Katz, Lindell
Introduction to Modern Cryptrography
Slides Chapter 3

Markus Bläser, Saarland University

Computational security

Goal: No adversary can break the scheme

- in "reasonable" time
- with "reasonable" success probability.

Question: Mathematical modelling of "reasonable"

Efficient algorithms

- Word-RAM, Turing machine, ...
- efficient $=$ polynomial running time
  (There is a polynomial p such that for all inputs x, the running time is bounded by $p(|x|)$.)
- algorithms are randomized, i.e., the algorithm can flip a fair coin at any time.
  (Equivalently, the algorithm gets a sufficiently long random string drawn uniformly at random as an additional input.)

Negligible success probability

Definition (3.4)

A function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is *negligible* if for every positive polynomial p there is an N such that for all $n > N$: $f(n) < \frac{1}{p(n)}$.

- $p(n) = n^c$ for all constants c is sufficient
- negligible functions will often be denoted by negl

Proposition (3.6)

Let negl_1 *and* negl_2 *be negligible functions and* p *be a positive polynomial.*

1. $\mathrm{negl}_1 + \mathrm{negl}_2$ *is negligible.*
2. $p \cdot \mathrm{negl}_1$ *is negligible.*

Why relaxations?

- Given c, you can run over all $k \in \mathcal{K}$ and compute $\mathsf{Dec}_k(c)$.
  This tells you which messages were *not* sent.
  Running time proportional to $|\mathcal{K}|$.
- Assume you know a pair (m_0, c_0) with $c_0 = \mathsf{Enc}_k(m_0)$. Then you can find the key k by testing $c_0 = \mathsf{Enc}_k(m_0)$.
  Running time proportional to $|\mathcal{K}|$.
- Or you can randomly guess a key k and check whether $c_0 = \mathsf{Enc}_k(m_0)$.
  Success probability: $1/|\mathcal{K}|$.

Private key encryption scheme

Definition (3.7)

A *private key encryption scheme* is a tuple of ppt algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ such that

1. Gen on input 1^n (n in unary) outputs a key k.
   W.l.o.g. $|k| \geq n$.
2. Enc on input k and $m \in \{0,1\}^*$ outputs a ciphertext c.
3. Dec on input k and c outputs a message m.

For every n, every k generated by Gen, and every $m \in \{0,1\}^*$,

$$\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m.$$

If for fixed n, Enc is only defined on messages of length $\ell(n)$ then the scheme if called a *fixed length private key encryption scheme*

- n = security parameter ("the larger, the more secure")
- scheme is *stateless*.
- Gen usually generates key uniformly at random.

Indistinguishability

The adversarial indistinguishability experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$:

1. On input 1^n, $\mathcal{A}$ outputs messages m_0, m_1 with $|m_0| = |m_1|$.
2. $k \leftarrow \mathsf{Gen}(1^n)$ and $b \in \{0,1\}$ is chosen uniformly at random. $c \leftarrow \mathsf{Enc}_k(m_b)$ is given to $\mathcal{A}$ ("challenge")
3. $\mathcal{A}$ outputs $b' \in \{0,1\}$.
4. $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = \begin{cases} 1 & b = b' \\ 0 & \text{otherwise} \end{cases}$

- $\mathcal{A}$ polynomial time bounded $\longrightarrow |m_i| = \mathrm{poly}(n)$.
- Π fixed length $\longrightarrow |m_i| = \ell(n)$.
- $\mathcal{A}$ sees only one ciphertext and no further interaction $\approx$ eavesdropping of one ciphertext.

Indistinguishability (2)

Definition (3.8)

$\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable encryptions in the presence of an eavesdropper* or is *EAV-secure* if for all ppt adversaries $\mathcal{A}$ there is a negligible functions negl such that for all n,

$$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

(Probability is taken over randomness of $\mathcal{A}$, k, b, and randomness of Enc.)

