Cybersecurity Fundamentals

Lecture 6

Symmetric Cryptography

Thomas Zacharias

University of Glasgow

Cryptography as a scientific field

The field of Computer Science that formally studies the problem of secure interaction among parties in the presence of a malicious entity (adversary), even in settings where the parties are not mutually trusted.

The role of Cryptography in Cybersecurity

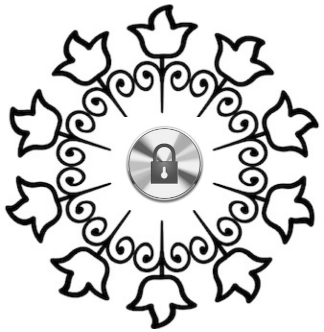
- Cryptographic primitives can serve as fundamental defence mechanisms with respect to all key Cybersecurity objectives.
  - Confidentiality (e.g., encryption).
  - Integrity (e.g., message authentication codes, digital signatures, hash functions).
  - Availability (e.g., threshold cryptography).
- Cryptography can contribute to both the prevention and the detection of attacks on a protected system.
- Security argumentation in modern Cryptography is provable, i.e., based on formal definitions and well-defined models.

Cryptography in real world applications

- Secure web browsing
- User and message authentication
- Electronic banking
- Mail encryption
- Disk encryption
- Secure messaging
- Cryptocurrencies
- Data privacy
- File checksums

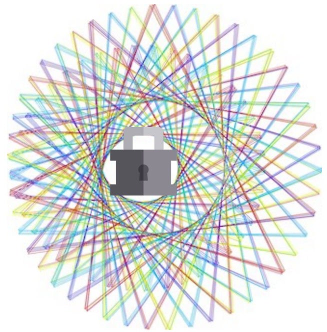
⋮

Symmetric Cryptography



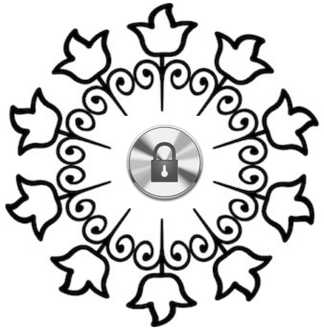
- The parties interact securely via the same cryptographic key.
- The key must be securely shared before interaction.
- Symmetric-key algorithms are fast.

Asymmetric Cryptography



- Secure interaction is via a public key and a corresponding private key.
- The private key never needs to be shared.
- Asymmetric-key algorithms are (currently) significantly slower.

In this lecture

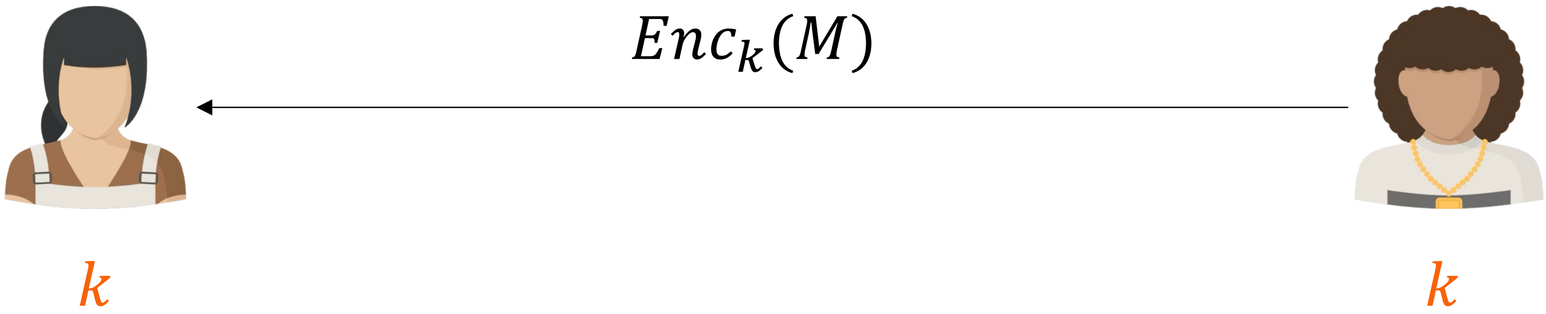


- Definition of symmetric encryption.
- Ciphers – Advanced Encryption Standard (AES).
- Modes of operation.
- Message authentication codes.
- Authenticated encryption.

