



Concrete Security of Symmetric-Key Encryption

Breno de Medeiros

Department of Computer Science
Florida State University

Security of Encryption

- ⑥ The “gold standard” security definition for encryption schemes is security against adaptive chosen-ciphertext attacks (IND-CCA2).
- ⑥ Weaker definitions are useful as well. For instance, proving security against (adaptively) chosen-plaintext attacks (IND-CPA) is often used as a “stepping stone” in the analysis of schemes that are secure under more stringent criteria (e.g, IND-CCA2).
- ⑥ The two definitions can be given in the same framework. As before, if K is the key for a symmetric encryption scheme, then $\mathcal{O}^K(\cdot)$ represents an encryption oracle, while $\mathcal{O}^{K^{-1}}(\cdot)$, a decryption oracle.

Several security definitions

- ⑥ In [1], four different adversarial-game frameworks are considered:
 - △ Left-or-Right (LoR) security, where an adversary must decide if the simulation always encrypts the adversary's left or right message;
 - △ Real-or-Random (RoR) security, where an adversary must decide if the simulation always provides a correct encryption or random values.
 - △ Find-Then-Guess (FtG) security, where an adversary plays a traditional two-stage game, with the goal to distinguish which of two chosen messages is encrypted in the simulation.
 - △ Semantic (SEM) security, where an adversary wins by computing some non-trivial function of the plaintext, given its ciphertext.
- ⑥ It is shown that all four notions are equivalent in terms of asymptotic security.

Definition of LoR experiment

- ⑥ A symmetric scheme \mathcal{SE} is given by its key generation \mathcal{K} , encryption \mathcal{E} , and decryption \mathcal{D} algorithms. At the beginning of each experiment, the simulator chooses a random bit b . $\text{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk-}b}$, where atk is one of cpa or cca , is defined as follows.
- ⑥ The simulator generates a new symmetric key K according to algorithm \mathcal{K} , then instantiates the encryption oracle $\mathcal{O}^{\text{LoR}, b, K}(\cdot)$ to respond to message-pair queries (m_0^i, m_1^i) by returning the encryption of m_b^i under K : $\mathcal{O}^{\text{LoR}, b, K}(m_0^i, m_1^i) = \mathcal{E}(K, m_b^i)$. If $\text{atk} = \text{cca}$, the decryption oracle $\mathcal{O}^{K^{-1}}(\cdot)$ is instantiated, and decrypts queried ciphertexts—however, it refuses to decrypt ciphertexts that have been previously computed as queries to the encryption oracle.
- ⑥ At the end of the simulation, the adversary outputs a guess bit b' of the value b used by the simulator.

Definition of LoR security

- ⑥ A (ϵ, t, q, μ, k) -LoR adversary for a symmetric scheme \mathcal{SE} is a probabilistic algorithm that:
 - △ on instances of security parameter k ;
 - △ using at most q queries to the oracles;
 - △ receiving at most μ bits total in the answers to its queries;
 - △ taking at most t computational steps;
 - △ Achieves advantage $\mathbf{Adv}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk}}$ at least ϵ , where

$$\mathbf{Adv}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk}} := \text{Prob} \left[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk-1}} = 1 \right] - \text{Prob} \left[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk-0}} = 1 \right].$$

- ⑥ Naturally, a symmetric scheme \mathcal{SE} is (ϵ, t, q, μ, k) -LoR secure if no (ϵ, t, q, μ, k) -LoR adversaries exist against it.

Definition of RoR experiment

- ⑥ Again, \mathcal{SE} is given by $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ algorithms. As before, at the beginning of each RoR experiment, the simulator chooses a random bit b . $\text{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{ror-atk-}b}$, where atk is one of cpa or cca , is defined as follows.
- ⑥ The simulator generates a new symmetric key K according to algorithm \mathcal{K} , then instantiates the encryption oracle $\mathcal{O}^{\text{RoR}, b, K}(\cdot)$ to respond to encryption queries m^i by returning either the requested encryption $\mathcal{E}(K, m^i)$ if $b = 1$, or that of a new random value $\mathcal{E}(K, r^i)$, if $b = 0$, where r^i has same length as m^i . If $\text{atk} = \text{cca}$, the decryption oracle $\mathcal{O}^{K^{-1}}(\cdot)$ is instantiated, and decrypts queried ciphertexts. Again, it refuses to decrypt ciphertexts previously generated by the encryption oracle.
- ⑥ At the end of the simulation, the adversary outputs a guess bit b' of the value b used by the simulator.