- perfectly secret encryption $\Longrightarrow$ EAV-secure
- goal: key shorter than message

Semantic security

Definition (3.12)

$(\mathsf{Enc}, \mathsf{Dec})$ is *semantically secure in the presence of an eavesdropper* if for every ppt algorithm $\mathcal{A}$ there is a ppt algorithm $\mathcal{A}'$ such that for every ppt algorithm Samp and polynomial time computable functions f and h,

$$|\Pr[\mathcal{A}(1^n, \mathsf{Enc}_k(m), h(m)) = f(m)] - \Pr[\mathcal{A}'(1^n, |m|, h(m)) = f(m)]| \leq \mathsf{negl}(n).$$

First probability is taken over uniform $k \in \{0,1\}^n$, $m \leftarrow \mathsf{Samp}(1^n)$, randomness of $\mathcal{A}$, randomness of Enc. Second probability is taken over $m \leftarrow \mathsf{Samp}(1^n)$ and randomness of $\mathcal{A}'$.

Semantic security (2)

- $\Pr[\mathcal{A}(1^n, \mathrm{Enc}_k(m), h(m)) = f(m)]$: adversary gets a ciphertext and has some information $h(m)$. $\mathcal{A}$ tries to guess the information $f(m)$.
- $\mathcal{A}'$ has almost the same chance of guessing $f(m)$ without knowing the chiphertext.
- "No (polynomial time computable) information is leaked."

Theorem (3.13)

A private-key encryption scheme is EAV-secure iff it is semantically secure.

EAV-secure is easier to work with.

Pseudorandom generators

Definition (3.14)

Let ℓ be a polynomial and let G be a deterministic polynomial time algorithm such that for each $s \in \{0,1\}^n$, $G(s) \in \{0,1\}^{\ell(n)}$. G is a *pseudorandom generator*, if

1. $\ell(n) > n$ for all n,
2. for any ppt algorithm D,

$$\left|\Pr_{s\in\{0,1\}^n}[D(G(s)) = 1] - \Pr_{r\in\{0,1\}^{\ell(n)}}[D(r) = 1]\right| \leq \mathsf{negl}(n)$$

for all n.

probability also taken over internal randomness of D

ℓ = expansion factor

A secure fixed-length encryption scheme

Construction (3.17)

G *prg with expansion factor* ℓ.

- Gen: *on input* 1^n, *return* $k \in \{0,1\}^n$ *uniformly at random.*
- Enc: *given key* k *and message* $m \in \{0,1\}^{\ell(n)}$, *output*

$$c := G(k) \oplus m.$$

- Dec: *given key* k *and ciphertext* $c \in \{0,1\}^{\ell(n)}$, *output*

$$c := G(k) \oplus c.$$

Theorem (3.18)

If G *is a prg, then Construction 3.17 is EAV-secure.*

Proofs by reduction

Assumption: Existence of pseudorandom generators.

- Assume that the scheme is not EAV-secure.
  Let $\mathcal{A}$ be an attacker with nonnegligible success probability.
- Construct a distinguisher who breaks the assumption, i.e., algorithm $\mathcal{D}$ who can distinguish the output of G from a uniform distribution with nonnegligable success probability.

Proof of Thm 3.18

Distinguisher $\mathcal{D}$

Input: $w \in \{0,1\}^{\ell(n)}$

1. Run $\mathcal{A}(1^n)$ to obtain $m_0, m_1 \in \{0,1\}^{\ell(n)}$.
2. Choose $b \in \{0,1\}$ at random. Set $c := w \oplus m_b$.
3. Give c to $\mathcal{A}$ and get output b'.
   Return 1 if $b' = b$ and 0 otherwise.

Stronger security notions

Multiple message eavesdropping experiment $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n)$**:**

1. $\mathcal{A}$ is given 1^n and it outputs two lists of messages $(m_{0,1}, \ldots, m_{0,t})$ and $(m_{1,1}, \ldots, m_{1,t})$ with $|m_{0,i}| = |m_{1,i}|$ for all i.
2. $k \leftarrow \mathsf{Gen}(1^n)$ is generated and $b \in \{0,1\}$ is chosen unformly at random. Compute $c_i \leftarrow \mathsf{Enc}_k(m_{b,i})$ and give $(c_1, \ldots, c_t)$ to $\mathcal{A}$.
3. $\mathcal{A}$ outputs a bit b'.
4. The output of the experiment is 1 if $b = b'$ and 0 otherwise.

