Public Key Encryption from trapdoor permutations

Public key encryption: definitions and security
Public key encryption

Bob: generates (PK, SK) and gives PK to Alice
Applications

Session setup  (for now, only eavesdropping security)

Non-interactive applications:  (e.g. Email)
• Bob sends email to Alice encrypted using  $pk_{alice}$
• Note: Bob needs $pk_{alice}$  (public key management)
**Public key encryption**

**Def:** A public-key encryption system is a triple of algs. \((G, E, D)\)

- **G:** Randomized alg. outputs a key pair \((pk, sk)\)
- **E:** Randomized alg. that takes \(m \in M\) and outputs \(c \in C\)
- **D:** Deterministic alg. that takes \(c \in C\) and outputs \(m \in M\) or \(\perp\)

**Consistency:** \(\forall (pk, sk)\) output by G:

\[\forall m \in M: \quad D(sk, E(pk, m)) = m\]
For \( b=0,1 \) define experiments \( \text{EXP}(0) \) and \( \text{EXP}(1) \) as:

\[
\text{Def: } E = (G,E,D) \text{ is sem. secure (a.k.a IND-CPA) if for all efficient } A:
\]

\[
\text{Adv}_{\text{SS}}[A, E] = \left| \Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1] \right| < \text{negligible}
\]
Recall: for symmetric ciphers we had two security notions:

- One-time security and many-time security (CPA)
- We showed that one-time security $\not\Rightarrow$ many-time security

For public key encryption:

- One-time security $\Rightarrow$ many-time security (CPA)
  (follows from the fact that attacker can encrypt by himself)
- Public key encryption must be randomized
Security against active attacks

What if attacker can tamper with ciphertext?

Attacker is given decryption of msgs that start with “to: attacker”
(pub-key) Chosen Ciphertext Security: definition

\[ E = (G, E, D) \] public-key enc. over \((M, C)\). For \(b=0,1\) define \(\text{EXP}(b)\):

1. **Chal.**
   - \((pk, sk) \leftarrow G()\)

2. **pk**
   - \(c_i \in C\)
   - \(m_i \leftarrow D(sk, c_i)\)

3. **challenge:**
   - \(m_0, m_1 \in M: |m_0| = |m_1|\)
   - \(c \leftarrow E(pk, m_b)\)

4. **CCA phase 1:**
   - \(c_i \in C\)
   - \(c_i \neq c\)

5. **CCA phase 2:**
   - \(c_i \in C\)
   - \(m_i \leftarrow D(sk, c_i)\)

6. **Adv. A**
   - \(b' \in \{0, 1\}\)
**Chosen ciphertext security: definition**

**Def:** $E$ is CCA secure (a.k.a IND-CCA) if for all efficient $A$:

$$\text{Adv}_{\text{CCA}} [A, E] = \left| \Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1] \right|$$

is negligible.

**Example:** Suppose

$$\text{Chal.} \quad (pk, sk) \leftarrow G()$$

$$\text{chal.:} \quad (\text{to: alice, 0}), (\text{to: alice, 1})$$

$$c \leftarrow E(pk, m_b)$$

CCA phase 2: $c' \neq c$

$$m' \leftarrow D(sk, c')$$
Active attacks: symmetric vs. pub-key

Recall: secure symmetric cipher provides **authenticated encryption**

[ chosen plaintext security & ciphertext integrity ]

- Roughly speaking: **attacker cannot create new ciphertexts**
- Implies security against chosen ciphertext attacks

In public-key settings:

- Attacker **can** create new ciphertexts using \( pk \)
- So instead: we directly require chosen ciphertext security
This and next module:

constructing CCA secure pub-key systems

End of Segment
Goal: construct chosen-ciphertext secure public-key encryption
**Trapdoor functions (TDF)**

**Def:** a trapdoor func. \( X \rightarrow Y \) is a triple of efficient algs. \((G, F, F^{-1})\)

- \(G()\): randomized alg. outputs a key pair \((pk, sk)\)
- \(F(pk, \cdot)\): det. alg. that defines a function \( X \rightarrow Y \)
- \(F^{-1}(sk, \cdot)\): defines a function \( Y \rightarrow X \) that inverts \( F(pk, \cdot) \)

More precisely: \( \forall (pk, sk) \) output by \( G \)

\[ \forall x \in X: \quad F^{-1}(sk, F(pk, x)) = x \]
Secure Trapdoor Functions (TDFs)

$(G, F, F^{-1})$ is secure if $F(pk, \cdot)$ is a “one-way” function:

can be evaluated, but cannot be inverted without $sk$

**Def:** $(G, F, F^{-1})$ is a secure TDF if for all efficient $A$:

$$\text{Adv}_{\text{OW}}[A, F] = \Pr[x = x'] < \text{negligible}$$
Public-key encryption from TDFs

- \((G, F, F^{-1})\): secure TDF \(X \rightarrow Y\)
- \((E_s, D_s)\): symmetric auth. encryption defined over \((K,M,C)\)
- \(H: X \rightarrow K\): a hash function

We construct a pub-key enc. system \((G, E, D)\):

