Provable Security for Public Key Schemes

Jürgen Ecker

FH Hagenberg, Computer- und Mediensicherheit
juergen.ecker@fh-hagenberg.at

Mathematik 2005, Klagenfurt
Outline

1. Introduction
   - Examples

2. Defining Security
   - What the attacker can
   - What the attacker wants

3. Relations among security properties
   - A concise definition of IND

4. Security of cryptosystems
   - Models
   - RSA
   - ElGamal
Examples
One-wayness
Examples
One-wayness
Examples
One-wayness

What’s inside the envelope?
Examples
Indistinguishability
Examples
Indistinguishability
Examples
Indistinguishability

Which of the two messages?
"'i love you'"? "'can you repare my toilet'"?
Examples
Non-malleability

\[ i \text{ love you} \]
Examples
Non-malleability

I don’t care what she writes, I only want to change it to something related.

→

i love you

→

I don’t care what she writes, I only want to change it to something related.
Examples
Non-malleability

I don’t care what she writes, I only want to change it to something related.
Examples
Non-malleability

I don’t care what she writes, I only want to change it to something related.
Examples
Non-malleability

I don’t care what she writes, I only want to change it to something related.

related!
Modelling the attacker
Information available to the attacker

- CPA (chosen plaintext attack): choose a plaintext and see its encryption.
- VCA (validity checking attack): choose a text and see if it is a valid ciphertext
- PCA (plaintext checking attack): choose a plaintext-ciphertext pair and see if they correspond.
- CCA1 (chosen ciphertext attack, lunchtime attack): choose a ciphertext and see the corresponding plaintext.
- CCA2 (adaptive chosen ciphertext attack): CCA1 after having seen the challenge ciphertext.

The attacker can run probabilistic polynomial time algorithms.
Modelling the attacker
Information available to the attacker

- CPA (chosen plaintext attack): choose a plaintext and see its encryption.
- VCA (validity checking attack): choose a text and see if it is a valid ciphertext
- PCA (plaintext checking attack): choose a plaintext-ciphertext pair and see if they correspond.
- CCA1 (chosen ciphertext attack, lunchtime attack): choose a ciphertext and see the corresponding plaintext.
- CCA2 (adaptive chosen ciphertext attack): CCA1 after having seen the challenge ciphertext.

The attacker can run probabilistic polynomial time algorithms.
Modelling the attacker
Information available to the attacker

- CPA (chosen plaintext attack): choose a plaintext and see its encryption.
- VCA (validity checking attack): choose a text and see if it is a valid ciphertext.
- PCA (plaintext checking attack): choose a plaintext-ciphertext pair and see if they correspond.
- CCA1 (chosen ciphertext attack, lunchtime attack): choose a ciphertext and see the corresponding plaintext.
- CCA2 (adaptive chosen ciphertext attack): CCA1 after having seen the challenge ciphertext.

The attacker can run probabilistic polynomial time algorithms.
Modelling the attacker
Information available to the attacker

- CPA (chosen plaintext attack): choose a plaintext and see its encryption.
- VCA (validity checking attack): choose a text and see if it is a valid ciphertext.
- PCA (plaintext checking attack): choose a plaintext-ciphertext pair and see if they correspond.
- CCA1 (chosen ciphertext attack, lunchtime attack): choose a ciphertext and see the corresponding plaintext.
- CCA2 (adaptive chosen ciphertext attack): CCA1 after having seen the challenge ciphertext.

The attacker can run probabilistic polynomial time algorithms.
Modelling the attacker

Information available to the attacker

- CPA (chosen plaintext attack): choose a plaintext and see its encryption.
- VCA (validity checking attack): choose a text and see if it is a valid ciphertext.
- PCA (plaintext checking attack): choose a plaintext-ciphertext pair and see if they correspond.
- CCA1 (chosen ciphertext attack, lunchtime attack): choose a ciphertext and see the corresponding plaintext.
- CCA2 (adaptive chosen ciphertext attack): CCA1 after having seen the challenge ciphertext.

