Security Properties of Symmetric Key Encryption

Murat Kantarcioglu

Our Goal

- Formalize what we mean by "security" in the context of symmetric encryption that involves block ciphers.
- Analyze the security properties of various modes of operation.
- Discuss possible attacks.
- Provide proofs of security for CTR mode under chosen plaintext attacks.

Preliminaries

- We will focus on computational security.
- We model the adversary as a polynomial time probabilistic TM
- Given the input and the internal coin tosses of the A, we denote the output as :

$$a \leftarrow A(x_1, x_2, \ldots, x_n)$$

- Note that output is a random variable !

Symmetric Encryption

- We represent symmetric encryption by specifying possibly randomized algorithms. Note that this time we are concerned about the efficiency as well.
- Key generation Algorithm: $K \stackrel{R}{\leftarrow} \mathcal{K}(k)$
- Encryption Algorithm: $C \stackrel{R}{\leftarrow} \mathcal{E}_K(M)$
- Decryption Algorithm: $M \leftarrow \mathcal{D}_K(M)$
- Symmetric Encryption:
- $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and $\mathcal{D}_K(\mathcal{E}_K(M)) = M$
- Note that decryption should deterministic (why?)

Possible Attacks…

- Try to recover the key by analyzing the plaintext/ciphertext pairs.
  - Not enough!
  - Even though keys may be hard to recover, important partial information could be revealed.
- Try to recover plaintext from ciphertext
- Try to obtain partial information by looking plaintext/ciphertext pairs.

Security Definitions: Informal

- We will try to formalize a security definition that is similar in the sprit of Shannon's perfect secrecy.
- Intuitively, we will say that adversary does not learn anything given the ciphertext.
- There are various formalizations that captures the above intuition.

Security Under Chosen Plaintext Attack: Informal

- Adversary will choose any two messages M_1 and M_2
- It will be pass those two messages to an oracle.
- Oracle will encrypt one of the messages (always either the first message or the second message) and will return the encrypted message to adversary.
- Adversary repeats the first three steps and finally tries to predict which of the messages are encrypted by the oracle.

Security Under Chosen Plaintext Attack

- Given a symmetric encryption system:

$$\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$$

- Let A_{cpa} be an adversary that has an access to oracle $\mathcal{E}_K(\mathcal{LR}(.,.,b))$ where b={0,1}
- Consider the following experiment:

$$
\begin{aligned}
&Experiment\ \boldsymbol{Exp}^{lor-cpa-b}_{\mathcal{SE},A_{cpa}}(k)\\
&\qquad K \stackrel{R}{\leftarrow} \mathcal{K}(k)\\
&\qquad d \leftarrow A_{cpa}^{\mathcal{E}_K(\mathcal{LR}(.,.,b))}(k)\\
&\qquad Return\ d
\end{aligned}
$$

Security Under Chosen Plaintext Attack

- The size of the messages passed to the oracle is the same for both messages.
- We define the advantage of adversary as:

$$\mathbf{Adv}^{lor-cpa}_{\mathcal{S}\varepsilon, A_{cpa}}(k) =$$
$$\Pr[\boldsymbol{Exp}^{lor-cpa-1}_{\mathcal{S}\varepsilon, A_{cpa}}(k) = 1] - Pr[\boldsymbol{Exp}^{lor-cpa-0}_{\mathcal{S}\varepsilon, A_{cpa}}(k) = 1]$$

- Given k, among all adversaries that runs with time t, at most q_e queries, total μ_e bits, define:

$$\mathbf{Adv}^{lor-cpa}_{\mathcal{S}\varepsilon}(k, t, q_e, \mu_e) = max\{\mathbf{Adv}^{lor-cpa}_{\mathcal{S}\varepsilon, A_{cpa}}(k)\}$$

Security of ECB

- We will prove that ECB is not secure by providing an adversary that has very high advantage. Consider the following adversary:

Adversary $A^{\mathcal{E}_K(\mathrm{LR}(\cdot,\cdot,b))}$
    $M_1 \leftarrow 0^{2n}$; $M_0 \leftarrow 0^n \,\|\, 1^n$
    $C[1]C[2] \leftarrow \mathcal{E}_K(\mathrm{LR}(M_0, M_1, b))$
    If $C[1] = C[2]$ **then return** 1 else return 0

Security of ECB

- Let us calculate the advantage of the above adversary.
- Note that $Pr[\boldsymbol{Exp}^{lor-cpa-1}_{\mathcal{SE},A_{cpa}}(k) = 1] = 1$
- Also note that $Pr[\boldsymbol{Exp}^{lor-cpa0}_{\mathcal{SE},A_{cpa}}(k) = 1] = 0$
- We can conclude that ECB is not secure

$$\mathbf{Adv}^{lor-cpa}_{\mathcal{SE},A_{cpa}}(k) = (\Pr[\boldsymbol{Exp}^{lor-cpa-1}_{\mathcal{SE},A_{cpa}}(k) = 1] - Pr[\boldsymbol{Exp}^{lor-cpa-0}_{\mathcal{SE},A_{cpa}}(k) = 1]) = 1$$

Pseudo-random Functions and Permutations

- We model block ciphers (e.g. DES,AES) as a pseudo-random permutations and/or functions.
- We formally define pseudo-random function family as: $F : \mathrm{Keys}(F) \text{ x } \mathrm{Dom}(F) \to \mathrm{Ran}(F)$
- We will choose a random element of the family as: $K \stackrel{R}{\leftarrow} \mathrm{Keys}\ (F) \quad f \leftarrow F_K$
- In short $f \stackrel{R}{\leftarrow} (F)$
- AES: $\{0,1\}^{128}$ x $\{0,1\}^{128} \to \{0,1\}^{128}$

Pseudo-random Functions and Permutations

- We consider the family of all functions from l bit strings to L bit strings $Rand^{l \to L}$
- Similarly, we consider the family of permutations from l bit strings to l bit strings $Perm^{l}$
- Let F be function family with input-length l and output-length L
- P be a permutation family with length l.
- $b \in \{0,1\}$.
- D_{fn}, D_{pn} be distinguishers that have access to the oracle $O_b(.)$

Pseudo-random Functions and Permutations

- Now, we consider the following experiments:

$$
\begin{array}{l}
Experiment\ \boldsymbol{Exp}^{prf-b}_{F,D_{fn}} \\
\quad O_0 \stackrel{R}{\leftarrow} \mathrm{Rand}^{l \to L}; O_1 \stackrel{R}{\leftarrow} F \\
\quad d \leftarrow D_{fn}^{O_b(.)} \\
\quad Return\ d
\end{array}
\qquad
\begin{array}{l}
Experiment\ \boldsymbol{Exp}^{prp-b}_{P,D_{pn}} \\
\quad O_0 \stackrel{R}{\leftarrow} \mathrm{Perm}^{l}; O_1 \stackrel{R}{\leftarrow} P \\
\quad d \leftarrow D_{pn}^{O_b(.)} \\
\quad Return\ d
\end{array}
$$

Pseudo-random Functions and Permutations

- Consider the advantages of the distinguishers D_{fn}, D_{pn} that have access to the oracle $O_b(.)$

$$\mathbf{Adv}^{prf}_{F,D_{fn}} = \Pr[\boldsymbol{Exp}^{prf-1}_{F,D_{fn}} = 1] - Pr[\boldsymbol{Exp}^{prf-0}_{F,D_{fn}} = 1]$$

$$\mathbf{Adv}^{prp}_{P,D_{pn}} = \Pr[\boldsymbol{Exp}^{prp-1}_{P,D_{pn}} = 1] - Pr[\boldsymbol{Exp}^{prp-0}_{P,D_{pn}} = 1]$$

Pseudo-random Functions and Permutations

- Now consider the best possible performance of a distinguisher.

