Recap: OWFs $\Rightarrow$ PRGs $\Rightarrow$ ONE-TIME SKE

PRFs $\Rightarrow$ CPA SKE

PRGs $\Rightarrow$ PRFs

The GGM Construction

Goldreich - Goldwasser - Micali

Let $G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a PRG and denote

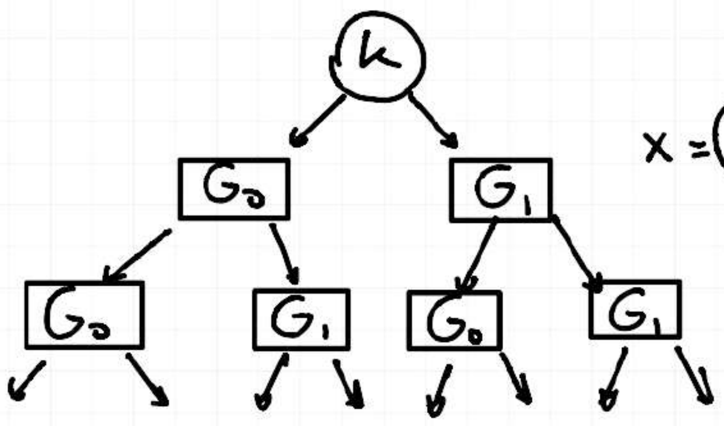
$$G(s) = (\underbrace{G_0(s), G_1(s)}_{\text{each outputs } \lambda \text{ bits}}).$$

This is a PRG on domain $\{0,1\}$

$$\begin{array}{c|c} x & y = F_k(x) \\ \hline 0 & G_0(k) \\ 1 & G_1(k) \end{array} \quad \approx_c \quad \begin{array}{c|l} x & y \\ \hline 1 & z_0 \leftarrow U_\lambda \\ 0 & z_1 \leftarrow U_\lambda \end{array}$$

$F_k : \{0,1\}^\lambda \to \{0,1\}^\lambda$

<u>GGM</u> : Extension to any $x \in \{0,1\}^{n(\lambda)}$



$x = (x_1, x_2 \ldots, x_n)$

You go down the tree for any bit of x

$\forall x \in \{0,1\}^n, k \in \{0,1\}^\lambda$

$$y = F_k(x) = G_{x_n}(G_{x_{n-1}}(G_{x_{n-2}}(\dots(G_{x_2}(G_{x_1}(k)))\dots)$$

THM Assuming $G: \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ a PRG, then above $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^\lambda\}_{k \in \{0,1\}^\lambda}$ is a PRF

Domain Extension for SKE

Until now, given a PRF with output $n(\lambda)$ bits, then I can do CPA-secure SKE on $\mathcal{M} = \{0,1\}^\lambda$

But what if $m = (m_1, m_2, \dots, m_t)$ where $m_i \in \{0,1\}^n$?

A solution to this problem is called a MODE OF OPERATION.
In practice we use AES (advanced encryption standard)

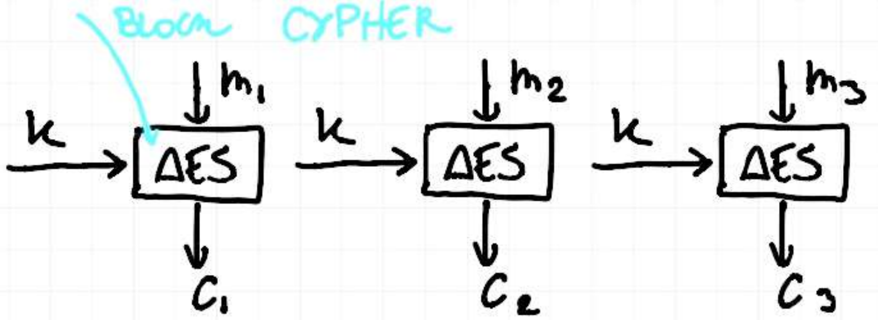
$$AES : \{0,1\}^{128} \times \{0,1\}^{256} \Rightarrow \{0,1\}^{256}$$

and there exists AES^{-1} that inverts encryption

(History of SKE : DES

AES → it isn't provably secure though...