Symmetric encryption

- Alice and Bob share a secret key k.
- Bob wans to send a message M through an insecure channel.

Symmetric encryption



Symmetric encryption

$$Dec_k(Enc_k(M))$$
$$\downarrow$$
$$M$$

k

k

Definition of symmetric encryption: Syntax and Correctness

- A symmetric encryption scheme is a triple of algorithms as follows:
  - A key-generation algorithm Gen that outputs a key k.
  - An encryption algorithm Enc that takes as input a key k and a message (plaintext) M and outputs a ciphertext C. We write $C \leftarrow Enc_k(M)$.
  - A decryption algorithm Dec that takes as input a key k and a ciphertext C and outputs a message M. We write $M \leftarrow Dec_k(C)$.

- **Correctness**: for every key k and every message M, it holds that

$$M \leftarrow Dec_k\left(Enc_k(M)\right)$$

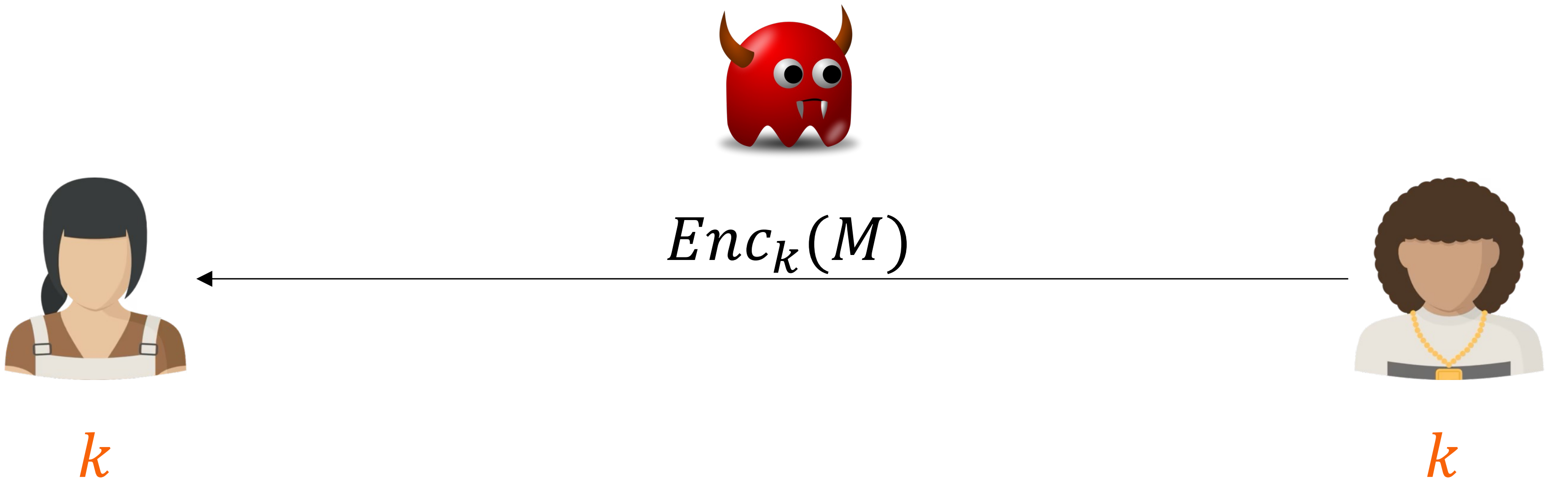
Definition of symmetric encryption: Security (Indistinguishability under chosen plaintext attack)

A symmetric encryption scheme achieves indistinguishability under chosen plaintext attack (IND-CPA security), if it leaks no information about Bob's message M against any adversary that

(1) observes the communication channel and

(2) can obtain encryptions for messages of its choice

(i.e., it has access to oracle $Enc_k(\cdot)$ but does not know k).