Definition of RoR security

- ⑥ A (ϵ, t, q, μ, k) -RoR adversary for a symmetric scheme \mathcal{SE} is a probabilistic algorithm that:
 - △ on instances of security parameter k ;
 - △ using at most q queries to the oracles;
 - △ receiving at most μ bits total in the answers to its queries;
 - △ taking at most t computational steps;
 - △ Achieves advantage $\mathbf{Adv}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{ror-atk}}$ at least ϵ , where

$$\mathbf{Adv}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{ror-atk}} := \text{Prob} \left[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{ror-atk-1}} = 1 \right] - \text{Prob} \left[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{ror-atk-0}} = 1 \right].$$

- ⑥ Naturally, a symmetric scheme \mathcal{SE} is (ϵ, t, q, μ, k) -RoR secure if no (ϵ, t, q, μ, k) -RoR adversaries exist against it.

LoR \geq RoR security

- ⑥ LoR and RoR security can be shown to be equivalent in a tight, concrete sense. Let \mathcal{A} be a RoR-adversary. A simulator \mathcal{S} can use it to implement an attacker \mathcal{B} against a LoR-game, as follows. If m^i is the i -th encryption query requested by \mathcal{A} , the simulator can forward the pair $(m_0^i, m_1^i) = (r^i, m^i)$ as \mathcal{B} 's query, where r^i is a newly generated random value. The simulator forwards replies from $\mathcal{O}^{LoR,b,K}(\cdot)$ to \mathcal{B} to \mathcal{A} .
- ⑥ Note that, if $\mathcal{O}^{LoR,b,K}(\cdot)$ corresponds to a left-oracle $b = 0$, random values are encrypted to \mathcal{A} , corresponding to a random \mathcal{A} -game (also $b = 0$). If, on the other hand, a right-oracle $b = 1$ was instantiated, correct encrypted values are forwarded to \mathcal{A} , consistent with a real \mathcal{A} -game ($b = 1$).
- ⑥ Therefore, if \mathcal{B} outputs as its guess bit b' the same value produced by \mathcal{A} , it has equal chances of success. We conclude that, whenever there exists a (ϵ, t, q, μ, k) -RoR adversary there also exists a (ϵ, t, q, μ, k) -LoR adversary.

RoR $\geq 1/2$ LoR security

- ⑥ Simulator \mathcal{S} may use an LoR attacker \mathcal{A} to implement an RoR attacker \mathcal{B} , as follows. First, \mathcal{S} chooses a random bit c . When \mathcal{A} queries with (m_0^i, m_1^i) , \mathcal{S} forwards m_c^i as \mathcal{B} 's query to the RoR oracle. It forwards to \mathcal{A} any responses that \mathcal{B} receives from $\mathcal{O}^{\text{RoR}, b, K}(\cdot)$.
- ⑥ If, at the end of the simulation, \mathcal{A} guesses $b' = c$, \mathcal{S} outputs $b'' = 1$ as \mathcal{B} 's guess. Otherwise, it outputs $b'' = 0$. We compute \mathcal{B} 's advantage:
$$\Pr[b'' = 1 | b = 1] - \Pr[b'' = 1 | b = 0].$$
- ⑥ When $b = 1$, the RoR oracle encrypts the given queries, and thus $b'' = 1$ when the adversary \mathcal{A} wins against a legitimate LoR game: $\Pr[b'' = 1 | b = 1] = 1/2 \left\{ \Pr[\mathbf{Exp}_{\mathcal{S}\mathcal{E}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk-1}} = 1] + \Pr[\mathbf{Exp}_{\mathcal{S}\mathcal{E}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk-0}} = 0] \right\} = 1/2 + 1/2 \mathbf{Adv}_{\mathcal{S}\mathcal{E}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk}}$.
- ⑥ When $b = 0$ the RoR oracle responds with random inputs. \mathcal{A} 's view is independent of the value c chosen by \mathcal{S} , and \mathcal{A} 's guess b' matches c exactly $1/2$ of the times. So $\Pr[b'' = 1 | b = 0] = 1/2$. Putting it all together, we get that the \mathcal{B} 's advantage in the RoR-game is $1/2$ that of \mathcal{A} against the LoR game.

