How to formalize the security of cryptographic primitives against certain kinds of attacks?

For concreteness, consider asymmetric encryption against chosen-plaintext attacks (CPA).

- A key pair (sk, pk) is generated, using the key generation algorithm $\mathcal{K}$.
- pk is given to the adversary $\mathcal{A}$.
- A message is encrypted. The cryptotext is given to the adversary.
- The adversary tries to deduce something about the message.

Success probability of the adversary $\mathcal{A}$ attacking a cryptosystem $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ should be small:

$$\Pr\left[\text{X has something to do with M\ldots} \;\middle|\; \begin{array}{l} (sk, pk) \leftarrow \mathcal{K}() \\ M \leftarrow \mathcal{D} \\ C \leftarrow \mathcal{E}_{pk}(M) \\ X \leftarrow \mathcal{A}(pk, C) \end{array} \right] \leqslant \varepsilon_0 + \varepsilon$$

Here ε_0 is the success probability of an adversary that "does not really try to find the correct X".

How small should ε_0 be?

ε is the advantage of $\mathcal{A}$. It characterizes how much C helps to say something about M.

"something to do with" — let there be a function $f : \{0,1\}^* \to \{0,1\}^*$ whose value $\mathcal{A}$ tries to find.

$$\Pr\left[\; X = f(M) \;\middle|\; \begin{array}{l} (sk, pk) \leftarrow \mathcal{K}() \\ M \leftarrow \mathcal{D} \\ C \leftarrow \mathcal{E}_{pk}(M) \\ X \leftarrow \mathcal{A}(pk, C) \end{array} \;\right] \leqslant \varepsilon_0 + \varepsilon$$

We get ε_0 by running $\mathcal{A}$ "without C".

$$\Pr\left[\, X = {\color{red} f}(M) \;\middle|\; \begin{array}{l} (sk, pk) \leftarrow \mathcal{K}() \\ M \leftarrow {\color{red}\mathcal{D}} \\ C \leftarrow \mathcal{E}_{pk}(M) \\ X \leftarrow \mathcal{A}(pk, C) \end{array} \,\right] \leqslant$$

$$\Pr\left[\, X = {\color{red} f}(M') \;\middle|\; \begin{array}{l} (sk, pk) \leftarrow \mathcal{K}() \\ M \leftarrow {\color{red}\mathcal{D}} \\ M' \leftarrow {\color{red}\mathcal{D}} \\ C \leftarrow \mathcal{E}_{pk}(M) \\ X \leftarrow \mathcal{A}(pk, C) \end{array} \,\right] + \varepsilon$$

Let $\mathbf{Exp}_{\Pi}^{\mathrm{CPA},b}(\mathcal{A})$ denote the following random variable (called experiment):

$$(sk, pk) \leftarrow \mathcal{K}()$$
$$M_0 \leftarrow \mathcal{D}$$
$$M_1 \leftarrow \mathcal{D}$$
$$C \leftarrow \mathcal{E}_{pk}(M_1)$$
$$X \leftarrow \mathcal{A}(pk, C)$$
$$\textbf{if } X = f(M_b) \textbf{ then } 1 \textbf{ else } 0 \ .$$

The inequality on the previous slide is then

$$\Pr[\mathbf{Exp}_{\Pi}^{\mathrm{CPA},1}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}_{\Pi}^{\mathrm{CPA},0}(\mathcal{A}) = 1] \leqslant \varepsilon \ .$$

Denote that difference by $\mathbf{Adv}_{\Pi}^{\mathrm{CPA}}(\mathcal{A})$.

Let $\mathcal{A}$ provide $\mathcal{D}$ and f. It now works in two stages, $\mathcal{A}_1$ and $\mathcal{A}_2$. $\mathbf{Exp}_{\Pi}^{\mathrm{CPA},b}(\mathcal{A})$ is then

$$\begin{aligned}
&(sk, pk) \leftarrow \mathcal{K}() \\
&(\mathcal{D}, s) \leftarrow \mathcal{A}_1(pk) \\
&M_0 \leftarrow \mathcal{D} \\
&M_1 \leftarrow \mathcal{D} \\
&C \leftarrow \mathcal{E}_{pk}(M_1) \\
&(f, X) \leftarrow \mathcal{A}_2(C, s) \\
&\textbf{if } X = f(M_b) \textbf{ then } 1 \textbf{ else } 0 \enspace .
\end{aligned}$$