Multiple encryptions

Definition (3.19)

$\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable multiple encryptions in the presence of an eavesdropper* if for all ppt $\mathcal{A}$,

$$\Pr\left[\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n).$$

(probability over randomness of $\mathcal{A}$ and $\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}$)

Theorem (3.21)

If Π *is a (stateless) encryption scheme in which* Enc *is deterministic, then* Π *cannot have indistinguishable multiple encryptions in the presence of an eavesdropper.*

Chosen plaintext attacks

CPA indistinguishability experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$

1. $k \leftarrow \mathsf{Gen}(1^n)$ is generated.
2. $\mathcal{A}$ is given 1^n and oracle access to $\mathsf{Enc}_k(.)$.
   He outputs two messages m_0, m_1 of the same length.
3. $b \in \{0,1\}$ is chosen uniformly at random and $c \leftarrow \mathsf{Enc}_k(m_b)$ is given to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b' (with oracle access to $\mathsf{Enc}_k(.)$).
5. The output of the experiment is 1 if $b = b'$ and 0 otherwise.

- oracle access to $\mathsf{Enc}_k \longrightarrow$ no knowledge of k!

Chosen plaintext attacks (2)

Definition (3.22)

$\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable encryptions under chosen-plaintext attacks* or is *CPA-secure* if for all ppt $\mathcal{A}$,

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi} = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n).$$

(probability over randomness of $\mathcal{A}$ and of experiment)

CPA-security for multiple encryptions

The LR-oracle experiment $\mathrm{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}$

1. $k \leftarrow \mathrm{Gen}(1^n)$ is generated.
2. $b \in \{0,1\}$ is chosen uniformly at random.
3. $\mathcal{A}$ is given 1^n and oracle access to $\mathrm{LR}_{k,b}(.,.)$.
4. $\mathcal{A}$ outputs b'.
5. The output of the experiment is 1 if $b = b'$ and 0 otherwise.

- $\mathrm{LR}_{k,b}(m_0, m_1)$ returns $c \leftarrow \mathrm{Enc}_k(m_b)$
- no knowledge of k or b!
- enables *adaptive* attacks

CPA-security for multiple encryptions (2)

Definition (3.23)

Π has *indistinguishable multiple encryptions under chosen-plaintext attacks* if for all ppt $\mathcal{A}$

$$\Pr\left[\mathsf{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n).$$

(probability over randomness of $\mathcal{A}$ and of experiment)

Theorem (3.24)

Any private-key encryption scheme that is CPA-secure is also CPA-secure for multiple encryptions.

Fixed-length versus arbitrary length

- $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ fixed length scheme which is CPA-secure
- Define arbitray length scheme $\Pi = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ as follows
- $\mathsf{Gen}' = \mathsf{Gen}$
- Cut message m into pieces $m_1, \ldots, m_t$ of length $\ell(n)$. $\mathsf{Enc}'_k(m) = \mathsf{Enc}_k(m_1), \ldots, \mathsf{Enc}_k(m_t)$
- Decryption Dec' is blockwise.

Theorem

Π' is CPA-secure if Π is CPA-secure.

Follows from Theorem 3.24.

Constructing CPA-secure encryption schemes

Keyed functions

- *keyed function* $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$.
- F is *efficient*, if $(k, x) \mapsto F(k, x)$ is polynomial time computable.
- $k \in \{0,1\}^*$ induces a function $F_k : \{0,1\}^* \to \{0,1\}^*$ by $F_k(x) := F(k, x)$.
- key length $\ell_{key}(n)$, input length $\ell_{in}(n)$, and output length $\ell_{out}(n)$
  restrict F_k to $\{0,1\}^{\ell_{in}(n)}$ and output has to be in $\{0,1\}^{\ell_{out}(n)}$.
- typically $\ell_{key}(n) = \ell_{in}(n) = \ell_{out}(n) = n$.

Pseudorandom functions

Definition (3.25)

Let F be an efficient, length-preserving, keyed function. F is *pseudorandom* if for all ppt distinguishers $\mathcal{D}$,

$$\left|\Pr[\mathcal{D}^{F_k(.)}(1^n) = 1] - \Pr[\mathcal{D}^{f(.)}(1^n) = 1]\right| \leq \mathsf{negl}(n).$$

(First probability over uniform choice of k, second probability over uniform choice of $f \in \mathsf{Func}_n$. Both over randomness of $\mathcal{D}$.)

