Limitations of Perfect Secrecy

Theorem: Let (Gen, Enc, Dec) be a perfectly-secret encryption scheme over a message space $\boldsymbol{M}$, and let $\boldsymbol{K}$ be the key space as determined by Gen. Then $|\boldsymbol{K}| \geq |\boldsymbol{M}|$.

The Computational Approach

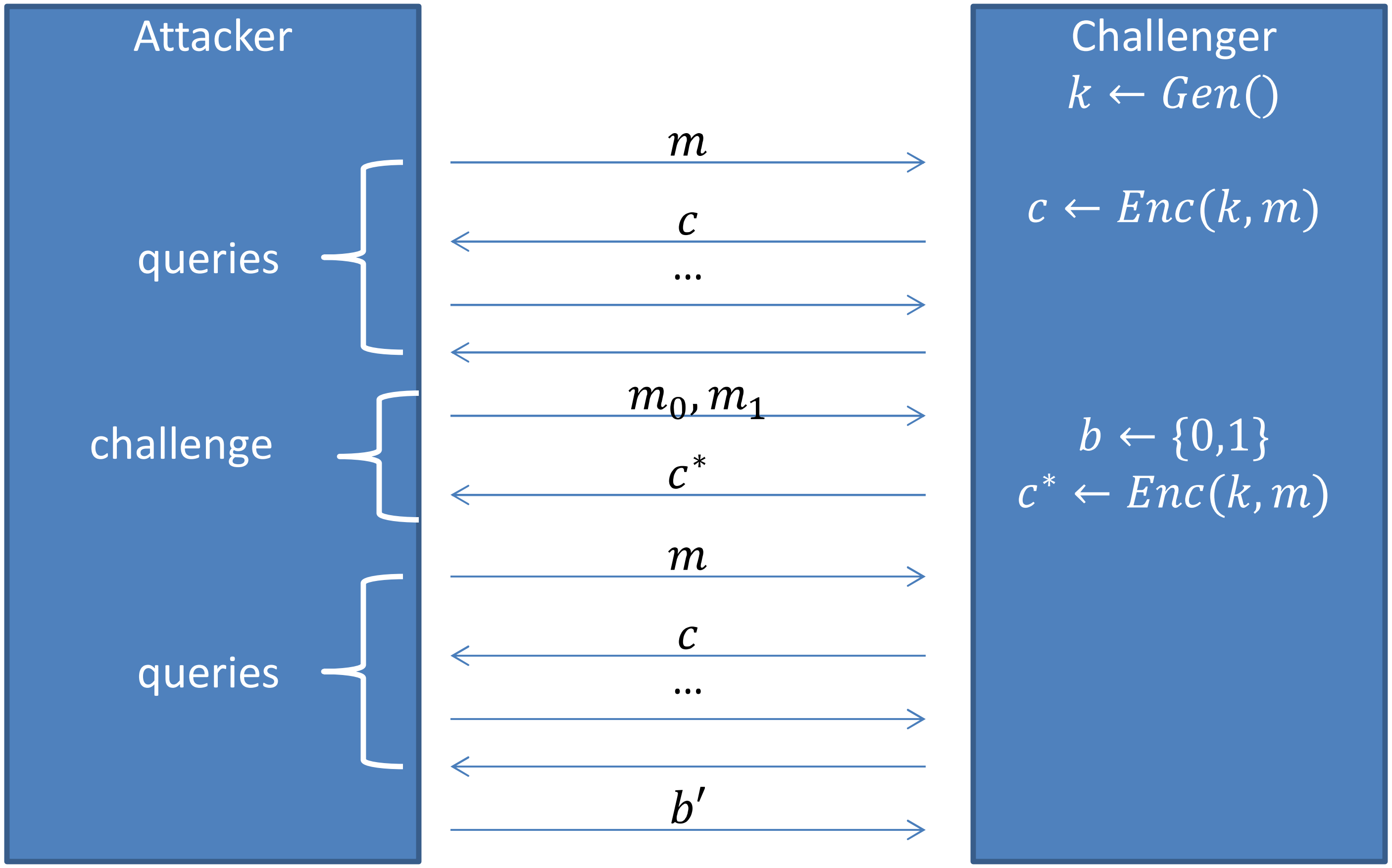
Two main relaxations:

1. Security is only guaranteed against efficient adversaries that run for some feasible amount of time.
2. Adversaries can potentially succeed with some very small probability.