Here s is the "internal state" of $\mathcal{A}$. Most probably it includes pk.

$$\mathbf{Adv}_{\Pi}^{\mathrm{CPA}}(\mathcal{A}) = \Pr[\mathbf{Exp}_{\Pi}^{\mathrm{CPA},1}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}_{\Pi}^{\mathrm{CPA},0}(\mathcal{A}) = 1]$$

We say that Π is (t, ε)-secure against CPA if $\mathbf{Adv}_{\Pi}^{\mathrm{CPA}}(\mathcal{A}) \leqslant \varepsilon$ for all adversaries $\mathcal{A}$ whose running time is at most t.

On running time: Assume that $\mathcal{A}$ is represented as a sequence of instructions. Accessing the i-th instruction of that sequence is forbidden before the i-th clock tick.

Exercise. What would happen if we allowed $\mathcal{A}$ to access up to $\Theta(2^i)$-th instruction at the i-th clock tick?

Exercise. Show that $\mathcal{E}$ has to be probabilistic for Π to satisfy that security definition (for reasonable t and ε).

This kind of definition is called semantic security (of an encryption system). There are several others (about) equivalent to it.

All have the form "$\mathbf{Adv}_{\Pi}^{\mathrm{XXX}}(\mathcal{A}) \leqslant \varepsilon$ for all $\mathcal{A}$ with running time at most t", where

$$\mathbf{Adv}_{\Pi}^{\mathrm{XXX}}(\mathcal{A}) = \Pr[\mathbf{Exp}_{\Pi}^{\mathrm{XXX},1}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}_{\Pi}^{\mathrm{XXX},0}(\mathcal{A}) = 1]$$

The def. on previous slide should be called "semantic security against CPA".

Example: find-then-guess security against CPA.

$\mathbf{Exp}_{\Pi}^{\mathrm{FtG},b}(\mathcal{A})$ is

$$
\begin{aligned}
&(sk, pk) \leftarrow \mathcal{K}() \\
&(M_0, M_1, s) \leftarrow \mathcal{A}_1(pk) \\
&C \leftarrow \mathcal{E}_{pk}(M_b) \\
&b^* \leftarrow \mathcal{A}_2(C, s) \\
&\mathbf{return}\ b^*
\end{aligned}
$$

I.e. $\mathcal{A}$ chooses two plaintexts, receives the encryption of one of them, and tries to guess, which.

How do the definitions for security against CPA for *symmetric cryptosystems* look like?

The adversary should still be able to obtain encryptions for chosen plaintexts.

Hence we give it the access to encryption functionality.

The adversary $\mathcal{A}$ will be an oracle (Turing) machine.

A oracle is something that takes queries and answers to them. It may be randomized and have internal state.

$\mathcal{A}$ may execute instructions of the form $M_1 := \mathsf{query}(M_2)$. The contents of the cell M_2 is then given to the oracle and the return value written to the cell M_1.

$\mathcal{A}$ with access to the oracle $\mathcal{O}(\cdot)$ is denoted $\mathcal{A}^{\mathcal{O}(\cdot)}$.

Experiment $\mathbf{Exp}_{\Pi}^{\text{s-FtG},b}(\mathcal{A})$ for a symmetric cryptosystem Π:

$$\begin{aligned}
&k \leftarrow \mathcal{K}() \\
&(M_0, M_1, s) \leftarrow \mathcal{A}_1^{\mathcal{E}_k(\cdot)}() \\
&C \leftarrow \mathcal{E}_k(M_b) \\
&b^* \leftarrow \mathcal{A}_2^{\mathcal{E}_k(\cdot)}(C, s) \\
&\mathbf{return}\ b^*
\end{aligned}$$

Access to the oracle is a resource (like running time). In security definitions we may want to discriminate based on its usage:

Π is (t, q, μ, ε)-FtG-secure against CPA if $\mathbf{Adv}_{\Pi}^{\text{s-FtG}}(\mathcal{A}) \leqslant \varepsilon$ for all adversaries $\mathcal{A}$ whose running time is at most t and who make at most q queries to the oracle, totalling at most μ bits.

