Lecture 12 – Public-Key Encryption Schemes

COSC-260 Codes and Ciphers
Adam O’Neill

Adapted from
http://cseweb.ucsd.edu/~mihir/cse107/
Today

We will see two types of public-key encryption schemes:
Today

We will see two types of public-key encryption schemes:

• Discrete-log based
Today

We will see two types of public-key encryption schemes:

• Discrete-log based
• RSA-based
Today

We will see two types of public-key encryption schemes:

- Discrete-log based
- RSA-based
Today

We will see two types of public-key encryption schemes:

• Discrete-log based
• RSA-based

Pay careful attention to efficiency, security model and assumptions needed to prove security.
DHIES

Let $G = \langle g \rangle$ be a cyclic group of order $m$ and $H: G \rightarrow \{0, 1\}^k$ a (public) hash function. The DHIES PKE scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined for messages $M \in \{0, 1\}^k$ via

$$
\begin{align*}
\text{Alg } & \mathcal{K} \\
x & \xleftarrow{\$} Z_m \\
X & \leftarrow g^x \\
\text{return } (X, x)
\end{align*}
\quad
\begin{align*}
\text{Alg } & \mathcal{E}_X(M) \\
y & \xleftarrow{\$} Z_m; Y \leftarrow g^y \\
K & \leftarrow X^y \\
W & \leftarrow H(K) \oplus M \\
\text{return } (Y, W)
\end{align*}
\quad
\begin{align*}
\text{Alg } & \mathcal{D}_X(Y, W) \\
K & = Y^x \\
M & \leftarrow H(K) \oplus W \\
\text{return } M
\end{align*}

Correct decryption is assured because $K = X^y = g^{xy} = Y^x$

**Note:** This is a simplified version of the actual scheme.
Which Hash Function to Use?

Our analysis will assume \( H \) is “perfect”

**Question:** What does this mean?

**Answer:** \( H \) will be modeled as a random oracle [BR93]
Random Oracle Model

A random oracle is a publicly-accessible random function

\[ W \xrightarrow{} H(W) \]

If \( H[W] = \perp \) then
\[ H[W] \leftarrow \{0, 1\}^k \]
Return \( H[W] \)

Oracle access to \( H \) provided to

- all scheme algorithms
- the adversary

The only access to \( H \) is oracle access.
Security of DHIES

The DHIES scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ associated to cyclic group $G = \langle g \rangle$ and (public) hash function $H$ can be proven IND-CPA assuming

- CDH is hard in $G$, and
- $H$ is a “random oracle,” meaning a “perfect” hash function.

In practice, $H(K)$ could be the first $k$ bits of the sequence

$$\text{SHA256}(0^8 || K) || \text{SHA256}(0^7 1 || K) || \cdots$$
ECIES

• DHIES scheme where $G$ is an appropriate elliptic curve group.
ECIES

• DHIES scheme where $G$ is an appropriate elliptic curve group.
• Attractive performance: ciphertext size 160 bits, encryption is 2 160-bit exponentiations.
ECIES

• DHIES scheme where $G$ is an appropriate elliptic curve group.
• Attractive performance: ciphertext size 160 bits, encryption is $2^{160}$-bit exponentiations.
• Widely standardized and used, prevailing scheme in next-generation Internet protocols.
A modulus $N$ and encryption exponent $e$ define the RSA function

$$f : \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$$

defined by

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

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

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

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

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

and

$$x^d = x \mod N$$

$$(x^e)^d = 1 \mod \varphi(N)$$

implies

$$x^{ed} \equiv 1 \pmod N$$

and

$$N = x \mod N$$
Let $N = 15$. So

\[
\mathbb{Z}_N^* = \{1, 2, 4, 7, 8, 11, 13, 14\}
\]

\[
\varphi(N) = 8
\]

\[
\mathbb{Z}_{\varphi(N)}^* = \{1, 3, 5, 7\}
\]

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

\[
ed \equiv 9 \equiv 1 \pmod{8}
\]

Let

\[
f(x) = x^3 \mod 15
\]

\[
g(y) = y^3 \mod 15
\]
Basic Idea for Usage

\[ pk = (N, e) \]

\[ RSA_{N, e}(x) = x^e \mod N = y \]

Alice

Bob

obtains \( y^d \) as the message

\[ sk = d \text{ s.t. } ed = 1 \mod \phi(N) \]
RSA Generators

An RSA generator with security parameter $k$ is an algorithm $K_{rsa}$ 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 $2^{k-1} \leq N \leq 2^k$
- $e \in \mathbb{Z}_{\varphi(N)}^*$ is called the encryption exponent
- $d \in \mathbb{Z}_{\varphi(N)}^*$ is called the decryption exponent
- $ed \equiv 1 \pmod{\varphi(N)}$

\[\text{Input: } 1^k\]
Building an RSA Generator

Typically one starts with a fixed $e$, e.g. $e = 3$ or $e = 2^{16}-1$.

