Introduction to Security Reduction

Lecture 9: Security Proofs

(Encryption under Decisional Hardness Assumption)

My IQ is up to 186.

My interest is breaking schemes.

You want me to help you solve problem?

Fool me first!

Lecture 12: Flaws in Papers
Lecture 11: Revision of Security Reduction
Lecture 10: Security Proofs for Encryption (Computational)
Lecture 9: Security Proofs for Encryption (Decisional)
Lecture 8: Security Proofs for Digital Signatures
Lecture 7: Analysis (Towards A Correct Reduction)
Lecture 6: Simulation and Solution
Lecture 5: Difficulties in Security Reduction
Lecture 4: Entry to Security Reduction
Lecture 3: Preliminaries (Hard Problem and Secure Scheme)
Lecture 2: Preliminaries (Field, Group, Pairing, and Hash Function)
Lecture 1: Definitions (Algorithm and Security Model)

Computational Complexity Theory

Outline

Outline

Proof Structure

- **Simulation.** The simulator uses the problem instance (X, Z) to generate a simulated scheme.
- **Solution.** The simulator solves the decisional hard problem using the adversary's guess c' of c, where the message in the challenge ciphertext is m_c. Guessing Z is the same in all security reductions.
  - If $c' = c$, the simulator outputs that Z is true.
  - Otherwise, $c' \neq c$, and it outputs that Z is false.
- **Analysis.** In this part, we need to provide the following analysis.
  1. The simulation is indistinguishable from the real attack if Z is true.
  2. The probability P_S of successful simulation.
  3. The probability P_T of breaking the challenge ciphertext if Z is true.
  4. The probability P_F of breaking the challenge ciphertext if Z is false.
  5. The advantage ϵ_R of solving the underlying hard problem.
  6. The time cost of solving the underlying hard problem.

Classification of the Challenge Ciphertext

The target Z in the instance of the underlying decisional hard problem is either true or false. The challenge ciphertext can be classified into the following two types.

- **True Challenge Ciphertext.** The challenge ciphertext created with the target Z is a true challenge ciphertext if Z is true. We have that the probability of breaking the true challenge ciphertext is

$$P_T = \Pr[c' = c | Z = \text{True}].$$

- **False Challenge Ciphertext.** The challenge ciphertext created with the target Z is a false challenge ciphertext if Z is false. We have that the probability of breaking the false challenge ciphertext is

$$P_F = \Pr[c' = c | Z = \text{False}].$$

Advantage Analysis (1)

The advantage of solving the underlying hard problem is

$$\begin{aligned} & \epsilon_R \\ = \; & \Pr\left[\text{Guess } Z = \text{True} | Z = \text{ True}\right] - \Pr\left[\text{Guess } Z = \text{True} | Z = \text{ False}\right] \\ = \; & \Pr\left[\begin{array}{c}\text{The simulator guesses} \\ Z \text{ is true}\end{array} \middle| Z = \text{True}\right] - \Pr\left[\begin{array}{c}\text{The simulator guesses} \\ Z \text{ is true}\end{array} \middle| Z = \text{False}\right]. \end{aligned}$$

- Let US be the event of unsuccessful simulation. If the simulation is unsuccessful, the simulator will randomly guess Z by itself.

$$\Pr\left[\begin{array}{c}\text{The simulator guesses} \\ Z \text{ is true}\end{array} \middle| US\right] = \frac{1}{2}.$$

- Let SS be the event of successful simulation. We have

$$\Pr\left[\begin{array}{c}\text{The simulator guesses} \\ Z \text{ is true}\end{array} \middle| SS\right] = \Pr[c' = c].$$

Advantage Analysis (2)

By applying the law of total probability, we have

$$
\begin{aligned}
& \Pr\left[\begin{array}{c}\text{The simulator guesses}\\ Z \text{ is true}\end{array} \middle| Z = \text{True}\right] \\
= \; & \Pr\left[\begin{array}{c}\text{The simulator guesses}\\ Z \text{ is true}\end{array} \middle| Z = \text{True} \wedge SS\right] \Pr[SS] + \Pr\left[\begin{array}{c}\text{The simulator guesses}\\ Z \text{ is true}\end{array} \middle| Z = \text{True} \wedge US\right] \Pr[US] \\
= \; & \Pr[c' = c | Z = \text{True}] \Pr[SS] + \frac{1}{2} \Pr[US] \\
= \; & P_T \cdot P_S + \frac{1}{2}(1 - P_S).
\end{aligned}
$$