Ciphers

- Ciphers are concrete algorithms for performing encryption or decryption.
- Types of modern ciphers:
  - **Block ciphers**: process the input message in fixed-size blocks and produce a block of ciphertext for each message block.
    - Examples: Data Encryption Standard (DES), Advanced Encryption Standard (AES), Serpent
    - Advantages: generally, they offer a higher level of security (keys can be reused)
    - Applications: blocks of data (e.g., file transfer, e-mail)
  - **Stream ciphers**: process input messages of variable length by performing encryption bit-by-bit (or byte-by-byte), typically via XOR with a pseudorandomly generated keystream.
    - Examples: RC4, Salsa20, Grain128a
    - Advantages: better performance
    - Applications: streams of data (e.g., a secure wireless connection)

Design principles of a secure cipher [Claude Shannon, 1945]

- **Confusion**: the relationship between the encryption key and the ciphertext should be as complex as possible. In particular, each ciphertext bit should depend on several parts of the key.
- **Diffusion**: each plaintext bit should affect as many ciphertext bits as possible. In particular,
  - if we change a bit of the plaintext, then several ciphertext bits should change and
  - if we change a bit of the ciphertext, then several plaintext bits should change.

Preliminaries: the Exclusive OR (XOR) operator

- The XOR operator for single bits is defined as follows:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

- Bitwise XOR for strings naturally extends the operator for single bits. E.g.,

$$\begin{array}{r} 11010010 \\ 01011001 \oplus \\ \hline 10001011 \end{array}$$

- Property of bitwise XOR: $(M \oplus k) \oplus k = M \oplus (k \oplus k) = M \oplus 0 = M$

The Advanced Encryption Standard (AES)

- Originally proposed as Rijndael, developed by Belgian cryptographers Vincent Rijmen and Joan Daemen.
- Standardised as AES in 2001 by US NIST after a 4 year competition among 15 candidates.
  - Available in standard crypto libraries.
  - Integrated in many modern processors.
- Block cipher specifications:
  - **Key length**: 128, 192, or 256 bits
  - **Block length**: 128 bits
  - **Number of rounds**: 10, 12, and 14 rounds for 128-bit keys, 192-bit keys, and 256-bit keys, respectively.

Overview of AES-128 (encryption)

- The 128-bit input message is arranged into 16 bytes $b_0 \cdots b_{15}$ according to the following 4×4 matrix:

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

- The above matrix initialises the state of the cipher, $STATE$, which is a 4×4 matrix modified at each step of the encryption process.

Overview of AES-128 (encryption)

- The 128-bit/16-byte key $k_0 \cdots k_{15}$ is arranged into 4 words K_0, K_1, K_2, K_3, where each word is a 4-byte column as follows:

$$K_0 = \begin{array}{|c|}\hline k_0 \\ \hline k_1 \\ \hline k_2 \\ \hline k_3 \\ \hline\end{array} \qquad K_1 = \begin{array}{|c|}\hline k_4 \\ \hline k_5 \\ \hline k_6 \\ \hline k_7 \\ \hline\end{array} \qquad K_2 = \begin{array}{|c|}\hline k_8 \\ \hline k_9 \\ \hline k_{10} \\ \hline k_{11} \\ \hline\end{array} \qquad K_3 = \begin{array}{|c|}\hline k_{12} \\ \hline k_{13} \\ \hline k_{14} \\ \hline k_{15} \\ \hline\end{array}$$

- K_0, K_1, K_2, K_3 are provided as input to a `KeyExpansion` function that outputs 44 words $W_0, W_1, \ldots, W_{43}$.
- The `KeyExpansion` function produces separate keys, where each key will be used in a single round.
- $W_0, W_1, \ldots, W_{43}$ are arranged as 11 round keys $[W_0\ W_1\ W_2\ W_3], [W_4\ W_5\ W_6\ W_7], \ldots, [W_{40}\ W_{41}\ W_{42}\ W_{43}]$, where each round key is a 4×4 matrix.

