\mathcal{E}_g(x)} \xrightarrow{\text{hard}} x$$

so to recover x, adversary must compute discrete logarithms, and we know it can't, so are we done?

Building a PKE Scheme

We would like security to result from the hardness of computing discrete logarithms.

Let the receiver's public key be g where $G = \langle g \rangle$ is a cyclic group. Let's let the encryption of x be g^x. Then

$$\underbrace{g^x}_{\mathcal{E}_g(x)} \xrightarrow{\text{hard}} x$$

so to recover x, adversary must compute discrete logarithms, and we know it can't, so are we done?

Problem: Legitimate receiver needs to compute discrete logarithm to decrypt too! But decryption needs to be feasible.

Building a PKE Scheme

We would like security to result from the hardness of computing discrete logarithms.

Let the receiver's public key be g where $G = \langle g \rangle$ is a cyclic group. Let's let the encryption of x be g^x. Then

$$\underbrace{g^x}_{\mathcal{E}_g(x)} \xrightarrow{\text{hard}} x$$

so to recover x, adversary must compute discrete logarithms, and we know it can't, so are we done?

Problem: Legitimate receiver needs to compute discrete logarithm to decrypt too! But decryption needs to be feasible.

Above, receiver has no secret key!

DH Key Exchange

Let $G = \langle g \rangle$ be a cyclic group of order m.

$$\begin{array}{ccc} \text{Alice} & & \text{Bob} \\ x \xleftarrow{\$} \mathbf{Z}_m;\, X \leftarrow g^x & \begin{array}{c} \xrightarrow{\quad X \quad} \\ \xleftarrow{\quad Y \quad} \end{array} & y \xleftarrow{\$} \mathbf{Z}_m;\, Y \leftarrow g^y \end{array}$$

Then

$$Y^x = (g^y)^x = g^{xy} = (g^x)^y = X^y$$

- Alice can compute $K = Y^x$
- Bob can compute $K = X^y$
- But adversary wanting to compute K is faced with

$$g^x, g^y \longrightarrow g^{xy}$$

which is exactly the CDH problem and is computationally hard.

So this enables Alice and Bob to get a common shared key which they can then use to secure their communications.

The El Gamal Scheme: Idea

We can turn DH key exchange into a public key encryption scheme via

- Let Alice have public key g^x and secret key x
- If Bob wants to encrypt M for Alice, he
  - Picks y and sends g^y to Alice
  - Encrypts M under $g^{xy} = (g^x)^y$ and sends ciphertext to Alice.
- But Alice can recompute $g^{xy} = (g^y)^x$ because
  - g^y is in the received ciphertext
  - x is her secret key

Thus she can decrypt and adversary is still faced with CDH .

EG Encryption, in Full

Let $G = \langle g \rangle$ be a cyclic group of order m. The EG PKE scheme $\mathcal{AE}_{\mathrm{EG}} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{K}$	Alg $\mathcal{E}_X(M)$	Alg $\mathcal{D}_x(Y, W)$
$x \xleftarrow{\$} \mathbf{Z}_m$	$y \xleftarrow{\$} \mathbf{Z}_m$; $Y \leftarrow g^y$	$K = Y^x$
$X \leftarrow g^x$	$K \leftarrow X^y$	$M \leftarrow W \cdot K^{-1}$
return (X, x)	$W \leftarrow K \cdot M$	return M
	return (Y, W)	

We assume the message $M \in G$ is a group element.

Correct decryption is assured because

$$K = X^y = g^{xy} = Y^x$$

Implementation uses several algorithms we have studied before: exponentiation, inverse.

Security of $\mathcal{AE}_{\mathrm{EG}}$

secret key $= x \in \mathbf{Z}_m$, where $m = |G|$
public key $= X = g^x \in G = \langle g \rangle$

$$
\begin{array}{l|l}
\text{algorithm } \mathcal{E}_X(M) & \\
y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y & \text{algorithm } \mathcal{D}_x(Y, W) \\
K \leftarrow X^y;\ W \leftarrow K \cdot M & K \leftarrow Y^x;\ M \leftarrow W \cdot K^{-1} \\
\text{return } (Y, W) & \text{return } M
\end{array}
$$

- To find x given X, adversary must solve DL problem

Security of $\mathcal{AE}_{\mathrm{EG}}$

secret key $= x \in \mathbf{Z}_m$, where $m = |G|$
public key $= X = g^x \in G = \langle g \rangle$

algorithm $\mathcal{E}_X(M)$ $y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y$ $K \leftarrow X^y;\ W \leftarrow K \cdot M$ return (Y, W)	algorithm $\mathcal{D}_x(Y, W)$ $K \leftarrow Y^x;\ M \leftarrow W \cdot K^{-1}$ return M

- To find x given X, adversary must solve DL problem
- To find M given $X, (Y, W)$, adversary must compute $K = g^{xy}$, meaning solve CDH problem

Security of $\mathcal{AE}_{\mathrm{EG}}$

secret key $= x \in \mathbf{Z}_m$, where $m = |G|$
public key $= X = g^x \in G = \langle g \rangle$

algorithm $\mathcal{E}_X(M)$ $y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y$ $K \leftarrow X^y;\ W \leftarrow K \cdot M$ return (Y, W)	algorithm $\mathcal{D}_x(Y, W)$ $K \leftarrow Y^x;\ M \leftarrow W \cdot K^{-1}$ return M

- To find x given X, adversary must solve DL problem
- To find M given $X, (Y, W)$, adversary must compute $K = g^{xy}$, meaning solve CDH problem
- But what prevents leakage of partial information about M? Is the scheme IND-CPA secure?

Security of $\mathcal{AE}_{\mathrm{EG}}$ in $\mathbf{Z}_p^*$

In $G = \mathbf{Z}_p^*$, where p is a prime

- DL, CDH are hard, yet
- There is an attack showing $\mathcal{AE}_{\mathrm{EG}}$ is NOT IND-CPA secure

Number theory is fun!

Squares

We say that a is a square (or quadratic residue) modulo p if there exists b such that $b^2 \equiv a \pmod{p}$.

We let

$$J_p(a) = \begin{cases} 1 & \text{if } a \text{ is a square mod } p \\ 0 & \text{if } a \bmod p = 0 \\ -1 & \text{otherwise} \end{cases}$$

be the Legendre or Jacobi symbol of a modulo p.

Let $p = 11$. Then

- Is 4 a square modulo p?

Squares

We say that a is a square (or quadratic residue) modulo p if there exists b such that $b^2 \equiv a \pmod{p}$.

We let

$$J_p(a) = \begin{cases} 1 & \text{if } a \text{ is a square mod } p \\ 0 & \text{if } a \bmod p = 0 \\ -1 & \text{otherwise} \end{cases}$$

be the Legendre or Jacobi symbol of a modulo p.

Let $p = 11$. Then

- Is 4 a square modulo p?
  YES because $2^2 \equiv 4 \pmod{11}$
- Is 5 a square modulo p?

Squares

We say that a is a square (or quadratic residue) modulo p if there exists b such that $b^2 \equiv a \pmod{p}$.

We let

$$J_p(a) = \begin{cases} 1 & \text{if } a \text{ is a square mod } p \\ 0 & \text{if } a \bmod p = 0 \\ -1 & \text{otherwise} \end{cases}$$

be the Legendre or Jacobi symbol of a modulo p.

Let $p = 11$. Then

- Is 4 a square modulo p?
  YES because $2^2 \equiv 4 \pmod{11}$
- Is 5 a square modulo p?
  YES because $4^2 \equiv 5 \pmod{11}$
- What is $J_{11}(5)$?

Squares

We say that a is a square (or quadratic residue) modulo p if there exists b such that $b^2 \equiv a \pmod{p}$.

We let

$$J_p(a) = \begin{cases} 1 & \text{if } a \text{ is a square mod } p \\ 0 & \text{if } a \bmod p = 0 \\ -1 & \text{otherwise} \end{cases}$$

be the Legendre or Jacobi symbol of a modulo p.

Let $p = 11$. Then

- Is 4 a square modulo p?
  YES because $2^2 \equiv 4 \pmod{11}$
- Is 5 a square modulo p?
  YES because $4^2 \equiv 5 \pmod{11}$
- What is $J_{11}(5)$?
  It equals $+1$

The set of squares

We let

$$\begin{aligned} \mathrm{QR}(\mathbf{Z}_p^*) &= \{a \in \mathbf{Z}_p^* \ : \ a \text{ is a square mod } p\} \\ &= \{a \in \mathbf{Z}_p^* \ : \ \exists b \in \mathbf{Z}_p^* \text{ such that } b^2 \equiv a \pmod{p}\} \end{aligned}$$

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$										

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1									

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4								

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9							

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5						

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5	3					

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5	3	3				

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5	3	3	5			

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5	3	3	5	9		

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5	3	3	5	9	4	

Example

Let $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$a^2 \bmod 11$	1	4	9	5	3	3	5	9	4	1

Then

$$\mathrm{QR}(\mathbf{Z}_p^*) = \{1, 3, 4, 5, 9\}$$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

Observe

- There are 5 squares and 5 non-squares.
- Every square has exactly 2 square roots.

Relation to discrete log

Recall that 2 is a generator of $\mathbf{Z}_{11}^*$

a	1	2	3	4	5	6	7	8	9	10
$\mathrm{DLog}_{\mathbf{Z}_{11}^*,2}(a)$	0	1	8	2	4	9	7	3	6	5
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

Relation to discrete log

Recall that 2 is a generator of $\mathbf{Z}_{11}^*$

a	1	2	3	4	5	6	7	8	9	10
$\mathrm{DLog}_{\mathbf{Z}_{11}^*,2}(a)$	0	1	8	2	4	9	7	3	6	5
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

so

$$J_{11}(a) = 1 \quad \text{iff} \quad \mathrm{DLog}_{\mathbf{Z}_{11}^*,2}(a) \text{ is even}$$

This makes sense because for any generator g,

$$g^{2j} = (g^j)^2$$

is always a square!

Squares and discrete logs

Fact: If $p \geq 3$ is a prime and g is a generator of $\mathbf{Z}_p^*$ then

$$\mathrm{QR}(\mathbf{Z}_p^*) = \{g^i \; : \; 0 \leq i \leq p-2 \text{ and } i \text{ is even}\}$$

Example: If $p = 11$ and $g = 2$ then $p - 2 = 9$ and the squares are

- $2^0 \bmod 11 = 1$
- $2^2 \bmod 11 = 4$
- $2^4 \bmod 11 = 5$
- $2^6 \bmod 11 = 9$
- $2^8 \bmod 11 = 3$

Computing the Legendre symbol

Is there an algorithm that given p and $a \in \mathbf{Z}_p^*$ returns $J_p(a)$, meaning determines whether or not a is a square mod p?

