Network Security and Cryptography
Symmetric-key cryptography

Lecture 5: Security of an encryption system

Mark Ryan

Defining the “security” of an encryption system

When should we say an encryption system is secure?

Candidate definition 1. “Given a ciphertext, there is no way that an attacker (that doesn’t have the key) can obtain the plaintext.”

What do you think about this definition?

- Maybe it’s too strong: maybe there is no encryption system that can meet this definition.
- Maybe it’s too weak: maybe an encryption system that meets it is still considered insecure.

Is the candidate definition too strong?

Technically, no. The one-time pad satisfies the definition. We can say that more formally:

Theorem

Let c be a ciphertext encrypted with a one time pad. Then any bitstring m of the same length as c has as much chance of being the plaintext corresponding to c as any other one.

Proof.

Let m and c be bitstrings of length n, and let k be a randomly-chosen key also of length n. Then

$$Pr[\mathrm{Enc}_k(m) = c] \;=\; \frac{1}{2^n}.$$

□

Is the candidate definition too strong?

In practice, yes. Even though it is be satisfied by the one-time pad, we still consider it too strong. The one-time pad is not a practical encryption system. In practice, we need encryption keys to be

- short
- reusable.

However, if the keys are short and reusable, that means that an attacker could enumerate all the possible keys and try them all. We don't want that to be considered a valid attack.

Is the candidate definition too weak?

Yes, it is too weak as well.

For example, 3DES in ECB mode would satisfy this definition, but we saw that 3DES in ECB mode is not secure because it leaks whether blocks in the plaintext are identical or not.

Candidate definition 2. “Given a ciphertext, the attacker can obtain the plaintext only with N units of work.”

(Units of work could be measured as hours/months/years of computation time on a fast computer; or more abstractly, could be measured as number of computation operations. Example: we could say “DES can be broken in 2^{56} operations.”)

Candidate definition 2. “Given a ciphertext, the attacker can obtain the plaintext only with N units of work.”

(Units of work could be measured as hours/months/years of computation time on a fast computer; or more abstractly, could be measured as number of computation operations. Example: we could say “DES can be broken in 2^{56} operations.”)

This is still not satisfactory. An attacker could get lucky, and guess the key. We don’t regard that as a valid attack either.

Candidate definition 3

Candidate definition 3 “Given a ciphertext, the attacker can obtain the plaintext only with N units of work, or guess the plaintext or the key with probability at most $\frac{1}{M}$.”

Example: the attacker can guess an DES key with probability $\frac{1}{2^{56}}$, but that is a very small probability.

What do you think about this definition?

- Too strong?
- Too weak?

Candidate definition 3 is too weak

Suppose the attacker could determine a few bits of the message. Or suppose the attacker could tell if some plaintexts were more likely than others. These are valid concerns. An encryption system should be considered insecure if an attacker can derive statistical information on what the plaintext is. But this is not captured in Definition 3.

Candidate definition 4. “Given a ciphertext, the attacker cannot tell anything about the plaintext, except with N units of work, or by guessing plaintext or the key with probability at most $\frac{1}{M}$.”

Defining security for encryption

We formalise the idea that the adversary “can’t tell anything” about the plaintext as a game between the attacker and a “challenger” that sets challenges.

Idea: suppose the challenger gives the attacker three values:

- A message m_1
- A message m_2, with the same length
- The encryption $c = \mathsf{Enc}_k(m_1)$ or the encryption $c = \mathsf{Enc}_k(m_2)$.

The attacker has to guess whether c is the encryption of m_1 or m_2. What is the attacker’s probability of getting it right?

Example

m_1 = “Meet at the agreed place at 16:00 on Tuesday”

m_2 = “The plan is cancelled. Don’t send the money!”

c = “UEFEX1VzZUNhc2VzQ29tbWVyY2lhbGlzYXRpb25UaG91Z2h0c19MdWx1ZmY=”

Do you think c is the ciphertext of m_1 or m_2?

Defining security for encryption

- How to rule out the attack of enumerating all the keys?

  Answer: by restricting the attacker to perform a number of computations that is "polynomial" in the key length. (There number of keys is "exponential" in the key length.)

- Can we strengthen the definition (= better security), i.e., make the game easier for the attacker?