Overview of AES-128 (encryption)

- **Initial step (round 0):**
  1. the $\texttt{AddRoundKey}(A, B)$ function outputs the bitwise XOR of A and B.
     - The state (currently, the input message) is XORed with the first round key.
     - $STATE \leftarrow \texttt{AddRoundKey}(STATE, [W_0\ W_1\ W_2\ W_3])$.
     - The function adds confusion by mixing the key with the state.

Overview of AES-128 (encryption)

- **Intermediary rounds** 1,…,9:
  1. the SubBytes function performs a byte-by-byte substitution of the state via a look up table (the S-box).
     - E.g., byte 00_{16} is substituted by 63_{16} , 01_{16} is substituted by $7c_{16}$, 02_{16} is substituted by 77_{16} ,…
     - $STATE \leftarrow \texttt{SubBytes}(STATE)$.
     - Non-linear (adds confusion). E.g., $\texttt{SubBytes}(A \oplus B) \neq \texttt{SubBytes}(A) \oplus \texttt{SubBytes}(B)$
  2. the ShiftRows function permutes the entries in the 2nd, 3rd, and 4th row of the state by performing a cyclical left shift of 1, 2, and 3 positions, respectively.
     - E.g., 2nd row: $s_{1,1}\ s_{1,2}\ s_{1,3}\ s_{1,0} \leftarrow s_{1,0}\ s_{1,1}\ s_{1,2}\ s_{1,3}$
     - $STATE \leftarrow \texttt{ShiftRows}(STATE)$.
     - Now the columns of the state are not encrypted independently, which produces diffusion.
  3. the MixColumns function that alters each byte in a column as a function of all of the bytes in the column.
     - $STATE \leftarrow \texttt{MixColumns}(STATE)$.
     - Now changing one byte affects other bytes of the state producing diffusion.
  4. at the end of the round, AddRoundKey is performed with the round key.

Overview of AES-128 (encryption)

- **Final round** 10:
  1. $STATE \leftarrow \texttt{SubBytes}(STATE)$.
  2. $STATE \leftarrow \texttt{ShiftRows}(STATE)$.
  3. $STATE \leftarrow \texttt{AddRoundKey}(STATE, [W_{40}\ W_{41}\ W_{42}\ W_{43}])$.
  4. Output ciphertext $C \leftarrow STATE$.

Overview of AES-128 (decryption)

- All functions, `AddRoundKey`, `SubBytes`, `ShiftRows`, and `MixColumns` are invertible.
  - Note: the inverse of `AddRoundKey(`round key,·) is itself.
- The decryption algorithm recovers the message by
  - making use of the round keys in reverse order.
  - running the inverses of the AES functions in reverse order.

