Review of Elementary Cryptography

For more material, see my notes of CSE 5351, available on my webpage

Outline

- Security (CPA, CCA, semantic security, indistinguishability)
- RSA
- ElGamal
- Homomorphic encryption

Two types of encryption schemes

- Private-key encryption schemes
  - Also called symmetric-key encryption schemes
- Public-key encryption schemes
  - Also called asymmetric-key encryption schemes
- We are more interested in the latter.

Symmetric-key encryption scheme

- Message space: $M \subseteq \{0,1\}^*$.
- Key generation algorithm G: On input 1^n, $G(1^n)$ outputs a key $k \leftarrow \{0,1\}^n$. ($K = \{0,1\}^n$; and n is the security parameter.)
- Encryption algorithm E: On input a key k and a plaintext $m \in M$, E outputs a ciphertext c. We write $c \leftarrow E(k,m)$ or $c \leftarrow E_k(m)$.
- Decryption algorithm D: On input a key k and a ciphertext c, D outputs a message m. We write $m := D(k,c)$ or $m := D_k(c)$.
- Correctness requirement: for each $k \in K$ and $m \in M$,

$$D_k\left(E_k(m)\right) = m.$$

- G, E, probabilistic algorithms. D, deterministic. All poly-time.

Public-key encryption scheme

- Key generation algorithm G: On input 1^n, $G(1^n)$ outputs a pair of keys, (pk, sk), each of length at least n.
- Encryption algorithm E: On input a public key pk and a plaintext $m \in M_{pk}$, E outputs a ciphertext c. We write $c \leftarrow E_{pk}(m)$. (The message space may depend on pk.)
- Decryption algorithm D: On input a secret key sk and a ciphertext c, D outputs a message m. We write $m := D_{sk}(c)$.
- Correctness requirement:

$$\Pr\left[D_{sk}\left(E_{pk}(m)\right) = m : \; m \leftarrow M_{pk} \right] = 1$$

except for a negligible portion of key pairs output by $G(1^n)$.

Symmetric vs. Asymmetric

- Symmetric encryptions are much faster than asymmetric ones.
  - AES is typically 100 times faster than RSA encryption and 1000 times faster than RSA decryption.
- Use asymmetric cipher to set up a session key and then use symmetric cipher to encrypt data.

Security Issues

What does it mean that an encryption scheme is secure (or insecure)?

- Semantic security
- Ciphertext-indistinguishability
- Non-malleability

Different types of attacks

- Different types of attacks (classified by the amount of information available to the attacker):
  - Ciphertext-only attack (eavesdropper)
  - Known-plaintext attack
  - Chosen-plaintext attack (CPA)
  - Chosen-ciphertext attack (CCA)

Negligible functions

- A nonegative function $f : N \to R$ is said to be negligible if for every positive polynomial $P(n)$, there is an integer n_0 such that

$$f(n) < \frac{1}{P(n)} \quad \text{for all } n > n_0 \ \text{ (i.e., for sufficiently large } n\text{).}$$

- Examples: 2^{-n}, $2^{-\sqrt{n}}$, $n^{-\log n}$ are negligible functions.
- Negligible functions approach zero faster than the reciprocal of every polynomial.
- We write negl(n) to denote an unspecified negligible function.

Security Parameter

- The security of an encryption scheme typically depends on its key length.
  - Is RSA secure if $|N| = 216,\ 512,\ \text{or } 1024$?
- In general, an encryption scheme is associated with an integer called its security parameter. (For now, you may think of it as key length.)
- When we say that the probability $\Pr(\lambda)$ of an encryption scheme being broken is negligible, it is w.r.t. the encryption scheme's security parameter λ.

Semantic Security (simplified)

- A private-key encryption scheme (G, E, D) with sesutiry parameter n is semantically secure against an eavesdropper if for every probabilistic polynomial-time (PPT) algorithm A there exists a PPT A' such that for all polynomial-time computable functions f and h:

$$\Big| \Pr\Big[A\big(1^n, E_k(m), h(m)\big) = f(m) : k \leftarrow G(1^n),\ m \leftarrow \{0,1\}^n \Big] - \Pr\Big[A'\big(1^n, h(m)\big) = f(m) : m \leftarrow \{0,1\}^n \Big] \Big| \le \operatorname{negl}(n).$$

Ciphertext-Indistinguishability

- Adversary: a polynomial-time eavesdropper.
- (G, E, D): an encryption scheme with security parameter *n*.
- Imagine a game played by Bob and Eve (adversary):
  - Eve is given input 1^n and outputs a pair of messages m_0, m_1 of the same length.
  - Bob chooses a key $k \leftarrow G(1^n)$ and $m \leftarrow_u \{m_0, m_1\}$. He computes $c \leftarrow E_k(m)$ and gives c to Eve.
  - Eve tries to determine whether c is the encryption of m_0 or m_1.
- An encryption scheme is ciphertext-indistinguishable against eavesdroppers if no adversary can succeed with probability non-negligibly greater than $1/2$.

