CSE 5852: Lecture 4

Chao Shang

September 12, 2016

1 Review of Last Class

The last class covered some background in probability theory and introduced the secrecy of a channel. We have learned the definition of perfect secrecy and Shannon secrecy. Today we will finish discussing the secrecy of channel and the one-time pad. We will then begin discussing active attackers and message authentication codes (MACs).

2 Secrecy of a channel

Theorem 1. *Let (Gen, Enc, Dec) be a Shannon Secrecy over a message space M $\{0,1\}^n$, and let K be the key space as determined by Gen. Then $|K| \geq |M| = 2^n$.*

Let's first consider the set of ciphertexts that can be created by each individual message. Denote by C_{m_1} the set of possible ciphertexts for a message m_1 (across the key space).

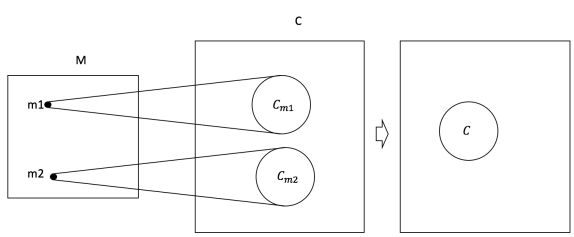

Figure 1: Relationship between C_{m_1} and C_{m_2}

Question 1: From above Figure 1, What is the relationship between C_{m_1} and C_{m_2}?

By the definition of perfect secrecy, for any m_1, m_2, $C_{m_1} = C_{m_2}$. That is, we can just consider the set C which will be same regardless of the message. If there was some c that was possible under some message (but not another) the adversary could always rule out a message based on that ciphertext. This violates perfect secrecy.

Recall that Dec function succeeds with probability 1. Since C is the same no matter the message, this means for any m there exists some k such that $\mathsf{Dec}(k, c) = m$. This means that $\mathsf{Dec}(k, \cdot)$ is an onto function. (That is, $\forall y, \exists x, s.t. \mathsf{Dec}(k, x) = y$.)

Furthermore, consider the truth table of the decryption function for a particular c. It must be true that for every c, m there exists some k such that $\mathsf{Dec}(c, k) = m$. (If not then C_m would not include c.) This means that for every c there exists the function $\mathsf{Dec}(c, \cdot)$ has range of size at least 2^n. This implies that $|\mathcal{K}| \geq 2^n$.

3 Active Attackers

3.1 What can Attacker do

We showed in the previous class that it is possible to provide perfect secrecy using the one-time pad or OTP [Ver19]. What does our adversary do now? Do they give up and go home?

If there is a attacker in the middle of sender and receiver on Figure 2, let's think about what Attacker can do. What set of actions might still be available to them?

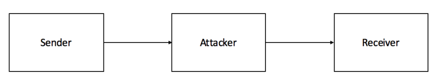

Figure 2: Attacker between sender and receiver

1. Learn about key
2. Take message directly from Receiver (by breaking into their computer)
3. Change C
4. Pretend to be one of the parties.
5. Not send C

Case 1: Two messages
Considering the case, two messages: m_1 = "Attack", m_2 = "Defend".

Attack	01100001	01110100	01110100	01100001	01100011	01101011
Key	10011010	11110010	00110010	11000110	00110010	00000110
Ciphertext	11111011	10000110	01000110	10100111	01010001	01101101
Defend	01100100	01100101	01100110	01100101	01101110	01100100
Mask	00000101	00010001	00010010	00000100	00001101	00001111
Ciphertext'	11111110	10010111	01010100	10100011	01011100	01100010

Based on “Attack” $\oplus$ “Defend”, we can add information to C:

$$C' = C \oplus (m_1 \oplus m_2) = k \oplus m_1 \oplus (m_1 \oplus m_2) = k \oplus m_2$$

So C have been changed in a way that the message will properly decrypt to “Defend.”