Overview of AES-128

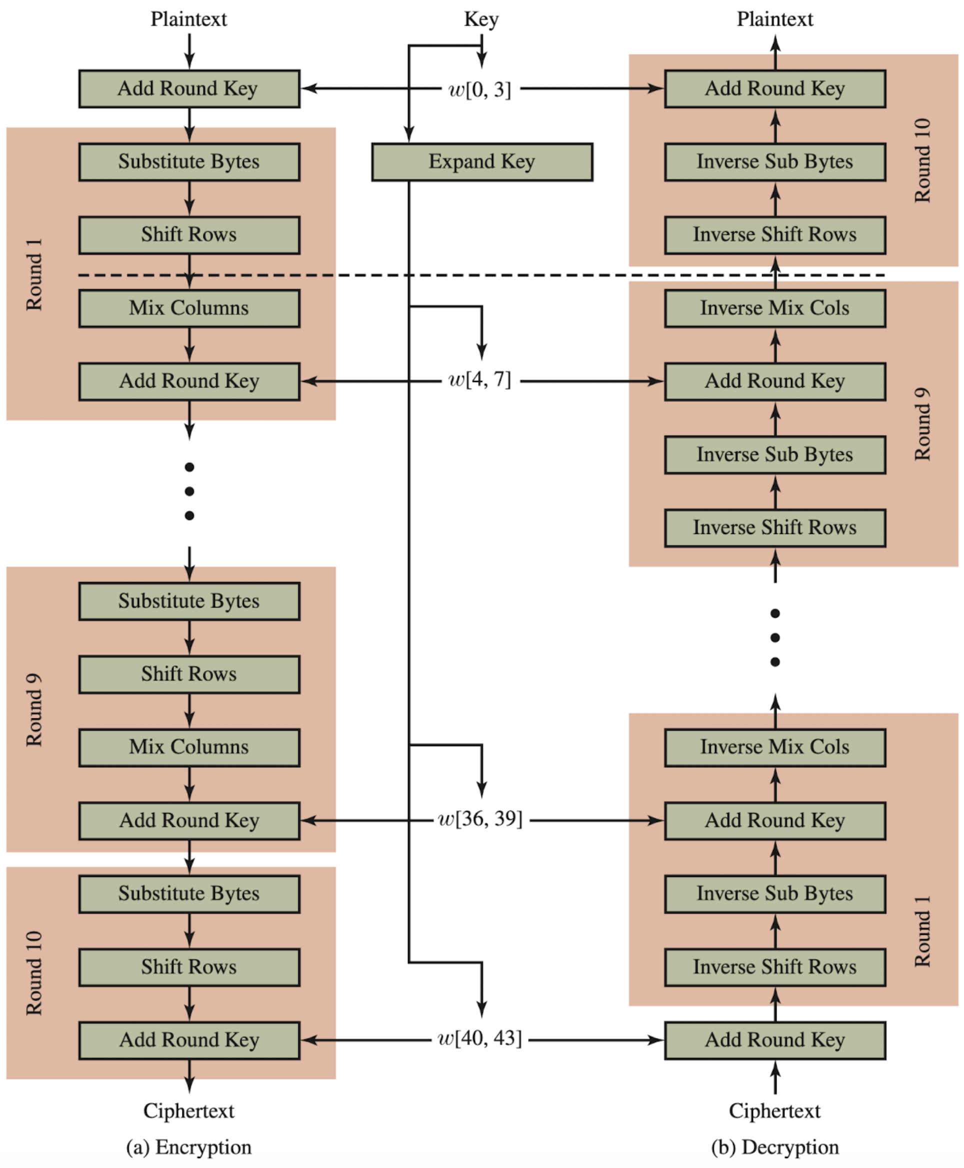


Figure source: "Computer Security Principles and Practice (5th Edition)" by Stallings and Brown

Block ciphers as pseudorandom permutations

- State-of-the-art block ciphers such as AES realise the cryptographic primitive named pseudorandom permutations (PRPs).
- Definition:

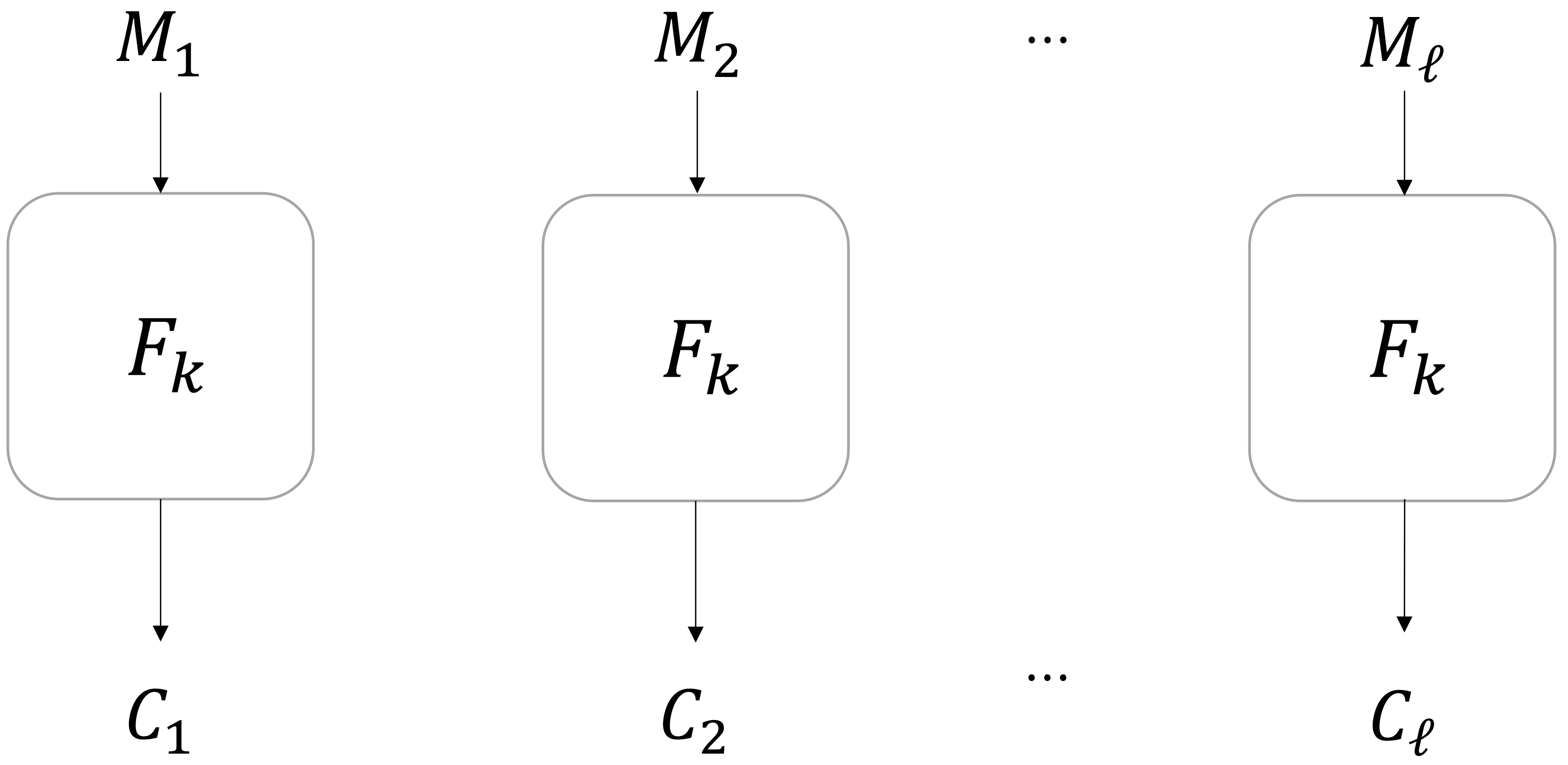
> A PRP is a keyed permutation $F_k\colon \{0,1\}^m \to \{0,1\}^m$ that behaves as a uniformly at random permutation $\sigma\colon \{0,1\}^m \to \{0,1\}^m$, in the view of any party that does not know k.
> Its inverse F_k^{-1} is efficiently computable.

Encryption of long messages: Modes of operation

- Block ciphers/PRPs can encrypt short fixed-size messages (e.g., AES message length $m = 128$ bits).
- Modes of operation are algorithms that use a block cipher to support encryption/decryption of long messages.
  - If needed, a message M is securely padded until its length is a multiple of m.
  - The (padded) message is M is processed as a sequence of blocks $M_1, M_2, \ldots, M_\ell$, where each block is of length m.

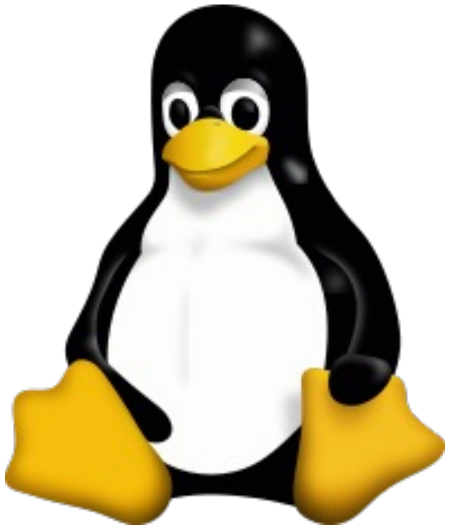
Electronic Code Book (ECB) mode

- Encryption of message $M = M_1 \mid\mid M_2 \mid\mid \cdots \mid\mid M_\ell$ under k:



Electronic Code Book (ECB) mode

- ECB is deterministic, so encrypting a repeated message block will result in a repeated ciphertext block!
- Insecure mode of encryption, as it leaks information to an eavesdropper.
- Illustrative example:



Original image

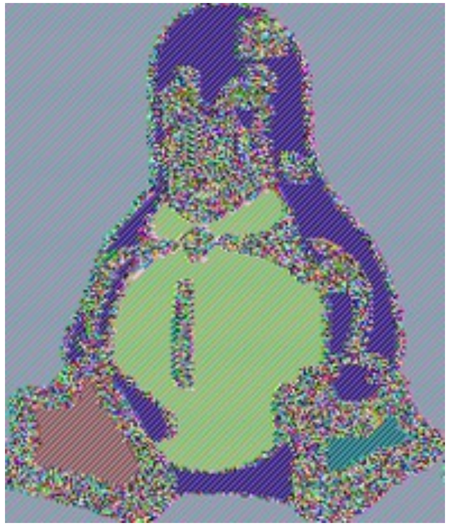
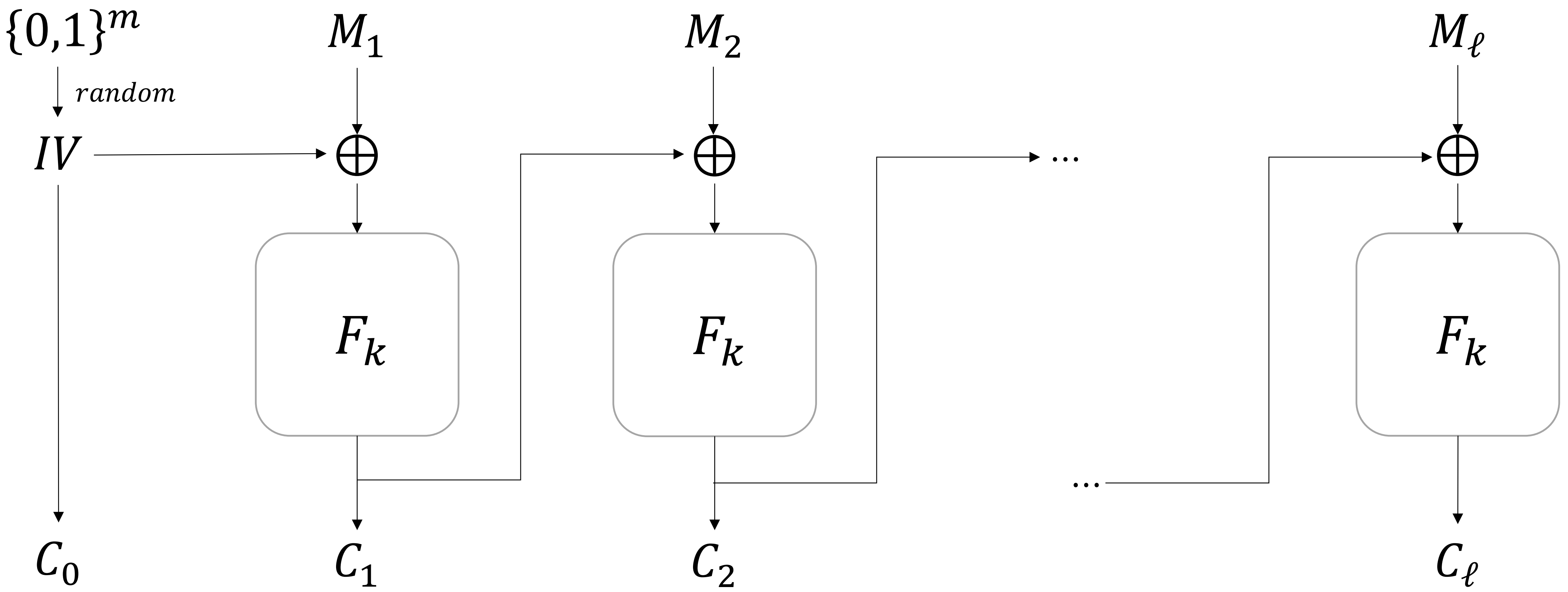