Key generation \(G\): same as \(G\) for TDF
Public-key encryption from TDFs

- \((G, F, F^{-1})\): secure TDF \(X \rightarrow Y\)
- \((E_s, D_s)\): symmetric auth. encryption defined over \((K, M, C)\)
- \(H: X \rightarrow K\) a hash function

\[
\begin{align*}
E(pk, m) : & \quad x \leftarrow^R X, \quad y \leftarrow F(pk, x) \\
& \quad k \leftarrow H(x), \quad c \leftarrow E_s(k, m) \\
& \quad \text{output } (y, c)
\end{align*}
\]

\[
\begin{align*}
D(sk, (y, c)) : & \quad x \leftarrow F^{-1}(sk, y), \\
& \quad k \leftarrow H(x), \quad m \leftarrow D_s(k, c) \\
& \quad \text{output } m
\end{align*}
\]
Security Theorem:

If \((G, F, F^{-1})\) is a secure TDF, \((E_s, D_s)\) provides auth. enc. and \(H: X \rightarrow K\) is a “random oracle” then \((G, E, D)\) is CCA\(^r\)o secure.
Incorrect use of a Trapdoor Function (TDF)

**Never** encrypt by applying $F$ directly to plaintext:

$$E(\text{pk}, m) : \quad \text{output } c \leftarrow F(\text{pk}, m)$$

$$D(\text{sk}, c) : \quad \text{output } F^{-1}(\text{sk}, c)$$

Problems:

- Deterministic: cannot be semantically secure!!
- Many attacks exist (next segment)
Next step: construct a TDF

End of Segment
Public Key Encryption from trapdoor permutations

The RSA trapdoor permutation
Review: trapdoor permutations

Three algorithms: \((G, F, F^{-1})\)

- **G**: outputs \(pk, sk\). \(pk\) defines a function \(F(pk, \cdot): X \rightarrow X\)
- **F(pk, x)**: evaluates the function at \(x\)
- **\(F^{-1}(sk, y)\)**: inverts the function at \(y\) using \(sk\)

**Secure** trapdoor permutation:

The function \(F(pk, \cdot)\) is one-way without the trapdoor \(sk\)
Review: arithmetic mod composites

Let \( N = p \cdot q \) where \( p, q \) are prime

\[ Z_N = \{0, 1, 2, \ldots, N-1\} ; \quad (Z_N)^* = \{\text{invertible elements in } Z_N\} \]

**Facts:** \( x \in Z_N \) is invertible \( \iff \) \( \gcd(x, N) = 1 \)

- Number of elements in \( (Z_N)^* \) is \( \varphi(N) = (p-1)(q-1) = N-p-q+1 \)

**Euler’s thm:** \( \forall x \in (Z_N)^* : \quad x^{\varphi(N)} = 1 \)
The RSA trapdoor permutation


Very widely used:

- SSL/TLS: certificates and key-exchange
- Secure e-mail and file systems
  ... many others
The RSA trapdoor permutation

\( G() \): choose random primes \( p, q \approx 1024 \text{ bits} \). Set \( N = pq \).

choose integers \( e, d \) s.t. \( e \cdot d = 1 \pmod{\varphi(N)} \)

output \( pk = (N, e) \), \( sk = (N, d) \)

\[ F^{-1}(sk, y) = y^d \quad ; \quad y^d = RSA(x)^d = x^{ed} = x^{k\varphi(N)+1} = (x^{\varphi(N)})^k \cdot x = x \]
The RSA assumption

RSA assumption: RSA is one-way permutation

For all efficient algs. A:

\[ \Pr \left[ A(N,e,y) = y^{1/e} \right] < \text{negligible} \]

where \( p,q \xleftarrow{\text{R}} \text{n-bit primes}, \ N \leftarrow pq, \ y \xleftarrow{\text{R}} \mathbb{Z}_N^* \)
Review: RSA pub-key encryption (ISO std)

$(E_s, D_s)$: symmetric enc. scheme providing auth. encryption.

$H: Z_N \rightarrow K$ where $K$ is key space of $(E_s, D_s)$

- **$G()$:** generate RSA params: $pk = (N,e)$, $sk = (N,d)$

- **$E(pk, m)$:**
  1. choose random $x$ in $Z_N$
  2. $y \leftarrow RSA(x) = x^e$, $k \leftarrow H(x)$
  3. output $(y, E_s(k,m))$

- **$D(sk, (y, c))$:** output $D_s(H(RSA^{-1}(y)), c)$
Textbook RSA is insecure

Textbook RSA encryption:

- public key: $\langle N, e \rangle$
- secret key: $\langle N, d \rangle$

Encrypt: $c \leftarrow m^e \quad \text{(in } \mathbb{Z}_N\text{)}$

Decrypt: $c^d \rightarrow m$

Insecure cryptosystem !!

- Is not semantically secure and many attacks exist

$\Rightarrow$ The RSA trapdoor permutation is not an encryption scheme!
Suppose $k$ is 64 bits: $k \in \{0,\ldots,2^{64}\}$. Eve sees: $c = k^e$ in $\mathbb{Z}_N$

If $k = k_1 \cdot k_2$ where $k_1, k_2 < 2^{34}$ (prob. $\approx 20\%$) then $c/k_1^e = k_2^e$ in $\mathbb{Z}_N$

Step 1: build table: $c/1^e, c/2^e, c/3^e, \ldots, c/2^{34}e$. time: $2^{34}$

Step 2: for $k_2 = 0,\ldots, 2^{34}$ test if $k_2^e$ is in table. time: $2^{34}$

Output matching $(k_1, k_2)$. Total attack time: $\approx 2^{40} \ll 2^{64}$
End of Segment