Experiment $\mathbf{Exp}_{\Pi}^{\text{s-LoR},b}(\mathcal{A})$ ("left or right"):

$$k \leftarrow \mathcal{K}()$$
$$b^* \leftarrow \mathcal{A}^{LR(\cdot,\cdot,k,b)}()$$
$$\textbf{return } b^*$$

where

$$LR(M_0, M_1, k, b) = \textbf{if } |M_0| = |M_1| \textbf{ then } \mathcal{E}_k(M_b) \textbf{ else error}$$

(LR is randomized because $\mathcal{E}$ is)

Also an often-used definition...

Here the length of a query (M_0, M_1) is defined as $|M_0|$.

In a chosen-ciphertext attack the adversary also has access to the decryption functionality.

Consider the following experiment $\mathbf{Exp}_{\Pi}^{\text{s-CCA},b}(\mathcal{A})$:

$$
\begin{aligned}
&k \leftarrow \mathcal{K}() \\
&b^* \leftarrow \mathcal{A}^{LR(\cdot,\cdot,k,b),\mathcal{D}(\cdot)}() \\
&\mathbf{return}\ b^*
\end{aligned}
$$

Exercise. Why cannot the CCA-security of symmetric cryptosystems be defined based on that experiment?

Actually, it can…

Π is $(t, q_\mathrm{e}, \mu_\mathrm{e}, q_\mathrm{d}, \mu_\mathrm{d}, \varepsilon)$-LoR-secure against CCA if $\mathbf{Adv}_\Pi^{\text{s-CCA}}(\mathcal{A}) \leqslant \varepsilon$ for all adversaries $\mathcal{A}$ whose running time is at most t, who make at most q_e queries to the first oracle (at most μ_e bits total), at most q_d queries to the second oracle (at most μ_d bits total) and do not query the second oracle with the bit-strings returned by the first.

We model a situation where the attacker can cause the system to decrypt some, but not all ciphertexts.

To give a definition that is as strong as possible, we only exclude ciphertexts whose decryption would immediately break the security.

For FtG-security against CCA consider the following experiment:

$$
\begin{aligned}
&k \leftarrow \mathcal{K}() \\
&(M_0, M_1, s) \leftarrow \mathcal{A}_1^{\mathcal{E}_k(\cdot),\mathcal{D}_k(\cdot)}() \\
&C \leftarrow \mathcal{E}_k(M_b) \\
&b^* \leftarrow \mathcal{A}_2^{\mathcal{E}_k(\cdot),\mathcal{D}_k(\cdot)}(C, s) \\
&\textbf{return } b^*
\end{aligned}
$$

where $\mathcal{A}_2$ may not invoke $\mathcal{D}_k(C)$.

We get two possible security definitions here, depending on whether $\mathcal{A}_2$ has access to $\mathcal{D}_k(\cdot)$ or not.

- $\mathcal{A}_2$ has access to $\mathcal{D}$ — security against adaptive CCA
  - Equivalent to LoR-security.
  - Also called “midnight attack”, CCA2.
- $\mathcal{A}_2$ does not have access to $\mathcal{D}$ — security against non-adaptive CCA
  - Also called “lunchtime attack”, CCA1.

For asymmetric cryptosystems consider the following experiment:

$$
\begin{aligned}
&(sk, pk) \leftarrow \mathcal{K}() \\
&(M_0, M_1, s) \leftarrow \mathcal{A}_1^{\mathcal{D}_{sk}(\cdot)}(pk) \\
&C \leftarrow \mathcal{E}_{pk}(M_b) \\
&b^* \leftarrow \mathcal{A}_2^{\mathcal{D}_{sk}(\cdot)}(C, s) \\
&\textbf{return } b^*
\end{aligned}
$$

where $\mathcal{A}_2$ may not invoke $\mathcal{D}_{sk}(C)$.

Again, two definitions are possible, depending on $\mathcal{A}_2$'s access to $\mathcal{D}$.

A block cipher E with block length l is a triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ of algorithms.

- $\mathcal{K}()$ is a probabilistic key generation algorithm;
- $\mathcal{E}_k(x)$ is a deterministic algorithm. For a fixed key k, $\mathcal{E}_k(\cdot)$ is a permutation of $\{0,1\}^*$.
- $\mathcal{D}_k(\cdot)$ is the inverse permutation of $\mathcal{E}_k(\cdot)$.

What is a suitable security definition for it?

Let Perm^l be the following probability distribution:

- its underlying set is the set of permutations of $\{0,1\}^l$;
- it is uniform.