$$
\begin{aligned}
& \Pr\left[\begin{array}{c}\text{The simulator guesses}\\ Z \text{ is true}\end{array} \middle| Z = \text{False}\right] \\
= \; & \Pr\left[\begin{array}{c}\text{The simulator guesses}\\ Z \text{ is true}\end{array} \middle| Z = \text{False} \wedge SS\right] \Pr[SS] + \Pr\left[\begin{array}{c}\text{The simulator guesses}\\ Z \text{ is true}\end{array} \middle| Z = \text{False} \wedge US\right] \Pr[US] \\
= \; & \Pr[c' = c | Z = \text{False}] \Pr[SS] + \frac{1}{2} \Pr[US] \\
= \; & P_F \cdot P_S + \frac{1}{2}(1 - P_S).
\end{aligned}
$$

Advantage Analysis (3)

The above analysis yields the advantage of solving hard problem

$$
\begin{aligned}
& \epsilon_R \\
= \quad & \Pr\left[\begin{array}{c}\text{The simulator guesses} \\ Z \text{ is true}\end{array} \middle| Z = \text{True}\right] - \Pr\left[\begin{array}{c}\text{The simulator guesses} \\ Z \text{ is true}\end{array} \middle| Z = \text{False}\right] \\
= \quad & \left(P_T \cdot P_S + \frac{1}{2}(1 - P_S)\right) - \left(P_F \cdot P_S + \frac{1}{2}(1 - P_S)\right) \\
= \quad & P_S(P_T - P_F).
\end{aligned}
$$

To solve a decisional hard problem with non-negligible advantage, we should program the security reduction in such a way that

- P_S is non-negligible.
- P_T is as large as possible, and
- P_F is as small as possible.

On the contrary, to make the security reduction fail, the aim of the malicious adversary is to achieve $P_T \approx P_F$ (useless attack).

Towards Non-Negligible Advantage

- **Correct Ciphertext.** A ciphertext is correct if it can be generated by the encryption algorithm.

  For example, taking as input $pk = (g, g_1, g_2) \in \mathbb{G}$ and message $m \in \mathbb{G}$, an encryption algorithm randomly chooses $r \in \mathbb{Z}_p$ and computes $CT = (g^r,\ g_1^r,\ g_2^r \cdot m)$. Any ciphertext that can be generated with a message m and a number r for pk is a correct ciphertext.

- **Incorrect Ciphertext.** A ciphertext is incorrect if it cannot be generated by the encryption algorithm.

  Continued from the above example, $(g^r, g_1^{r+1}, g_2^r \cdot m)$ is an incorrect ciphertext because it cannot be generated by the encryption algorithm with (pk, m) as the input using any random number.

Towards Non-Negligible Advantage

$$\epsilon_R = P_S(P_T - P_F).$$

$$P_T = \Pr[c' = c | Z = \text{True}] \quad (1)$$
$$P_F = \Pr[c' = c | Z = \text{False}] \quad (2)$$

- If the challenge ciphertext is generated and independent of Z, the guess of the message in the challenge ciphertext should be independent of Z, and thus $P_T = P_F$.
- Therefore, the simulator must embed the target Z in the challenge ciphertext such that it satisfies $P_T \neq P_F$. (The value Z affects the adversary's success probability.)

Towards Non-Negligible Advantage

$$\epsilon_R = P_S(P_T - P_F).$$

$$P_T = \Pr[c' = c | Z = \text{True}] \quad (3)$$

$$P_F = \Pr[c' = c | Z = \text{False}] \quad (4)$$

- If Z is true, the true challenge ciphertext is a correct ciphertext whose encrypted message is $m_c \in \{m_0, m_1\}$, and c is randomly chosen by the simulator. We should program indistinguishable simulation and then the adversary can guess the encrypted message correctly with non-negligible advantage defined in the breaking assumption.
- If Z is false, the false challenge ciphertext can be either a correct ciphertext or an incorrect ciphertext. However, the challenge ciphertext cannot be an encryption of the message m_c from the point of view of the adversary. We program the simulation in such a way that the adversary cannot guess the encrypted message correctly except with negligible advantage.

