Security Notions for Public-Key Encryption Schemes

Giuseppe Persiano
Dip. Informatica ed Appl.
Università di Salerno
pino.persiano@unisa.it

Abstract

An introducion to the notions of security for public-key encryption schemes prepared for FOSAD 01

Definition 1
A **public-key encryption scheme** *is a triple of* **efficient** *algorithms* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *such that*

1. *For all constants* c*, for all sufficiently large* n *and all* **cleartexts** $m \in \{0,1\}^n$

$$\mathbf{Pr}[(pk, sk) \leftarrow \mathcal{K}(1^n); ct \leftarrow \mathcal{E}(pk, m) : \mathcal{D}(sk, ct) = m] > 1 - \frac{1}{n^c}$$

Definition 2 (Informal One-way security)
A public-key encryption scheme is **one-way secure** *if the probability that an* **efficient** *adversary is able to invert a* **ciphertext** *is* **very small.**

Definition 3 (One-way security)
A public-key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *is* **one-way secure** *if for all probabilistic polynomial time algorithms* $\mathcal{A}$ *for all constants* c *and for all sufficiently large* n

$$\mathbf{Pr}[(pk, sk) \leftarrow \mathcal{K}(1^n); m \leftarrow \{0,1\}^n; ct \leftarrow \mathcal{E}(pk, m) : \mathcal{A}(pk, ct) = m] < \frac{1}{n^c}$$

The RSA public-key cryptosystem

Key Generation

Input: security parameter k

1. pick two random primes p, q of length k;
2. set $N = p \cdot q$ and let $\phi(N) = (p - 1) \cdot (q - 1)$;
3. pick random e s.t. $(e, \phi(N)) = 1$;
4. set $pk = (e, N)$ and $sk = (d, N)$;

Encryption

Input: public key $pk = (e, N)$, cleartext $m \in Z_N^*$;

1. set $ct = m^e \text{mod } N$;

Decryption:

Input: secret key $pk = (e, N)$, ciphertext $ct \in Z_N^*$;

1. set $m' = ct^d \text{mod } N$;

What one would like to be able to show

Theorem 1 *The RSA public-key cryptosystem is 1-way secure.*

Informal statement: no polynomial-time algorithm can invert RSA except with negligible probabilty.

What we can actually do

Conjecture 1 *The RSA public-key cryptosystem is 1-way secure.*

Conjecture 2 *There exists a 1-way secure public-key cryptosystem.*

1-way security is not enough

1. only guarantees the whole cleartext cannot be recovered without the secret key;

   does not rule, say, that it is possible to recover half of the bits of the cleartexts by looking at the ciphertext without having access to the secret key;

2. does not say anything for sparse message spaces;

   for example: encrypting one bit, encrypting italian sentences, ...

Definition 4 (Indistinguishability for bits)
A public-key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *is* **1-bit secure in the indistinguishable sense** *if for all probabilistic polynomial time algorithms* $\mathcal{A}$ *for all constants* c *and for all sufficiently large* n

$$Succ^{\mathcal{A}}_{Dist}(n) < \frac{1}{2} + \frac{1}{n^c}$$

where

$$\begin{aligned} Succ^{\mathcal{A}}_{Dist}(n) = \mathbf{Pr}[\ & (pk, sk) \leftarrow \mathcal{K}(1^n); b \leftarrow \{0, 1\}; \\ & ct \leftarrow \mathcal{E}(pk, b) : \mathcal{A}(pk, ct) = b] \end{aligned}$$

Definition 5 (Hard-core Predicate)
Let $B = \{B_n\}$ *where* $B_n : \{0,1\}^n \rightarrow \{0,1\}$ *be a family of efficiently computable predicate.*

B *is a* **hard-core predicate** *for* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *if for all probabilistic polynomial time algorithms* $\mathcal{A}$ *for all constants* c *and for all sufficiently large* n

$$Succ_B^{\mathcal{A}}(n) < \frac{1}{2} + \frac{1}{n^c}$$

where

$$\begin{aligned} Succ_B^{\mathcal{A}}(n) = \mathbf{Pr}[\ & (pk, sk) \leftarrow \mathcal{K}(1^n); m \leftarrow \{0,1\}^n; \\ & ct \leftarrow \mathcal{E}(pk, b) : \mathcal{A}(pk, ct) = B_n(m)] \end{aligned}$$