If $\pi \leftarrow \mathsf{Perm}^l$ then we say that π is a random permutation (over l-bit strings).

- "random" is not a property of a permutation, but rather of its choice.

We want $\mathcal{E}_k(\cdot)$ (for k chosen according to $\mathcal{K}$) to look like a random permutation.

Exercise. How to "implement" a random permutation?

Let the experiment $\mathbf{Exp}_E^{\mathrm{PRP},1}(\mathcal{A})$ be

$$k \leftarrow \mathcal{K}();\ b^* \leftarrow \mathcal{A}^{\mathcal{E}_k(\cdot)};\ \mathbf{return}\ b^*$$

and the experiment $\mathbf{Exp}_E^{\mathrm{PRP},0}(\mathcal{A})$ be

$$\pi \leftarrow \mathsf{Perm}^l;\ b^* \leftarrow \mathcal{A}^{\pi(\cdot)};\ \mathbf{return}\ b^*$$

Let

$$\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) = \Pr[\mathbf{Exp}_E^{\mathrm{PRP},1}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}_E^{\mathrm{PRP},0}(\mathcal{A}) = 1]$$

Block cipher E is a (t, q, ε)-pseudorandom permutation if $\mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) \leqslant \varepsilon$ for all adversaries $\mathcal{A}$ of running time at most t and making at most q oracle queries.

A related notion is (pseudo)random function.

Let $\mathsf{Rand}^{l \to L}$ be the uniform probability distribution over all functions from $\{0,1\}^l$ to $\{0,1\}^L$.

Exercise. How to "implement" it?

Let $\mathbf{Exp}_E^{\mathrm{PRF},1}(\mathcal{A}) = \mathbf{Exp}_E^{\mathrm{PRP},1}(\mathcal{A})$ and $\mathbf{Exp}_E^{\mathrm{PRF},0}(\mathcal{A})$ be

$$\pi \leftarrow \mathsf{Rand}^{l \to l};\ b^* \leftarrow \mathcal{A}^{\pi(\cdot)};\ \mathbf{return}\ b^*$$

Let

$$\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) = \Pr[\mathbf{Exp}_E^{\mathrm{PRF},1}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}_E^{\mathrm{PRF},0}(\mathcal{A}) = 1]$$

Block cipher E is a (t, q, ε)-pseudorandom function if $\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) \leqslant \varepsilon$ for all adversaries $\mathcal{A}$ of running time at most t and making at most q oracle queries.

A block cipher is a pseudorandom function iff it is a pseudorandom permutation.

Theorem. $|\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) - \mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A})| \leqslant q(q-1)/2^{l+1}$ where q is the number of oracle queries made by $\mathcal{A}$.

Proof.

$$\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{A}) - \mathbf{Adv}_E^{\mathrm{PRP}}(\mathcal{A}) = \Pr[\mathbf{Exp}_E^{\mathrm{PRP},0}(\mathcal{A}) = 1] - \Pr[\mathbf{Exp}_E^{\mathrm{PRF},0}(\mathcal{A}) = 1]$$