To generate $p, q$, just pick $1/2$-bit numbers at random until
- $p$ is prime
- $q$ is prime
- $\gcd(e, (p-1)(q-1)) = 1$.

Find $d$ using extended GCD alg.
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.
Let $\mathcal{K}_{rsa}$ be a RSA generator and $I$ an adversary.

**Game $\text{OW}_{\mathcal{K}_{rsa}}$**

**procedure Initialize**

$(N, p, q, e, d) \leftarrow \mathcal{K}_{rsa}$

$x \leftarrow \mathcal{Z}_N^*; \ y \leftarrow x^e \mod N$

return $N, e, y$

**procedure Finalize($x'$)**

return $(x = x')$

The $\text{ow}$-advantage of $I$ is

$$\text{Adv}^{\text{ow}}_{\mathcal{K}_{rsa}}(I) = \Pr[\text{OW}_{\mathcal{K}_{rsa}}^I \Rightarrow \text{true}]$$
Inverting RSA

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

\[
\text{EASY because } f^{-1}(y) = y^d \pmod{N}
\]

- Know $d$

\[
\text{EASY because } d = e^{-1} \mod \varphi(N)
\]

- Know $\varphi(N)$

\[
\text{EASY because } \varphi(N) = (p - 1)(q - 1)
\]

- Know $p, q$

\[
\text{EASY because } f^{-1}(y) = y^d \pmod{N}
\]

- Know $N$
Factoring and RSA

Naive factoring alg:

Factor $(N)$ // want to output $P_1 q$ s.t. $N = p q$

For $i = 2$ to $N$

If $i$ divides $N$

Output $(i, N/i)$

Conjecture: For "good" RSA there is no better attack on one-wayness than factoring $N$. 

## Best Algorithms and Implication

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time taken to factor $N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive</td>
<td>$O(e^{0.5 \ln N})$</td>
</tr>
<tr>
<td>Quadratic Sieve (QS)</td>
<td>$O\left(e^{c(\ln N)^{1/2}(\ln \ln N)^{1/2}}\right)$</td>
</tr>
<tr>
<td>Number Field Sieve (NFS)</td>
<td>$O\left(e^{1.92(\ln N)^{1/3}(\ln \ln N)^{2/3}}\right)$</td>
</tr>
</tbody>
</table>

80-bit security $\Rightarrow$ 1029-bit modulus
The plain RSA PKE scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ associated to RSA generator $\mathcal{K}_{rsa}$ is

\[
\begin{align*}
\textbf{Alg } & \mathcal{K} \\
(N, p, q, e, d) & \leftarrow \mathcal{K}_{rsa} \\
\text{pk} & \leftarrow (N, e) \\
\text{sk} & \leftarrow (N, d) \\
\text{return } (\text{pk}, \text{sk})
\end{align*}
\]

\[
\begin{align*}
\textbf{Alg } & \mathcal{E}_{pk}(M) \\
C & \leftarrow M^e \text{ mod } N \\
\text{return } C
\end{align*}
\]

\[
\begin{align*}
\textbf{Alg } & \mathcal{D}_{sk}(C) \\
M & \leftarrow C^d \text{ mod } N \\
\text{return } M
\end{align*}
\]

The “easy-backwards with trapdoor” property implies

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

for all $M \in \mathbb{Z}_N^*$. 
Security Analysis

Is this scheme IND-CPA secure assuming $K_{rsa}$ is one-way?