Practical Implications of Computational Security

- For key size n, any adversary running in time $2^{n/2}$ breaks the scheme with probability $1/2^{n/2}$.
- Meanwhile, Gen, Enc, Dec each take time n^2.
- If $n = 128$ then:
  - Gen, Enc, Dec take time 16,384
  - Adversarial run time is $2^{64} \approx 10^{18}$
- If n = 256 then:
  - Gen, Enc, Dec quadruples--takes time 65,536
  - Adversary run time is multiplied by 2^{64}. Becomes $2^{128} \approx 10^{38}$

CPA Security



Attacker

queries

challenge

queries

m

c

…

m_0, m_1

c^*

m

c

…

b'

Challenger

$k \leftarrow Gen()$

$c \leftarrow Enc(k, m)$

$b \leftarrow \{0,1\}$

$c^* \leftarrow Enc(k, m)$

Attacker “wins” if $b' = b$.

CPA Security: Any efficient attacker wins with probability at most $\frac{1}{2} + negligible$

CPA-secure Encryption Must Be Probabilisitic

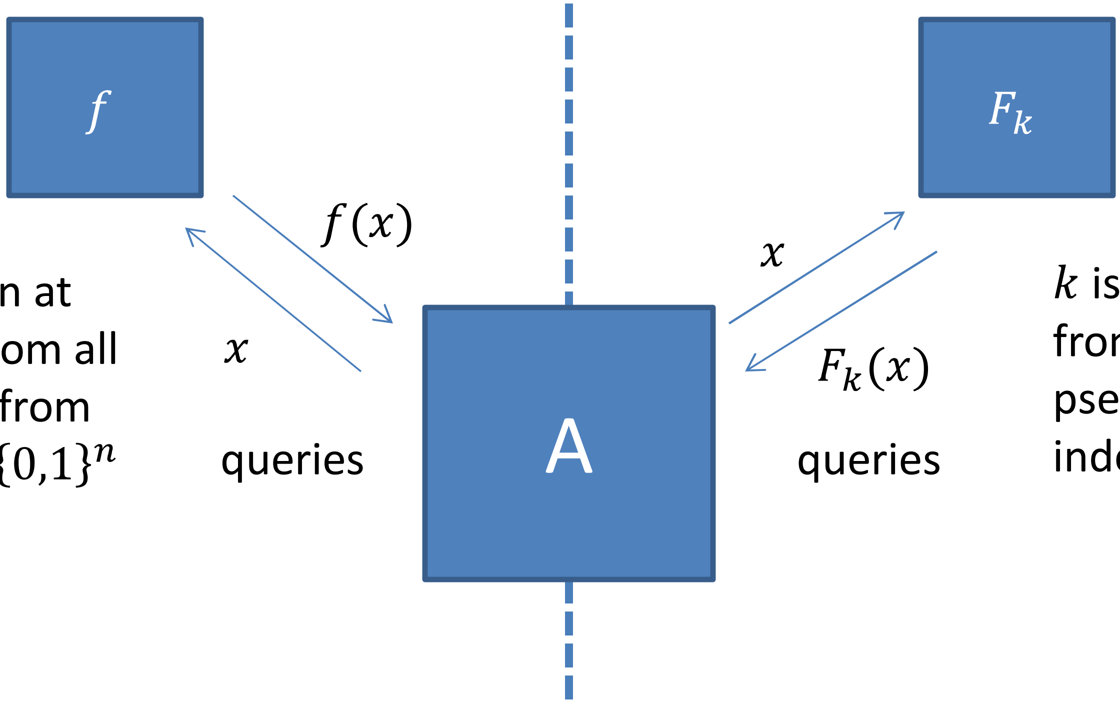
Theorem: If $\Pi = (Gen, Enc, Dec)$ is an encryption scheme in which Enc is a deterministic function of the key and the message, then Π cannot be CPA-secure.

Why not?

Pseudorandom Function

Definition: A keyed function $F: \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ is a two-input function, where the first input is called the key and denoted k.

Pseudorandom Function (PRF)



f is chosen at random from all functions from $\{0,1\}^n$ to $\{0,1\}^n$

k is chosen at random from $\{0,1\}^n$. F_k is the pseudorandom function indexed by k.