- Definition: An encryption scheme is ciphertext-indistinguishable against eavesdroppers if for every PPT algorithm A and all $m_0, m_1 \in M$, $|m_0| = |m_1|$, it holds:

$$\Pr\left[A(1^n, m_0, m_1, E_k(m)) = m : \; m \leftarrow_u \{m_0, m_1\}, \; k \leftarrow G(1^n) \right]$$

$$\leq \frac{1}{2} + \mathrm{negl}(n)$$

Equivalence of semantic security and ciphertext-indistinguishability

- Theorem: Against an eavesdropper, an encryption scheme is semantically secure iff it is ciphertext-indistinguishable.

- Theorem: Under CPA, CCA1 or CCA2, an encryption scheme is semantically secure if and only if it is ciphertext-indistinguishable.

Chosen-plaintext attacks (CPA)

- Informally, we may describe CPA as follows:
  - Given: $(m_1, c_1),\ (m_2, c_2),\ \ldots,\ (m_t, c_t)$, where $m_1,\ m_2,\ \ldots,\ m_t$ are chosen by the adversary; and a new ciphertext c.
  - Q: what is the plaintext of c?
- Adaptively-chosen-plaintext attack: $m_1,\ m_2,\ \ldots,\ m_t$ are chosen adaptively.
- Now we describe CPA in terms of oracle.

Chosen-plaintext attacks (CPA)

A CPA on an encryption scheme (G, E, D) is modeled as follows.

1. A key $k \leftarrow G(1^n)$ is generated.
2. The adversary is given input 1^n and oracle access to E_k. She may request the oracle to encrypt plaintexts of her choice.
3. The adversary chooses two message m_0, m_1 with $|m_0| = |m_1|$; and is given a challenge ciphertext $c \leftarrow E_k(m_b)$, where $b \leftarrow_u \{0,1\}$.
4. The adversary continues to have oracle access and may request the encryptions of additional plaintexts of her choice, even m_0 and m_1.
5. The adversary finally answers 0 or 1.

Note: The CPA here actually refers to an adaptive CPA.

Ciphertext-indistinguishability against CPA

- An encryption scheme (G, E, D) is IND-CPA if no polynomial-time adversary can answer correctly with probability non-negligibly greater than $1/2$.

- Definition: an encryption scheme (G, E, D) is IND-CPA if for ever polynomial adversary A it holds that:

$$\left| \Pr\left[A^{E_k}\left(1^n, m_0, m_1, E_k(m)\right) = m : \; k \leftarrow G(1^n), \; m \leftarrow_u \{m_0, m_1\}, \; m_0, m_1 \leftarrow_A M \right] \right|$$

$$\leq \; \frac{1}{2} + \mathrm{negl}(n)$$

Chosen-ciphertext attacks (CCA)

- Informally we may describe CCA as follows:
  - Given: $(m_1, c_1),\ (m_2, c_2),\ \ldots,\ (m_t, c_t)$, where $c_1,\ c_2,\ \ldots,\ c_t$ are chosen by the adversary; and a new ciphertext c.
  - Q: what is the plaintext of c?
- Adaptively-chosen-plaintext attack: $c_1,\ c_2,\ \ldots,\ c_t$ are chosen adaptively.
- Now we describe CCA in terms of oracle.
- We will allow a CCA adversary to also have CPA capability. (So, by CCA we mean CCA+CPA, rather than pure CCA.)

Chosen-ciphertext attacks (CCA)

A CCA on an encryption scheme (G, E, D) is modeled as follows.

1. A key $k \leftarrow G(1^n)$ is generated.
2. The adversary is given input 1^n and oracle access to E_k and D_k. She may request the oracles to perform encryptions and/or decryptions for her.
3. The adversary chooses two message m_0, m_1 with $|m_0| = |m_1|$; and is given a challenge ciphertext $c \leftarrow E_k(m_b)$, where $b \leftarrow_u \{0,1\}$.
4. The adversary continues to have oracle access to E_k and D_k, but is not allowed to request the decryption of c.
5. The adversary finally answers 0 or 1.

Ciphertext-indistinguishability against CCA

- An encryption scheme (G, E, D) is IND-CCA if no polynomial-time adversary can answer correctly with probability non-negligibly greater than $1/2$.