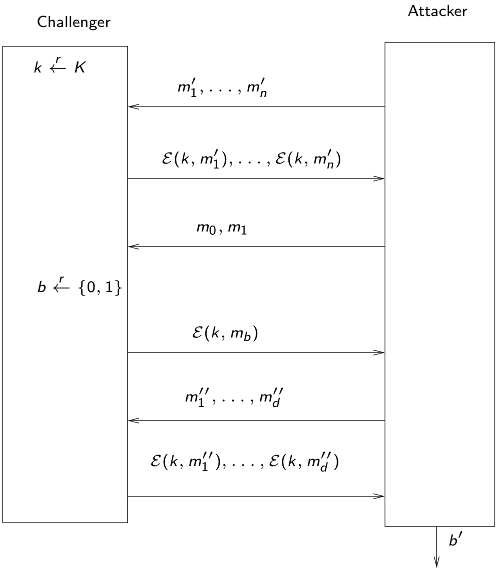
  Answer: by allowing the attacker to perform other encryptions, and allowing the attacker to chose the messages m_1 and m_2.

Definition (IND-CPA)

Let $(\mathrm{Enc}, \mathrm{Dec})$ be an encryption system with key space K. We define the game *indistinguishability under chosen-plaintext* (IND-CPA) between challenger and attacker as follows:

- The challenger generates a key $k \in K$ at random.
- The attacker can perform a polynomial number of computations, and can ask the challenger for the encryption by Enc of a polynomial number of arbitrary messages.
- The attacker submits two messages m_0 and m_1 to the challenger.
- The challenger selects a bit $b \in \{0, 1\}$ at random.
- The challenger returns the encryption $\mathrm{Enc}_k(m_b)$ to the attacker
- The attacker can again perform polynomially many computations, and ask for polynomially-many encryptions.
- The attacker outputs a bit b'.

The attacker wins this game if $b' = b$.



Challenger

Attacker

$k \xleftarrow{r} K$

$m'_1, \ldots, m'_n$

$\mathcal{E}(k, m'_1), \ldots, \mathcal{E}(k, m'_n)$

m_0, m_1

$b \xleftarrow{r} \{0, 1\}$

$\mathcal{E}(k, m_b)$

$m''_1, \ldots, m''_d$

$\mathcal{E}(k, m''_1), \ldots, \mathcal{E}(k, m''_d)$

b'

An encryption system is secure if the attacker cannot do any better than to *guess* the bit b.

Definition

Suppose the keys in K have length n. Let $Pr[b = b']$ be the probability that the attacker wins the IND-CPA game, taken over all encryption keys in K and all bits b. An encryption system satisfies *indistinguishability under chosen-plaintext attack* (IND-CPA) if

$$\left| Pr[b = b'] - \frac{1}{2} \right|$$

is negligible in n.

(A function $f : N \to R$ is negligible in n if, as n grows, the function becomes exponentially small. Formally: f is negligible if for all c, there exists n_0 such that for all $n > n_0$, $|f(n)| < \frac{1}{n^c}$.)

The attacker's power in security games

In security games, attacker can only do efficient operations, and only "efficiently" many of them

Formally: attacker is *probabilistic polynomial-time Turing machine* (PPT)

Importantly: attacker cannot search through all keys, as the number of possible keys increases exponentially with the length of the key.

ECB is not secure

Let (E, D) be a secure block cipher, and let Enc be encryption using using E in ECB mode.

The attacker can easily win the IND-CPA game. He can get the encryption of m_1 and m_2 in the first part (or even in the last part) of the game, and hence can easily distinguish which one the challenger chose.

CBC and CTR are secure

If (E, D) is a secure block cipher with key space X, then using (E, D) in CBC mode or CTR mode is secure.

There are limits on how many messages you should encrypt with the same key, and the maximum length of those messages, but these limits are generous. Let q be the number of messages encrypted with the same key k and L is the maximum length of the messages, and Adv is the advantage of the attacker in the game for the secure block cipher. Then the adversary's advantage in CTR mode is bounded by

$$\frac{2q^2L}{|X|} + 2Adv$$

Suppose $|X| = 2^{128}$. Then, for example, it means you should change keys after encrypting 2^{16} message of length 2^{16} each to obtain advantage of $\frac{1}{2^{80}}$.

A variation of CBC that is not secure

Let (E, D) be a secure block cipher, and let Enc be encryption using the following variation of CBC mode:

Each time an encryption is done, the IV that is used is the previous IV plus 1.

(In CBC, one should use a random IV each time. Here, we are simply incrementing the IV used last time, instead of taking a fresh random one.)

Explain why this is insecure.