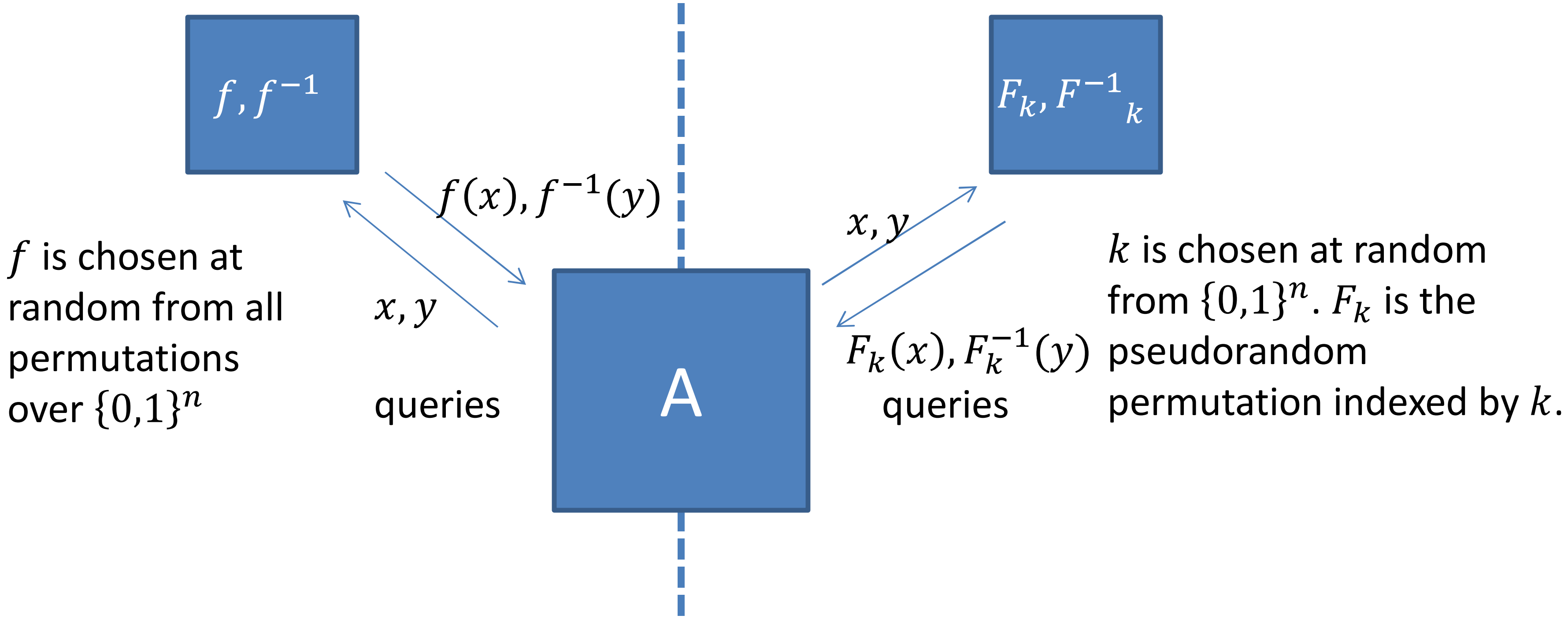
PRF: Any efficient A cannot tell which world it is in.

$$\left|\Pr[A^{f}(\,) = 1] - \Pr[A^{F_k}(\,) = 1]\right| \leq negligible$$

Block Ciphers/Pseudorandom Permutations

Definition: Pseudorandom Permutation is exactly the same as a Pseudorandom Function, except for every key k, F_k must be a permutation and it must be indistinguishable from a random permutation.

Pseudorandom Permutation (PRP) Block Cipher



f, f^{-1}

$f(x), f^{-1}(y)$

x, y

queries

f is chosen at random from all permutations over $\{0,1\}^n$

A

$F_k, F^{-1}{}_k$

x, y

$F_k(x), F_k^{-1}(y)$

queries

k is chosen at random from $\{0,1\}^n$. F_k is the pseudorandom permutation indexed by k.

PRP: Any efficient A cannot tell which world it is in.

$$\left|\Pr[A^{f}(\,) = 1] - \Pr[A^{F_k}(\,) = 1]\right| \leq negligible$$

Modes of Operation—Block Cipher

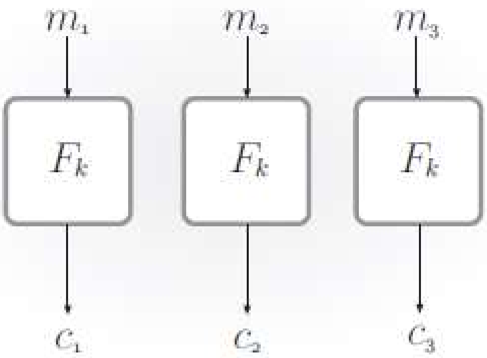


FIGURE 3.5: Electronic Code Book (ECB) mode.