Outline

Outline

Probability P_T

- Suppose there exists an adversary who can break the proposed scheme in polynomial time with non-negligible advantage ϵ.
- If the message m_c is encrypted in the real scheme, according to the definition of the advantage in the security model, we have

$$\epsilon = 2\left(\Pr[c' = c] - \frac{1}{2}\right).$$

- That is, the adversary can correctly guess the message in the challenge ciphertext of the real scheme with probability

$$\Pr[c' = c] = \frac{1}{2} + \frac{\epsilon}{2}.$$

If Z is true and the simulation is indistinguishable, the adversary will correctly guess the encrypted message in the challenge ciphertext with probability $\frac{1}{2} + \frac{\epsilon}{2}$. That is, we have

$$P_T = \Pr[c' = c | Z = \text{True}] = \frac{1}{2} + \frac{\epsilon}{2}.$$

Probability P_F

- The challenge ciphertext in the real scheme should be a correct ciphertext whose encrypted message is from $\{m_0, m_1\}$.
- If Z is false, the false challenge cipheretext should be an incorrect ciphertext or a correct ciphertext whose encrypted message is not equal to m_c.
- Therefore, the adversary knows that the given scheme is not a real scheme, but a simulated scheme. (Distinguishable)

Since the adversary is malicious, it will not abort but try its best to guess c using what it knows in order to have $P_F \approx P_T$. Therefore,

$$P_F = \Pr[c' = c | Z = \text{False}] \geq \frac{1}{2},$$

which is highly dependent on the simulation and is no smaller than $\frac{1}{2}$. The probability $\frac{1}{2}$ is obtained by a random guess.

Question: What happen if the adversary simply abort?

Advantage Analysis

$$\epsilon_R = P_S(P_T - P_F) = P_S\left(\frac{1}{2} + \frac{\epsilon}{2} - P_F\right).$$

The advantage is non-negligible if and only if $P_F \approx \frac{1}{2}$. Otherwise, the advantage of solving the hard problem will be zero.

A correct security reduction requires that

- P_S is non-negligible.
- The simulated scheme is indistinguishable if Z is true.
- $P_F \approx \frac{1}{2}$, which means that the malicious adversary has almost no advantage in breaking the false challenge ciphertext.

The ideal probability $P_F = \frac{1}{2}$ holds if and only if the message m_c is encrypted with a one-time pad from the point of view of the adversary.

Outline

One-Time Pad

Definition (One-Time Pad)

Let $E(m, r, R)$ be an encryption of a given message m with a public parameter R and a secret parameter r. The ciphertext $E(m, r, R)$ is a one-time pad if, for any two distinct messages m_0, m_1 from the same message space, CT can be seen as

- an encryption of message m_0 with the secret parameter r_0, or
- an encryption of message m_1 with the secret parameter r_1

under the public parameter R with the same probability:

$$\Pr\left[CT = E(m_0, r_0, R)\right] = \Pr\left[CT = E(m_1, r_1, R)\right].$$

Note: We need to prove that the false challenge ciphertext is a one-time pad from the point of view of the adversary.

One-Time Pad

Lemma

Let CT be a general ciphertext defined as follows, where $m_c \in \{m_0, m_1\}$ and the adversary knows the group $(\mathbb{G}, g, p)$ and messages $m_0, m_1 \in \mathbb{G}$:

$$CT = \left(g^{x_1},\ g^{x_2},\ g^{x_3},\ \cdots,\ g^{x_n},\ g^{x^*} \cdot m_c\right).$$

The ciphertext CT is a one-time pad if

- *x^* is a random number from $\mathbb{Z}_p$, and*
- *x^* is independent of $x_1, x_2, \cdots, x_n$.*

The ciphertext CT is not a one-time pad if

- *x^* is not a random number from $\mathbb{Z}_p$, or*
- *x^* is dependent on $x_1, x_2, \cdots, x_n$.*

One-Time Pad: Example (1)

Suppose the adversary knows the following information.

- The cyclic group $(\mathbb{G}, g, h, p)$.
- Two messages m_0, m_1 from $\mathbb{G}$.
- How the ciphertext is created.

In the following ciphertexts, $c \in \{0, 1\}$ and $x, y, z \in \mathbb{Z}_p$ are randomly chosen by the simulator. The aim of the adversary is to guess c in CT.

$$CT = \left(g^x, \ g^4 \cdot m_c\right) \tag{5}$$

$$CT = \left(g^x, \ g^y \cdot m_c\right) \tag{6}$$

One-Time Pad: Example (1)

Suppose the adversary knows the following information.