Adversary $A(\mathcal{N}, e)$

$c \leftarrow LR(X_1, X_2)$ where $X_1, X_2$ are arbitrary distinct elements of $\mathbb{Z}_N^*$

$c' \leftarrow x_1^e \mod N$

If $c = c'$

ret 0

else ret 1
“Simple RSA” Encryption Scheme

The SRSA PKE scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ associated to RSA generator $\mathcal{K}_{rsa}$ and (public) hash function $H: \{0, 1\}^* \rightarrow \{0, 1\}^k$ encrypts $k$-bit messages via:

\[
\text{Alg } \mathcal{K} \\
(N, p, q, e, d) \leftarrow \mathcal{K}_{rsa} \\
pk \leftarrow (N, e) \\
\text{sk} \leftarrow (N, d) \\
\text{return } (pk, sk)
\]

\[
\text{Alg } \mathcal{E}_{N,e}(M) \\
x \leftarrow \mathbb{Z}_N^* \\
K \leftarrow H(x) \\
C_a \leftarrow x^e \mod N \\
\boxed{Cs \leftarrow K \oplus M} \\
\text{return } (C_a, Cs)
\]

\[
\text{Alg } \mathcal{E}_{N,d}(C_a, Cs) \\
x \leftarrow C_a^d \mod N \\
K \leftarrow H(x) \\
M \leftarrow Cs \oplus K \\
\text{return } M
\]

$Cs \leftarrow E_K(m)$ when $E$ is an authenticated encryption scheme.
The SRSA PKE scheme $\mathcal{A}\mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ associated to RSA generator $\mathcal{K}_{rsa}$ and (public) hash function $H$: $\{0, 1\}^* \rightarrow \{0, 1\}^k$ can be proven IND-CPA assuming

- $\mathcal{K}_{rsa}$ is one-way
- $H$ is a “random oracle,” meaning a “perfect” hash function.

In practice, $H(K)$ could be the first $k$ bits of the sequence

$$SHA256(0^8\|K)\|SHA256(0^71\|K)\| \cdots$$
Choose $r$ of appropriate length

$c \leftarrow (r || x)^e \mod N$

Non-trivial attacks show that length of $r$ needs to be very long.
Security Analysis

Is this scheme IND-CPA assuming \(K_{os}\) is one-way?
RSA-OAEP (PKCS #1 v2.1) [BR’94]

Receiver keys: $pk = (N, e)$ and $sk = (N, d)$ where $|N| = 1024$

Hash functions: $G$: $\{0, 1\}^{128} \rightarrow \{0, 1\}^{894}$ and $H$: $\{0, 1\}^{894} \rightarrow \{0, 1\}^{128}$

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

- $r \leftarrow \{0, 1\}^{128}$; $p \leftarrow 765 - |M|$
- $r \leftarrow 128$
- $10^p$  // $|M| \\ |
- $s \leftarrow 0^{128} \parallel M \parallel 10^p$
- $G$
- $H$
- $s$
- $t$
- $x \leftarrow s \parallel t$
- $C \leftarrow x^e \mod N$
- return $C$

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

- $x \leftarrow C^d \mod N$
- $s \parallel t \leftarrow x$
- $s \leftarrow 128$
- $10^p$
- $G$
- $H$
- $r$
- $a \parallel M \parallel 10^p$
- if $a = 0^{128}$ then return $M$
- else return ⊥
Attractive Features

• Same ciphertext length as RSA PKCS v1.5.
Attractive Features

• Same ciphertext length as RSA PKCS v1.5.
• But has provable security guarantees:
Attractive Features

• Same ciphertext length as RSA PKCS v1.5.
• But has provable security guarantees:
  • IND-CPA in the RO model assuming RSA is one-way [BR’94].
Attractive Features

• Same ciphertext length as RSA PKCS v1.5.

• But has provable security guarantees:
  • IND-CPA in the RO model assuming RSA is one-way [BR’94].
  • IND-CCA in the RO model assuming RSA is one-way [FOPS’00].
Attractive Features

• Same ciphertext length as RSA PKCS v1.5.
• But has provable security guarantees:
  • IND-CPA in the RO model assuming RSA is one-way [BR’94].
  • IND-CCA in the RO model assuming RSA is one-way [FOPS’00].
  • IND-CPA in the standard model assuming RSA is “lossy” [KOS’10].
Careful in Practice

• Attacks are possible if $d$ is too small, timing information leaks, etc. (cf. “Twenty Years of Attacks on RSA” by Dan Boneh).
Careful in Practice

• Attacks are possible if *d is too small*, *timing information leaks*, *etc.* (cf. “Twenty Years of Attacks on RSA” by Dan Boneh).

• Lenstra *et al.* recently found many keys *share a common divisor* due to buggy randomness!!
Careful in Practice

- Attacks are possible if $d$ is too small, timing information leaks, etc. (cf. “Twenty Years of Attacks on RSA” by Dan Boneh).
- Lenstra et al. recently found many keys share a common divisor due to buggy randomness!!
- Use open-source, publicly scrutinized implementations!
PKE Summary
PKE Summary