In this lecture we mainly try to apply various relaxations to the definition of perfect secrecy. We would like to show that under these relaxations the notion of perfect secrecy still holds. For this purpose we present various modified definitions and prove that they are perfectly secret.

1 Separate Theorem

If \((\text{keyGen}, \text{Enc}, \text{Dec})\) is perfectly secret, then \(|k| \geq |m|\).

1.1 Proof

The condition of perfect secrecy specifies that, \(\forall m \in M, \forall c \in C\) for which \(Pr[C = c] > 0\):

\[
Pr[M = m | C = c] = Pr[M = m]
\]

We prove our stated theorem by contradiction. Let’s assume that \(|k| < |m|\). We begin by fixing a ciphertext. Let’s now look at

\[
m(c) = \{m | \exists k \in K \text{ such that } \text{Enc}_k(m) = c\}
\]

Roughly the above equation is saying that there exists a key \(k\) such that the encryption of the message \(m\) with \(k\) yields ciphertext \(c\). We are going to try to figure out which messages correspond to specific ciphertext. Looking at the set of decryption

\[
\{\text{Dec}_k(c) : k = K \text{ for } a \in C\}
\]

implies \(|m(c)| \leq |k| < |m|\).

To understand why it is less than or equal to the number of keys, realize that we are decrypting using all the keys so maybe we get same message twice but at best we will get a different message for each key and our assumption stated that number of keys is less than message so that explains the last inequality. This also suggests that there exists \(m\) that cannot be recovered from \(c\) meaning that \(m\) cannot be encrypted to \(c\) no matter what the key. Mathematically, \(\exists \tilde{m} \in M\) such that \(\tilde{m} \notin M(c)\). But then \(Pr[M = \tilde{m}|C = c] = 0 \neq Pr[M = \tilde{m}]\) and so the scheme is not perfectly secret. This is because the probability that the message \(\tilde{m}\) is the message becomes 0 if there is no way to produce cyphertext \(c\) from it. The conclusion of this proof is that in order for a scheme to be perfectly secret there must be at least as many possible keys as possible messages.

We need an alternate(equivalent) definition of perfect secrecy that can be relaxed. We are going to relax the definition so that it worries only about the practical attacks. Practical attacks are bound by computation and time.
2 Variation 1

An encryption scheme \((\text{Gen, Enc, Dec})\) over a message space \(\mathcal{M}\) is perfectly secret if and only if for every probability distribution over \(\mathcal{M}\), every message \(m \in \mathcal{M}\) and every ciphertext \(c \in \mathcal{C}\):

\[
Pr[C = c | M = m] = Pr[C = c]
\]

2.1 Proof

Fix a distribution over \(\mathcal{M}\) and arbitrary \(m \in \mathcal{M}\) and \(c \in \mathcal{C}\). Say

\[
Pr[C = c | M = m] = Pr[C = c]
\]

Multiplying both sides of the equation by \(\frac{Pr[M = m]}{Pr[C = c]}\) gives

\[
\frac{Pr[C = c | M = m] \cdot Pr[M = m]}{Pr[C = c]} = Pr[M = m]
\]

Using Baye’s theorem, the left hand-side is exactly equal to \(Pr[M = m | C = c]\). Thus, \(Pr[M = m | C = c] = Pr[M = m]\) and the scheme is perfectly secret.

To prove in the other direction we start with \(Pr[M = m | C = c] = Pr[M = m]\) and multiply both sides by \(\frac{Pr[C = c]}{Pr[M = m]}\). It gives

\[
\frac{Pr[M = m | C = c] \cdot Pr[C = c]}{Pr[M = m]} = Pr[C = c]
\]

which is equal to our starting equation, \(Pr[C = c | M = m] = Pr[C = c]\), hence proving both definitions equivalent.

3 Variation 2: Perfect Indistinguishability

We now use the above theorem to obtain another useful and equivalent formulation of perfect secrecy. This formulation states that the probability distribution over \(\mathcal{C}\) is independent of the plaintext. That is, let \(\mathcal{C}(m)\) denote the distribution of the ciphertext when the message being encrypted is \(m \in \mathcal{M}\) (this distribution depends on the choice of the key, as well as the randomness of the encryption algorithm in case it is probabilistic). Then the claim is that for every \(m_0, m_1 \in \mathcal{M}\) the distributions \(\mathcal{C}(m_0)\) and \(\mathcal{C}(m_1)\) are the same. This is just another way of saying that the ciphertext contains no information about the plaintext. We refer to this formulation as perfect indistinguishability because it implies that it is impossible to distinguish an encryption of \(m_0\) from an encryption of \(m_1\) (due to the fact that the distribution over the ciphertext is the same in each case).