- The cyclic group $(\mathbb{G}, g, h, p)$.
- Two messages m_0, m_1 from $\mathbb{G}$.
- How the ciphertext is created.

In the following ciphertexts, $c \in \{0, 1\}$ and $x, y, z \in \mathbb{Z}_p$ are randomly chosen by the simulator. The aim of the adversary is to guess c in CT.

$$CT = \left(g^x, \ g^4 \cdot m_c\right) \tag{5}$$

$$CT = \left(g^x, \ g^y \cdot m_c\right) \tag{6}$$

- **5 No.** We have $x^* = 4$ which is not random.
- **6 Yes.** We have $(x_1, x^*) = (x, y)$. x^* is random and independent of x_1, because (x, y) are both random numbers.

One-Time Pad: Example (2)

$$CT = \left(g^x,\ h^x \cdot m_c\right) \tag{7}$$

$$CT = \left(g^x,\ g^y,\ g^{xy} \cdot m_c\right) \tag{8}$$

One-Time Pad: Example (2)

$$CT = \left(g^x,\ h^x \cdot m_c\right) \tag{7}$$

$$CT = \left(g^x,\ g^y,\ g^{xy} \cdot m_c\right) \tag{8}$$

- **7 No.** We have

$$(x_1, x^*) = (x, x \log_g h).$$

  x^* is dependent on x_1 and $\log_g h$, satisfying $x^* = x_1 \log_g h$.

- **8 No.** We have

$$(x_1, x_2, x^*) = (x, y, xy).$$

  x^* is dependent on x_1 and x_2, satisfying the equation $x^* = x_1 x_2$.

One-Time Pad: Example (3)

$$CT = \left(g^x,\ h^{x+y} \cdot m_c\right) \tag{9}$$

$$CT = \left(g^{2x+y+z},\ g^{x+3y+z},\ g^{4x+7y+3z} \cdot m_c\right) \tag{10}$$

One-Time Pad: Example (3)

$$CT = \left(g^x,\ h^{x+y} \cdot m_c\right) \tag{9}$$

$$CT = \left(g^{2x+y+z},\ g^{x+3y+z},\ g^{4x+7y+3z} \cdot m_c\right) \tag{10}$$

- **9 Yes.** We have

$$(x_1, x^*) = (x, x \log_g h + y \log_g h).$$

  x^* is independent of x_1, because y is a random number that only appears in x^*.

- **10 No.** We have

$$(x_1, x_2, x^*) = (2x + y + z, x + 3y + z, 4x + 7y + 3z).$$

  The determinant of the coefficient matrix is zero, and x^* is dependent on (x_1, x_2), satisfying $x^* = x_1 + 2x_2$.

Analysis of One-Time Pad

- It is not sufficient to prove that the false challenge ciphertext is a one-time pad.
- Instead, we must prove that breaking the false challenge ciphertext is as hard as breaking a one-time pad from the point of view of the adversary (based on what it knows).
- We have to analyze in this way because (public key and decryption results in CCA security) might help the adversary break the false challenge ciphertext.

Analysis of One-Time Pad

For example:

- The following ciphertext is a one-time pad.

$$CT = \left(g^x, \quad h^{x+y} \cdot m_c\right)$$

- Suppose the public key is $pk = (g, h, g^y)$.
- The above ciphertext is no longer a one-time pad.

Outline

Ciphertexts

We define valid ciphertext and invalid ciphertext associated with decryption result.

- **Valid Ciphertext.** A ciphertext is valid if the decryption of the ciphertext returns a message.
- **Invalid Ciphertext.** A ciphertext is invalid if the decryption of the ciphertext returns an error $\perp$, without returning any message.

Note: We stress that the message returned from the decryption on valid ciphertext can be any message as long as the output is not $\perp$.

Outline

Simulation of Decryption: Z is True

$$\epsilon_R = P_S(P_T - P_F) = P_S\left(\frac{1}{2} + \frac{\epsilon}{2} - P_F\right).$$

If Z is true, the simulation of decryption requires that

- The adversary will still break the challenge ciphertext with

$$\frac{1}{2} + \frac{\epsilon}{2}.$$

- Therefore, the decryption simulation should be indistinguishable. That is, the simulated scheme will accept correct ciphertexts and reject incorrect ciphertexts the same as the real scheme.
- We don't care whether the decryption simulation helps the adversary in breaking the challenge ciphertext.