The attacker can run probabilistic polynomial time algorithms.
Modelling the attacker
Information available to the attacker

- CPA (chosen plaintext attack): choose a plaintext and see its encryption.
- VCA (validity checking attack): choose a text and see if it is a valid ciphertext
- PCA (plaintext checking attack): choose a plaintext-ciphertext pair and see if they correspond.
- CCA1 (chosen ciphertext attack, lunchtime attack): choose a ciphertext and see the corresponding plaintext.
- CCA2 (adaptive chosen ciphertext attack): CCA1 after having seen the challenge ciphertext.

The attacker can run probabilistic polynomial time algorithms.
Modelling the attacker
Goals of an attacker

- OW (one-wayness): given a ciphertext find the corresponding plaintext.
- SEM.SEC (semantic security, polynomial security): given a ciphertext find partial information about the plaintext.
- IND (indistinguishability): distinguish between encryptions of two chosen plaintexts.
- SEM.SEC.REL: given a ciphertext find a plaintext which is in some way related to the plaintext corresponding to the given ciphertext.
- NM (non-malleability): given a ciphertext find another ciphertext such that the two ciphertexts are related.
Modelling the attacker
Goals of an attacker

- **OW** (one-wayness): given a ciphertext find the corresponding plaintext.
- **SEM.SEC** (semantic security, polynomial security): given a ciphertext find partial information about the plaintext.
- **IND** (indistinguishability): distinguish between encryptions of two chosen plaintexts.
- **SEM.SEC.REL**: given a ciphertext find a plaintext which is in some way related to the plaintext corresponding to the given ciphertext
- **NM** (non-malleability): given a ciphertext find another ciphertext such that the two ciphertexts are related.
**Modelling the attacker**

**Goals of an attacker**

- **OW (one-wayness):** given a ciphertext find the corresponding plaintext.
- **SEM.SEC (semantic security, polynomial security):** given a ciphertext find partial information about the plaintext.
- **IND (indistinguishability):** distinguish between encryptions of two chosen plaintexts.
- **SEM.SEC.REL:** given a ciphertext find a plaintext which is in some way related to the plaintext corresponding to the given ciphertext.
- **NM (non-malleability):** given a ciphertext find another ciphertext such that the two ciphertexts are related.
Modelling the attacker

Goals of an attacker

- OW (one-wayness): given a ciphertext find the corresponding plaintext.
- SEM.SEC (semantic security, polynomial security): given a ciphertext find partial information about the plaintext.
- IND (indistinguishability): distinguish between encryptions of two chosen plaintexts.
- SEM.SEC.REL: given a ciphertext find a plaintext which is in some way related to the plaintext corresponding to the given ciphertext.
- NM (non-malleability): given a ciphertext find another ciphertext such that the two ciphertexts are related.
Modelling the attacker
Goals of an attacker

- **OW** (one-wayness): given a ciphertext find the corresponding plaintext.
- **SEM.SEC** (semantic security, polynomial security): given a ciphertext find partial information about the plaintext.
- **IND** (indistinguishability): distinguish between encryptions of two chosen plaintexts.
- **SEM.SEC.REL**: given a ciphertext find a plaintext which is in some way related to the plaintext corresponding to the given ciphertext.
- **NM** (non-malleability): given a ciphertext find another ciphertext such that the two ciphertexts are related.
Defining IND

Informally,

- key generator generates a key pair
- message finder $F$ get the public key and chooses two messages $m_0, m_1$
- line tapper $T$ gets the two plaintexts and the encryption of one of them and answers 0 if he thinks that the encrypted message is $m_0$ and 1 otherwise
- both message finder and line tapper may encrypt (polynomially many) plaintexts with arbitrary keys (CPA)
- when $m_0$ is encrypted the probability that $T$ answers 0 should be negligibly bigger than when $m_1$ is encrypted
A concise definition of IND