3.1 Theorem

An encryption scheme \((\text{Gen, Enc, Dec})\) over a message space \(\mathcal{M}\) is perfectly secret if and only if for every probability distribution over \(\mathcal{M}\), every \(m_0, m_1 \in \mathcal{M}\), and every \(c \in \mathcal{C}\):

\[
Pr[C = c | M = m_0] = Pr[C = c | M = m_1]
\]
3.2 Proof

Assume that encryption scheme is perfectly secret and fix messages \( m_0, m_1 \in \mathcal{M} \) and ciphertext \( c \in \mathcal{C} \). By earlier theorem we have,

\[
Pr[C = c | M = m_0] = Pr[C = c] = Pr[C = c | M = m_1],
\]

completing the proof in the first direction. Assume next that for every distribution over \( \mathcal{M} \), every \( m_0, m_1 \in \mathcal{M} \), and every \( c \in \mathcal{C} \) it holds that \( Pr[C = c | M = m_0] = Pr[C = c | M = m_1] \). Fix some distribution over \( \mathcal{M} \), and an arbitrary \( m_0 \in \mathcal{M} \) and \( c \in \mathcal{C} \). Define \( p \overset{\text{def}}{=} Pr[C = c | M = m_0] \). Since \( Pr[C = c | M = m] = Pr[C = c | M = m_0] = p \) for all \( m \), we have

\[
Pr[C = c] = \sum_{m \in \mathcal{M}} Pr[C = c | M = m] \cdot Pr[M = m] = \sum_{m \in \mathcal{M}} p \cdot Pr[M = m] = p \cdot \sum_{m \in \mathcal{M}} Pr[M = m] = p = Pr[C = c | M = m_0].
\]

4 One Time Pad

Here we are going to use the One Time Pad as an example. First we will describe the system and then we will prove that it has perfect secrecy by Shannon Theorem.

4.1 Description

We will define the one time pad scheme as having:

\( \mathcal{M} = \{0, 1\}^l, \mathcal{K} = \{0, 1\}^l, \mathcal{C} = \{0, 1\}^l \) where \( l \) is the message length.

Keygen = outputs a uniform random key in \( \{0, 1\}^l \)

Encryption = \( E(m, k) = m \oplus k \)

Decryption = \( D(c, k) = c \oplus k \)

Alternatively it can be seen that \( m \oplus k = c \) so \( m \oplus k \oplus k = c \oplus k \)
4.2 Proving Perfect Secrecy

To prove perfect secrecy let us start with:
∀ \( m \in M \) and ∀ \( c \in C \) then \( Pr[M = m | C = c] = Pr[M = m] \)

\[
Pr[M = m | C = c] = \frac{Pr[C = c | M = m] \cdot Pr[M = m]}{Pr[C = c]}
\]

\[
= \frac{Pr[k \oplus m = c | M = m] \cdot Pr[M = m]}{\sum_{(m', k') \in S(c)} Pr[M = m' \land K = k']}
\]

\[
= \frac{\sum_{m' \in \{0,1\}^l} Pr[M = m'] \cdot Pr[K = c \oplus m]}{Pr[M = m]}
\]

\[
= \frac{\sum_{m' \in \{0,1\}^l} Pr[M = m']}{Pr[M = m]}
\]

Where the set of ciphertexts \( S(c) = \{(m', k')|m' \in M, k' \in K, m' \oplus k' = c\} \)

4.3 Recap

One Time Pad is a strong example because it offers provable perfect secrecy (by Shannon Theorem) at the cost of some serious limitations. The first being the one time use of keys, and the second being that the keys must be as long as the messages.

Keys must be regenerated or refreshed after each use in order to maintain secrecy because:

\[
m_0 \oplus m_1 = (K \oplus m_0 \oplus (K \oplus m_1)) = m_0 \oplus m_1
\]

Smaller keys can also not be used to encrypt larger messages. Despite any illusion of success, this would simply be exactly the same as splitting a larger message into chunks, and encrypting each chunk with an identical key. This not only doesn’t defeat the problem with key-length, but also runs into the problem of not regenerating keys.