Simulation of Decryption: Z is True

Indistinguishable simulation of decryption:

- The simulator can decrypt ciphertexts queried by the adversary.
- The decryption result by simulator and by challenger are the same.

The meaning of "the same"

- A queried ciphertext rejected by challenger should be also rejected by the simulator.
- A queried ciphertext accepted by challenger should be also accepted by the simulator.

Outline

Simulation of Decryption: Z is False

$$\epsilon_R = P_S(P_T - P_F) = P_S\left(\frac{1}{2} + \frac{\epsilon}{2} - P_F\right).$$

If Z is false, the simulation of decryption requires that

- P_F is still as small as $\frac{1}{2}$.
- We do care whether the decryption simulation helps the adversary in breaking the challenge ciphertext.
- We don't care that the decryption simulation is distinguishable due to a false Z.

Simulation of Decryption: Z is False

We care whether decryption results from decryption queries will help the adversary break the false challenge ciphertext.

The queried ciphertext from the adversary can be

- Correct ciphertext (can be generated by encryption algorithm). The simulator must accept correct ciphertexts as valid ciphertexts. Otherwise, the simulation is distinguishable if Z is true.
- Incorrect ciphertext (cannot be generated by encryption algorithm). **This case is a bit complicated because the simulator can either accept it or reject it depending on scheme and reduction.**

Simulation of Decryption: Z is False

A CCA-secure encryption scheme must be well constructed satisfying the following requirements.

- Decryption on correct ciphertexts cannot help the adversary.
- Decryption on incorrect ciphertexts
  - cannot help the adversary, or
  - can help the adversary but they will be rejected, or
  - can help the adversary but are accepted with negligible probability.

Simulation of Decryption: Z is False

We can simplify the analysis for **some encryption schemes** as follows.

Decryption on correct ciphertexts cannot help the adversary.

\+

Decryption on incorrect ciphertexts cannot help the adversary.

=

Decryption key cannot help the adversary.

Example (1)

$pk = (g, h)$ and $sk = \alpha$ where $h = g^{\alpha}$ and $\alpha \in \mathbb{Z}_p$ is randomly chosen.

- Suppose the false challenge ciphertext is equal to

$$CT^* = (C_1^*, C_2^*) = \left(g^x,\ Z \cdot m_c\right),$$

  where the target (false) Z is randomly chosen from $\mathbb{G}$.

- This challenge ciphertext is a one-time pad from the point of view of the adversary if and only if Z is random and unknown to the adversary.

Example (1)

$pk = (g, h)$ and $sk = \alpha$ where $h = g^{\alpha}$ and $\alpha \in \mathbb{Z}_p$ is randomly chosen.

- Suppose the false challenge ciphertext is equal to

$$CT^* = (C_1^*, C_2^*) = \left(g^x,\ Z \cdot m_c\right),$$

  where the target (false) Z is randomly chosen from $\mathbb{G}$.

- This challenge ciphertext is a one-time pad from the point of view of the adversary if and only if Z is random and unknown to the adversary.

- Even if the challenge decryption key α can be computed by the adversary, it doesn't help the adversary guess the encrypted message in the challenge ciphertext.

Example (2)

$pk = (g, g_1, g_2, g_3, h)$ and $sk = (\alpha, \beta, \gamma)$ where $h = g_1^{\alpha} g_2^{\beta} g_3^{\gamma}$.

- Suppose the false challenge ciphertext is equal to

$$CT^* = (C_1^*, C_2^*, C_3^*) = \left(g_1^x,\ Z,\ Z^{\alpha} \cdot m_c\right),$$

  where the (false) target Z is randomly chosen from $\mathbb{G}$. The message is encrypted with Z^{α}, and Z is also given in the ciphertext.

- Therefore, this challenge ciphertext is a one-time pad from the point of view of the adversary if and only if α is random and unknown to the adversary.

Example (2)

$pk = (g, g_1, g_2, g_3, h)$ and $sk = (\alpha, \beta, \gamma)$ where $h = g_1^{\alpha} g_2^{\beta} g_3^{\gamma}$.

- Suppose the false challenge ciphertext is equal to

$$CT^* = (C_1^*, C_2^*, C_3^*) = \left(g_1^x,\ Z,\ Z^{\alpha} \cdot m_c\right),$$

  where the (false) target Z is randomly chosen from $\mathbb{G}$. The message is encrypted with Z^{α}, and Z is also given in the ciphertext.