Computing the Legendre symbol

Is there an algorithm that given p and $a \in \mathbf{Z}_p^*$ returns $J_p(a)$, meaning determines whether or not a is a square mod p?

Sure!

Alg $\mathrm{TEST\text{-}SQ}(p, a)$
Let g be a generator of $\mathbf{Z}_p^*$
Let $i \leftarrow \mathrm{DLog}_{\mathbf{Z}_p^*, g}(a)$
if i is even $\mathrm{then\ return}\ 1\ \mathrm{else\ return}\ -1$

Computing the Legendre symbol

Is there an algorithm that given p and $a \in \mathbf{Z}_p^*$ returns $J_p(a)$, meaning determines whether or not a is a square mod p?

Sure!

Alg $\mathrm{TEST\text{-}SQ}(p, a)$
Let g be a generator of $\mathbf{Z}_p^*$
Let $i \leftarrow \mathrm{DLog}_{\mathbf{Z}_p^*, g}(a)$
if i is even $\mathrm{then\ return}\ 1\ \mathrm{else\ return}\ -1$

This is correct, but

- How do we find g?
- How do we compute $\mathrm{DLog}_{\mathbf{Z}_p^*, g}(a)$?

Fermat's Theorem

Fact: If $p \geq 3$ is a prime then for any a

$$J_p(a) \equiv a^{\frac{p-1}{2}} \pmod{p}$$

Example: Let $p = 11$.

- Let $a = 5$. We know that 5 is a square, meaning $J_{11}(5) = 1$. Now compute

$$a^{\frac{p-1}{2}} \equiv 5^5 \equiv (25)(25)(5) \equiv 3 \cdot 3 \cdot 5 \equiv 45 \equiv 1 \pmod{11}.$$

- Let $a = 6$. We know that 6 is not a square, meaning $J_{11}(6) = -1$. Now compute

$$a^{\frac{p-1}{2}} \equiv 6^5 \equiv (36)(36)(6) \equiv 3 \cdot 3 \cdot 6 \equiv 54 \equiv -1 \pmod{11}.$$

Fermat's Theorem

Fact: If $p \geq 3$ is a prime then for any a

$$J_p(a) \equiv a^{\frac{p-1}{2}} \pmod{p}$$

This yields a cubic-time algorithm to compute the Legendre symbol, meaning determine whether or not a given number is a square:

Alg $\text{TEST-SQ}(p, a)$
$s \leftarrow a^{\frac{p-1}{2}} \bmod p$
$\text{if } s = 1 \text{ then return } 1 \text{ else return } -1$

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5						

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6					

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8				

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1			

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1		

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2						

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2	7					

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2	7	3				

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2	7	3	-1			

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2	7	3	-1	-1		

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2	7	3	-1	-1	1	

Multiplicity of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any a, b

$$J_p(ab) = J_p(a) \cdot J_p(b)$$

Example: Let $p = 11$.

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	b	ab	$J_{11}(a)$	$J_{11}(b)$	$J_{11}(ab)$	$J_{11}(a) \cdot J_{11}(b)$
5	6	8	1	-1	-1	-1
2	7	3	-1	-1	1	1

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3			

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4		

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4	1	

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4	1	1

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4	1	1
7			

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4	1	1
7	8		

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4	1	1
7	8	-1	

Inversion of Legendre symbol

Fact: If $p \geq 3$ is a prime then for any $a \in \mathbf{Z}_p^*$

$$J_p(a^{-1}) = J_p(a)$$

Example: $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$J_{11}(a)$	1	-1	1	1	1	-1	-1	-1	1	-1

a	a^{-1}	$J_{11}(a)$	$J_{11}(a^{-1})$
3	4	1	1
7	8	-1	-1

Legendre symbol of EG key

Fact: Let $p \geq 3$ be a prime and $x, y \in \mathbf{Z}_{p-1}$. Let $X = g^x$ and $Y = g^y$ and $K = g^{xy}$. Then

$$J_p(K) = \begin{cases} 1 & \text{if } J_p(X) = 1 \text{ or } J_p(Y) = 1 \\ -1 & \text{otherwise} \end{cases}$$

In particular one can determine $J_p(K)$ given $J_p(X)$ and $J_p(Y)$

Proof:

$$\begin{aligned} J_p(K) &= J_p(g^{xy}) = \begin{cases} 1 & \text{if } xy \text{ is even} \\ -1 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } x \text{ is even or } y \text{ is even} \\ -1 & \text{otherwise} \end{cases} \\ &= \begin{cases} 1 & \text{if } J_p(g^x) = 1 \text{ or } J_p(g^y) = 1 \\ -1 & \text{otherwise} \end{cases} \end{aligned}$$

EG modulo a prime

Let p be a prime and g a generator of $\mathbf{Z}_p^*$. The EG PKE scheme $\mathcal{AE}_{\mathrm{EG}} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{K}$	**Alg** $\mathcal{E}_X(M)$	**Alg** $\mathcal{D}_x(Y, W)$
$x \stackrel{\$}{\leftarrow} \mathbf{Z}_{p-1}$	$y \stackrel{\$}{\leftarrow} \mathbf{Z}_{p-1}$; $Y \leftarrow g^y$	$K = Y^x$
$X \leftarrow g^x$	$K \leftarrow X^y$	$M \leftarrow W \cdot K^{-1}$
return (X, x)	$W \leftarrow K \cdot M$	return M
	return (Y, W)	

The weakness: Suppose $(Y, W) \stackrel{\$}{\leftarrow} \mathcal{E}_X(M)$. Then we claim that given

- the public key X
- the ciphertext (Y, W)

an adversary can easily compute $J_p(M)$.

This represents a loss of partial information.

EG modulo a prime

Suppose (Y, W) is an encryption of M under public key $X = g^x$, where $Y = g^y$. Then

- $W = K \cdot M$
- $K = g^{xy}$

So

$$\begin{aligned} J_p(M) &= J_p(W \cdot K^{-1}) = J_p(W) \cdot J_p(K^{-1}) = J_p(W) \cdot J_p(K) \\ &= J_p(W) \cdot s \end{aligned}$$

where $s = \begin{cases} 1 & \text{if } J_p(X) = 1 \text{ or } J_p(Y) = 1 \\ -1 & \text{otherwise.} \end{cases}$

So we can compute $J_p(M)$ via

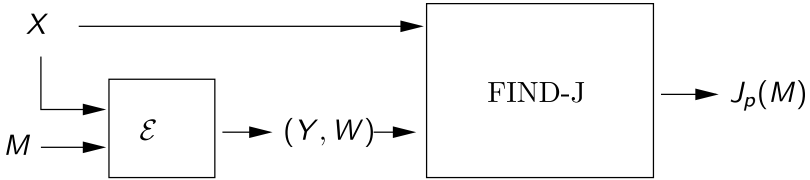
Alg $\mathrm{FIND\text{-}J}(X, Y, W)$
if $J_p(X) = 1$ or $J_p(Y) = 1$ then $s \leftarrow 1$ else $s \leftarrow -1$
return $J_p(W) \cdot s$

EG modulo a prime

Let p be a prime and g a generator of $\mathbf{Z}_p^*$. The EG PKE scheme $\mathcal{AE}_{\mathrm{EG}} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined by

Alg $\mathcal{K}$	**Alg** $\mathcal{E}_X(M)$	**Alg** $\mathcal{D}_x(Y, W)$
$x \xleftarrow{\$} \mathbf{Z}_{p-1}$	$y \xleftarrow{\$} \mathbf{Z}_{p-1}$; $Y \leftarrow g^y$	$K = Y^x$
$X \leftarrow g^x$	$K \leftarrow X^y$	$M \leftarrow W \cdot K^{-1}$
return (X, x)	$W \leftarrow K \cdot M$	return M
	return (Y, W)	

The weakness: There is an algorithm FIND-J



IND-CPA attack

Given public key X

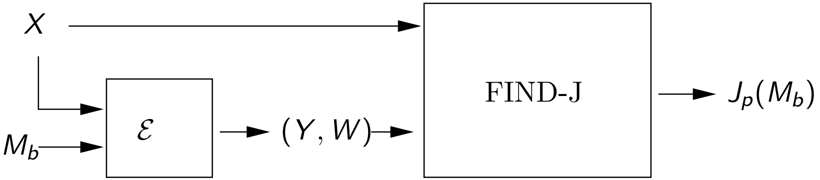
- Produce two messages M_0, M_1
- Receive encrytion (Y, W) of M_b
- Figure out b

IND-CPA attack

Given public key X

- Produce two messages M_0, M_1
- Receive encrytion (Y, W) of M_b
- Figure out b

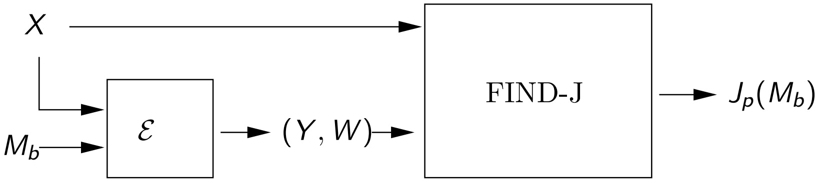
How? Use:



IND-CPA attack

Given public key X

- Let M_0, M_1 be such that $J_p(M_0) = -1$ and $J_p(M_1) = 1$
- Receive encryption (Y, W) of M_b



- if FIND-J$(X, Y, W) = 1$ then return 1 else return 0

IND-CPA attack on EG

Let $\mathcal{AE}_{\text{EG}} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the EG PKE scheme over $\mathbf{Z}_p^*$ where p is a prime.

Left world

A $\xrightarrow{M_0, M_1}$ **LR**: $C \xleftarrow{\$} \mathcal{E}_{pk}(M_0)$ $\xrightarrow{C}$ A

Right world

A $\xrightarrow{M_0, M_1}$ **LR**: $C \xleftarrow{\$} \mathcal{E}_{pk}(M_1)$ $\xrightarrow{C}$ A

adversary $A(X)$

$M_1 \leftarrow 1\,;\; M_0 \leftarrow g$

$(Y, W) \xleftarrow{\$} \mathbf{LR}(M_0, M_1)$

if $\text{FIND-J}(X, Y, W) = 1$ then return 1 else return 0

Then

$$\begin{aligned}\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}_{\text{EG}},A} &= \Pr\left[\text{Right}^A_{\mathcal{AE}_{\text{EG}}} \Rightarrow 1\right] - \Pr\left[\text{Left}^A_{\mathcal{AE}_{\text{EG}}} \Rightarrow 1\right] \\ &= 1 - 0 = 1\end{aligned}$$

IND-CPA security of EG

We have seen that EG is not IND-CPA over groups $G = \mathbf{Z}_p^*$ for prime p.