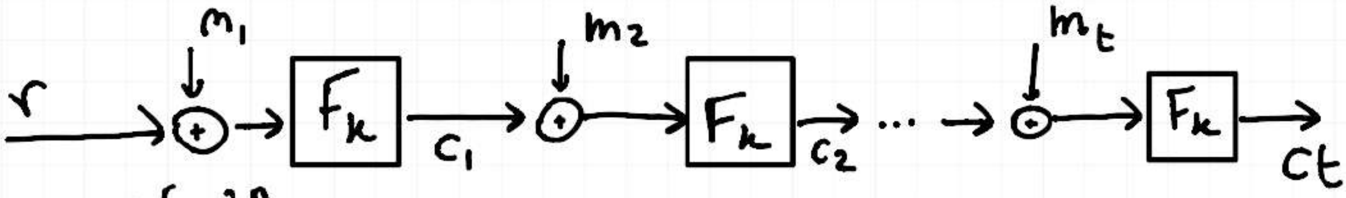
AES is not CPA secure (as it is deterministic), but is really efficient.
So we do as follows



This is NOT CPA-secure!

This is called ECB (Electronic Code Block) MODE

There also is CBC (Cipher Block Chaining) MODE



$r \in \leftarrow \$ \{0,1\}^n$

$$\text{Ciphertext} = C = (c_0 = r, c_1, c_2, \dots, c_t)$$

To decrypt it's needed to evaluate F_k^{-1}.
i.e. F_k is a PERMUTATION $\forall k \in \{0,1\}^\lambda$

Is it CPA-secure? Yes, assuming that F_k is a PSEUDORANDOM PERMUTATION (PRP)

How to instantiate CBC:

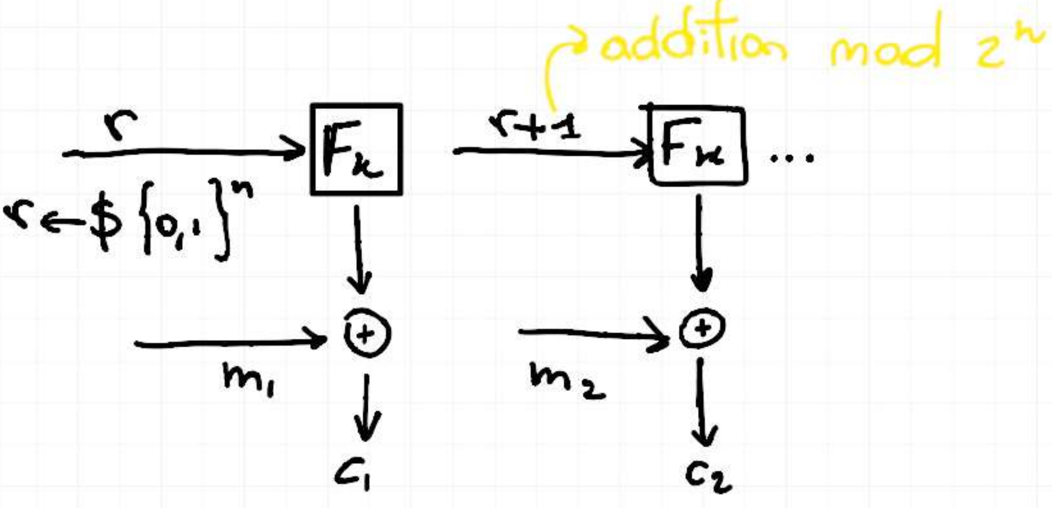
1. PRACTITIONER: $F_k \equiv AES_k$
   (not provably secure, but efficient)

2. THEORETICIAN: $F_k \equiv PRP$

OWFs ⇒ PRGs ⇒ OWFs ⇒ PRPs

↳ we'll see this more later...