A Hard Core Predicate for RSA

Theorem 2 (Alexi, Chor, Goldreich, Schnorr)

If

there exist a constant $c > 0$ *and a probabilistic polynomial-time algorithm that, on input an RSA public key* (e, N) *and a ciphertext ct, correctly computes the* least significant bit *of the cleartext relative to ct with probability greater than* $1/2 + |N|^{-c}$

then

there exists a polynomial-time algorithm for inverting the RSA function.

A 1-bit Secure Cryptosystem based on RSA

Key Generation:
Input: security parameter k;

1. pick two random primes p, q of length k;
2. set $N = p \cdot q$ and let $\phi(N) = (p-1) \cdot (q-1)$;
3. pick random e s.t. $(e, \phi(N)) = 1$;
4. set $pk = (e, N)$ and $sk = (d, N)$;

Encryption:
Input: public key $pk = (e, N)$, cleartext $b \in \{0, 1\}$;

1. pick a random $m \in Z_N^*$ such that $\mathrm{LSB}(m) = b$;
2. set $ct = m^e \bmod N$;

A 1-bit Secure Cryptosystem based on RSA

Decryption:

Input: secret key $sk = (d, N)$, ciphertext $ct \in Z_N^*$;

1. set $m' = ct^d \bmod N$;
2. output LSB(m');

Definition 6 (Indistinguishability)
A public-key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *is* **secure in the indistinguishable sense** *if for all probabilistic polynomial time algorithms* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ *for all constants* c *and for all sufficiently large* n

$$Succ^{\mathcal{A}}_{Dist}(n) < \frac{1}{2} + \frac{1}{n^c}$$

where

$$\begin{aligned} Succ^{\mathcal{A}}_{Dist}(n) = \mathbf{Pr}[\ & (pk, sk) \leftarrow \mathcal{K}(1^n); (m_0, m_1, aux) \leftarrow \mathcal{A}_0(pk); \\ & b \leftarrow \{0,1\}; ct \leftarrow \mathcal{E}(pk, m_b) : \\ & \mathcal{A}_1(pk, m_0, m_1, aux, ct) = m_b \wedge |m_0| = |m_1|] \end{aligned}$$

From 1 bit to many bits

Let $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a 1-bit secure cryptosystem in the indistinguishable sense. Then the cryptosystem $(\mathcal{K}, t\mathcal{E}, t\mathcal{D})$ is a t-bit secure cryptosystem in the indistinguishable sense.

Algorithm t$\mathcal{E}$

Input: $pk, m = m_0 \circ \cdots \circ m_{t-1}$;

1. **for** $i = 0$ **to** $t - 1$
   (a) $ct_i = \mathcal{E}(pk, m_i)$;
2. **Output:** $ct = ct_0 \circ \cdots \circ ct_{t-1}$

Algorithm t$\mathcal{D}$

Input: $sk, ct = ct_0 \circ \cdots \circ ct_{t-1}$;

1. **for** $i = 0$ **to** $t - 1$
   (a) $m_i = \mathcal{D}(sk, ct_i)$;
2. **Output:** $m = m_0 \circ \cdots \circ m_{t-1}$

Definition 7 (Semantic Security)
A public-key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *is* **semantically secure** *if for all predicates* p *and all probabilistic polynomial time algorithms* $\mathcal{A}$ *for all constants* c *and for all sufficiently large* n

$$Succ^{\mathcal{A}}_{Dist}(n) < p^*(n) + \frac{1}{n^c}$$

where

$$\begin{aligned} Succ^{\mathcal{A}}_{Dist}(n) = \mathbf{Pr}[\ & (pk, sk) \leftarrow \mathcal{K}(1^n); m \leftarrow \{0,1\}^n; \\ & ct \leftarrow \mathcal{E}(pk, m) : \mathcal{A}(pk, ct) = p(m)] \end{aligned}$$

and

$$\begin{aligned} p^*(n) = \mathbf{Pr}[\ & (pk, sk) \leftarrow \mathcal{K}(1^n); m \leftarrow \{0,1\}^n; \\ & ct \leftarrow \mathcal{E}(pk, m) : \mathcal{A}(pk) = p(m)]. \end{aligned}$$

Theorem 3 (Goldwasser, Micali)
A cryptosystem is semantically secure if and only if it is secure in the indistinguishable sense.