Formally:

Definition ([Dolev, Dwork & Naor, 1995], [Fuchsbauer, 2005])

The public key cryptosystem \((KG, E, D)\) is IND-CPA secure iff

\[
\forall F, T \in PPTM \ \forall c > 0 \ \exists n_c \in \mathbb{N} \ \forall n > n_c : \\
Pr[Succ_{T, E, n}(KG_1(1^n), F(KG_1(1^n))) \geq \frac{1}{n^c}] < \frac{1}{n^c},
\]

where

\[
Succ_{T, E, n}(e, (m_0, m_1)) := | Pr[T(e, m_0, m_1, E_e(m_0, R_n)) = 0] \\
- Pr[T(e, m_0, m_1, E_e(m_1, R_n)) = 0] |.
\]
Defining SEM.SEC.REL

Informally,

- key generator generates a key pair
- $A_1$ gets the public key and generates a description of a probability distribution $\hat{M}$ of messages
- a message $m$ is chosen according to $\hat{M}$ and encrypted with the public key
- $A_2$ gets the public key, $\hat{M}$, and the encryption of $m$, and returns $\beta$; measure $\pi_n(A, R) = Pr[R(m, \beta) = 1]$
- an adversary simulator has to do the same, but without seeing an encryption; measure $\pi'_n(A', R) = Pr[R(m', \beta') = 1]$
- for any relation $R$, $\text{Adv}_A := |\pi_n(A, R) - \pi'_n(A', R)|$ should be negligible
A concise definition of SEM.SEC.REL-CPA

Definition

Adversary

\[ \pi_n(A, R) := Pr[R(m, \beta) = 1 :: e \leftarrow KG_1(1^n); \hat{M} \leftarrow A_1(e); \]
\[ m \leftarrow U(\hat{M}); r \in_R \{0, 1\}^{p(n)}; \gamma := E_e(m, r); \]
\[ \beta \leftarrow A_2(e, \hat{M}, \gamma, hist(m))] \]

Adversary simulator

\[ \pi'_n(A', R) := Pr[R(m, \beta) = 1 :: e \leftarrow KG_1(1^n); \hat{M} \leftarrow A'_1(e); \]
\[ m \leftarrow U(\hat{M}); \beta \leftarrow A'_2(e, \hat{M}, hist(m))] \]
A concise definition of SEM.SEC.REL-CPA

Definition ([Dolev, Dwork & Naor, 1995], [Fuchsbauer, 2005])

The public key cryptosystem \((KG, E, D)\) is \textit{semantically secure with respect to relations} iff

\[
\forall A \exists A' \ \forall R \ \forall \text{hist} \ \forall c > 0 \ \exists n_c \in \mathbb{N} \ \forall n > n_c : \\
| \pi_n(A, R) - \pi'_n(A', R) | < \frac{1}{n^c}
\]
Theorem

For any public key cryptosystem, SEM.SEC, IND, and SEM.SEC.REL are equivalent under CPA.

Proof.

- SEM.SEC vs. IND: [Goldwasser & Micali, 1984], [Goldreich, 2004]
- IND vs. SEM.SEC.REL: [Dolev, Dwork & Naor, 1995], [Fuchsbauer, 2005]
A bigger picture

implication: ← separation: ⤵

OW − CPA → IND − CPA → NM − CPA

OW − VCA

OW − PCA

OW − CCA1 → IND − CCA1 → NM − CCA1

OW − CCA2 → IND − CCA2 → NM − CCA2
Ideal worlds

To prove the security of efficient schemes, cryptographic primitives are replaced by their ideal counterparts.

- random oracle model: hash function = random function
- ideal cipher model: block cipher = random permutation
- generic model: concrete group = blackbox group

There are cryptosystems which are (provably) secure in the random oracle model, but insecure when the random function is replaced by a concrete hash function. All known examples are pathological.
Ideal worlds

To prove the security of efficient schemes, cryptographic primitives are replaced by their ideal counterparts.