However it is IND-CPA secure over any group G where the DDH problem is hard.

This is not a contradiction because if p is prime then the DDH problem in $\mathbf{Z}_p^*$ is easy even though DL, CDH seem to be hard.

We can in particular securely implement EG over

- Appropriate prime-order subgroups of $\mathbf{Z}_p^*$ for a prime p
- Elliptic curve groups of prime order

Security of $\mathcal{AE}_{\mathrm{EG}}$

Fact: If DDH is hard in G then $\mathcal{AE}_{\mathrm{EG}}$ is IND-CPA secure

Note: DDH is NOT hard in $\mathbf{Z}_p^*$ (p is a prime)

DDH based security of $\mathcal{AE}_{\mathrm{EG}}$

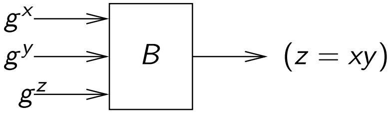
Theorem: Let $\mathcal{AE}_{\mathrm{EG}} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be the El Gamal asymmetric encryption scheme over a cyclic group $G = \langle g \rangle$. Let A be an $\mathrm{ind\text{-}cpa}$ adversary making 1 **LR** query. Then there is a ddh adversary B such that

$$\mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{AE}_{\mathrm{EG}}}(A) \leq 2 \cdot \mathbf{Adv}^{\mathrm{ddh}}_{G,g}(B)$$

Furthermore the running time of B is that of A.

Proof Intuition

Given A want to design



B will let $b \xleftarrow{\$} \{0,1\}$; $pk \leftarrow g^x$ and provide A challenge ciphertext $(g^y, M_b \cdot g^z)$. Then

- If $z = xy$ the ciphertext is correct, so A will have advantage $\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}_{\text{EG}}}(A)$ in computing b
- If $z \xleftarrow{\$} \mathbf{Z}_m$ the ciphertext leaks no information about b so A will have zero advantage in computing b

By seeing whether or not A successfully computes b, adversary B can tell how z was chosen.

Games in proof

Game G_0

procedure Initialize
$x \xleftarrow{\$} \mathbf{Z}_m;\ X \leftarrow g^x;\ b \xleftarrow{\$} \{0,1\}$
return X

procedure LR(M_0, M_1)
$y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y;\ Z \leftarrow g^{xy}$
return $(Y, M_b \cdot Z)$

procedure Finalize(b')
return $(b = b')$

Game G_1

procedure Initialize
$x \xleftarrow{\$} \mathbf{Z}_m;\ X \leftarrow g^x;\ b \xleftarrow{\$} \{0,1\}$
return X

procedure LR(M_0, M_1)
$y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y;\ z \xleftarrow{\$} \mathbf{Z}_m;\ Z \leftarrow g^z$
return $(Y, M_b \cdot Z)$

procedure Finalize(b')
return $(b = b')$

Claim 1: $\Pr\left[G_1^A \Rightarrow \text{true}\right] = \frac{1}{2}$

Claim 2: We can design B so that

$$\Pr\left[G_0^A \Rightarrow \text{true}\right] - \Pr\left[G_1^A \Rightarrow \text{true}\right] \leq \mathbf{Adv}^{\mathrm{ddh}}_{G,g}(B)$$

Analysis

$$\Pr\left[G_0^A \Rightarrow \text{true}\right] = \underbrace{\Pr\left[G_1^A \Rightarrow \text{true}\right]}_{1/2} + \underbrace{\Pr\left[G_0^A \Rightarrow \text{true}\right] - \Pr\left[G_1^A \Rightarrow \text{true}\right]}_{\leq \mathbf{Adv}^{\mathrm{ddh}}_{G,g}(B)}$$

So,

$$\begin{aligned}
\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(A) &= 2 \cdot \Pr\left[G_0^A \Rightarrow \text{true}\right] - 1 \\
&\leq 2 \cdot \left(\frac{1}{2} + \mathbf{Adv}^{\mathrm{ddh}}_{G,g}(B)\right) - 1 \\
&= 2 \cdot \mathbf{Adv}^{\mathrm{ddh}}_{G,g}(B)
\end{aligned}$$

Proof of Claim 1

Game G_1

procedure Initialize
$x \xleftarrow{\$} \mathbf{Z}_m;\ X \leftarrow g^x;\ b \xleftarrow{\$} \{0,1\}$
return X

procedure LR(M_0, M_1)
$y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y;\ z \xleftarrow{\$} \mathbf{Z}_m;\ Z \leftarrow g^z$
return $(Y, M_b \cdot Z)$

procedure Finalize(b')
return $(b = b')$

Game G_2

procedure Initialize
$x \xleftarrow{\$} \mathbf{Z}_m;\ X \leftarrow g^x;\ b \xleftarrow{\$} \{0,1\}$
return X

procedure LR(M_0, M_1)
$y \xleftarrow{\$} \mathbf{Z}_m;\ Y \leftarrow g^y;\ w \xleftarrow{\$} \mathbf{Z}_m;\ W \leftarrow g^w$
return (Y, W)

procedure Finalize(b')
return $(b = b')$

$$\Pr\left[G_1^A \Rightarrow \mathsf{true}\right] = \Pr\left[G_2^A \Rightarrow \mathsf{true}\right] = \frac{1}{2}$$

Proof of Claim 2

adversary $B(X, Y, Z)$
$b \xleftarrow{\$} \{0,1\}$
$b' \xleftarrow{\$} A^{\mathsf{LRSIM}}(X)$
if $(b = b')$ then return 1
else return 0

subroutine $\mathrm{LRSIM}(M_0, M_1)$
return $(Y, M_b \cdot Z)$

Then

$$\Pr\left[\mathrm{DDH1}_{G,g}^{B} \Rightarrow \mathsf{true}\right] = \Pr\left[G_0^A \Rightarrow \mathsf{true}\right]$$

$$\Pr\left[\mathrm{DDH0}_{G,g}^{B} \Rightarrow \mathsf{true}\right] = \Pr\left[G_1^A \Rightarrow \mathsf{true}\right]$$

Message encoding in $\mathcal{AE}_{\mathrm{EG}}$

The $\mathcal{AE}_{\mathrm{EG}}$ asymmetric encryption scheme assumes that messages can be encoded as elements of the underlying group G. But

- Messages may be of large and varying lengths, but we want the group to be fixed beforehand and as small as possible
- For some groups this encoding is hard even if the messages are short

Speed

Asymmetric cryptography is orders of magnitude slower than symmetric cryptography

An exponentiation in a 160-bit elliptic curve group costs about the same as 3000-4000 hashes or block cipher operations

Hybrid encryption

Build an asymmetric encryption scheme by combining symmetric and asymmetric techniques:

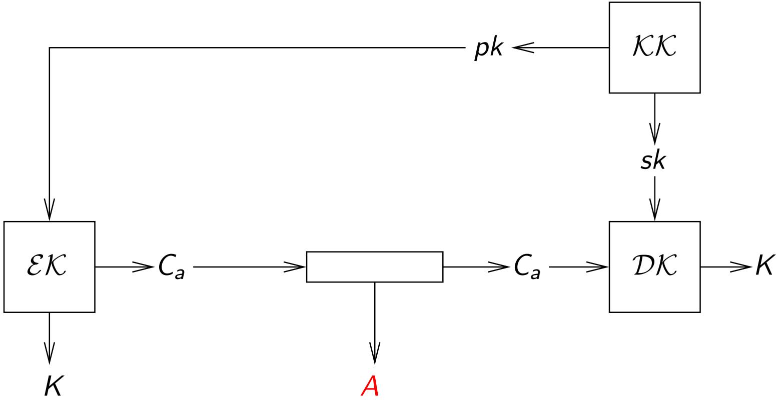
- Symmetrically encrypt data under a key K
- Asymmetrically encrypt K

Benefits:

- Speed
- No encoding problems

Key Encapsulation Mechanisms (KEMs)

A KEM $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ is a triple of algorithms



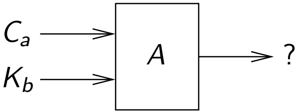
$K \in \{0,1\}^k$ is a symmetric key of some key length k associated to $\mathcal{KEM}$

KEM Security

Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ be a KEM with key length k. Security requires that if we let

$$(K_1, C_a) \stackrel{\$}{\leftarrow} \mathcal{EK}(pk)$$

then K_1 should look "random". Somewhat more precisely, if we also generate $K_0 \stackrel{\$}{\leftarrow} \{0,1\}^k$; $b \stackrel{\$}{\leftarrow} \{0,1\}$ then



A has a hard time figuring out b

KEM IND-CPA security

Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ be a KEM with key length k, and A an adversary.

Game $\mathrm{INDCPA}_{\mathcal{KEM}}$

procedure Initialize
$(pk, sk) \xleftarrow{\$} \mathcal{KK}$; $b \xleftarrow{\$} \{0,1\}$
return pk

procedure Finalize(b')
return $(b = b')$

procedure Enc
$K_0 \xleftarrow{\$} \{0,1\}^k$; $(K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$
return (K_b, C_a)

We allow only one call to **Enc**. The ind-cpa advantage of A is

$$\mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{KEM}}(A) = 2 \cdot \Pr\left[\mathrm{INDCPA}^{A}_{\mathcal{KEM}} \Rightarrow \mathsf{true}\right] - 1$$

Alternative formulation of KEM IND-CPA security

Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ be a KEM with key length k, and A an adversary.

Game $\mathrm{INDCPA0}_{\mathcal{KEM}}$	Game $\mathrm{INDCPA1}_{\mathcal{KEM}}$
procedure Initialize	**procedure Initialize**
$(pk, sk) \xleftarrow{\$} \mathcal{KK}$	$(pk, sk) \xleftarrow{\$} \mathcal{KK}$
return pk	return pk
procedure Enc	**procedure Enc**
$K_0 \xleftarrow{\$} \{0,1\}^k$; $(K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$	$K_0 \xleftarrow{\$} \{0,1\}^k$; $(K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$
return (K_0, C_a)	return (K_1, C_a)

We allow only one call to **Enc**. The ind-cpa advantage of A is

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(A) = \Pr\left[\mathrm{INDCPA1}^{A}_{\mathcal{KEM}} \Rightarrow 1\right] - \Pr\left[\mathrm{INDCPA0}^{A}_{\mathcal{KEM}} \Rightarrow 1\right]$$

KEM IND-CCA security

Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ be a KEM with key length k, and A an adversary.