Definition 8 (Chosen Ciphertext Attack)
A public-key encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *is* **secure in the indistinguishable sense** *with respect to chosen-ciphertext attacks (CCA secure) if for all pairs of oracle probabilistic polynomial time algorithms* $\mathcal{A} = (A_1, A_2)$ *for all constants* c *and for all sufficiently large* n

$$Succ^{\mathcal{A}}_{CCA}(n) < \frac{1}{2} + \frac{1}{n^c}$$

where

$$\begin{aligned} Succ^{\mathcal{A}}_{CCA}(n) = \mathbf{Pr}[\ & (pk, sk) \leftarrow \mathcal{K}(1^n); \texttt{state} \leftarrow A_1^{\mathcal{D}}(pk); \\ & b \leftarrow \{0,1\}^n; ct \leftarrow \mathcal{E}(pk, b) : A_2(pk, ct, \texttt{state}) = b] \end{aligned}$$

Computing Square Roots in Z_n^* with prime n

Given $x \in Z_n^*$ *compute (if it exists)*
$w \in Z_n^*$ *s.t.* $w^2 \equiv x \pmod{n}$.

1. half of the elements of Z_n^* are *quadratic residues* (*i.e.*, have a square root);
2. x is a quadratic residue iff
$$x^{\frac{n-1}{2}} = 1;$$
3. each quadratic residue has two square roots w and $n - w$;
4. there exists a randomized polynomial time algorithm for computing square roots in Z_n^*;

Computing Square Roots in Z_n^* with n product of two primes

1. one quarter of the elements of Z_n^* are quadratic residues;
2. x is a quadratic residue iff x is a quadratic residue modulo p and modulo q;
3. each quadratic residue has four square roots w_1 and $n - w_1$, w_2 and $n - w_2$;
4. computing square roots is easy if factorization of n is known;
5. if there exists an efficient algorithm that on input x and n outputs a square root of n modulo x then there exists an efficient algorithm that factors x;

Cryptosystem based on Squaring (M. O. Rabin)

Key Generation:

Input: security parameter k

1. pick two random primes p, q of length k;

2. set $pk = (n)$ and $sk = (p, q)$;

Encryption:

Input: public key $pk = (n)$, cleartext $m \in Z_n^*$;

1. set $ct = m^2 \text{mod } n$;

Decryption:

Input: secret key $sk = (p, q)$, ciphertext $ct \in Z_n^*$;

1. compute a square root of ct modulo n;

Nota bene: squaring is not 1-to-1 but there are ways around it.

Equivalence of Extracting Square Roots and Factoring

Suppose that

1. Let n be an integer product of two odd distinct primes of unknown factorization;
2. Let x be a quadratic residue modulo n;
3. x has four square roots $w_1, w_2 < n/2$ and their opposites.
4. SQ is an algorithm that given x and n outputs either w_1 or w_2.

then the following algorithm factors

Factoring Algorithm(n)

1. pick a random $w < n/2$ and compute $x \equiv w^2 \pmod{n}$;
2. run SQ on input x and n and receive a square root w' of x;
3. if $w = w'$ then goto 1;
4. GCD$(w - w', n)$ is a proper factor of n.

Constructing CCA secure cryptosystems

1. Naor & Yung
   - Use two semantically secure cryptosystems E_1 and E_2.
   - Encrypt message m twice: once using E_1 and once using E_2 obtaining two ciphertexts ct_0, ct_1.
   - Add a zero-knowledge proof that the two ciphertexts encrypt the same message.
2. DeSantis & P
   - Use one semantically secure cryptosystems E_1.
   - Encryption contains zero-knowledge proof that sender knows cleartext.

Other topics

1. Non Malleability.
2. Stronger attack: CCA2.