- **random oracle model**: hash function = random function
- **ideal cipher model**: block cipher = random permutation
- **generic model**: concrete group = blackbox group

There are cryptosystems which are (provably) secure in the random oracle model, but insecure when the random function is replaced by a concrete hash function. All known examples are pathological.
Ideal worlds

To prove the security of efficient schemes, cryptographic primitives are replaced by their ideal counterparts.

- random oracle model: hash function = random function
- ideal cipher model: block cipher = random permutation
- generic model: concrete group = blackbox group

There are cryptosystems which are (provably) secure in the random oracle model, but insecure when the random function is replaced by a concrete hash function. All known examples are pathological.
Ideal worlds

To prove the security of efficient schemes, cryptographic primitives are replaced by their ideal counterparts.

- random oracle model: hash function = random function
- ideal cipher model: block cipher = random permutation
- generic model: concrete group = blackbox group

There are cryptosystems which are (provably) secure in the random oracle model, but insecure when the random function is replaced by a concrete hash function. All known examples are pathological.
Ideal worlds

To prove the security of efficient schemes, cryptographic primitives are replaced by their ideal counterparts.

- random oracle model: hash function = random function
- ideal cipher model: block cipher = random permutation
- generic model: concrete group = blackbox group

There are cryptosystems which are (provably) secure in the random oracle model, but insecure when the random function is replaced by a concrete hash function. All known examples are pathological.
Ideal worlds

To prove the security of efficient schemes, cryptographic primitives are replaced by their ideal counterparts.

- random oracle model: hash function = random function
- ideal cipher model: block cipher = random permutation
- generic model: concrete group = blackbox group

There are cryptosystems which are (provably) secure in the random oracle model, but insecure when the random function is replaced by a concrete hash function. All known examples are pathological.
RSA

Standard (textbook) RSA ([Rivest, Shamir & Adleman, 1978])

- **Keys**: \( p, q \in \mathbb{P}, n = pq, e \in \mathbb{Z}_{(p-1)(q-1)}^*, d = e^{-1} \mod (p-1)(q-1); (n, e) \text{ public, } (n, d) \text{ private} \)
- **Encryption**: \( c = m^e \mod n \)
- **Decryption**: \( m = c^d \mod n \)

**Security**:
- OW-CPA (RSA-problem)
- OW-PCA: reduction to RSA-problem
- OW-CCA2: no (ciphertext blinding)
- IND-CPA: no (encrypt the two plaintexts and compare)
Standard (textbook) RSA ([Rivest, Shamir & Adleman, 1978])
- Keys: $p, q \in \mathbb{P}$, $n = pq$, $e \in \mathbb{Z}^*_{(p-1)(q-1)}$, $d = e^{-1} \mod (p-1)(q-1)$; $(n, e)$ public, $(n, d)$ private
- Encryption: $c = m^e \mod n$
- Decryption: $m = c^d \mod n$

Security:
- OW-CPA (RSA-problem)
- OW-PCA: reduction to RSA-problem
- OW-CCA2: no (ciphertext blinding)
- IND-CPA: no (encrypt the two plaintexts and compare)
RSA
Probabilistic Encryption

Micali–Goldwasser–Encryption
- Encryption (basically):
  \[ 1 \mapsto \text{a random square modulo } n \]
  \[ 0 \mapsto \text{a random non-square modulo } n \]

Security:
- IND-CPA: reduction to Quadratic Reduosity Assumption
- OW-CCA2: no ("multiplying-y-attack")
RSA
Probabilistic Encryption

Micali–Goldwasser–Encryption
- Encryption (basically):
  - $1 \mapsto$ a random square modulo $n$
  - $0 \mapsto$ a random non-square modulo $n$