Game $\mathrm{INDCCA}_{\mathcal{KEM}}$

procedure Initialize
$(pk, sk) \xleftarrow{\$} \mathcal{KK}\,;\ b \xleftarrow{\$} \{0,1\}$
$S \leftarrow \emptyset\,;\ \mathrm{return}\ pk$

procedure Finalize(b')
$\mathrm{return}\ \ (b = b')$

procedure Enc
$K_0 \xleftarrow{\$} \{0,1\}^k;\ (K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$
$S \leftarrow S \cup \{C_a\}$
$\mathrm{return}\ \ (K_b, C_a)$

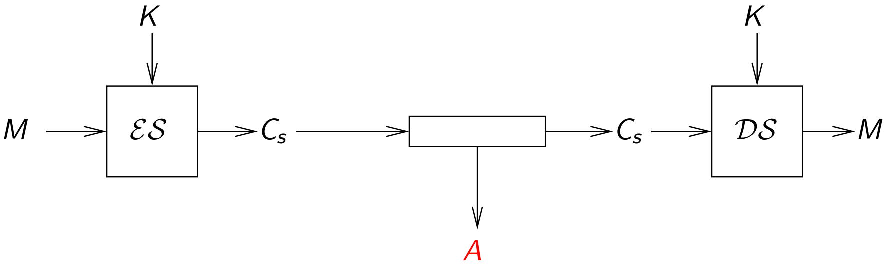
procedure Dec(C_a)
$\mathrm{if}\ \ C_a \in S\ \mathrm{then}\ K \leftarrow \perp$
$\mathrm{else}\ \ K \leftarrow \mathcal{DK}_{sk}(C_a)$
$\mathrm{return}\ \ K$

We allow only one call to **Enc**. The ind-cca advantage of A is

$$\mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{KEM}}(A) = 2 \cdot \Pr\left[\mathrm{INDCCA}^{A}_{\mathcal{KEM}} \Rightarrow \mathsf{true}\right] - 1$$

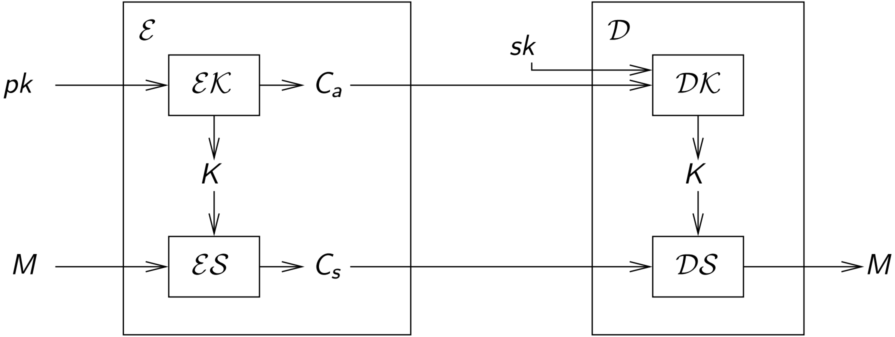
Data Encapsulation Mechanisms (DEMs)

A DEM is simply a symmetric encryption scheme $\mathcal{SE} = (\mathcal{KS}, \mathcal{ES}, \mathcal{DS})$ where $\mathcal{K}$ returns $K \xleftarrow{\$} \{0,1\}^k$ for some k called the key length.



KEM + DEM asymmetric encryption

Given $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ and DEM $\mathcal{SE} = (\mathcal{KS}, \mathcal{ES}, \mathcal{DS})$ both with key length k, define the asymmetric encryption scheme $\mathcal{AE} = (\mathcal{KK}, \mathcal{E}, \mathcal{D})$ as follows:



Ciphertext $C = (C_a, C_s)$

KEM + DEM works

If the KEM is	and the DEM is	then the constructed $\mathcal{AE}$ scheme is
IND-CPA	IND-CPA	IND-CPA
IND-CCA	IND-CCA	IND-CCA

Theorem: Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ and DEM $\mathcal{SE} = (\mathcal{KS}, \mathcal{ES}, \mathcal{DS})$ both have key length k, and let $\mathcal{AE} = (\mathcal{KK}, \mathcal{E}, \mathcal{D})$ be the corresponding asymmetric encryption scheme. Let A be an $\mathrm{ind\text{-}cpa}$ adversary making $1\ \mathrm{LR}$ query and having running time t. Then there are $\mathrm{ind\text{-}cpa}$ adversaries B_a, B_s such that

$$\mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{AE}}(A) \leq 2 \cdot \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{KEM}}(B_a) + \mathbf{Adv}^{\mathrm{ind\text{-}cpa}}_{\mathcal{SE}}(B_s)$$

Furthermore B_a makes one Enc query, B_s makes one LR query, and both have running time about t.

Note: Since $\mathcal{SE}$ is only required to be $1\text{-}\mathrm{query}$ secure we can instantiate it with a deterministic scheme like CBC with fixed IV

KEM + DEM: CCA Security

Theorem: Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ and DEM $\mathcal{SE} = (\mathcal{KS}, \mathcal{ES}, \mathcal{DS})$ both have key length k, and let $\mathcal{AE} = (\mathcal{KK}, \mathcal{E}, \mathcal{D})$ be the corresponding asymmetric encryption scheme. Let A be an $\mathrm{ind\text{-}cca}$ adversary making 1 LR query, q_d decryption queries and having running time t. Then there are $\mathrm{ind\text{-}cca}$ adversaries B_a, B_s such that

$$\mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{AE}}(A) \leq 2 \cdot \mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{KEM}}(B_a) + \mathbf{Adv}^{\mathrm{ind\text{-}cca}}_{\mathcal{SE}}(B_s)$$

Furthermore B_a makes one Enc query, B_s makes one LR query, and both have running time about t and make q_d decryption queries

Proof of KEM + DEM security: Intuition

With $b \xleftarrow{\$} \{0,1\}$; $K_0 \xleftarrow{\$} \{0,1\}^k$; $(K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$

Game	Challenge ciphertext	Adversary goal
G_0	$C_a, \mathcal{ES}_{K_1}(M_b)$	Compute b
G_1	$C_a, \mathcal{ES}_{K_0}(M_b)$	Compute b

- A unlikely to win in G_1 because of security of symmetric scheme (DEM)
- A is about as likely to win in G_1 as in G_0 due to KEM security

Games for proof of KEM + DEM

Game G_0

procedure Initialize

$(pk, sk) \xleftarrow{\$} \mathcal{KK};\ b \xleftarrow{\$} \{0,1\}$

return pk

procedure LR(M_0, M_1)

$K_0 \xleftarrow{\$} \{0,1\}^k;\ (K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$

$C_s \xleftarrow{\$} \mathcal{ES}_{K_1}(M_b)$

return (C_a, C_s)

procedure Finalize(b')

return $(b = b')$

Game G_1

procedure Initialize

$(pk, sk) \xleftarrow{\$} \mathcal{KK};\ b \xleftarrow{\$} \{0,1\}$

return pk

procedure LR(M_0, M_1)

$K_0 \xleftarrow{\$} \{0,1\}^k;\ (K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$

$C_s \xleftarrow{\$} \mathcal{ES}_{K_0}(M_b)$

return (C_a, C_s)

procedure Finalize(b')

return $(b = b')$

Claim 1

We can design B_s so that

$$2 \cdot \Pr\left[G_1^A \Rightarrow \mathsf{true}\right] - 1 \leq \mathbf{Adv}_{\mathcal{SE}}^{\mathrm{ind\text{-}cpa}}(B_s)$$

Idea:

- Key in B_s's IND-CPA game plays role of K_0
- Challenge bit in B_s's IND-CPA game plays role of b
- B_s itself picks pk, sk, K_1, C_a
- B_s invokes its LR oracle to get C_s

B_s, details

Claim 1: Adversary B_s below satisfies

$$2 \cdot \Pr\left[G_1^A \Rightarrow \mathsf{true}\right] - 1 \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(B_s)$$

adversary B_s $(pk, sk) \xleftarrow{\$} \mathcal{KK}$ $b' \leftarrow A^{\mathrm{LRSIM}}(pk)$ return b'	**subroutine** $\mathrm{LRSIM}(M_0, M_1)$ $(K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$ $C_s \xleftarrow{\$} \mathbf{LR}(M_0, M_1)$ return (C_a, C_s)

Then

$$\Pr\left[G_1^A \Rightarrow \mathsf{true}\right] = \Pr\left[\mathrm{INDCPA}^{B_s}_{\mathcal{SE}} \Rightarrow \mathsf{true}\right]$$

But by definition

$$2 \cdot \Pr\left[\mathrm{INDCPA}^{B_s}_{\mathcal{SE}} \Rightarrow \mathsf{true}\right] - 1 = \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(B_s)$$

Claim 2

We can design B_a so that

$$\Pr\left[G_0^A \Rightarrow \mathsf{true}\right] - \Pr\left[G_1^A \Rightarrow \mathsf{true}\right] \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(B_a)$$

Idea:

- K_0, K_1, C_a from B_a's Enc oracle
- pk from B_a's ind-cpa game
- b choosen by B_a

B_a, details

Claim 2: Adversary B_a below satisfies

$$\Pr\left[G_0^A \Rightarrow \text{true}\right] - \Pr\left[G_1^A \Rightarrow \text{true}\right] \leq \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(B_a)$$

adversary $B_a(pk)$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow A^{\mathrm{LRSIM}}$
if $(b = b')$ then return 1
else return 0

subroutine $\mathrm{LRSIM}(M_0, M_1)$
$(K_d, C_a) \xleftarrow{\$} \mathsf{Enc}()$
$C_s \xleftarrow{\$} \mathcal{ES}_{K_d}(M_b)$
return (C_a, C_s)

If $d = 1$ then A gets environment of G_0 so

$$\Pr\left[\mathrm{INDCPA1}^{B_a}_{\mathcal{KEM}} \Rightarrow \text{true}\right] = \Pr\left[G_0^A \Rightarrow \text{true}\right]$$

If $d = 0$ then A gets environment of G_1 so

$$\Pr\left[\mathrm{INDCPA0}^{B_a}_{\mathcal{KEM}} \Rightarrow \text{true}\right] = \Pr\left[G_1^A \Rightarrow \text{true}\right]$$