$$\mathbf{Adv}^{prf}_{F}(t,q) = \max_{D_{fn}}\{\mathbf{Adv}^{prf}_{F,D_{fn}}\}$$

$$\mathbf{Adv}^{prp}_{P}(t,q) = \max_{D_{pn}}\{\mathbf{Adv}^{prp}_{P,D_{pn}}\}$$

- We informally say a block-cipher is secure if the best possible advantage of a distinguisher is low under reasonable time and query constraints

Pseudo-random Functions and Permutations

- In practice we can bound the difference between PRF and PRP advantages.
- More specifically:

Proposition 8 [PRPs are PRFs]:
For any permutation family P with length l,

$$\mathbf{Adv}_P^{prf}(t,q) \leq \mathbf{Adv}_P^{prp}(t,q) + q^2 2^{-l-1}$$

XOR and CTR Encryption

- First, we fix a pseudo-random function family from l bits to L bits with k bits key.
- Using an element of the pseudo-random function family, we can define the XOR encryption by specifying key generation, encryption and decryption algorithms:

$$\mathrm{XOR[F]} = (\mathrm{E–XOR},\ \mathrm{D–XOR},\ \mathrm{K–XOR})$$

- Key generation is simple, just generate a key for the pseudo-random function family

XOR Encryption/Decryption

function E–XOR$^f(x)$
r$\leftarrow \{0,1\}^l$
for i=1,...,n do $y_i = f(r+i) \oplus x_i$
return $r \| y_1 y_2 ... y_n$

function D–XOR$^f(z)$
Parse z as $r \| y_1 y_2 ... y_n$
for i=1,...,n do $x_i = f(r+i) \oplus y_i$
return $x = x_1 x_2 ... x_n$

Security Analysis of XOR Scheme

- First we will prove a lower bound on the security of the XOR scheme assuming that F is pseudo-random family
- Then, we will analyze the security properties of the XOR scheme when F belongs to a random family $Rand^{l \to L}$
- Finally, we will conclude that if there exists an adversary that attacks the XOR scheme successfully, then we can find a distinguisher that can distinguish the pseudo-random function

Lower Bound on Insecurity

Proposition 9 [Lower bound on insecurity of XOR using a random function]:
Suppose $R = Rand^{l \leftarrow L}$. Then, for any q_e, μ_e such that $\mu_e q_e / L \le 2^l$,

$$\mathbf{Adv}^{lor-cpa}_{XOR[R]}(., t, q_e, \mu_e) \ge 0.316.\frac{\mu_e.(q_e-1)}{L.2^l}.$$

Lower Bound on Insecurity

- To prove the claim, we specify an adversary:

Algorithm $A^{\mathcal{O}(\cdot,\cdot)}(k)$

(1) Let $n = \mu/(Lq)$. (This will be the number of blocks in all queried messages.)

(2) Choose messages $N_1, \ldots, N_q$, all n blocks long, such that $N_i[k] \ne N_j[k']$ for all $i, j = 1, \ldots, q$ and $k, k' = 1, \ldots, n$ satisfying $(i, k) \ne (j, k')$. (For example, set $N_i[k]$ to the L-bit binary encoding of the integer $n(i-1)+k$ for $i = 1, \ldots, q$ and $k = 1, \ldots, n$.)

(3) For $i = 1, \ldots, q$ do: $(r_i, y_i[1] \ldots y_i[n]) \leftarrow \mathcal{O}(0^{nL}, N_i)$. We call r_i the i'th nonce.

(4) If there is some $i \ne j$ that $|r_i - r_j| < n$ (treat r_i, r_j as integers here!) then determine the values $k, k' \in \{1, \ldots, n\}$ such that $r_i + k = r_j + k'$. Output 0 if $y_i[k] = y_j[k']$ and 1 otherwise.

(5) If there is no $i \ne j$ that $|r_i - r_j| < n$, output a coin flip.