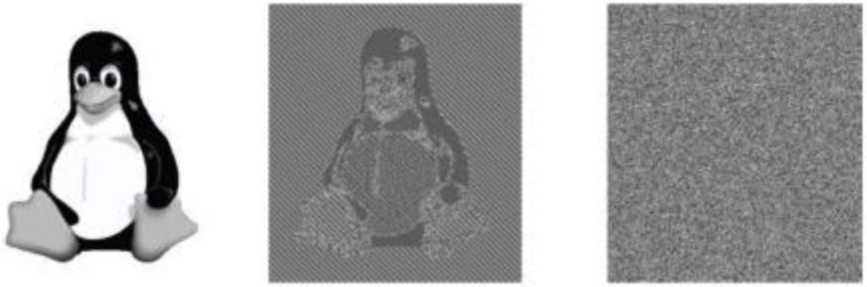


FIGURE 3.6: An illustration of the dangers of using ECB mode. The middle figure is an encryption of the image on the left using ECB mode; the figure on the right is an encryption of the same image using a secure mode.

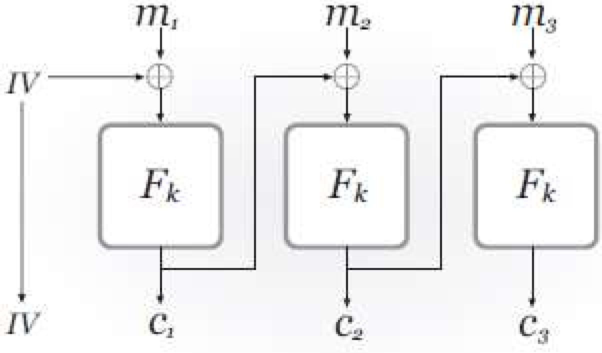


FIGURE 3.7: Cipher Block Chaining (CBC) mode.

Modes of Operation—Block Cipher

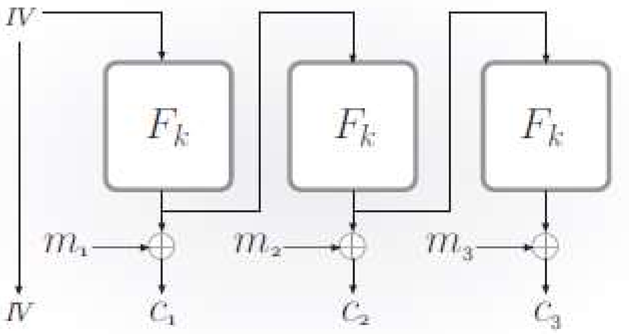


FIGURE 3.9: Output Feedback (OFB) mode.

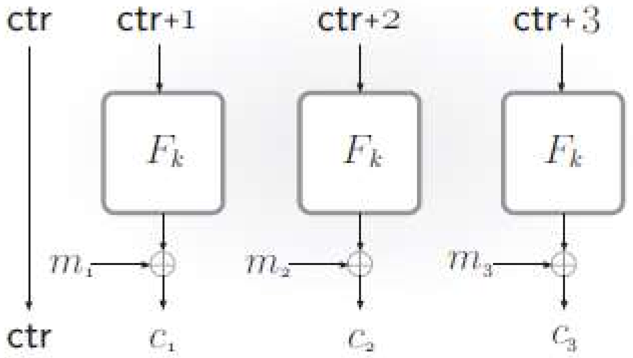


FIGURE 3.10: Counter (CTR) mode.

Details on DES

- The Data Encryption Standard was developed in the 1970s by IBM (with help from the National Security Agency), and adopted in 1977 as a Federal Information Processing Standard for the US.
- DES is no longer considered secure due to its short key length of 56 bits which makes it vulnerable to brute-force attacks.
- It remains in wide use today in the strengthened form of triple-DES, described in Section 6.2.4.
- DES is of great historical significance. It has undergone intensive scrutiny within the cryptographic community, arguably more than any other cryptographic algorithm in history. The common consensus is that, relative to its key length, DES is an extremely well designed cipher.
  - To date, the best known attack on DES in practice is an exhaustive search over all 2^{56} possible keys.