- Definition: an encryption scheme (G, E, D) is IND-CCA if for ever polynomial-time adversary A, it holds that:

$$\Big| \Pr\Big[A^{E_k, D_k}\big(1^n, m_0, m_1, E_k(m)\big) = m : \ k \leftarrow G(1^n), \ m \leftarrow_u \{m_0, m_1\}, \quad m_0, m_1 \leftarrow_A M \Big] \Big|$$

$$\leq \ \frac{1}{2} + \mathrm{negl}(n)$$

CCA1 vs. CCA2

- The CCA described above is also called CCA2.
- If in item #4 the adversary has no access to the decryption oracle, the CCA is called CCA1.

Non-malleability

- An encryption scheme (G, E, D) is non-malleable if given a ciphertext $c = E(m)$, it is computationally infeasible for an adversary to produce a ciphertext c' such that $m' = D(c')$ has some known relation with m.
- RSA is malleable.
- malleable $\Rightarrow$ not IND-CCA2.
- Every homomorphic encryption scheme is malleable, and hence cannot be IND-CCA2.
  - Highest security level possible for homomorphic encryption scheme: IND-CCA1.

Homomorphic Encryption

RSA is homomorphic

- $\mathrm{RSA}(m_1 \cdot m_2) = \mathrm{RSA}(m_1) \cdot \mathrm{RSA}(m_2)$

  where $\cdot$ is the multiplication in Z_n^* (i.e., modulo n).

- Easy to verify:
  - $\mathrm{RSA}(m_1 \cdot m_2) = \left(m_1 \cdot m_2\right)^e$
  - $\mathrm{RSA}(m_1) = m_1{}^e$
  - $\mathrm{RSA}(m_2) = m_2{}^e$
  - $\mathrm{RSA}(m_1) \cdot \mathrm{RSA}(m_2) = m_1{}^e \cdot m_2{}^e = \left(m_1 \cdot m_2\right)^e$

Homomorphic encryption

M : message space

C : ciphertext space

$\odot_M$: some binary operation in M

$\odot_C$: some binary operation in C

Definition: An encryption scheme is $\odot_M$ -homomorphic if the encryption function E satisfies

$$E_{pk}(m_1 \odot_M m_2) = E_{pk}(m_1) \odot_C E_k(m_2)$$

for all keys pk and messages $m_1, m_2 \in M$.

Comment: doesn't work for probabilistic encryptions.

ElGamal encryption is homomorphic

- $E(m_1 \cdot m_2) \leftarrow E(m_1) \cdot E(m_2),$ in the following sense:

  $E(m_1) \cdot E(m_2)$ is a valid encryption of $m_1 m_2$.

- Verification:

  If $E(m_1) = \left(g^{k_1},\ m_1 y^{k_1}\right)$ and $E(m_2) = \left(g^{k_2},\ m_2 y^{k_2}\right)$, then

  $$E(m_1) \cdot E(m_2) = \left(g^{k_1},\ m_1 y^{k_1}\right) \cdot \left(g^{k_2},\ m_2 y^{k_2}\right)$$
  $$= \left(g^{k_1+k_2},\ m_1 m_2 y^{k_1+k_2}\right)$$

  is a valid encryption of $m_1 m_2$.

Homomorphic encryption redefined

M : message space

C : ciphertext space

$\odot_M$: some binary operation in M

$\odot_C$: some binary operation in C

Definition: An encryption scheme is $\odot_M$ -homomorphic if the encryption function E satisfies

$$m_1 \odot_M m_2 = D_{sk}\left(E_{pk}(m_1) \odot_C E(m_2)\right)$$

for all messages $m_1, m_2 \in M$ and all encryption/decryption key pairs (pk, sk).

A generalized definition

Definition: An encryption scheme is homomorphic w.r.t $\odot_M$ if there is a polynomial time algorithm A such that

$$m_1 \odot_M m_2 = D_{sk}\left(A_{pk}\left(E(m_1), E(m_2)\right)\right)$$

for all messages $m_1, m_2 \in M$ and all encryption/decryption key pairs.

Question: How to further generalize it?

Various homomorphic encryptions

- An encryption scheme is
  - additively homomorphic if it is homomorphic w.r.t $+_M$
  - multiplicatively homomorphic if it is homomorphic w.r.t $\cdot_M$
  - algebraicly homomorphic if it is homomorphic w.r.t both $+_M$ and $\cdot_M$
- RSA (1977) and ElGamal (1984) are multiplicatively homomorphic.
- Padded RSA is not homomorphic.
- Goldwasser-Micali (1982): additively homomorphic

Fully homomorphic encryption

- In 2009, Craig Gentry proposed a fully homomorphic encryption scheme.
- Informally: An encryption scheme is homomorphic w.r.t A_M if there is a polynomial time algorithm A_C such that

$$A_M(m_1, \ldots, m_t) = D_{pk}\left(A_C\left(pk, E(m_1), \ldots, E(m_t)\right)\right)$$

- Informally: An encryption scheme is fully homomorphic if it is homomorphic w.r.t. any algorithm A_M.