Demonstrating Attacks Against Encryption Schemes

October 16, 2022

1 How to Formally Describe Attacks

To formally describe an attack against an encryption scheme you need to write an algorithm for an adversary then analyze your adversary to show that it wins the security game with non-negligible advantage. What security game the adversary is playing depends on the *security definition* that you want to disprove. Each security definition has its own security game.

- **One-time uniform ciphertext security**: In this game the adversary chooses a message m and sends it to the challenger. The challenger then flips a bit b, and if $b == 0$ sends back $c = \mathsf{Enc}(\mathsf{KeyGen}(1^n), m)$. If $b == 1$ the challenger instead samples a random ciphertext $c \overset{\$}{\leftarrow} \mathcal{C}$ and sends c to the adversary. The adversary then guesses a bit b', and wins the game if $b == b'$. The security property states that no adversary can win this game with probability greater than $1/2$.

- **One-time uniform perfect security**: In this game the adversary chooses two messages (m_0, m_1) and sends them to the challenger. The challenger then flips a bit b, and if $b == 0$ sends back $c = \mathsf{Enc}(\mathsf{KeyGen}(1^n), m_0)$. If $b == 1$ the challenger instead sends back $c = \mathsf{Enc}(\mathsf{KeyGen}(1^n), m_1)$. The adversary then guesses a bit b', and wins the game if $b == b'$. The security property states that no adversary can win this game with probability greater than $1/2$.

- **Indistinguishability Security**: This game is similar to the above, except the adversary gets to *adaptively* query *multiple* pairs of messages. At the start of the game the challenger flips a bit b, and samples a key $k \leftarrow \mathsf{KeyGen}(1^n)$. Then, the adversary gets to query a pair of messages (m_0^1, m_1^1). If $b == 0$, the challenger sends back $c = \mathsf{Enc}(k, m_0)$. If $b == 1$ the challenger instead sends back $c = \mathsf{Enc}(k, m_1)$. Then, the adversary may query another pair of messages (m_0^2, m_1^2) and receive a response generated in the same way. The game then proceeds in this way until the adversary decides to stop querying messages and outputs a bit b'. The adversary wins the game if $b == b'$. Note that for the adversary to run in polynomial time it can only query a polynomial number of message pairs. The security property states that no PPT adversary can win this game with probability greater than $1/2 + \mathsf{negl}(n)$, where n is the security parameter.

For **One-time uniform ciphertext security** and **One-time perfect security** you need to prove that for your adversary $\mathcal{A}$:

$$\Pr[\mathcal{A} \text{ correctly guesses } b] > 1/2$$

This is equivalent to proving:

$$|\Pr[\mathcal{A} \text{ outputs } 0 | b == 0] - \Pr[\mathcal{A} \text{ outputs } 0 | b == 1]| > 0$$

Which is often easier to analyze.

For **Indistinguishability security** you need to prove that for your adversary $\mathcal{A}$:

$$\Pr[\mathcal{A} \text{ correctly guesses } b] > 1/2 + \mathsf{negl}(n)$$

This is equivalent to proving:

$$|\Pr[\mathcal{A} \text{ outputs } 0 | b == 0] - \Pr[\mathcal{A} \text{ outputs } 0 | b == 1]| > \mathsf{negl}(n)$$

Which is often easier to analyze.

Every attack proof has three components:

1. Determine which security notion you are disproving
2. Write an adversary attacking the scheme in the game defined by the security notion
3. Analyze the advantage of your adversary, showing that it is sufficient to disprove the security notion.

2 Example

Let's describe an insecure encryption scheme and then show an attack on it.

Our encryption scheme is as follows:

- $\mathsf{KeyGen}(1^n)$:
  - $k \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{Enc}(k, m)$:
  - $c := (k \oplus m) || m[0]$
- $\mathsf{Dec}(k, c)$:
  - $m = k \oplus m[0, n]$

i.e. just OTP but leaking the first bit of the message.

2.1 Attack Against One-time Uniform Ciphertext Security

First we can demonstrate an adversary showing that the scheme does not satisfy **One-time Uniform Ciphertext Security**.

- $\mathcal{A}()$:
  - Set $m := 0^n$
  - Send m to the challenger and get back c
  - If $c[n] == 0$ output 0
  - Else output 1

Finally, we can analyze the advantage of our adversary:

$\Pr[\mathcal{A}$ outputs $0 | b == 0] = 1$, as the last bit of the ciphertext will always be 0 when $b == 0$.

$\Pr[\mathcal{A}$ outputs $0 | b == 1] = 1/2$, as when $b == 1$ the ciphertext is sampled randomly, and so the probability that the last bit is 0 is $1/2$.

Therefore, the advantage of our adversary is $1 - 1/2 = 1/2$, which is clearly greater than 0, and the proof is complete.

2.2 Attack Against One-time Perfect Security

Now, we can demonstrate an adversary showing that the scheme does not satisfy **One-time Perfect Security**.

- $\mathcal{A}()$:
  - Set $m_0 := 0^n$
  - Set $m_1 := 1^n$
  - Send (m_0, m_1) to the challenger and get back c
  - If $c[n] == 0$ output 0
  - Else output 1

Finally, we can analyze the advantage of our adversary:

$\Pr[\mathcal{A}$ outputs $0 | b == 0] = 1$, as the last bit of the ciphertext will always be 0 when $b == 0$.

$\Pr[\mathcal{A}$ outputs $0 | b == 1] = 0$, as the last bit of the ciphertext will always be 1 when $b == 1$.

Therefore, the advantage of our adversary is $1 - 0 = 1$, which is clearly greater than 0, and the proof is complete.