Definition of FtG experiment

- ⑥ Again, \mathcal{SE} is given by $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ algorithms. First, the simulator \mathcal{S} chooses a random bit b . $\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{ftg-atk-}b}$, where atk is one of cpa or cca , is defined as follows.
- ⑥ The simulator generates a new key K according to algorithm \mathcal{K} , then instantiates a regular encryption oracle $\mathcal{O}^K(\cdot)$, and if $\text{atk} = \text{cca}$, also a regular decryption oracle $\mathcal{O}^{K^{-1}}(\cdot)$.
- ⑥ After interacting with the oracles at its disposal, the attacker \mathcal{A} chooses two messages m_0, m_1 , and the simulator responds with $C = \mathcal{E}(K, m_b)$.
- ⑥ \mathcal{A} interacts with the oracles—except that it is not allowed to ask for a decryption of C , and then outputs a guess b' of which message was encrypted by \mathcal{S} , winning if $b' = b$.

LoR \geq FtG security

- ⑥ It is trivial to show that a FtG-adversary \mathcal{A} can implement a LoR-adversary \mathcal{B} with the same winning advantage. The simulator only has to translate \mathcal{A} 's encryption queries of the type m^i into \mathcal{B} 's queries as (m^i, m^i) —i.e., left and right messages are equal. The exception is when \mathcal{A} 's presents the pair (m_0, m_1) , which is forwarded unmodified as \mathcal{B} 's query. If \mathcal{B} then presents \mathcal{A} 's guess as its own, it wins as often as \mathcal{A} does.
- ⑥ It is far more difficult to use an LoR-adversary \mathcal{A} to implement a FtG one \mathcal{B} . This is because \mathcal{A} 's queries of the form (m_0^i, m_1^i) must be arbitrarily unraveled into a single query for \mathcal{B} of type $m_{c_i}^i$. (Except for \mathcal{B} 's special query that has two elements.)
- ⑥ For an attacker \mathcal{A} that makes q queries, we consider hybrid experiments $\text{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}i}$, with $0 \leq i \leq q$, where any of \mathcal{A} 's queries (m_0^j, m_1^j) , $1 \leq j \leq i$ will result in m_0^j being encrypted, while any of \mathcal{A} 's queries (m_0^j, m_1^j) with $i < j \leq q$ will result in m_1^j being encrypted.

$FtG \geq q$ LoR security

- Let b indicate which value m_b will be encrypted in the FtG game. At the beginning the simulator chooses a random value i , $1 \leq i \leq q$, to indicate which of \mathcal{A} 's queries will be forwarded as \mathcal{B} 's special one. For values $j < i$, the value m_0^j will be sent as an encryption query, while m_1^j sent for $j > i$.

- \mathcal{B} 's advantage is the average of its advantages for $i = 1, \dots, q$. Now, if $b = 0$, the above game is identical to $\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}i}$, while if $b = 1$, it is identical to $\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}(i-1)}$. So the contribution of the i -th game is:

$$Pr[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}(i-1)} = 1] - Prob[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}i} = 1].$$

- Adding it over all i there are cancellations, resulting (after dividing by q to get average) in:

$$Adv(\mathcal{B}) = 1/q \left\{ Prob[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}0} = 1] - Prob[\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}q} = 1] \right\}$$

- It can be easily seen that $\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}0}$ is a right game, while $\mathbf{Exp}_{\mathcal{SE}, \mathcal{A}}^{\text{hyb-atk-}1}$ is a left game, and therefore the above is simply $1/q \mathbf{Adv}_{\mathcal{SE}, \mathcal{A}_{\text{atk}}}^{\text{lor-atk}}$.

References

References

- [1] M. Bellare, A. Desai, E. Jorjani, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proc. of the 38th IEEE Symposium on Foundations of Computer Science (FOCS 1997)*, IEEE Press, 1997.