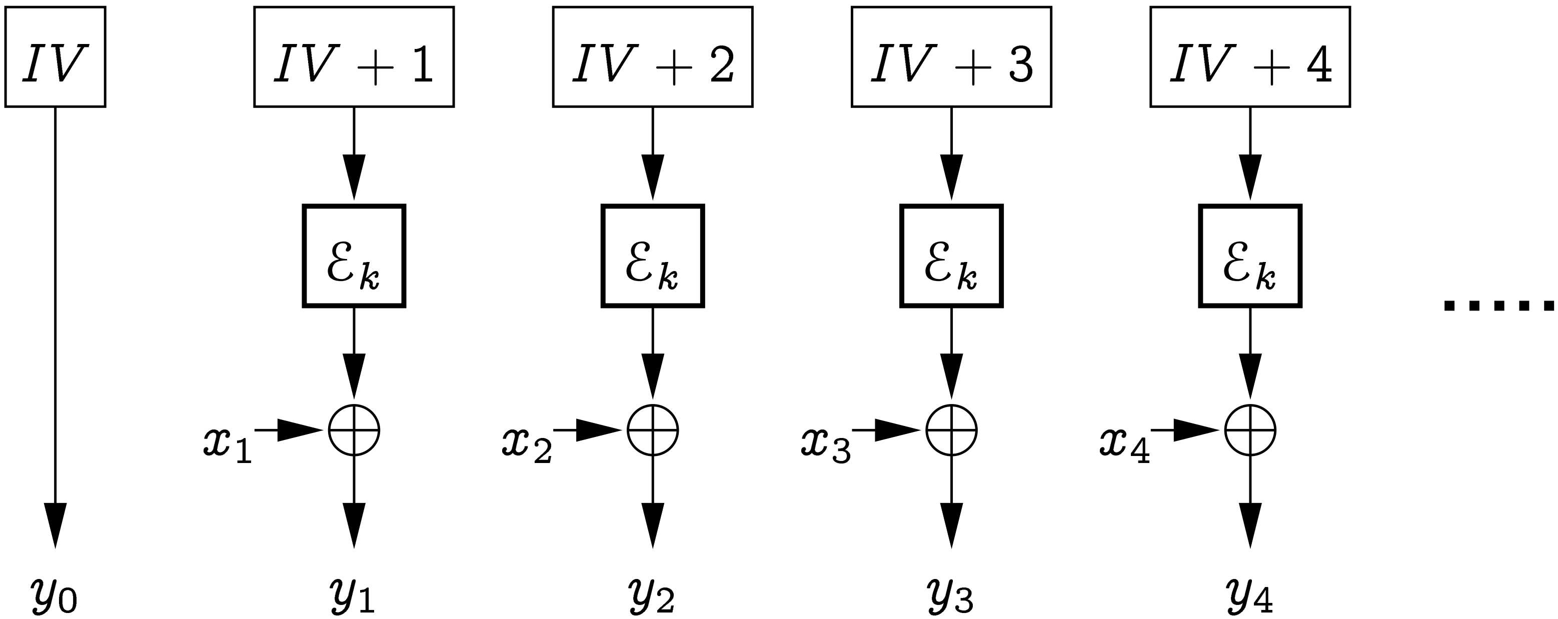
The responses to oracle queries in experiments for PRPs and PRFs are the same, except that for PRPs, they all must be different.

Among q queries, there are $q(q-1)/2$ query pairs. The probability that a pair of different queries produces the same answer for PRF is $1/2^l$.

Theorem. If a block cipher is a (t, q, ε)-pseudorandom permutation then it is also a $(t, q, \varepsilon + \frac{q(q-1)}{2^l})$-pseudorandom function.

Theorem. If a block cipher is a (t, q, ε)-pseudorandom function then it is also a $(t, q, \varepsilon + \frac{q(q-1)}{2^l})$-pseudorandom permutation.

Recall the counter mode of operation of block ciphers:



We show that if E is a good pseudorandom function then the resulting symmetric encryption system is secure agaist CPA (in left-or-right sense).

Block cipher $E = (\mathcal{K}^E, \mathcal{E}^E, \mathcal{D}^E)$.

Symmetric encryption system $\Pi = (\mathcal{K}^\Pi, \mathcal{E}^\Pi, \mathcal{D}^\Pi)$ defined by

- $\mathcal{K}^\Pi = \mathcal{K}^E$;
- $\mathcal{E}_k^\Pi(x_1 \cdots x_n)$, where $x_i \in \{0,1\}^l$ is
  - $IV \in_R \{0,1\}^l$; $y_0 := IV$
  - $y_i := \mathcal{E}_k^E(IV + i) \oplus x_i$
  - return $y_0 \cdots y_n$.
- $\mathcal{D}_k^\Pi = \mathcal{E}_k^\Pi$.

(Denote $\Pi = \mathbf{XOR}[E]$)

Theorem. If E is a (t, q, ε)-pseudorandom function then Π is a symmetric encryption system that is $(t', q', \mu', \varepsilon')$-LoR-secure against CPA, where

$$t' = \dots$$

$$q' = \dots$$

$$\mu' = \dots$$

$$\varepsilon' = \dots$$

Let $\mathcal{RO}$ be the following oracle: on input x of length nl generate $y \in_R \{0,1\}^{(n+1)l}$ and return it.