KEM + DEM security proof: Conclusion

$$
\begin{aligned}
&\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE}}(A) \\
&= \; 2 \cdot \Pr\left[G_0^A \Rightarrow \mathsf{true}\right] - 1 \\
&= \; 2 \cdot \left(\Pr\left[G_1^A \Rightarrow \mathsf{true}\right] + \Pr\left[G_0^A \Rightarrow \mathsf{true}\right] - \Pr\left[G_1^A \Rightarrow \mathsf{true}\right]\right) - 1 \\
&= \; 2 \cdot \Pr\left[G_1^A \Rightarrow \mathsf{true}\right] - 1 + 2 \cdot \left(\Pr\left[G_0^A \Rightarrow \mathsf{true}\right] - \Pr\left[G_1^A \Rightarrow \mathsf{true}\right]\right) \\
&\leq \; \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{SE}}(B_s) + 2 \cdot \mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(B_a)
\end{aligned}
$$

$\mathcal{AE}_{\mathrm{EG}}$ as a KEM + DEM

Let $G = \langle g \rangle$ be a cyclic group of order m and let $sk = x$ and $pk = X = g^x$ be $\mathcal{AE}_{\mathrm{EG}}$ keys

algorithm $\mathcal{E}_X(M)$	algorithm $\mathcal{D}_x(Y, W)$
$y \xleftarrow{\$} \mathbf{Z}_m$; $C_a \leftarrow g^y$	$K \leftarrow {C_a}^x$; $M \leftarrow C_s \cdot K^{-1}$
$K \leftarrow X^y$; $C_s \leftarrow K \cdot M$	return M
return (C_a, C_s)	

Is a KEM + DEM with

- Symmetric key $K = g^{xy} = X^y = {C_a}^x$
- DEM $\mathcal{ES}_K(M) = K \cdot M$

$\mathcal{AE}_{\mathrm{EG}}$ as a KEM

Let $G = \langle g \rangle$ be a cyclic group of order m and let $sk = x$ and $pk = X = g^x$ be $\mathcal{AE}_{\mathrm{EG}}$ keys. Then $\mathcal{AE}_{\mathrm{EG}}$ can be viewed as a KEM + DEM with

algorithm $\mathcal{EK}_X()$ $y \xleftarrow{\$} \mathbf{Z}_m;\ \ C_a \leftarrow g^y$ $K \leftarrow X^y$ return (K, C_a)	algorithm $\mathcal{ES}_K(M)$ return $K \cdot M$

But this DEM has many drawbacks as we saw before.

$\mathcal{AE}_{\mathrm{EG}}$ as a KEM

Let $G = \langle g \rangle$ be a cyclic group of order m and let $sk = x$ and $pk = X = g^x$ be $\mathcal{AE}_{\mathrm{EG}}$ keys. Then $\mathcal{AE}_{\mathrm{EG}}$ can be viewed as a KEM + DEM with

algorithm $\mathcal{EK}_X()$ $y \xleftarrow{\$} \mathbf{Z}_m;\ C_a \leftarrow g^y$ $K \leftarrow X^y$ return (K, C_a)	algorithm $\mathcal{ES}_K(M)$ return $K \cdot M$

But this DEM has many drawbacks as we saw before.

Can we substitue the DEM with (say) $\mathrm{AES\text{-}CBC}$ to solve these problems?

Difficulty: The key for $\mathrm{AES\text{-}CBC}$ needs to be a 128 bit string, not a group element.

An alternative to $\mathcal{AE}_{\mathrm{EG}}$ KEM

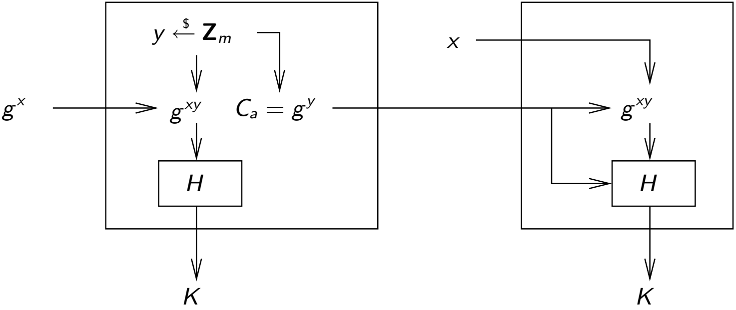
Let the symmetric key be $H(g^y \| g^{xy})$ rather than merely g^{xy}, where $H : \{0,1\}^* \to \{0,1\}^k$ is a hash function

For use with $\mathrm{AES\text{-}CBC}$, set $k = 128$

The $\mathcal{AE}_{\mathrm{EG}}$ KEM

Let $G = \langle g \rangle$ be a cyclic group of order m and define $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ by

algorithm $\mathcal{KK}$	algorithm $\mathcal{EK}_X()$	algorithm $\mathcal{DK}_x(C_a)$
$x \xleftarrow{\$} \mathbf{Z}_m$	$y \xleftarrow{\$} \mathbf{Z}_m$; $C_a \leftarrow g^y$	$Z \leftarrow C_a{}^x$
$X \leftarrow g^x$	$Z \leftarrow X^y$	$K \leftarrow H(C_a \| Z)$
return (X, x)	$K \leftarrow H(C_a \| Z)$	return K
	return (K, C_a)	



Here $H : \{0,1\}^* \to \{0,1\}^k$

What H is suitable?

Our analysis will assume H is "perfect"

Question: What does this mean?

Answer: H will be modeled as a random oracle

Random Oracle Model

A random oracle is a publicly-accessible random function

$$\xrightarrow{\quad W \quad} \qquad \begin{array}{|l|}\hline \text{If } H[W] = \perp \text{ then} \\ \quad H[W] \xleftarrow{\$} \{0,1\}^k \\ \text{Return } H[W] \\ \hline \end{array}$$

$$\xleftarrow{\quad H(W) \quad}$$

Oracle access to H provided to

- all scheme algorithms
- the adversary

The only access to H is oracle access.

The RO EG KEM

Let $G = \langle g \rangle$ be a cyclic group of order m and define the RO-model KEM $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ by

algorithm $\mathcal{KK}$	algorithm $\mathcal{EK}_X^H()$	algorithm $\mathcal{DK}_x^H(C_a)$
$x \stackrel{\$}{\leftarrow} \mathbf{Z}_m$	$y \stackrel{\$}{\leftarrow} \mathbf{Z}_m;\ C_a \leftarrow g^y$	$Z \leftarrow {C_a}^x$
$X \leftarrow g^x$	$Z \leftarrow X^y$	$K \leftarrow H(C_a \| Z)$
return (X, x)	$K \leftarrow H(C_a \| Z)$	return K
	return (K, C_a)	

RO model KEM CPA security

Let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ be a RO model KEM with key length k and A an adversary

Game $\mathrm{INDCPA}_{\mathcal{KEM}}$

procedure Initialize
$(pk, sk) \xleftarrow{\$} \mathcal{KK};\ b \xleftarrow{\$} \{0,1\}$
return pk

procedure Finalize(b')
return $(b = b')$

procedure $H(W)$
if $H[W] = \perp$ then $H[W] \xleftarrow{\$} \{0,1\}^k$
return $H[W]$

procedure Enc
$K_0 \xleftarrow{\$} \{0,1\}^k;\ (K_1, C_a) \xleftarrow{\$} \mathcal{EK}_{pk}()$
return (K_b, C_a)

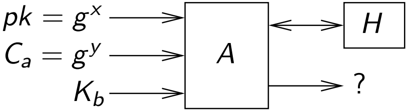
We allow only one call to **Enc**. The ind-cpa advantage of A is

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(A) = 2 \cdot \Pr\left[\mathrm{INDCPA}^{A}_{\mathcal{KEM}} \Rightarrow \mathsf{true}\right] - 1$$

RO model security of our EG KEM

Claim: The $\mathcal{AE}_{\mathrm{EG}}$ KEM is IND-CPA secure in the RO model

In the IND-CPA game

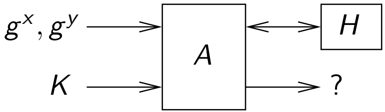


where

$$b \xleftarrow{\$} \{0,1\};\ K_0 \xleftarrow{\$} \{0,1\}^k;\ K_1 \leftarrow H(g^y \| g^{xy})$$

We are saying A has a hard time figuring out b. Why?

Intuition



where

$$x, y \xleftarrow{\$} \mathbf{Z}_m;\;\; b \xleftarrow{\$} \{0,1\};\;\; K_0 \xleftarrow{\$} \{0,1\}^k;$$

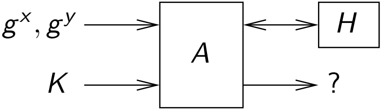
$$K_1 \leftarrow H(g^y \| g^{xy});\;\; K \leftarrow K_b$$

Possible strategy for A:

- Query $g^y \| g^{xy}$ to H to get back $Z = H(g^y \| g^{xy})$
- If $Z = K$ then return 1 else return 0

This startegy works! So why do we say that A can't figure out b?

Intuition



where

$$x, y \xleftarrow{\$} \mathbf{Z}_m;\ \ b \xleftarrow{\$} \{0,1\};\ \ K_0 \xleftarrow{\$} \{0,1\}^k;$$

$$K_1 \leftarrow H(g^y \| g^{xy});\ \ K \leftarrow K_b$$

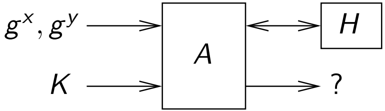
Possible strategy for A:

- Query $g^y \| g^{xy}$ to H to get back $Z = H(g^y \| g^{xy})$
- If $Z = K$ then return 1 else return 0

This startegy works! So why do we say that A can't figure out b?

Problem: A can't compute g^{xy} hence can't make the query

Intuition



where

$$x, y \xleftarrow{\$} \mathbf{Z}_m;\ \ b \xleftarrow{\$} \{0,1\};\ \ K_0 \xleftarrow{\$} \{0,1\}^k;$$

$$K_1 \leftarrow H(g^y \| g^{xy});\ \ K \leftarrow K_b$$

Observation:

- If A does not query $g^y \| g^{xy}$ to H then it cannot predict $H(g^y \| g^{xy})$ and hence has no chance at all to determine whether $K = H(g^y \| g^{xy})$ or K is random
- If A does query $g^y \| g^{xy}$ to H it has solved the CDH problem

In the second case, we can "see" a solution to CDH by "watching" A's oracle queries

RO model security of EG KEM

Theorem: Let $G = \langle g \rangle$ be a cyclic group of order m and let $\mathcal{KEM} = (\mathcal{KK}, \mathcal{EK}, \mathcal{DK})$ be the RO model of EG KEM over G with key length k. Let A be an ind-cpa adversary making 1 LR query and q queries to the RO H and having running time at most t. Then there is a cdh adversary B such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(A) \leq q \cdot \mathbf{Adv}^{\text{cdh}}_{G,g}(B).$$

Furthermore B has running time about t.

