Introduction to Cryptology

Lecture 12

Announcements

- Midterm solutions and grades up on Canvas
- Homework 6 will be out on Thursday

Agenda

- This time:
  - CCA Security (3.7)
  - New topic: Message Integrity (4.1)
  - Message Authentication Codes (MAC) (4.2)

Chosen Ciphertext Security

CCA Security

The CCA Indistinguishability Experiment $PrivK^{cca}_{A,\Pi}(n)$:

1. A key k is generated by running $Gen(1^n)$.
2. The adversary A is given input 1^n and oracle access to $Enc_k(\cdot)$ and $Dec_k(\cdot)$, and outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $b \leftarrow \{0,1\}$ is chosen, and then a challenge ciphertext $c \leftarrow Enc_k(m_b)$ is computed and given to A.
4. The adversary A continues to have oracle access to $Enc_k(\cdot)$ and $Dec_k(\cdot)$, but is not allowed to query the latter on the challenge ciphertext itself. Eventually, A outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

CCA Security

A private-key encryption scheme $\Pi = (Gen, Enc, Dec)$ has indistinguishable encryptions under a chosen-ciphertext attack if for all ppt adversaries A there exists a negligible function $negl$ such that

$$\Pr\left[PrivK^{cca}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the random coins used by A, as well as the random coins used in the experiment.

Vulnerability of Construction to CCA Attacks

- Consider the following attack:
  - Attacker asks for an encryption of two messages m_0, m_1 where lsb of $m_0 = 0$ and lsb of $m_1 = 1$.
  - Attacker receives ciphertext $c = \langle r, s \rangle$.
  - Attacker sets ciphertext s' to be equal to s with the lsb flipped.
  - Attacker asks decryption oracle to decrypt $c' = \langle r, s' \rangle$, receiving m' in return.
  - If the lsb of m' is 1, return $\mathrm{b}' = 0$. Otherwise, return $\mathrm{b}' = 1$.
- Why does this attack work?

Message Integrity

- Secrecy vs. Integrity

- Encryption vs. Message Authentication

Message Authentication Codes

Definition: A message authentication code (MAC) consists of three probabilistic polynomial-time algorithms $(Gen, Mac, Vrfy)$ such that:

1. The key-generation algorithm Gen takes as input the security parameter 1^n and outputs a key k with $|k| \geq n$.
2. The tag-generation algorithm Mac takes as input a key k and a message $m \in \{0,1\}^*$, and outputs a tag t. $t \leftarrow Mac_k(m)$.
3. The deterministic verification algorithm $Vrfy$ takes as input a key k, a message m, and a tag t. It outputs a bit b with $b = 1$ meaning valid and $b = 0$ meaning invalid. $b := Vrfy_k(m, t)$.

It is required that for every n, every key k output by $Gen(1^n)$, and every $m \in \{0,1\}^*$, it holds that $Vrfy_k\big(m, Mac_k(m)\big) = 1$.

Security of MACs

The message authentication experiment $MACforge_{A,\Pi}(n)$:

1. A key k is generated by running $Gen(1^n)$.
2. The adversary A is given input 1^n and oracle access to $Mac_k(\cdot)$. The adversary eventually outputs (m, t). Let Q denote the set of all queries that A asked its oracle.
3. A succeeds if and only if (1) $Vrfy_k(m, t) = 1$ and (2) $m \notin Q$. In that case, the output of the experiment is defined to be 1.

Security of MACs

Definition: A message authentication code $\Pi = (Gen, Mac, Vrfy)$ is existentially unforgeable under an adaptive chosen message attack if for all probabilistic polynomial-time adversaries A, there is a negligible function neg such that:

$$\Pr\left[MACforge_{A,\Pi}(n) = 1\right] \leq neg(n)\,.$$