Details on DES

- The DES block cipher is a 16-round Feistel network with a block length of 64 bits and a key length of 56 bits. The same round function $\hat{f}$ is used in each of the 16 rounds.
- Round function takes a 48-bit sub-key and, as in a (balanced) Feistel network, a 32-bit input
- The key schedule of DES is used to derive a sequence of 48-bit sub-keys $k_1, \ldots, k_{16}$ from the 56-bit master key.

3DES (Triple Encryption)

- First Idea: increase the key length by doing a double-encryption, thereby increasing complexity of brute-force attack from 2^{56} to 2^{112}.
- Let F be a block-cipher with an n-bit key length and ℓ-bit block length.
  - Define the following block cipher with $2n$-bit key: $F'_{k_1,k_2}(x) \coloneqq F_{k_2}(F_{k_1}(x))$
- Problem: Meet in the middle attack

Meet in the Middle Attack on Double DES

Adversary is given a single input/output pair (x, y) where $y = F_{k^*{}_1, k^*{}_2}(x)$ for unknown k_1, k_2. The adversary does the following:

- For each $k_1 \in \{0,1\}^n$, compute $z := F_{k_1}(x)$ and store (z, k_1) in a list L.
- For each $k_2 \in \{0,1\}^n$, compute $z := F_{k_2}{}^{-1}(y)$ and store (z, k_2) in a list L'.
- Sort L and L', respectively, by their first components.
- Entries $(z_1, k_1) \in L$ and $(z_2, k_2) \in L'$ are a match if $z_1 = z_2$. For each match of this sort, add (k_1, k_2) to a set S.

Expected number of elements in S is $2^{2n-\ell}$. Can use a few more input/output pairs to reduce to a single (k_1, k_2) .

Triple DES

Two variants:

- ${F'}_{k_1,k_2,k_3}(x) \coloneqq F_{k_3}({F_{k_2}}^{-1}(F_{k_1}(x)))$
- ${F'}_{k_1,k_2}(x) \coloneqq F_{k_1}({F_{k_2}}^{-1}(F_{k_1}(x)))$

- Middle cipher is reversed for backwards compatibility: setting $k_1 = k_2 = k_3$ results in a single invocation of F using key k_1.

Security of Triple-DES

- Security of the first variant: The cipher is susceptible to a meet-in-the-middle attack just as in the case of double encryption, though the attack now takes time 2^{2n}. This is the best known attack.
- Security of the second variant. There is no known attack with time complexity better than 2^{2n} when the adversary is given only a small number of input/output pairs. Thus, two-key triple encryption is a reasonable choice in practice.

Disadvantage of both Triple-DES variants: Fairly slow since it requires 3 invocations of DES.

Details on AES

- In January 1997, the United States National Institute of Standards and Technology (NIST) announced a competition to select a new block cipher—to be called the Advanced Encryption Standard, or AES
- 15 submissions from all over the world. Each team’s candidate cipher was intensively analyzed by members of NIST, the public, and (especially) the other teams. Two workshops were held (‘98, ’99) to analyze the various submissions. Following the second workshop, NIST narrowed the field down to 5 “finalists” and the second round of the competition began. A third AES workshop was held in April 2000, inviting additional scrutiny on the five finalists.
- In October 2000, NIST announced that the winning algorithm was Rijndael (a block cipher designed by Belgian cryptographers Vincent Rijmen and Joan Daemen)

Details on AES

A 4-by-4 array of bytes called the **state** is modified in a series of rounds. The state is initialized to the input to the cipher (128 bits = 16 bytes). The following operations are then applied in each round:

1. Stage 1 – AddRoundKey: A 128-bit sub-key is derived from the master key, and is interpreted as a 4-by-4 array of bytes. **state** updated by XORing it with this sub-key.
2. Stage 2 – SubBytes: Each byte of **state** is replaced by another byte according to a single fixed lookup table S. This substitution table (or S-box) is a bijection over $\{0, 1\}^8$.
3. Stage 3 – ShiftRows: The bytes in each row of **state** are cyclically shifted to the left as follows: the first row of the array is untouched, the second row is shifted one place to the left, the third row is shifted two places to the left, and the fourth row is shifted three places to the left. All shifts are cyclic so that, e.g., in the second row the first byte becomes the fourth byte.
4. Stage 4 – MixColumns: An invertible transformation is applied to the four bytes in each column. (linear transformation—i.e., matrix multiplication—over an appropriate field.)

   If two inputs differ in b > 0 bytes, then transformation yields two outputs differing in at least 5 − b bytes.

   In the final round, MixColumns is replaced with AddRoundKey. Why?

- To date, no practical cryptanalytic attacks significantly better than a exhaustive search.