Games for proof

Game G_0, $\boxed{G_1}$

procedure Initialize
$x, y \xleftarrow{\$} \mathbf{Z}_m$; $K \xleftarrow{\$} \{0,1\}^k$
return g^x

procedure Enc
return (K, g^y)

procedure $H(W)$
$H[W] \xleftarrow{\$} \{0,1\}^k$; $Y||Z \leftarrow W$
if $Z = g^{xy}$ and $Y = g^y$ then
  $bad \leftarrow true$; $\boxed{H[W] \leftarrow K}$
return $H[W]$

Assume (wlog) that A never repeats a H-query. Then

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(A) = \Pr[G_1^A \Rightarrow true] - \Pr[G_0^A \Rightarrow true]$$
$$\leq \Pr[G_0^A \text{ sets bad}]$$

Bounding the probability of setting bad

We would like to design B so that $\Pr[G_0^A \textit{ sets bad}] \leq \mathbf{Adv}^{\mathrm{cdh}}_{G,g}(B)$

adversary $B(g^x, g^y)$
$K \xleftarrow{\$} \{0,1\}^k$
$b' \leftarrow A^{\mathrm{EncSIM,HSIM}}(g^x)$

subroutine EncSIM
return K, g^y

subroutine HSIM(W)
$H[W] \xleftarrow{\$} \{0,1\}^k$; $Y || Z \leftarrow W$
if $Z = g^{xy}$ and $Y = g^y$ then
  output W and halt
return $H[W]$

Bounding the probability of setting bad

We would like to design B so that $\Pr[G_0^A \textit{ sets bad}] \leq \mathbf{Adv}^{\mathrm{cdh}}_{G,g}(B)$

adversary $B(g^x, g^y)$
$K \xleftarrow{\$} \{0,1\}^k$
$b' \leftarrow A^{\mathrm{EncSIM,HSIM}}(g^x)$

subroutine EncSIM
return K, g^y

subroutine HSIM(W)
$H[W] \xleftarrow{\$} \{0,1\}^k$; $Y||Z \leftarrow W$
if $Z = g^{xy}$ and $Y = g^y$ then
  output W and halt
return $H[W]$

Problem: B can't do the test since it does not know g^{xy}.

The generalized CDH problem

Let $G = \langle g \rangle$ be a cyclic group of order m and B' an adversary with q outputs.

Game $\mathrm{CDH}_{G,g}$

procedure Initialize	**procedure Finalize**$(Z_1, \ldots, Z_q)$
$x, y \xleftarrow{\$} \mathbf{Z}_m$	for $i = 1, \ldots, q$ do
return g^x, g^y	if $Z_i = g^{xy}$ then $win \leftarrow true$
	return win

The cdh-advantage of B' is

$$\mathbf{Adv}^{\mathrm{cdh}}_{G,g}(B') = \Pr[\mathrm{CDH}^{B'}_{G,g} \Rightarrow true]$$

Reducing generalized CDH to CDH

Lemma: Let $G = \langle g \rangle$ be a cyclic group and B' a cdh-adversary that has q outputs and running time t. Then there is a cdh-adversary B that has 1 output, running time about t, and

$$\mathbf{Adv}^{\mathrm{cdh}}_{G,g}(B') \leq q \cdot \mathbf{Adv}^{\mathrm{cdh}}_{G,g}(B)$$

Proof:

$$\begin{array}{l} \text{Adversary } B(g^x, g^y) \\ (Z_1, \ldots, Z_q) \stackrel{\$}{\leftarrow} B'(g^x, g^y) \\ i \stackrel{\$}{\leftarrow} \{1, \ldots, q\} \\ \mathrm{return}\ Z_i \end{array}$$

Bounding the probability of setting bad

We design a q-output cdh adversary B' so that

$$\Pr[G_0^A \textit{ sets bad}] \leq \mathbf{Adv}^{\mathrm{cdh}}_{G,g}(B')$$

adversary $B(g^x, g^y)$ $K \xleftarrow{\$} \{0,1\}^k$ $i \leftarrow 0$ $b' \leftarrow A^{\mathrm{EncSIM},\mathrm{HSIM}}(g^x)$ return $Z_1, \ldots, Z_q$	**subroutine** EncSIM return K, g^y **subroutine** $\mathrm{HSIM}(W)$ $H[W] \xleftarrow{\$} \{0,1\}^k$; $Y \| Z \leftarrow W$ $i \leftarrow i+1$; $Z_i \leftarrow Z$ return $H[W]$

Then the cdh-adversary B of the theorem is obtained by applying the lemma to B'.

DHIES and ECIES [ABR]

The asymmetric encryption scheme derived from KEM + DEM with

- The RO EG KEM
- Some suitable mode of operation DEM (e.g. CBC) is standardized as DHIES and ECIES

ECIES features:

Operation	**Cost**
encryption	2 160-bit exp
decryption	1 160-bit exp
ciphertext expansion	160-bits

ciphertext expansion = (length of ciphertext) - (length of plaintext)

But what about H?

We have studied the EG KEM in an abstract model where H is a random function accessible only as an oracle. To get a "real" scheme we need to instantiate H with a "real" function

How do we do this securely?

PRF-based RO

We know that PRFs approximate random functions, meaning if $F : \{0,1\}^s \times \{0,1\}^k \to \{0,1\}^k$ is a PRF then the I/O behavior of F_K is like that of a random function.

So can we instantiate H via F?

PRF-based RO

We know that PRFs approximate random functions, meaning if $F : \{0,1\}^s \times \{0,1\}^k \to \{0,1\}^k$ is a PRF then the I/O behavior of F_K is like that of a random function.

So can we instantiate H via F?

F_K depends on a key K. Who will have K? Since the sender needs to be able to encrypt given just pk, we need to put K in pk.

PRF-based RO

We know that PRFs approximate random functions, meaning if $F : \{0,1\}^s \times \{0,1\}^k \to \{0,1\}^k$ is a PRF then the I/O behavior of F_K is like that of a random function.

So can we instantiate H via F?

F_K depends on a key K. Who will have K? Since the sender needs to be able to encrypt given just pk, we need to put K in pk.

Problem: The adversary has pk and PRFs don't preserve security when the key is known to the adversary.

RO paradigm

- Design and analyze schemes in RO model
- In instantiation, replace RO with a hash-function based construct.

Example: $H(W) =$ first 128 bits of SHA1(W). More generally if we need ℓ output bits:

$H(W) =$ first ℓ bits of SHA1$(1||W)$ $||$ SHA1$(2||W)$ $||$ $\ldots$

RO paradigm

There is no proof that the instantiated scheme is secure based on some "standard" assumption about the hash function.

The RO paradigm is a heuristic that seems to work well in practice.

The RO model is a model, not an assumption on H. To say

"Assume SHA1 is a RO"

makes no sense: it isn't.

PRF paradigm vs RO paradigm

PRF paradigm: For symmetric cryptography

- Design scheme in a model where parties (sender and receiver) have oracle access to a random function, but the adversary does not.
- Provable security in maintained when the oracle is replaced by F_K where F is a PRF and K is held by the parties, but not given to the adversary.

RO paradigm: For asymmetric cryptography

- Design scheme in a model where everyone, adversary included, has oracle access to a random function.
- Instantiation results in a scheme that is heuristically good, but not provably so.

Instantiating ROs

There are schemes which are

- Secure in the RO model
- But insecure for all instantiations of the RO by real (families of) functions.

However, these counter-example schemes are all artificial, contrived to fail.

So far it seems that the RO paradigm works (yields secure instantiated schemes) for “real and natural” schemes.

But there is no proof of this.

Why the RO paradigm?

It yields practical, natural schemes with provable support that has held up well in practice.

Cryptanalysts will often attack schemes assuming the hash functions in them are random, and a RO proof indicates security against such attacks.

Bottom line on RO paradigm:

- Use, but use with care
- Have a balanced perspective: understand both strengths and limitations
- Research it!

A counter-example

Let $\mathcal{AE}' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ be an IND-CPA asymmetric encryption scheme. We modify it to a RO model asymmetric encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, which

- Is IND-CPA secure in the RO model
- Not IND-CPA secure for any instantiation of the RO.

Programs are strings, and vice versa

Any (computable) function $H : \{0,1\}^* \to \{0,1\}^k$ has a string representation as a program $\langle H \rangle$.

Any string S can be parsed as the representation of a program P.

Counter-example

Given $\mathcal{AE}' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ we define $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ via

algorithm $\mathcal{E}^H_{pk}(M)$
Parse M as $\langle h \rangle$ where $h : \{0,1\}^* \to \{0,1\}^k$
$x \xleftarrow{\$} \{0,1\}^k$
if $H(x) = h(x)$ then return M
else return $\mathcal{E}'_{pk}(M)$

If H is a RO then for any $M = \langle h \rangle$

$$\Pr[H(x) = h(x)] \le \frac{q}{2^k}$$

for an adversary making q queries to H, and hence security is hardly affected.

Counter-example

Given $\mathcal{AE}' = (\mathcal{K}, \mathcal{E}', \mathcal{D}')$ we define $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ via

algorithm $\mathcal{E}^H_{pk}(M)$
Parse M as $\langle h \rangle$ where $h : \{0,1\}^* \to \{0,1\}^k$
$x \xleftarrow{\$} \{0,1\}^k$
if $H(x) = h(x)$ then return M
else return $\mathcal{E}'_{pk}(M)$

Now let $h : \{0,1\}^* \to \{0,1\}^k$ be any fixed function, and instantiate H with h. Then if we encrypt $M = \langle h \rangle$ we have

$$\mathcal{E}^h_{pk}(\langle h \rangle) = M$$

so the scheme is insecure.

RSA Math

Recall that $\varphi(N) = |\mathbf{Z}_N^*|$.

Claim: Suppose $e, d \in \mathbf{Z}_{\varphi(N)}^*$ satisfy $ed \equiv 1 \pmod{\varphi(N)}$. Then for any $x \in \mathbf{Z}_N^*$ we have

$$(x^e)^d \equiv x \pmod{N}$$

Proof:

$$(x^e)^d \equiv x^{ed \bmod \varphi(N)} \equiv x^1 \equiv x$$

modulo N

The RSA function

A modulus N and encryption exponent e define the RSA function $f : \mathbf{Z}_N^* \to \mathbf{Z}_N^*$ defined by

$$f(x) = x^e \mod N$$

for all $x \in \mathbf{Z}_N^*$.

A value $d \in Z_{\varphi(N)}^*$ satisfying $ed \equiv 1 \pmod{\varphi(N)}$ is called a decryption exponent.