- Func_n set of all functions $\{0,1\}^n \to \{0,1\}^n$, $|\mathsf{Func}_n| = 2^{n2^n}$
- $|\text{key space}| = 2^n$
- input of $\mathcal{D}$ is the oracle
- k is *not* given to $\mathcal{D}$

Pseudorandom permutations

keyed permutation: $\ell_{in}(n) = \ell_{out}(n)$, F_k bijective for all k

pseudorandom: indistinguishable from random permutation

Proposition (3.27)

If F is a pseudorandom permutation and $\ell_{in}(n) \geq n$, then F is also a pseudorandom function.

Definition

A keyed permutation F is *efficient* if there is a polynomial time algorithm computing $(k, x) \mapsto F_k(x)$ and a polynomial time algorithm computing $(k, y) \mapsto F_k^{-1}(y)$.

Pseudorandom permutations (2)

Definition

Let F be an efficient, length-preserving, keyed permutation. F is *strongly pseudorandom* if for all ppt $\mathcal{D}$

$$\left|\Pr[\mathcal{D}^{F_k(.),F_k^{-1}(.)}(1^n) = 1] - \Pr[\mathcal{D}^{f(.),f^{-1}(.)}(1^n) = 1]\right| \leq \mathsf{negl}(n).$$

(first probability over k, second over $f \in \mathsf{Perm}_n$, both over $\mathcal{D}$)

- Perm_n set of all permutations on $\{0,1\}^n$, $|\mathsf{Perm}_n| = (2^n)!$

Pseudorandom functions versus generators

F pseudorandom $\longrightarrow$ prg ("stream chipher")

- Choose $s, I \in \{0,1\}^n$
- Repeat until we produced the desired number of bits:
  - output $F_s(I)$
  - $I := I + 1$.

---

prg G with expansion factor $n2^{t(n)}$
$\longrightarrow$ prf $\{0,1\}^n \times \{0,1\}^{t(n)} \to \{0,1\}^n$

- interpret $G(k)$ as a table of values
- larger block lengths are possible but harder to achieve (Ch. 7.5)

CPA-secure encryption from prfs

Construction (3.30)

Let F *be a prf.*

- Gen*: returns* $k \in \{0,1\}^n$ *uniformly at random*
- Enc*: on key* $k \in \{0,1\}^n$ *and message* $m \in \{0,1\}^n$*, choose* $r \in \{0,1\}^n$ *uniformly at random and output*

$$c := \langle r, F_k(r) \oplus m \rangle.$$

- Dec*: on key* $k \in \{0,1\}^n$ *and ciphertext* $c = \langle r, s \rangle$*, output*

$$m := F_k(r) \oplus s.$$

CPA-secure encryption—proof

Theorem (3.31)

If F is a prf, then Construction 3.30 is CPA-secure for messages of length n.

Distinguisher $\mathcal{D}$: has oracle access to $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$

1. Run $\mathcal{A}(1^n)$. When $\mathcal{A}$ queries its oracle, then:
   - Query $\mathcal{O}(r)$ on random $r \in \{0,1\}^n$ and obtain answer y.
   - Return $\langle r, y \oplus m \rangle$ to $\mathcal{A}$.
2. When $\mathcal{A}$ outputs m_0, m_1, choose $b \in \{0,1\}$ uniformly at random:
   - Query $\mathcal{O}(r)$ on random $r \in \{0,1\}^n$ and obtain answer y.
   - Return challenge $\langle r, y \oplus m_b \rangle$ to $\mathcal{A}$
3. Answer oracle queries of $\mathcal{A}$ as above. When $\mathcal{A}$ outputs b', then output 1, if $b = b'$ and 0 otherwise.

Stream ciphers and block ciphers

In practice:

- Stream ciphers produce a stream of pseudorandom bits
- CPA-secure, variable-length schemes based on prg-like construction

- block ciphers are practical implementations of prfs (or prps)
- they are put into a "mode of operation" for repeated use.

(Definitions are somewhat blurry...)