Define the experiment $\mathbf{Exp}_{\Pi}^{\mathrm{RF},1}(\mathcal{A})$:

$$k \leftarrow \mathcal{K}^{\Pi}();\ b^* \leftarrow \mathcal{A}^{\mathcal{E}_k(\cdot)}();\ \mathbf{return}\ b^*$$

and $\mathbf{Exp}_{\Pi}^{\mathrm{RF},0}(\mathcal{A})$:

$$b^* \leftarrow \mathcal{A}^{\mathcal{RO}(\cdot)}();\ \mathbf{return}\ b^*$$

$\mathcal{RO}$ may be considered as a symmetric encryption system.
($\mathcal{K}^{\mathcal{RO}}$ returns a constant and $\mathcal{D}^{\mathcal{RO}}$ does not exist
($\mathcal{D}$ is not necessary for talking about CPA))

Recall the experiment $\mathbf{Exp}_{\Pi}^{\text{s-LoR},b}(\mathcal{A})$:

$$k \leftarrow \mathcal{K}()$$
$$b^* \leftarrow \mathcal{A}^{LR(\cdot,\cdot,k,b)}()$$
$$\textbf{return } b^*$$

where

$$LR(M_0, M_1, k, b) = \textbf{if } |M_0| = |M_1| \textbf{ then } \mathcal{E}_k(M_b) \textbf{ else error}$$

Lemma. $\mathbf{Adv}_{\mathcal{RO}}^{\text{s-LoR}}(\mathcal{A}) = 0$ for any adversary $\mathcal{A}$.

Proof. The distribution of $\mathcal{E}^{\mathcal{RO}}(M)$ does not depend on M. Hence the values returned by LR do not depend on b.

Let $\Xi = \mathbf{XOR}[\mathsf{Rand}^{l \to l}]$. I.e.

- $\mathcal{K}^{\Xi}()$ picks a random function f from $\{0,1\}^l$ to $\{0,1\}^l$;
- $\mathcal{E}^{\Xi}_f(x_1 \cdots x_n) = y_0 \cdots y_n$ where y_0 is random and $y_i = f(y_0 + i) \oplus x_i$;
- $\mathcal{D}^{\Xi}_f = \mathcal{E}^{\Xi}_f$.

Lemma. For all adversaries $\mathcal{A}$ that make at most q oracle queries with μ bits in total,

$$\Pr[\mathcal{A}^{\mathcal{E}^{\Xi}_f(\cdot)}() = 1 \mid f \leftarrow \mathsf{Rand}^{l \to l}] - \Pr[\mathcal{A}^{\mathcal{RO}(\cdot)}() = 1] \leqslant \ldots$$

The answers from $\mathcal{RO}(\cdot)$ are completely random.

The answers from $\mathcal{E}_f^{\Xi}(\cdot)$ are also completely random, as long as f is not invoked twice on the same argument.

There are up to q queries to $\mathcal{E}_f^{\Xi}(\cdot)$. Assume that the i-th query is n_i blocks long. Then $n_i \geqslant 0$ and $\sum_{i=1}^{q} n_i \leqslant \mu/l$.

Let $IV_1, \ldots, IV_q$ be independent, uniformly distributed random variables over $\{0,1\}^l$. What is the probability that some of the following numbers are equal?

$$\begin{array}{cccc}
IV_1 + 1 & IV_1 + 2 & \cdots & IV_1 + n_1 \\
IV_2 + 1 & IV_2 + 2 & \cdots & IV_2 + n_2 \\
\vdots & \vdots & \ddots & \vdots \\
IV_q + 1 & IV_q + 2 & \cdots & IV_q + n_q
\end{array}$$

- When choosing IV_1, there are 0 possibilities (out of 2^l) to create a collision.
- When choosing IV_2, there are $n_1 + n_2$ possibilities to create a collision.
- When choosing IV_3, there are $\leqslant (n_1 + n_3) + (n_2 + n_3)$ possibilities to create a collision.
- When choosing IV_i, there are $\leqslant \sum_{j=1}^{i-1}(n_j + n_i)$ possibilities to create a collision.

Summing $i = 1, \ldots, q$: there are $\leqslant (q-1)\sum_{i=1}^{q} n_i$ possibilities (out of 2^l) to create a collision.

Hence

$$\Pr[\mathcal{A}^{\mathcal{E}^{\Xi}_{f}(\cdot)}() = 1 \mid f \leftarrow \mathsf{Rand}^{l \to l}] - \Pr[\mathcal{A}^{\mathcal{RO}(\cdot)}() = 1] \leqslant \frac{(q-1)\sum_{i=1}^{q} n_i}{2^l} \leqslant \frac{(q-1)\mu}{2^l \cdot l}$$

(that's what our lemma claimed).