Claim: The RSA function $f : \mathbf{Z}_N^* \to \mathbf{Z}_N^*$ is a permutation with inverse $f^{-1} : \mathbf{Z}_N^* \to \mathbf{Z}_N^*$ given by

$$f^{-1}(y) = y^d \mod N$$

Proof: For all $x \in \mathbf{Z}_N^*$ we have

$$f^{-1}(f(x)) \equiv (x^e)^d \equiv x \pmod{N}$$

by previous claim.

Example

Let $N = 15$. So

$$\begin{aligned}\mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= \end{aligned}$$

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4		
7		
8		
11		
13		
14		

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4	4	
7		
8		
11		
13		
14		

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4	4	
7	13	
8		
11		
13		
14		

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4	4	
7	13	
8	2	
11		
13		
14		

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1,2,4,7,8,11,13,14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1,3,5,7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4	4	
7	13	
8	2	
11	11	
13		
14		

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4	4	
7	13	
8	2	
11	11	
13	7	
14		

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	
2	8	
4	4	
7	13	
8	2	
11	11	
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	
4	4	
7	13	
8	2	
11	11	
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	
7	13	
8	2	
11	11	
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1,2,4,7,8,11,13,14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1,3,5,7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	4
7	13	
8	2	
11	11	
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1,2,4,7,8,11,13,14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1,3,5,7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	4
7	13	7
8	2	
11	11	
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	4
7	13	7
8	2	8
11	11	
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	4
7	13	7
8	2	8
11	11	11
13	7	
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1, 2, 4, 7, 8, 11, 13, 14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1, 3, 5, 7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	4
7	13	7
8	2	8
11	11	11
13	7	13
14	14	

Example

Let $N = 15$. So

$$\begin{aligned} \mathbf{Z}_N^* &= \{1,2,4,7,8,11,13,14\} \\ \varphi(N) &= 8 \\ \mathbf{Z}_{\varphi(N)}^* &= \{1,3,5,7\} \end{aligned}$$

Let $e = 3$ and $d = 3$. Then

$$ed \equiv 9 \equiv 1 \pmod 8$$

Let

$$\begin{aligned} f(x) &= x^3 \bmod 15 \\ g(y) &= y^3 \bmod 15 \end{aligned}$$

x	$f(x)$	$g(f(x))$
1	1	1
2	8	2
4	4	4
7	13	7
8	2	8
11	11	11
13	7	13
14	14	14

RSA usage

- $pk = N, e$; $sk = N, d$
- $\mathcal{E}_{pk}(x) = x^e \mod N = f(x)$
- $\mathcal{D}_{sk}(y) = y^d \mod N = f^{-1}(y)$

Security will rely on it being hard to compute f^{-1} without knowing d.

RSA is a trapdoor, one-way permutation:

- Easy to invert given trapdoor d
- Hard to invert given only N, e

RSA generators

An RSA generator with security parameter k is an algorithm $\mathcal{K}_{rsa}$ that returns N, p, q, e, d satisfying

- p, q are distinct odd primes
- $N = pq$ and is called the (RSA) modulus
- $|N| = k$, meaning $2^{k-1} \leq N \leq 2^k$
- $e \in \mathbf{Z}^*_{\varphi(N)}$ is called the encryption exponent
- $d \in \mathbf{Z}^*_{\varphi(N)}$ is called the decryption exponent
- $ed \equiv 1 \pmod{\varphi(N)}$

Plan

- Building RSA generators
- Basic RSA security
- Encryption with RSA

Some more math

Fact: If p, q are distinct primes and $N = pq$ then $\varphi(N) = (p-1)(q-1)$.

Proof:

$$\begin{aligned}
\varphi(N) &= |\{1, \ldots, N-1\}| - |\{ip : 1 \leq i \leq q-1\}| - |\{iq : 1 \leq i \leq p-1\}| \\
&= (N-1) - (q-1) - (p-1) \\
&= N - p - q + 1 \\
&= pq - p - q + 1 \\
&= (p-1)(q-1)
\end{aligned}$$

Example:

- $15 = 3 \cdot 5$
- $\mathbf{Z}_{15}^{*} = \{1, 2, 4, 7, 8, 11, 13, 14\}$
- $\varphi(15) = 8 = (3-1)(5-1)$

Recall

Given $\varphi(N)$ and $e \in \mathbf{Z}^*_{\varphi(N)}$, we can compute $d \in \mathbf{Z}^*_{\varphi(N)}$ satisfying $ed \equiv 1 \pmod{\varphi(N)}$ via

$$d \leftarrow \mathrm{MOD\text{-}INV}(e, \varphi(N)).$$

We have algorithms to efficiently test whether a number is prime, and a random number has a pretty good chance of being a prime.

Building RSA generators

Say we wish to have $e = 3$ (for efficiency). The generator $\mathcal{K}_{rsa}^{3}$ with (even) security parameter k:

repeat

$\quad p, q \stackrel{\$}{\leftarrow} \{2^{k/2-1}, \ldots, 2^{k/2} - 1\}$; $N \leftarrow pq$; $M \leftarrow (p-1)(q-1)$

until

$\quad N \geq 2^{k-1}$ and p, q are prime and $gcd(e, M) = 1$

$d \leftarrow \mathrm{MOD\text{-}INV}(e, M)$

return N, p, q, e, d

One-wayness of RSA

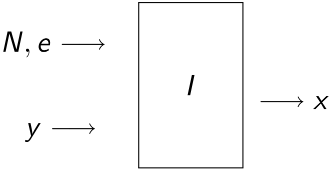
The following should be hard:

Given: N, e, y where $y = f(x) = x^e \mod N$

Find: x

Formalism picks x at random and generates N, e via an RSA generator.

ow-adversaries



wins if $x = f^{-1}(y)$, meaning $x^e \equiv y \pmod{N}$.

One-wayness of RSA, formally

Let K_{rsa} be a RSA generator and I an adversary.

Game $\mathrm{OW}_{K_{rsa}}$

procedure Initialize
$(N, p, q, e, d) \xleftarrow{\$} K_{rsa}$
$x \xleftarrow{\$} \mathbf{Z}_N^*$; $y \leftarrow x^e \mod N$
return N, e, y

procedure Finalize(x')
return $(x = x')$

The ow-advantage of I is

$$\mathbf{Adv}^{\mathrm{ow}}_{K_{rsa}}(I) = \Pr\left[\mathrm{OW}^{I}_{K_{rsa}} \Rightarrow \mathsf{true}\right]$$

Inverting RSA

Inverting RSA : given N, e, y find x such that $x^e \equiv y \pmod{N}$

Inverting RSA

Inverting RSA : given N, e, y find x such that $x^e \equiv y \pmod{N}$

EASY ↑ because $f^{-1}(y) = y^d \bmod N$

Know d

Inverting RSA

Inverting RSA : given N, e, y find x such that $x^e \equiv y \pmod{N}$

$\uparrow$ EASY — because $f^{-1}(y) = y^d \bmod N$

Know d

$\uparrow$ EASY — because $d = e^{-1} \bmod \varphi(N)$

Know $\varphi(N)$

Inverting RSA

Inverting RSA : given N, e, y find x such that $x^e \equiv y \pmod{N}$

$\uparrow$ EASY because $f^{-1}(y) = y^d \bmod N$

Know d

$\uparrow$ EASY because $d = e^{-1} \bmod \varphi(N)$

Know $\varphi(N)$

$\uparrow$ EASY because $\varphi(N) = (p-1)(q-1)$

Know p, q

Inverting RSA

Inverting RSA : given N, e, y find x such that $x^e \equiv y \pmod{N}$

$\uparrow$ EASY — because $f^{-1}(y) = y^d \bmod N$

Know d

$\uparrow$ EASY — because $d = e^{-1} \bmod \varphi(N)$

Know $\varphi(N)$

$\uparrow$ EASY — because $\varphi(N) = (p-1)(q-1)$

Know p, q

$\uparrow$?

Know N

Factoring Problem

Given: N where $N = pq$ and p, q are prime

Find: p, q

If we can factor we can invert RSA. We do not know whether the converse is true, meaning whether or not one can invert RSA without factoring.

A factoring algorithm

Alg $\mathrm{FACTOR}(N)$ // $N = pq$ where p, q are primes
for $i = 2, \ldots, \left\lceil \sqrt{N} \right\rceil$ do
  if $N \bmod i = 0$ then
    $p \leftarrow i$; $q \leftarrow N/i$; return p, q

This algorithm works but takes time

$$\mathcal{O}(\sqrt{N}) = \mathcal{O}(e^{0.5 \ln N})$$

which is prohibitive.

Factoring algorithms

Algorithm	**Time taken to factor N**
Naive	$O(e^{0.5 \ln N})$
Quadratic Sieve (QS)	$O(e^{c(\ln N)^{1/2}(\ln \ln N)^{1/2}})$
Number Field Sieve (NFS)	$O(e^{1.92(\ln N)^{1/3}(\ln \ln N)^{2/3}})$

Factoring records

Number	bit-length	Factorization	alg	MIPS years
RSA-400	400	1993	QS	830
RSA-428	428	1994	QS	5000
RSA-431	431	1996	NFS	1000
RSA-465	465	1999	NFS	2000
RSA-515	515	1999	NFS	8000
RSA-576	576	2003	NFS	

How big is big enough?

Current wisdom: For 80-bit security, use a 1024 bit RSA modulus

80-bit security: Factoring takes 2^{80} time.

Factorization of RSA-1024 seems out of reach at present.

Estimates vary, and for more security, longer moduli are recommended.

RSA: what to remember

The RSA function $f(x) = x^e \mod N$ is a trapdoor one way permutation:

- Easy forward: given N, e, x it is easy to compute $f(x)$
- Easy back with trapdoor: Given N, d and $y = f(x)$ it is easy to compute $x = f^{-1}(y) = y^d \mod N$
- Hard back without trapdoor: Given N, e and $y = f(x)$ it is hard to compute $x = f^{-1}(y)$

Plain-RSA encryption

The plain RSA asymmetric encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ associated to RSA generator K_{rsa} is

Alg $\mathcal{K}$ $(N, p, q, e, d) \xleftarrow{\$} K_{rsa}$ $pk \leftarrow (N, e)$ $sk \leftarrow (N, d)$ return (pk, sk)	Alg $\mathcal{E}_{pk}(M)$ $C \leftarrow M^e \mod N$ return C	Alg $\mathcal{D}_{sk}(C)$ $M \leftarrow C^d \mod N$ return M

The "easy-back with trapdoor" property implies

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(M)) = M$$

for all $M \in \mathbf{Z}_N^*$.