5 Adversarial/Statistical Indistinguishability

We conclude this section by presenting another equivalent definition of perfect secrecy. The definition is based on an experiment involving an adversary \( A \), and formalizes \( A \)'s inability to distinguish the encryption of one plaintext from the encryption of another; we thus call it adversarial indistinguishability. This definition will serve as our starting point when we introduce the notion of computational security in the next chapter. Throughout the book we will often use experiments in order to define security. These "experiments" are essentially a game played between an adversary trying to break a cryptographic scheme and an imaginary tester who wishes to see if the adversary succeeds. We define an experiment that we call \( \text{PrivK}^{\text{eav}} \) since it considers the setting of private-key encryption and an eavesdropping adversary (the adversary is eavesdropping because it only receives
a ciphertext \( c \) and then tries to determine something about the plaintext). The experiment is defined for any encryption scheme \( \prod = (\text{Gen}, \text{Enc}, \text{Dec}) \) over message space \( \mathcal{M} \) and for any adversary \( \mathcal{A} \). We let \( \text{PrivK}_{A,\prod}^{\text{eav}} \) denote an execution of the experiment for a given \( \prod \) and \( \mathcal{A} \). The experiment is defined as follows:

1. The adversary \( \mathcal{A} \) outputs a pair of messages \( m_0, m_1 \in \mathcal{M} \).
2. The random key is generated by running \( \text{Gen} \), and a random bit \( b \leftarrow \{0,1\} \) is chosen. (These are chosen by some imaginary entity that is running the experiment with \( \mathcal{A} \).) Then, a ciphertext \( c \leftarrow \text{Enc}_k(m_b) \) is computed and given to \( \mathcal{A} \).
3. \( \mathcal{A} \) outputs a bit \( b' \).
4. The output of the experiment is defined to be 1 if \( b' = b \), and 0 otherwise. We write \( \text{PrivK}_{A,\prod}^{\text{eav}} = 1 \) if the output is 1 and in this case we say that \( \mathcal{A} \) succeeded.

One should think of \( \mathcal{A} \) as trying to guess the value of \( b' \) that is chosen in the experiment, and \( \mathcal{A} \) succeeds when its guess \( b' \) is correct. Observe that it is always possible for \( \mathcal{A} \) to succeed in the experiment with probability one half by just guessing \( b' \) randomly. The question is whether it is possible for \( \mathcal{A} \) to do better than this. The alternate definition that we give now states that an encryption scheme is perfectly secret if no adversary \( \mathcal{A} \) can succeed with probability any better than one half. We stress that, as is the case throughout this chapter, there is no limitation whatsoever on the computational power of \( \mathcal{A} \).

**An encryption scheme** \( (\text{Gen}, \text{Enc}, \text{Dec}) \) over a message space \( \mathcal{M} \) is perfectly secret if for every adversary \( \mathcal{A} \) it holds that

\[
\Pr[\text{PrivK}_{A,\prod}^{\text{eav}} = 1] = \frac{1}{2}
\]

**5.1 Proof**

if \( \prod \) is perfectly secret, then \( \prod \) is statistically indistinguishable. \( \forall \mathcal{A}, \forall m_0, m_1 \in \mathcal{M} \)

\[
\Pr_k[A(c) = b'] = \frac{1}{2}.
\]

We use the lemma:

\[
\Pr_k[A(c) = 1|M = m_0] = \Pr_k[A(c) = 1|M = m_1].
\]

Bit can be either 1 or 0.

\[
\Pr_k[A(c) = B] = \Pr_k[A(c) = B|B = 0] \cdot \Pr_k[B = 0] + \Pr_k[A(c) = B|B = 1] \cdot \Pr_k[B = 1]
\]

\[
\Pr_k[B = 0] = \Pr_k[B = 1] = \frac{1}{2}
\]

\[
\Pr_k[A(c) = B] = \frac{1}{2}\Pr_k[A(c) = B|B = 0] + \frac{1}{2}\Pr_k[A(c) = B|B = 1]
\]

\[
\frac{1}{2}[\Pr_k[A(c) = 0|B = 0] + \frac{1}{2}\Pr_k[A(c) = 1|B = 1]]
\]

\[
= \frac{1}{2}
\]

defining \( c = \text{Enc}_k(m_B) \)
5.2 More Formal Definitions of Perfect Secrecy

Now let us list some ways we can explicitly define perfect secrecy, since we have described it with multiple expressions:

\[
\begin{align*}
Pr[M = m] &= Pr[M = m|C = c] \\
\text{and} \\
Pr[C = c] &= Pr[C = c|M = m] \\
\text{and} \\
Pr[C = c|M = m_0] &= Pr[C = c|M = m_1]
\end{align*}
\]

References