Lemma. For any adversary $\mathcal{A}$ that makes at most q oracle queries totalling μ bits,

$$\mathbf{Adv}_{\Xi}^{\text{s-LoR}}(\mathcal{A}) \leqslant \frac{(q-1)\mu}{2^{l-1} \cdot l} \quad .$$

Proof. Construct the following algorithm $\mathcal{B}^{\mathcal{O}(\cdot)}$:

- Generate $d \in_R \{0, 1\}$;
- Let $b \leftarrow \mathcal{A}^{(\cdot,\cdot)}()$;
  - Whenever $\mathcal{A}$ makes an oracle query (M_0, M_1), return $\mathcal{O}(M_d)$.
- If $d = b$, return 1, else return 0.

We see that $\mathcal{B}$ makes as many oracle queries as $\mathcal{A}$, with the same total length.

$$\frac{(q-1)\mu}{2^l \cdot l} \geqslant$$

$$\Pr[\mathcal{B}^{\mathcal{E}^{\Xi}_{f}(\cdot)}() = 1 \mid f \leftarrow \mathsf{Rand}^{l \to l}] - \Pr[\mathcal{B}^{\mathcal{RO}(\cdot)}() = 1] =$$

$$\Pr[d=0]\cdot\Pr[\mathbf{Exp}_{\Xi}^{\text{s-LoR},0}(\mathcal{A}) = 0] + \Pr[d=1]\cdot\Pr[\mathbf{Exp}_{\Xi}^{\text{s-LoR},1}(\mathcal{A}) = 1] -$$

$$\Pr[d=0]\cdot\Pr[\mathbf{Exp}_{\mathcal{RO}}^{\text{s-LoR},0}(\mathcal{A}) = 0] - \Pr[d=1]\cdot\Pr[\mathbf{Exp}_{\mathcal{RO}}^{\text{s-LoR},1}(\mathcal{A}) = 1] =$$

$$\frac{1}{2}(1 - \Pr[\mathbf{Exp}_{\Xi}^{\text{s-LoR},0}(\mathcal{A}) = 1] + \Pr[\mathbf{Exp}_{\Xi}^{\text{s-LoR},1}(\mathcal{A}) = 1]) -$$

$$\frac{1}{2}(1 - \Pr[\mathbf{Exp}_{\mathcal{RO}}^{\text{s-LoR},0}(\mathcal{A}) = 1] + \Pr[\mathbf{Exp}_{\mathcal{RO}}^{\text{s-LoR},1}(\mathcal{A}) = 1]) =$$

$$\frac{1}{2}(\mathbf{Adv}_{\Xi}^{\text{s-LoR}}(\mathcal{A}) - \mathbf{Adv}_{\mathcal{RO}}^{\text{s-LoR}}(\mathcal{A})) = \frac{1}{2}\mathbf{Adv}_{\Xi}^{\text{s-LoR}}(\mathcal{A})$$

Lemma. For any $b \in \{0,1\}$ and any adversary $\mathcal{A}$ with running time at most t, whose oracle queries total at most μ bits there is an adversary $\mathcal{B}$ with running time at most $O(t)$ that makes at most μ/l oracle queries and satisfies

$$\mathbf{Adv}_{\Pi}^{\text{s-LoR}}(\mathcal{A}) \leqslant \ldots \cdot \mathbf{Adv}_{E}^{\text{PRF}}(\mathcal{B}) + \ldots$$

Proof. Given such $\mathcal{A}$, let $\mathcal{B}^{\mathcal{O}(\cdot)}$ be

$$\begin{aligned}&\textbf{let } \mathcal{Q} = \mathbf{XOR}[\mathcal{O}]\\ &d \in_R \{0,1\}\\ &b \leftarrow \mathcal{A}^{LR(\cdot,\cdot,d)}()\\ &\textbf{if } d = b \textbf{ then } 1 \textbf{ else } 0\end{aligned}$$

where

$$LR(M_0, M_1, d) = \textbf{if } |M_0| = |M_1| \textbf{ then } \mathcal{Q}(M_d) \textbf{ else error}$$