Stream Ciphers

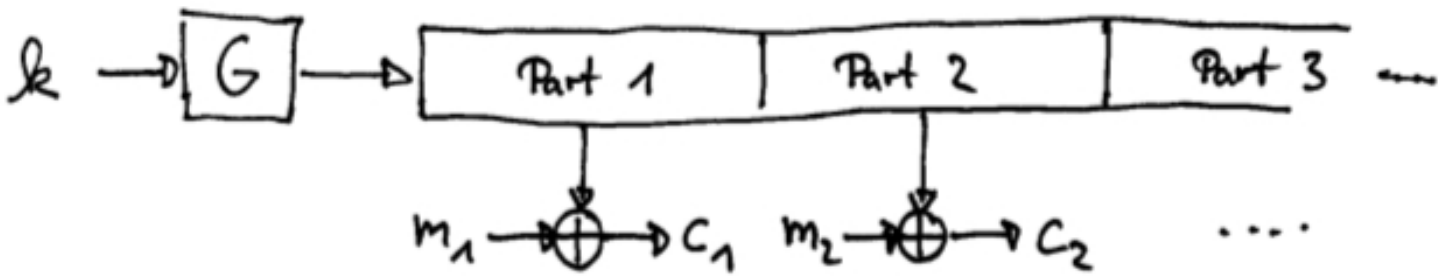
Algorithm (3.16)

Input: seed s, *initialisation vector* IV
Output: $y_1, \ldots, y_\ell$

1. $\mathrm{st}_0 := \mathrm{Init}(s, \mathrm{IV})$
2. *for* $i := 1$ *to* ℓ *do*
3. $\quad (y_i, \mathrm{st}_i) := \mathrm{GetBits}(\mathrm{st}_{i-1})$
4. *return* $y_1, \ldots, y_\ell$

produces pseudorandom bits one after another

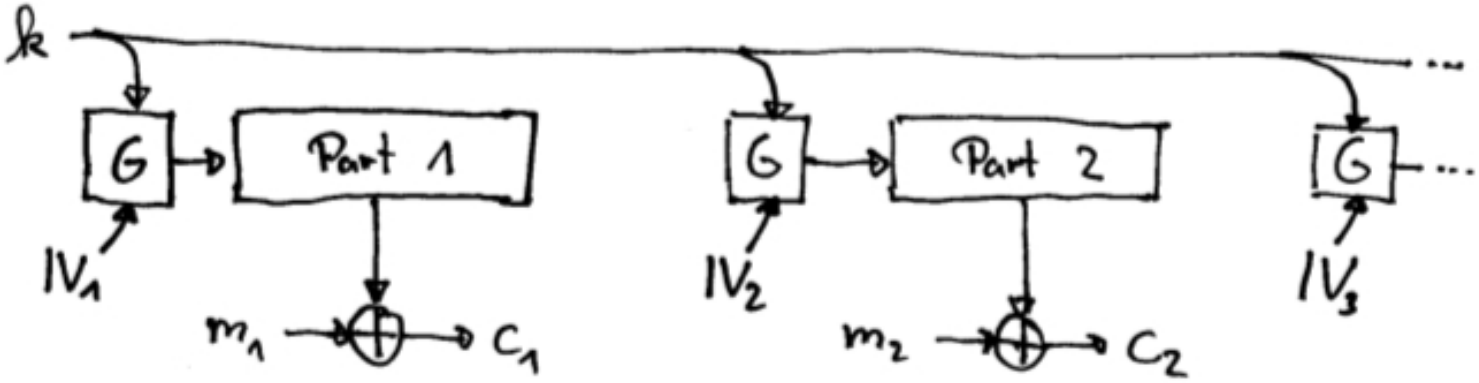
Synchronized mode



Prg G_∞ with variable output length:

- $G_\infty(s, 1^\ell)$ returns ℓ bits like in Construction 3.17
- Encryption: $c := G_\infty(k, 1^{|m|}) \oplus m$
- Decryption: $m := G_\infty(k, 1^{|c|}) \oplus c$
- can be even used to encrypt/decrypt multiple messages by sharing the current state of the stream cipher
- no initialisation vector is needed