- Therefore, this challenge ciphertext is a one-time pad from the point of view of the adversary if and only if α is random and unknown to the adversary.

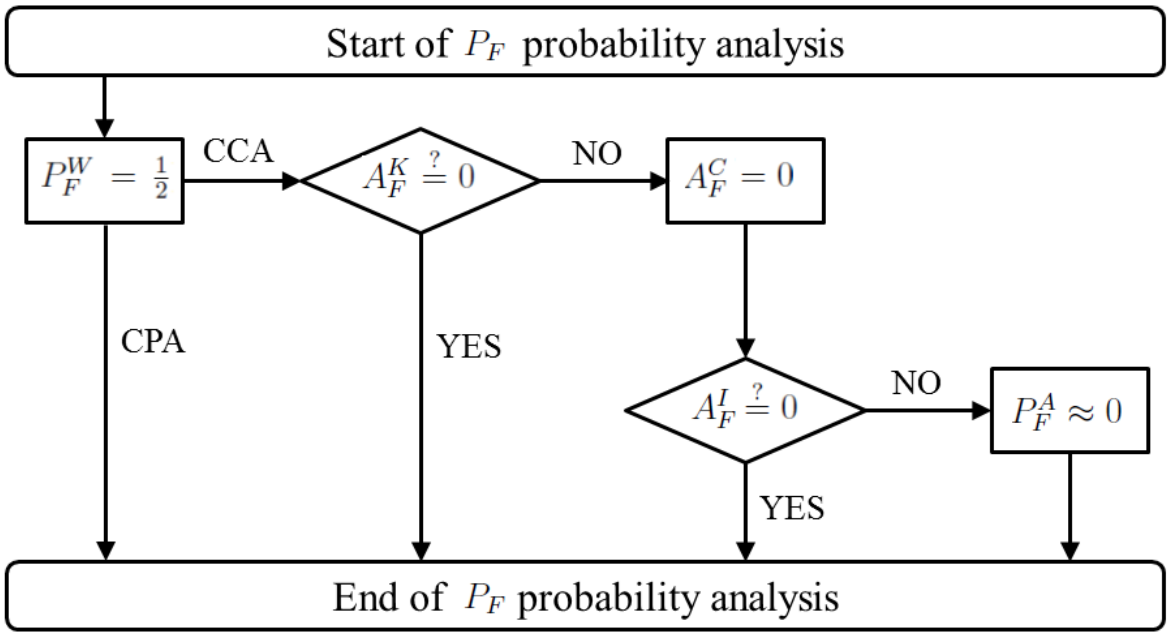
- It is easy to see that with the help of (α, β, γ), the adversary can easily break the false challenge ciphertext.