The number of oracle queries made by $\mathcal{B}$ is μ/l. The running time of $\mathcal{B}$ consists of the time to run $\mathcal{A}$, to implement the XOR-mode, and to generate and compare d.

$$
\begin{aligned}
&\mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{B}) = \\
&\quad \Pr[\mathbf{Exp}_E^{\mathrm{PRF},1}(\mathcal{B}) = 1] - \Pr[\mathbf{Exp}_E^{\mathrm{PRF},0}(\mathcal{B}) = 1] = \\
&\frac{1}{2}(\Pr[\mathbf{Exp}_\Pi^{\text{s-LoR},0}(\mathcal{A}) = 0] + \Pr[\mathbf{Exp}_\Pi^{\text{s-LoR},1}(\mathcal{A}) = 1]) - \\
&\frac{1}{2}(\Pr[\mathbf{Exp}_\Xi^{\text{s-LoR},0}(\mathcal{A}) = 0] + \Pr[\mathbf{Exp}_\Xi^{\text{s-LoR},1}(\mathcal{A}) = 1]) = \\
&\qquad\qquad\qquad \frac{1}{2}(\mathbf{Adv}_\Pi^{\text{s-LoR}}(\mathcal{A}) - \mathbf{Adv}_\Xi^{\text{s-LoR}}(\mathcal{A})) \ .
\end{aligned}
$$

Hence

$$
\mathbf{Adv}_\Pi^{\text{s-LoR}}(\mathcal{A}) \leqslant 2 \cdot \mathbf{Adv}_E^{\mathrm{PRF}}(\mathcal{B}) + \frac{(q-1)\mu}{2^{l-2} \cdot l} \ .
$$

Theorem. If E is a (t, q, ε)-pseudorandom function then Π is a symmetric encryption system that is $(t', q', \mu', \varepsilon')$-LoR-secure against CPA, where

$$
\begin{aligned}
t' &= \frac{t}{O(1)} \\
q' &= q \\
\mu' &= q \cdot l \\
\varepsilon' &= 2 \cdot \varepsilon + \frac{q-1}{2^{l-2}}
\end{aligned}
$$

What is an appropriate definition of security for message authentication codes?

Consider an active adversary. It may obtain the tag of certain messages of its choice. (chosen-message attack)

Adversary is successful if it can construct the tag of some message that has not been MAC'ed before (existential forgery).

A MAC $(\mathcal{K}, sig, ver)$ is (t, q, μ, ε)-secure against EF-CMA if for all adversaries $\mathcal{A}$ whose running time is bounded by t, and who make no more than q oracle queries totalling no more than μ bits,

$$\Pr\left[\begin{array}{l|l} ver_k(M, \sigma) = \text{true and} & k \leftarrow \mathcal{K}() \\ \mathcal{A} \text{ did not query } M & (M, \sigma) \leftarrow \mathcal{A}^{sig_k(\cdot)}() \end{array}\right] \leqslant \varepsilon$$

Security def. for digital signatures is similar, but $\mathcal{A}$ gets the verification key, too.

Sometimes we do not just have a single cryptographic primitive Π, but an entire family of primitives $\{\Pi_k\}_{k \in \mathbb{N}}$.

This k is related to the security of the primitive. Larger k means more security.

Example: various primitives based on number-theoretic problems. k is the size of the moduli.

We want to talk about the rate at which security increases if we increase k.

Let the adversary $\mathcal{A}$ also take the parameter k.

Let there be some polynomial p, such that the running time of $\mathcal{A}(k, \ldots)$ is at most $p(k)$.

$\mathbf{Adv}_{\Pi_k}(\mathcal{A}(k, \ldots))$ is then also a function of k.

We want this function to be negligible:

$$\forall q \in \mathbb{N}[x]\ \exists k_0 \in \mathbb{N}\ \forall k \geqslant k_0 : \mathbf{Adv}_{\Pi_k}(\mathcal{A}(k, \ldots))(k) \leqslant \frac{1}{q(k)} \ .$$

Alternatively, we may consider for each k the function $\varepsilon_k(t) = \max_{\mathcal{A}} \mathbf{Adv}_{\Pi_k}(\mathcal{A})$ where max is taken over all adversaries with running time $\leqslant t$.

For each polynomial p we then demand the mapping

$$k \mapsto \varepsilon_k(p(k))$$

to be negligible.

Exercise. What is the difference between those two definitions?