Unsynchronized mode



- allows stateless CPA-secure encryption of arbitrary length messages
- $\mathrm{G}_\infty(s, \mathrm{IV}, 1^\ell)$ returns ℓ bits like in Construction 3.17
- Encryption: $c := \langle \mathrm{IV}, \mathrm{G}_\infty(s, \mathrm{IV}, 1^{|m|}) \oplus m \rangle$, IV chosen uniformly at random.
- Decryption of $\langle \mathrm{IV}, c' \rangle$: $m := \mathrm{G}_\infty(s, \mathrm{IV}, 1^{|m|}) \oplus c'$.
- CPA-secure, if $\mathrm{F}_k(\mathrm{IV}) := \mathrm{G}_\infty(k, \mathrm{IV}, 1^\ell)$ is a prf for any $\ell = \mathrm{poly}(n)$.

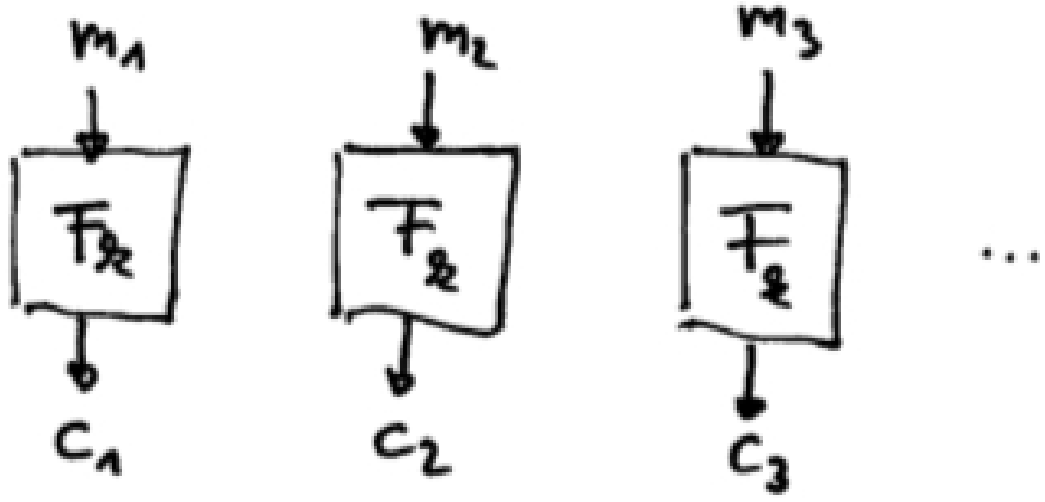
Block ciphers—modes of operation

Recall Construction 3.10:

- Prf F, encode m as $\langle r, F_k(r) \oplus m \rangle$
- Drawback: message length is doubled