Outline

Advantage and Probability

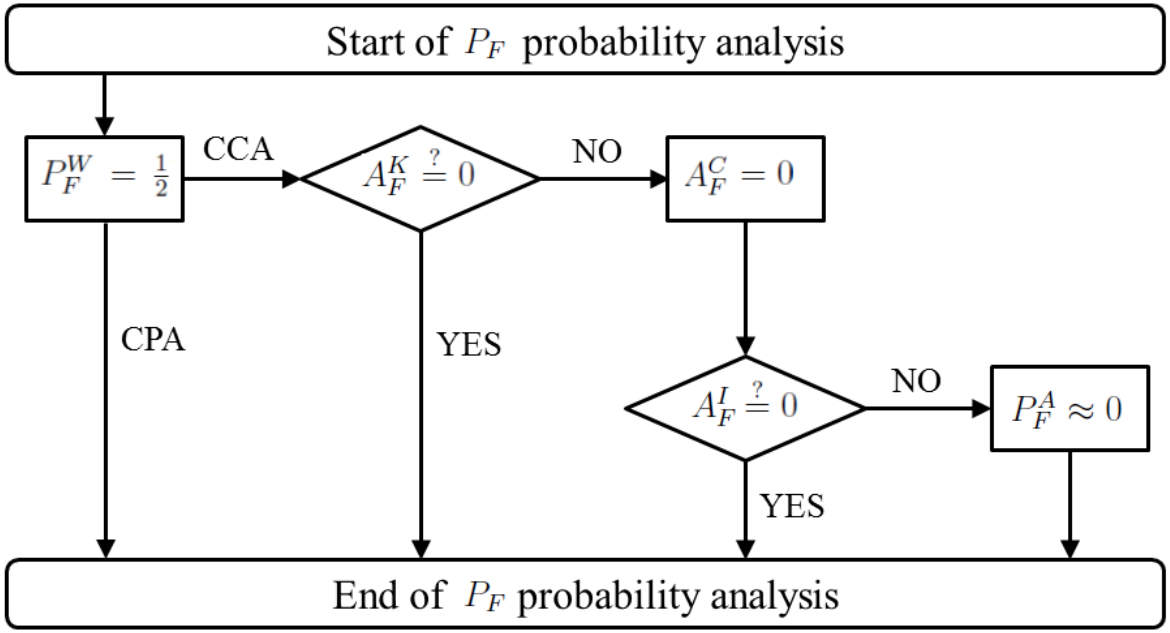
- **Probability $P_F^W = \frac{1}{2}$.** The probability P_F^W of breaking the false challenge ciphertext without any decryption query is $\frac{1}{2}$.
- **Advantage $A_F^K \stackrel{?}{=} 0$.** The advantage A_F^K of breaking the false challenge ciphertext with the help of the challenge decryption key is either 0 or 1.
- **Advantage $A_F^C = 0$.** The advantage A_F^C of breaking the false challenge ciphertext with the help of decryption queries on correct ciphertexts is 0.
- **Advantage $A_F^I \stackrel{?}{=} 0$.** The advantage A_F^I of breaking the false challenge ciphertext with the help of decryption queries on incorrect ciphertexts is either 0 or 1.
- **Probability $P_F^A \approx 0$.** The probability P_F^A of accepting an incorrect ciphertext for decryption query is negligible.

Advantage and Probability



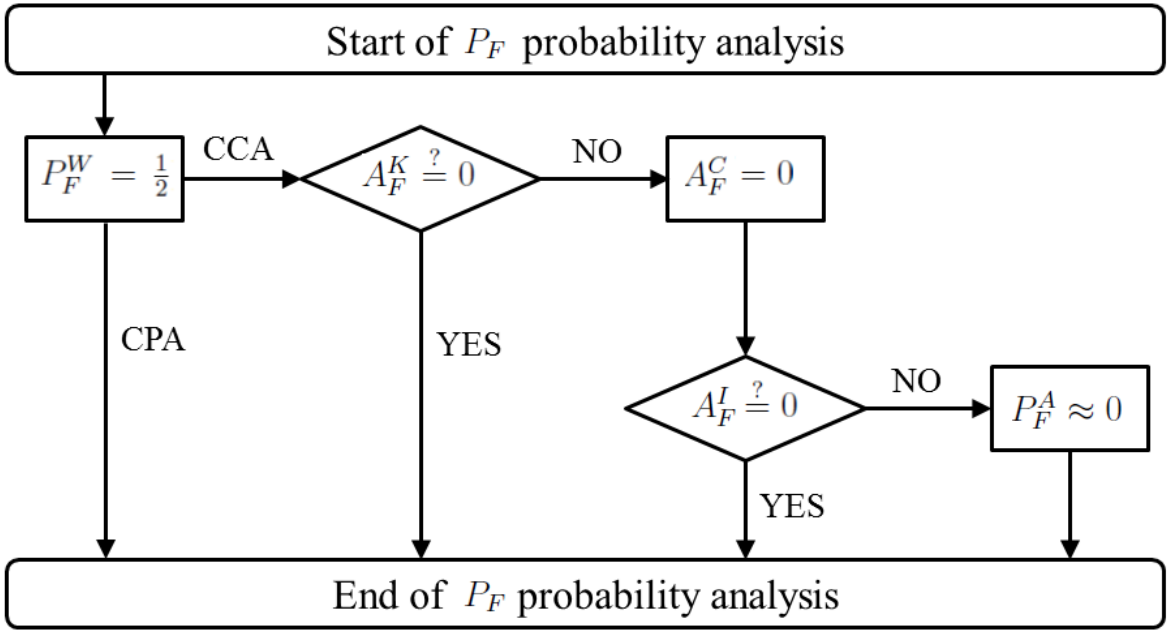
- **Probability $P_F^W = \frac{1}{2}$.** The probability P_F^W of breaking the false challenge ciphertext without any decryption query is $\frac{1}{2}$.