CTR (Counter) MODE



$C = (r, c_1, c_2, \dots c_t)$

Note:

1. Efficient (1 block overhead)
2. No need to calculate F_k^{-1} for decryption
   you just need r and F_k, in this way you can XOR the result to get the original blocks back

THM: Assuming $\mathcal{F}$ is a PRF family, CTR mode is CPA-secure for VARIABLE-INPUT-LENGTH messages

PROOF: We consider a sequence of experiments

$$\mathrm{GAME}^{CPA}_{\Pi,A}(\lambda, b) \equiv H_0(\lambda, b)$$

$A(1^\lambda)$ $\qquad\qquad$ $C(1^\lambda)$

$$\xrightarrow{m = m_1 || m_2 || \dots || m_t} \quad k \leftarrow \$ \{0,1\}^\lambda$$

NOTE: t not fixed

$$\xleftarrow{\hspace{4cm}} \quad r \leftarrow \$ \mathbb{N}$$

$$c = (r, c_1, c_2, \dots c_t) \qquad \forall i \in [t] : c_i = F_k(r + i - 1 \oplus m_i)$$

$$\xrightarrow{m_0^*,\, m_1^* \,(t^* \text{blocks})} \quad r^* \leftarrow \$ \mathbb{N}$$

$$\xleftarrow{\hspace{4cm}} \quad \forall i \in [t^*] : c_i^* = F_k(r^* + i - 1 \oplus \underbrace{m_i^*})$$

$$c^* = (r^*, c_1^*, \dots c_{t^*}^*)$$

$$\xrightarrow[b']{\hspace{4cm}}$$

the i-th block in m_b^*

We need to show $\{H_0(\lambda, 0)\} \approx_c \{H_0(\lambda, 1)\}$

We define $H_1(\lambda, b)$ we don't pick a random key k but a function $R \leftarrow \$ \mathcal{R}(\lambda, n, m)$.

In this way $H_1(\lambda, b)$

$\mathcal{A}(1^\lambda)$ $\qquad\qquad\qquad$ $\mathcal{C}(1^\lambda)$

$$\xrightarrow{m = m_1 || m_2 || \dots || m_t} \quad R \leftarrow \$ \mathcal{R}(\lambda, n, m)$$

$$\xleftarrow{\hspace{4cm}} \quad r \leftarrow \$ \mathbb{N}$$

$$c = (r, c_1, c_2, \dots c_t) \qquad \forall i : c_i = R(r + i - 1 \oplus m_i)$$

$$\xrightarrow{m_0^*,\, m_1^* \,(t^* \text{blocks})} \quad r^* \leftarrow \$ \mathbb{N}$$

$$\xleftarrow{\hspace{4cm}} \quad \forall i : c_i^* = R(r^* + i - 1 \oplus m_i^*)$$

$$c^* = (r^*, c_1^*, \dots c_{t^*}^*)$$

$$\xrightarrow[b']{\hspace{4cm}}$$

LEMMA: $\forall b \in \{0,1\} : \{H_0(\lambda, b)\} \approx_c \{H_1(\lambda, b)\}$

Proof by security of PRFs.

We define $H_2(\lambda, b)$ as a modification of H_1

$\mathcal{A}(1^\lambda)$ $\qquad\qquad\qquad$ $\mathcal{C}(1^\lambda)$

$\xrightarrow{m = m_1 || m_2 || \dots || m_t}$ $\quad R \leftarrow \$ \mathcal{R}(\lambda, n, m)$

$\xleftarrow{\qquad}$ $\quad r \leftarrow \$ N$

$c = (r, c_1, c_2, \dots c_t)$ $\quad \forall i : c_i = R(r+i-1 \oplus m_i)$

$\xrightarrow{m_0^*, m_1^* \ (t^* \text{ blocks})}$

Note $H_2(\lambda, 0) \equiv H_2(\lambda, 1)$

$\xleftarrow{\qquad}$ $\quad c^* = (r^*, c_1^*, \dots c_{t^*}^*)$

$c^* = (r^*, c_1^*, \dots c_{t^*}^*)$ $\quad \leftarrow \$ U_{(t^*+1)n}$

$\xrightarrow[b']{\qquad}$

independent of b

LEMMA: $\forall b \in \{0,1\} : \{H_1(\lambda, b)\} \approx_c \{H_2(\lambda, b)\}$

PROOF: For each encryption query, let $r_i \leftarrow \$ [N]$ be the initial counter and assume the plaintext is made of $t_i \in \mathbb{N}$ blocks.