Plain-RSA encryption security

Alg $\mathcal{K}$	Alg $\mathcal{E}_{pk}(M)$	Alg $\mathcal{D}_{sk}(C)$
$(N, p, q, e, d) \xleftarrow{\$} K_{rsa}$	$C \leftarrow M^e \mod N$	$M \leftarrow C^d \mod N$
$pk \leftarrow (N, e)$	return C	return M
$sk \leftarrow (N, d)$		
return (pk, sk)		

Getting *sk* from *pk* involves factoring N.

Plain-RSA encryption security

Alg $\mathcal{K}$	Alg $\mathcal{E}_{pk}(M)$	Alg $\mathcal{D}_{sk}(C)$
$(N, p, q, e, d) \xleftarrow{\$} K_{rsa}$	$C \leftarrow M^e \mod N$	$M \leftarrow C^d \mod N$
$pk \leftarrow (N, e)$	return C	return M
$sk \leftarrow (N, d)$		
return (pk, sk)		

Alg $\mathcal{E}$ is deterministic so we can detect repeats and the scheme is not IND-CPA secure.

A message recovery attack

Suppose sender encrypts M and $M+1$ under public key $N, 3$. Adversary has

$$C_1 = M^3 \mod N \text{ and } C_2 = (M+1)^3 \mod N$$

Then modulo N we have

$$\frac{C_2 + 2C_1 - 1}{C_2 - C_1 + 2} \quad =$$

A message recovery attack

Suppose sender encrypts M and $M+1$ under public key $N, 3$. Adversary has

$$C_1 = M^3 \mod N \text{ and } C_2 = (M+1)^3 \mod N$$

Then modulo N we have

$$\begin{aligned} \frac{C_2 + 2C_1 - 1}{C_2 - C_1 + 2} &= \frac{(M+1)^3 + 2M^3 - 1}{(M+1)^3 - M^3 + 2} \\ &= \end{aligned}$$

A message recovery attack

Suppose sender encrypts M and $M+1$ under public key $N, 3$. Adversary has

$$C_1 = M^3 \mod N \text{ and } C_2 = (M+1)^3 \mod N$$

Then modulo N we have

$$\begin{aligned}
\frac{C_2 + 2C_1 - 1}{C_2 - C_1 + 2} &= \frac{(M+1)^3 + 2M^3 - 1}{(M+1)^3 - M^3 + 2} \\
&= \frac{(M^3 + 3M^2 + 3M + 1) + 2M^3 - 1}{(M^3 + 3M^2 + 3M + 1) - M^3 + 2} \\
&=
\end{aligned}$$

A message recovery attack

Suppose sender encrypts M and $M+1$ under public key $N, 3$. Adversary has

$$C_1 = M^3 \mod N \text{ and } C_2 = (M+1)^3 \mod N$$

Then modulo N we have

$$\begin{aligned}
\frac{C_2 + 2C_1 - 1}{C_2 - C_1 + 2} &= \frac{(M+1)^3 + 2M^3 - 1}{(M+1)^3 - M^3 + 2} \\
&= \frac{(M^3 + 3M^2 + 3M + 1) + 2M^3 - 1}{(M^3 + 3M^2 + 3M + 1) - M^3 + 2} \\
&= \frac{3M^3 + 3M^2 + 3M}{3M^2 + 3M + 3} =
\end{aligned}$$

A message recovery attack

Suppose sender encrypts M and $M+1$ under public key $N, 3$. Adversary has

$$C_1 = M^3 \mod N \text{ and } C_2 = (M+1)^3 \mod N$$

Then modulo N we have

$$\begin{aligned}
\frac{C_2 + 2C_1 - 1}{C_2 - C_1 + 2} &= \frac{(M+1)^3 + 2M^3 - 1}{(M+1)^3 - M^3 + 2} \\
&= \frac{(M^3 + 3M^2 + 3M + 1) + 2M^3 - 1}{(M^3 + 3M^2 + 3M + 1) - M^3 + 2} \\
&= \frac{3M^3 + 3M^2 + 3M}{3M^2 + 3M + 3} = \frac{M(3M^2 + 3M + 3)}{3M^2 + 3M + 3} = M
\end{aligned}$$

so adversary an recover M.

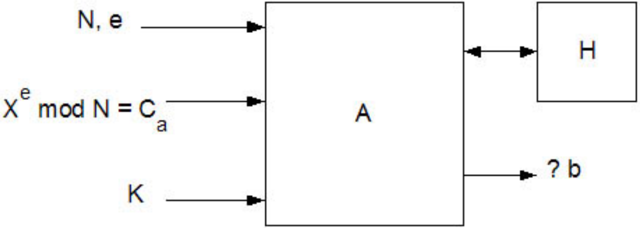
The RO model simple RSA KEM

The SRSA KEM $\mathcal{KEM} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ associated to RSA generator K_{rsa} is

Alg $\mathcal{K}$	Alg $\mathcal{E}^{H}_{pk}$	Alg $\mathcal{D}^{H}_{sk}(C_a)$
$(N, p, q, e, d) \xleftarrow{\$} K_{rsa}$	$x \xleftarrow{\$} \mathbf{Z}^*_N$	$x \leftarrow C_a^d \mod N$
$pk \leftarrow (N, e)$	$K \leftarrow H(x)$	$K \leftarrow H(x)$
$sk \leftarrow (N, d)$	$C_a \leftarrow x^e \mod N$	return K
return (pk, sk)	return K, C_a	

where $H : \{0,1\}^x \rightarrow \{0,1\}^k$ is a RO.

KEM security: Intuition



Here $x \xleftarrow{\$} \mathbf{Z}_N^*$; $b \xleftarrow{\$} \{0,1\}$; $K_0 \xleftarrow{\$} \{0,1\}^k$; $K_1 = H(x)$; $K \leftarrow K_b$;
If A queries x to H it can get $H(x)$ and test whether $K = H(x)$, but

- To find x it must invert RSA at C_a
- Without querying x it has 0 advantage in determining b
- If it queries x we can "see" this and invert RSA

SRSA KEM security: Result

Theorem: Let K_{rsa} be a RSA generator and $\mathcal{KEM} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ the associated SRSA KEM in the RO model. Let A be an ind-cpa adversary that makes 1 LR query and q queries to the RO H. Then there is a OW-adversary I such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{KEM}}(A) \leq \mathbf{Adv}^{\text{ow}}_{K_{rsa}}(I)$$

Furthermore the running time of I is about that of A plus the time for q RSA encryptions.

PKCS #1

Receiver keys: $pk = (N, e)$ and $sk = (N, d)$ where $n = |N|_8 = 128$

Alg $\mathcal{E}_{N,e}(M)$ // $|M|_8 \le n - 11$

$Pad \xleftarrow{\$} (\{0,1\}^8 - \{00\})^{n-m-3}$
$x \leftarrow 00||02||Pad||00||M$
$C \leftarrow x^e \bmod N$
return C

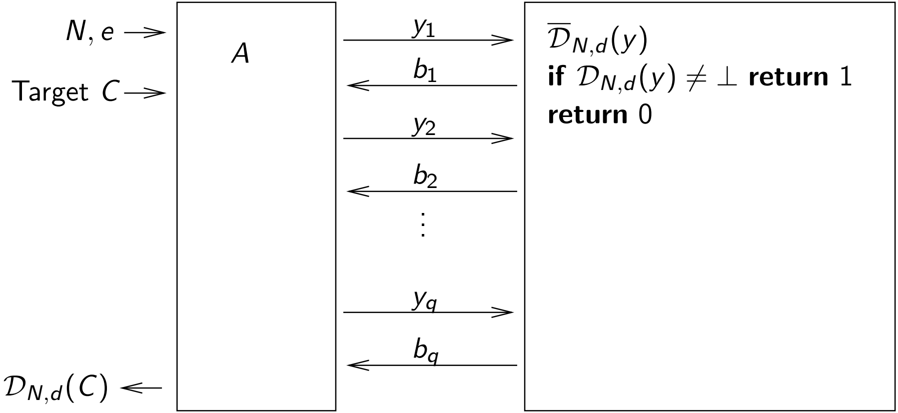
Alg $\mathcal{D}_{N,d}(C)$ // $C \in \mathbb{Z}_N^*$

$x \leftarrow C^d \bmod N$
$aa||bb||w \leftarrow x$
if $aa \neq 00$ or $bb \neq 02$ or $00 \notin w$ **then**
  return $\perp$
$Pad||00||M \leftarrow w$ where $00 \notin Pad$
return M

$x =$

00	02	*Pad*	00	*M*

Attack on PKCS #1 [Bl98]



The attack A succeeds in decrypting C after making $q \approx 1$ million clever queries to the box.

Attack on PKCS #1 and response

This is a (limited) chosen-ciphertext attack in which the oracle does not fully decrypt but indicates whether or not the decryption is valid.

The attack can be mounted on SSL.

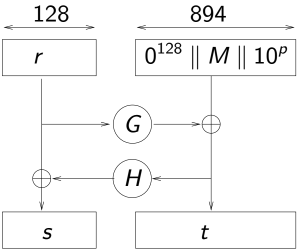
Use of an IND-CCA scheme would prevent the attack.

OAEP [BR94]

Receiver keys: $pk = (N, e)$ and $sk = (N, d)$ where $|N| = 1024$
ROs: G: $\{0,1\}^{128} \to \{0,1\}^{894}$ and H: $\{0,1\}^{894} \to \{0,1\}^{128}$

Algorithm $\mathcal{E}_{N,e}(M)$ $\quad$ // $|M| \leq 765$

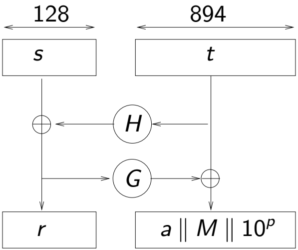
$r \xleftarrow{\$} \{0,1\}^{128}$; $p \leftarrow 765 - |M|$



$x \leftarrow s||t$
$C \leftarrow x^e \bmod N$
return C

Algorithm $\mathcal{D}_{N,d}(C)$ $\quad$ // $C \in \mathbb{Z}_N^*$

$x \leftarrow C^d \bmod N$
$s||t \leftarrow x$



if $a = 0^{128}$ **then return** M
else return $\perp$

OAEP security

If RSA is 1-way and H, G are random oracles then

- OAEP is IND-CPA secure [BR94]
- OAEP is IND-CCA secure [FOPS00]

RSA OAEP usage

Protocols:

- SSL ver. 2.0, 3.0 / TLS ver. 1.0, 1.1
- SSH ver 1.0, 2.0
- ...

Standards:

- RSA PKCS #1 versions 1.5, 2.0
- IEEE P1363
- NESSIE (Europe)
- CRYPTREC (Japan)
- ...