Lower Bound on Insecurity

- First, let us denote the probability that the condition on the line 4 is satisfied as p
- Note that the advantage of the adversary is p. $Pr_b[A=1]$ denotes the probability that Adversary outputs 1 when it is in world b.

$$\begin{aligned}\mathbf{Adv}^{lor-cpa}_{XOR[R],A}(.) &= Pr_1[A=1]-Pr_0[A=1] \\ &= (p.1+(1-p).\frac{1}{2})-(p.0+(1-p).\frac{1}{2}) \\ &= p \end{aligned} \quad (1)$$

Lower Bound on Insecurity

- We need to find a lower bound on p
- Consider the probability D_i that ith query does not cause any overlap.

$Pr[D_{i+1}|D_i] \leq \frac{2^l - in}{2^l} = 1 - \frac{in}{2^l}$.

Fact For any real number x with $0 \leq x \leq 1$ we have $(1-e^{-1})x \leq 1-e^{-x} \leq x$

- Let us calculate an upper bound on prob. that no query overlaps.

$Pr[D_q] =$
$\prod_{i=1}^{q-1} Pr[D_{i+1}|D_i] \leq \prod_{i=1}^{q-1}(1-\frac{in}{2^l}) \leq \prod_{i=1}^{q-1} e^{-in/2^l}$
$= e^{-nq(q-1)/2^{l+1}}$

Lower Bound on Insecurity

- Now using the Fact given above. We can conclude that

$$\begin{aligned} p &= Pr[OverlapNonce] \\ &= 1 - Pr[D_q] \\ &\geq 1 - e^{-nq(q-1)/2^{l+1}} \\ &\geq 1 - e^{-(1/2).\mu(q-1)/(L2^l)} \\ &\geq (1 - \frac{1}{e}).\frac{1}{2}.\frac{\mu(q-1)}{L2^l} \end{aligned}$$

Upper bound on insecurity of XOR scheme using Random Function

- Again we fix R= $Rand^{l \to L}$ and for any t, q, μ. We prove that

$$\mathbf{Adv}^{lor-cpa}_{XOR[R]}(., t, q_e, \mu_e) \leq \frac{\mu_e.(q_e-1)}{L.2^l}.$$

Upper bound on insecurity of XOR scheme using Random Function

- Let D be the event that no collusion occurs in the inputs to the random function.
- Let $Pr_b[E]$ is the probability of event E occurring game b for any event E.
- Note that $Pr_0[\bar{D}] = Pr_1[\bar{D}]$
- Also note that $Pr_0[A = 1|D] = Pr_1[A = 1|D]$
  - If there is no collusion and f is random function then the outputs of function f will be totally random. XOR scheme becomes like one-time pad.

Upper bound on insecurity of XOR scheme using Random Function

- We can find an upper bound the advantage of A as follows:

$$\begin{aligned} Adv^{lor-cpa}_{XOR[R],A}(.) &= Pr_1[A = 1] - Pr_0[A = 1] \\ &= (Pr_1[A = 1|\bar{D}].Pr_1[\bar{D}] + Pr_1[A = 1|D].Pr_1[D]) - \\ &\quad (Pr_0[A = 1|\bar{D}].Pr_0[\bar{D}] - Pr_0[A = 1|D].Pr_0[D]) \\ &= (Pr_1[A = 1|\bar{D}] - Pr_0[A = 1|\bar{D}]).Pr_1[\bar{D}] \\ &\leq Pr_1[\bar{D}] \end{aligned}$$

Upper bound on insecurity of XOR scheme using Random Function

- To conclude, we need to bound $Pr_0[\bar{D}] = Pr_1[\bar{D}]$
- Let p_i be the probability that ith query's random starting number colludes with one of the previous queries.
- Then we know that

$$p_i \leq \frac{\sum_{j=1}^{i-1}(n_j + n_i - 1)}{2^l} = \frac{(i-1)(n_i-1) + \sum_{j=1}^{i-1} n_j}{2^l}$$