Look at the sequence

$$R(r^*), R(r^*+1), \dots R(r^* + t^* - 1)$$

$$R(r_i), R(r_i+1), \dots, R(r_i+t_i-1)$$

Let E be the event that the sequence

$r^*, r^*+1, \dots, r^*+t^*-1$ is FRESH (it never overlaps with other sequencies)

Then c^* is uniformly random and thus $H_2 \equiv H_1$

By a previous lemma:

$$SD(H_1(\lambda,b); H_2(\lambda,b)) \le \Pr[E]$$

BAD event $\neg E \equiv$ OVERLAP

$\Rightarrow \exists\, i, j, j'$ with $j \le t_i$, $j' \le t^*$ such that

$$r_i + j = r^* + j'$$

Let OVERLAP_i be the event that OVERLAP happens with encryption query $i \in [q(\lambda)]$ with $q(\lambda) = poly(\lambda)$

$r_i \leftarrow \$ \, N$

$r^* \quad r^*+1 \quad \dots \quad r^*+t^*-1$

To simplify, assume that $t_i, t^* = q(\lambda) = poly(\lambda)$

Now, in order for OVERLAP_i to happen for fixed r^* we need

$$r^*-q+1 \le r_i \le r^*+q-1$$

$$\Rightarrow \Pr[\text{OVERLAP}_i] \leq \frac{(r^* + q - 1) - (r^* - q + 1) + 1}{2^n}$$

$$= \frac{2q(\lambda) - 1}{2^n} = negl(\lambda)$$

By UNION BOUND:

$$\Pr[\text{OVERLAP}] \leq \sum_i^q \Pr[\text{OVERLAP}_i] \leq negl(\lambda)$$

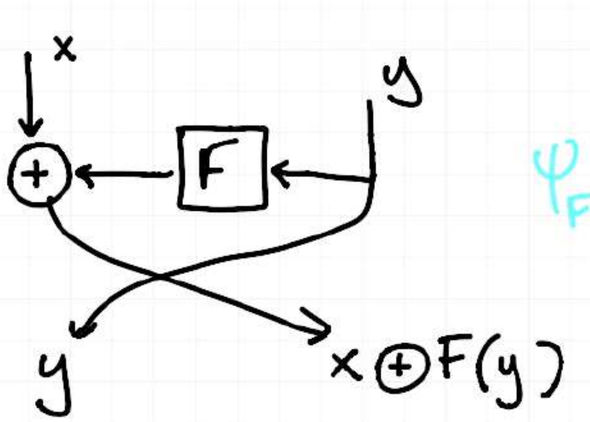
$\hookrightarrow = \Pr[\exists i \in [q] : \text{Overlap}_i]$

In this way, we solved symmetric encryption!
We have an encryption system that takes a key $k << m$, and that can be reused multiple times!

One more thing...

PRFs $\Rightarrow$ PRP

Let $\mathcal{F} = \{F_k : \{0,1\}^n \to \{0,1\}^n\}$ be a PRF.
How to get a PRP?
Through the FEINSTEIL construction



$$\Psi_F = \overbrace{(x, y)}^{2n \text{ bits}} =$$

$$= (y, x \oplus F(y))$$

Note: $\Psi_F^{-1}(x', y') = (F(x') \oplus y', x')$

and you can do multiple ROUNDS, with each round that uses a different key for F

FACT (LUBY-Rackoff) 3-ROUND FEISTEL using PRF family with independent keys yields a PRP.