Lecture Note: Definitions of Security for Encryption

February 24, 2017

1 Encryption scheme

Any symmetric encryption scheme $\mathsf{Enc}, \mathsf{Dec}$ has a symmetric secret key that is shared between the send and reciever. The sender encrypts the message m to obtain the ciphertext $c = \mathsf{Enc}_k(m)$. The reciever decrypts the ciphertext to obtain the plaintext as $m' = \mathsf{Dec}_k(c)$.

Any encryption scheme must satisfy the correctness property, *i.e.,* $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$.

There are various definitions of security for encryption schemes, listed below.

1.1 Ciphertext only (COA) security.

When playing the following game with a Challenger, no efficient Adversary should be able to win with probability much more than half:

- A key k for the encryption scheme is chosen uniformly at random by the Challenger.
- The Adversary is given access to a set of ciphertexts $c_1, ..., c_n$, where each $c_i = \mathsf{Enc}_k(m_i)$. The adversary does not know k or m_i.
- The Adversary chooses two messages of equal length, m_0 and m_1, and sends them to the Challenger.
- The Challenger chooses a random bit b, produces the challenge ciphertext $c^* = \mathsf{Enc}_k(m_b)$, and sends c^* to the adversary.
- The Adversary outputs $b' = 0$ if it thinks that c^* was the encryption of m_0, and $b' = 1$ otherwise.
- The Adversary wins if $b' = b$.

1.2 Known-Plaintext-Attack (KPA) security.

When playing the following game with a Challenger, no efficient Adversary should be able to win with probability much more than half:

- A key k for the encryption scheme is chosen uniformly at random by the Challenger.
- The Adversary, which does not know k, is given access to set of (plaintext, ciphertexts) pairs $(m_1, c_1), ..., (m_n, c_n)$, where each $c_i = \mathsf{Enc}_k(m_i)$. The adversary does not know k.
- The Adversary chooses two messages of equal length, m_0 and m_1, and sends them to the Challenger.

- The Challenger chooses a random bit b, produces the challenge ciphertext $c^* = \mathsf{Enc}_k(m_b)$, and sends c^* to the adversary.
- The Adversary outputs $b' = 0$ if it thinks that c^* was the encryption of m_0, and $b' = 1$ otherwise.
- The Adversary wins if $b' = b$.

1.3 Chosen-Plaintext-Attack (CPA) security.

When playing the following game with a Challenger, no efficient Adversary should be able to win with probability much more than half:

- A key k for the encryption scheme is chosen uniformly at random by the Challenger.
- The Adversary, which does not know k, is given access to an oracle that computes $\mathsf{Enc}_k(\cdot)$ on a message m of the adversary's choice.
- The Adversary chooses two messages of equal length, m_0 and m_1, and sends them to the Challenger.
- The Challenger chooses a random bit b, produces the challenge ciphertext $c^* = \mathsf{Enc}_k(m_b)$, and sends c^* to the adversary.
- The Adversary may continue to send any message of its choice to the the oracle that computes $\mathsf{Enc}_k(\cdot)$.
- The Adversary outputs $b' = 0$ if it thinks that c^* was the encryption of m_0, and $b' = 1$ otherwise.
- The Adversary wins if $b' = b$.

1.4 Chosen-Cipher-Attack (CCA2) security.

When playing the following game with a Challenger, no efficient Adversary should be able to win with probability much more than half:

- A key k for the encryption scheme is chosen uniformly at random by the Challenger.
- The Adversary, which does not know k, is given access to an oracle that computes $\mathsf{Enc}_k(\cdot)$ on a message m of the adversary's choice.
- The Adversary is also given access to an oracle that computes $\mathsf{Dec}_k(\cdot)$ on a ciphertext c of the adversary's choice.
- The Adversary chooses two messages of equal length, m_0 and m_1, and sends them to the Challenger.
- The Challenger chooses a random bit b, produces the challenge ciphertext $c^* = \mathsf{Enc}_k(m_b)$, and sends c^* to the adversary.
- The Adversary may continue to send any message of its choice to the the oracle that computes $\mathsf{Enc}_k(\cdot)$ and the oracle that computes $\mathsf{Dec}_k(\cdot)$

- The Adversary outputs $b' = 0$ if it thinks that c^* was the encryption of m_0, and $b' = 1$ otherwise.
- The Adversary wins if $b' = b$.