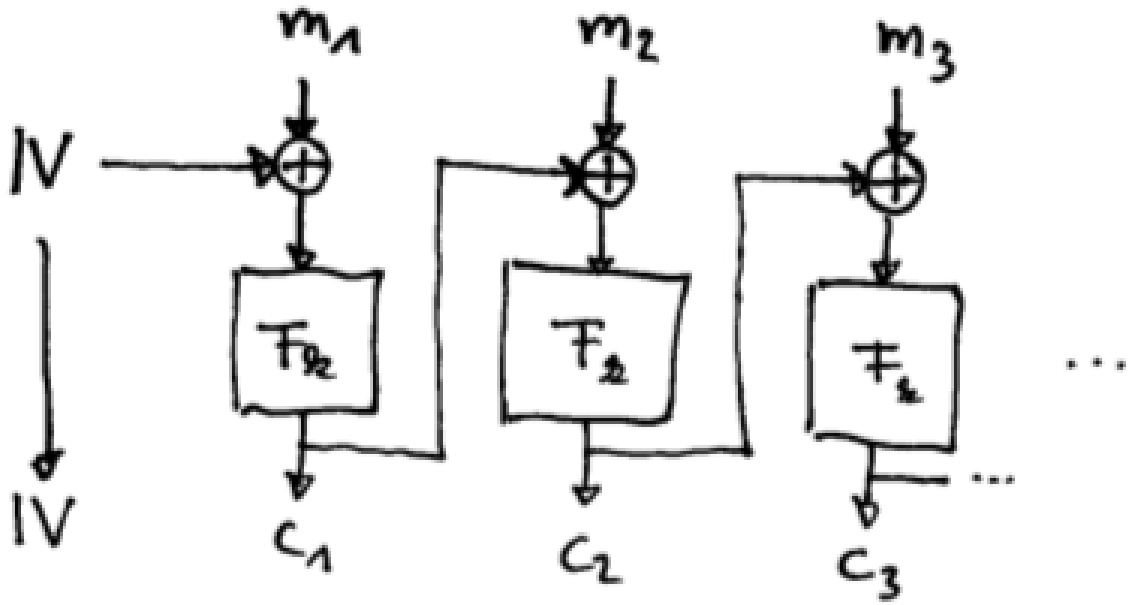
Solution: Block ciphers

Electronic code book mode (ECB)



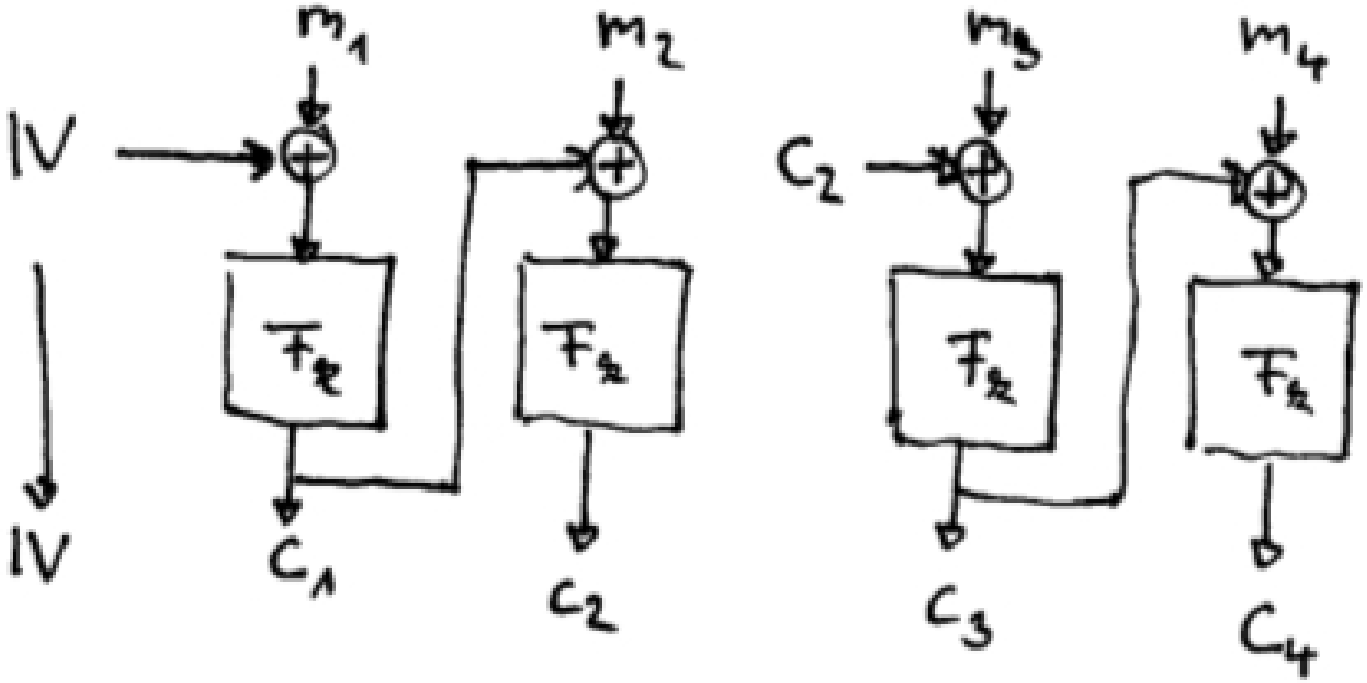
Crap.