Security:
- IND-CPA: reduction to Quadratic Reduosity Assumption
- OW-CCA2: no ("multiplying-y-attack")
RSA

RSA-OAEP

RSA-OAEP (Optimal Asymmetric Encryption Padding) ([Bellare & Rogaway, 1994]); the basic idea:

plaintext $\parallel 0 \ldots 0 \quad$ random bits

\[ h_1 \leftarrow \]
\[ h_2 \rightarrow \]
\[ s \leftarrow \]
\[ t \rightarrow \]
\[ \text{RSA} \]
RSA
RSA-OAEP

Security:

- IND-CCA1: reduction to “RSA is a one-way trapdoor permutation” ([Bellare & Rogaway, 1994], [Shoup, 2000/01])
- IND-CCA2: reduction to “RSA is partial domain one-way” ([Shoup, Pointcheval 2001])
- all reductions in the random oracle model
RSA
RSA-OAEP

Security:
- IND-CCA1: reduction to “RSA is a one-way trapdoor permutation” ([Bellare & Rogaway, 1994], [Shoup, 2000/01])
- IND-CCA2: reduction to “RSA is partial domain one-way” ([Shoup, Pointcheval 2001])
- all reductions in the random oracle model
Security:

- IND-CCA1: reduction to “RSA is a one-way trapdoor permutation” ([Bellare & Rogaway, 1994], [Shoup, 2000/01])
- IND-CCA2: reduction to “RSA is partial domain one-way” ([Shoup, Pointcheval 2001])
- all reductions in the random oracle model
ElGamal

Standard (textbook) ElGamal ([ElGamal, 1985])

- Keys: public key \((p, g, A)\), private key \(a, A = g^a \mod p\)
- Encryption: \(B = g^k \mod p, c = mA^k \mod p\)
- Decryption: \(m = cB^{-a} \mod p\)

Security:

- OW-CPA: reduction to CDH (from \(g^x\) and \(g^y\) compute \(g^{xy}\))
- IND-CPA: reduction to DDH (distinguish \((g^x, g^y, g^{xy})\) from a random triple \((a, b, c)\)) (for "suitable" parameters)
- OW-PCA: reduction to GDH (solve CDH with a DDH-oracle)
- NM-CPA: no, plaintexts of \((B, c)\) and \((B, 2c)\) are related
- OW-CCA2: no, ciphertext blinding
ElGamal

Standard (textbook) ElGamal ([ElGamal, 1985])

- Keys: public key \((p, g, A)\), private key \(a, A = g^a \mod p\)
- Encryption: \(B = g^k \mod p, c = mA^k \mod p\)
- Decryption: \(m = cB^{-a} \mod p\)

Security:

- OW-CPA: reduction to CDH (from \(g^x\) and \(g^y\) compute \(g^{xy}\))
- IND-CPA: reduction to DDH (distinguish \((g^x, g^y, g^{xy})\) from a random triple \((a, b, c)\)) (for "suitable" parameters)
- OW-PCA: reduction to GDH (solve CDH with a DDH-oracle)
- NM-CPA: no, plaintexts of \((B, c)\) and \((B, 2c)\) are related
- OW-CCA2: no, ciphertext blinding
ElGamal
Cramer–Shoup Cryptosystem

Cramer–Shoup Cryptosystem ([Cramer & Shoup, 1998])
- Encryption: 4 exponentiations, 2 multiplications, 1 hash function evaluation
- Decryption: 2 exponentiation, 2 multiplications, 1 inversion, 1 hash function evaluation

Security:
- IND-CCA2: reduction to DDH
For Further Reading I


For Further Reading II


For Further Reading III

G. Fuchsbauer.
Diplomarbeit, Linz, 2005.

S. Goldwasser, S. Micali.
Probabilistic Encryption.
*Journal of Computer and System Sciences*, 270–299, 1984

V. Shoup.
OAEP Reconsidered.
*Crypto ’01*, Springer, 2001