Case 2: Three messages

Based on m_1, m_2, m_3, our attack still works a fraction of the time. For example, consider the mask $m_1 \oplus m_3$.

$$k \oplus m_1 \oplus m_1 \oplus m_3 = k \oplus m_3$$

$$k \oplus m_2 = m_1 \oplus m_2 \oplus m_3$$

$$k \oplus m_3 = m_1 \oplus k$$

Thus, the attack succeeds with a nonzero probability but it is not always successful. Since in perfectly secure schemes the ciphertext does not depend on the key it is easy to change C.

New goal: Detect when C is changed.

3.2 Algorithms to prevent an adversary

Message authentication code (MAC)

The aim of a message authentication code is to prevent an adversary from modifying a message sent by one party to another, without the parties detecting that a modification has been made.

Definition 1. *(Message authentication code): A message authentication code or MAC is a tuple of probabilistic polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vfy})$ *fulfilling the following:*

1. Gen *gives the key k on input 1^n, where n is the security parameter.*

2 .Mac *outputs a tag t on the key k and the input string c.*

$$\mathsf{Mac}(\alpha, c) = t$$

3. Vfy *outputs accepted or rejected on inputs: the key k, the string c and the tag t.* Vfy *outputs either* 1 *or* 0 *(representing true or false).*

The Informal Goal is : $Verify(\alpha, c, t) = 1$ iff c hasn't charged. Note that we don't care if an adversary changed t but kept x constant.[1] We now turn to trying to define security.

Message authentication experiment Mac-forge

The message authentication experiment Mac-forge is :

1. A random key α is chosen.
2. The attacker A creates a message c and receives $t = \mathsf{Mac}(\alpha, c)$.

[1]We use the term c since we were previously discussed how to protect integrity of an encryption scheme. However Mac algorithms can also be used on plaintext messages.

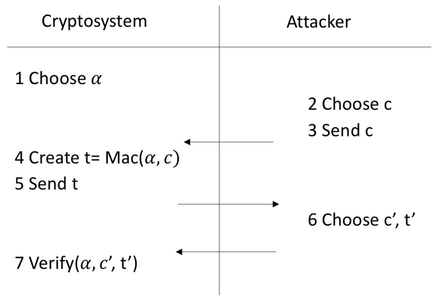

Figure 3: Message authentication experiment Mac-forge

3. The output of the experiment is defined to be 1 if and only if the adversary can output a new message and a correct tag, that is,

$$c' \neq c$$

$$Verify(\alpha, c', t') = 1$$

Question 3: When should we say the attacker won?
$c' \neq c$ an $Verify(\alpha, c', t') = True$
Question 4: $\forall A, Pr_\alpha[Mac\text{-}forge^{A,Mac} = 1]$ **=0?**
$\forall A, Pr_\alpha[Mac\text{-}forge^{A,Mac} = 1] < \epsilon$
Do we have any hope that the adversary never wins this game? There have to be some other m', t' pairs. For any particular message there must be at least one good tag t'. Thus, the adversary's success probability is at least $1/|t|$. Thus, our definition will now have a parameter. We'll say a scheme $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Vfy})$ is ϵ-unforgeable, if all adversaries $\mathcal{A}$ win the $\mathtt{Mac} - \mathtt{forge}$ game with probability at most ϵ. Or more formally,

Definition 2. *A scheme* $(\mathsf{Mac}, \mathsf{Vfy})$ *is ϵ-unforgeable under chosen message attack if*

$$\forall \mathcal{A}, \Pr_K[\mathtt{Mac} - \mathtt{forge}^{\mathcal{A},\mathsf{Mac}} = 1] < \epsilon.$$

The definition states that no adversary should succeed in the above experiment with probability greater than ϵ.

We'll now turn to trying to construct such an object. Informally our goal is the following. **Goal**: t' is independent of c, c', t.

References

[Ver19] Gilbert S Vernam. Secret signaling system, July 22 1919. US Patent 1,310,719.