Advantage and Probability



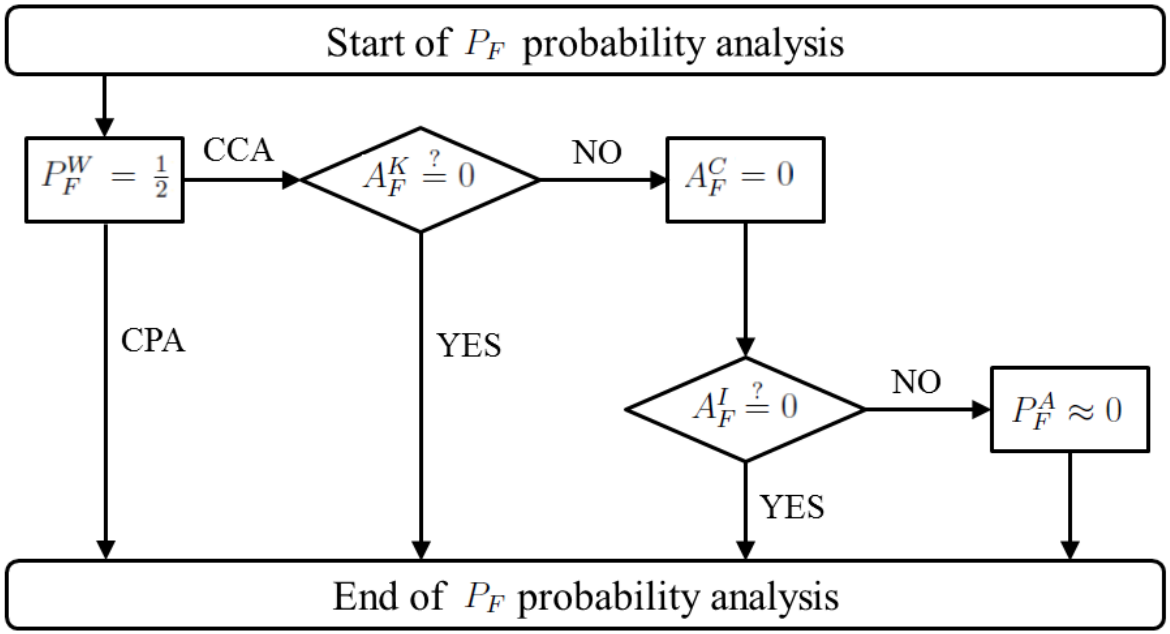
- **Advantage $A_F^K \stackrel{?}{=} 0$.** The advantage A_F^K of breaking the false challenge ciphertext with the help of the challenge decryption key is either 0 or 1.

Advantage and Probability



- **Advantage $A_F^C = 0$.** The advantage A_F^C of breaking the false challenge ciphertext with the help of decryption queries on correct ciphertexts is 0.

Advantage and Probability



- **Advantage $A_F^I \stackrel{?}{=} 0$.** The advantage A_F^I of breaking the false challenge ciphertext with the help of decryption queries on incorrect ciphertexts is either 0 or 1.
- **Probability $P_F^A \approx 0$.** The probability P_F^A of accepting an incorrect ciphertext for decryption query is negligible.

Outline

Simulation of Challenge Decryption Key

It is not necessary for the simulator to program the simulation without knowing the challenge decryption key. Currently, there are two methods.

- In the first method, the simulator knows the challenge decryption key. It is easy for the simulator to simulate the decryption by following the decryption algorithm. It is not easy to simulate the challenge ciphertext. Otherwise, the simulator can use the known key to decrypt the challenge ciphertext and solve hard problem.
- In the second method, the simulator doesn't know the challenge decryption key. It is not easy to simulate the decryption, but the simulator has to be able to simulate the decryption.

We cannot adaptively choose one of them to program a security reduction for a proposed scheme. Which method can be used is dependent on the proposed scheme and the underlying hard problem.

Summary

A correct security reduction should satisfy the following conditions.

- The underlying hard problem is a decisional problem.
- The simulator uses the adversary's guess to solve hard problem.
- The simulation is indistinguishable from the real attack if Z is true.
- The probability of successful simulation is non-negligible.
- The advantage of breaking the true challenge ciphertext is ϵ.
- The advantage of breaking the false challenge ciphertext is negligible.
- The advantage ϵ_R of solving hard problem is non-negligible.
- The time cost of the simulation is polynomial time.


END