Cipher block chaining mode (CBC)



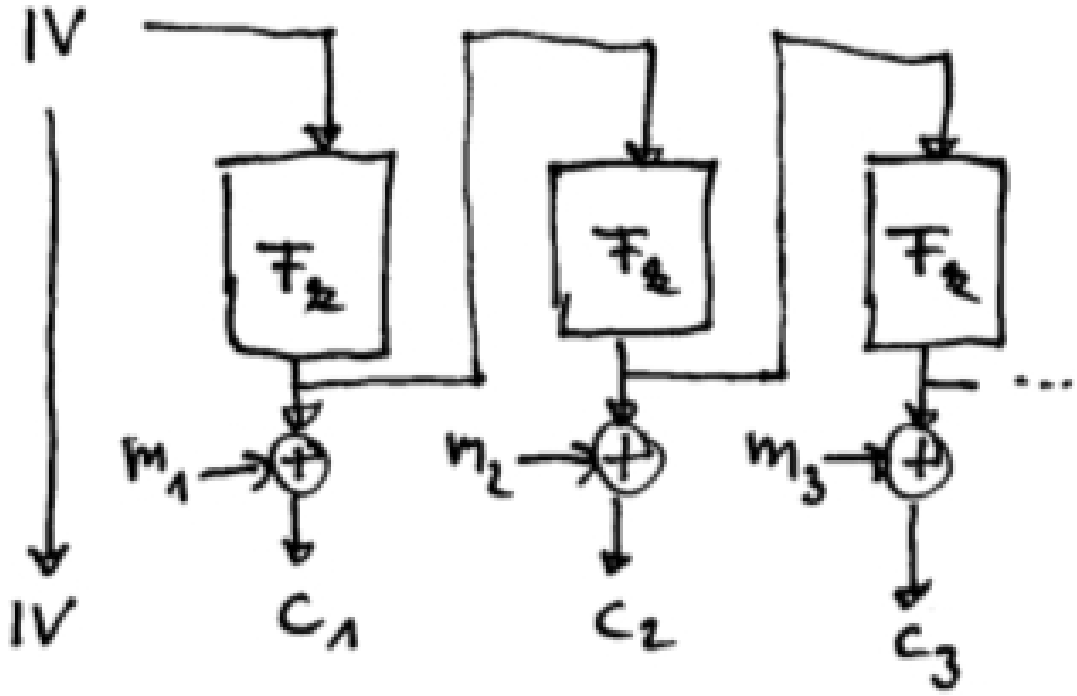
- Encryption: $c_i := F_k(c_{i-1} \oplus m_i)$
- Decryption: $m_i := F_k^{-1}(c_i) \oplus c_{i-1}$
- IV needs to be included, IV is random
- CPA-secure if F is prp.

Chained CBC



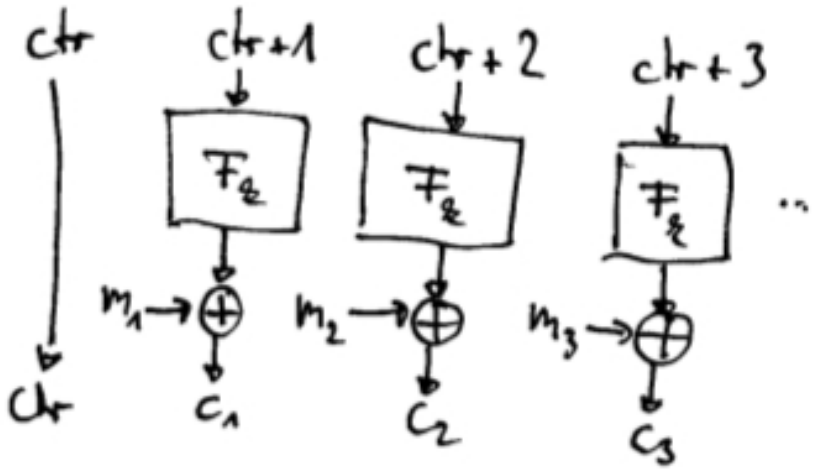
- Stateful variant of CBC
- can be attacked

Output feedback mode (OFB)



- CPA-secure if F is a prf.
- Evaluation of F can be done before actual encryption

Counter mode (CTR)



- $\mathrm{ctr} \in \{0,1\}^n$ is chosen uniformly at random.
- Encryption: $c_i := m_i \oplus F_k(\mathrm{ctr} + i)$.
- CTR can be parallelized.
- ith block can be decrypted individually with only one evaluation of F.

Theorem

If F is a prf, then CTR is CPA-secure.