- Also, we know that

$$Pr[\bar{D}] = Pr[p_1 \cup p_2 \cup \cdots \cup p_q] \leq \sum_{i=1}^{q} p_i$$

Upper bound on insecurity of XOR scheme using Random Function

- Finally,

$$Pr[\bar{D}] \leq \sum_{i=1}^{q} p_i \leq \sum_{i=1}^{q} \frac{((i-1)(n_i-1) + \sum_{j=1}^{i-1} n_j)}{2^l} = \frac{\frac{\mu}{L}(q-1) - \frac{q(q-1)}{2}}{2^l}$$
$$\leq \frac{\mu(q-1)}{L.2^l}$$

- Putting everything together we have

$$\mathbf{Adv}_{XOR[R],A}^{lor-cpa}(\cdot) \leq \frac{\mu(q-1)}{L.2^l}$$

Security of XOR Using a Pseudo-random Function

- Suppose F is pseudo-random function family with input length l and output length L. Then for any t, q_e, μ_e=L.q', we prove that

$$\mathbf{Adv}^{lor-cpa}_{XOR[F]}(.,t,q_e,\mu_e) \leq 2.\mathbf{Adv}^{prf}_{F}(t,q\prime) + \frac{\mu_e.(q_e-1)}{L.2^l}.$$

Security of XOR Using a Pseudo-random Function

- Intuitively, since we know that XOR[R] is secure, if XOR[F] were not secure, this would imply that F is not a good PRF family.
- Proof Idea: Assume that we have an Adversary A that attacks XOR[F] successfully under chosen plaintext attack. Then we show that we can create a distinguisher that can attack the pseudo-random function family successfully . Assuming that pseudo-random function family is secure, this will create a contradiction.

Security of XOR Using a Pseudo-random Function

- Consider the distinguisher on the next slide attacking PRF family F that uses an adversary A that attacks XOR encryption

Security of XOR Using a Pseudo-random Function

Algorithm $D^f(k)$
(1) $b \leftarrow \{0,1\}$. (This represents a choice to play either left or right oracle for A.)
(2) Run A, responding to its oracle queries as follows. When A makes an oracle query (M_1, M_2), let $z \leftarrow \varepsilon\text{--}XOR^f(M_b)$, and return z to A as the answer to the oracle query.
(It is important here that D can implement the encryption function given an oracle for f.)
(3) Eventually A stops and outputs a guess d to indicate whether it thought its oracle was the left oracle or the right oracle. If $d = b$ then output 1, else output 0.

Security of XOR Using a Pseudo-random Function

- Note that to answer a query (M0, M1) given by A, D asks |M0|/L queries to f
- D asks total μ/L queries
- Let Correct(G) be the probability that A correctly identifies its oracle when function underlying the encryption scheme is
  $f \stackrel{R}{\leftarrow} G$ where $G \in \{F, R\}$
- It is easy to show that

$$Correct(G) = (1/2).[1 + \mathbf{Adv}^{lor-cpa}_{XOR(G),A}(.)]$$

Security of XOR Using a Pseudo-random Function

- Now we can bound the advantage of distinguisher as

$$\begin{aligned} \mathbf{Adv}^{prf}_{F,D} &= Correct(F) - Correct(R) \\ &= (1/2).[\mathbf{Adv}^{lor-cpa}_{XOR(F),A}(.) - \mathbf{Adv}^{lor-cpa}_{XOR(R),A}(.)] \end{aligned}$$

- Now using the above equation

$$\begin{aligned} \mathbf{Adv}^{lor-cpa}_{XOR(F),A}(.) &= 2.\mathbf{Adv}^{prf}_{F,D} + \mathbf{Adv}^{lor-cpa}_{XOR(R),A}(.) \\ &\leq 2.\mathbf{Adv}^{prf}_{F,D} + \frac{\mu_e.(q_e - 1)}{L.2^l} \end{aligned}$$