Image encrypted using ECB

Figure source: Wikipedia

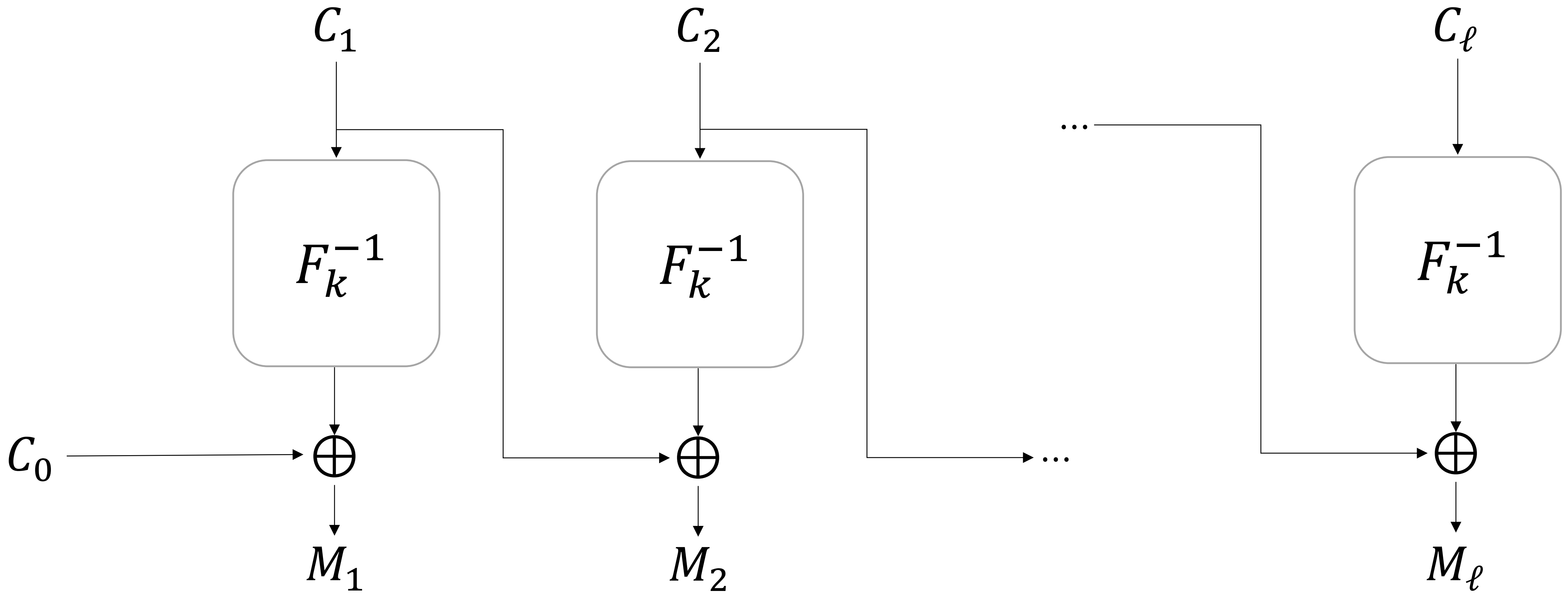
Cipher Block Chained (CBC) mode

- Encryption of message $M = M_1 \,||\, M_2 \,||\, \cdots \,||\, M_\ell$ under k:



Cipher Block Chained (CBC) mode

- Decryption of ciphertext $C = C_0 \,||\, C_1 \,||\, C_2 \cdots \,||\, C_\ell$ under k:



Cipher Block Chained (CBC) mode

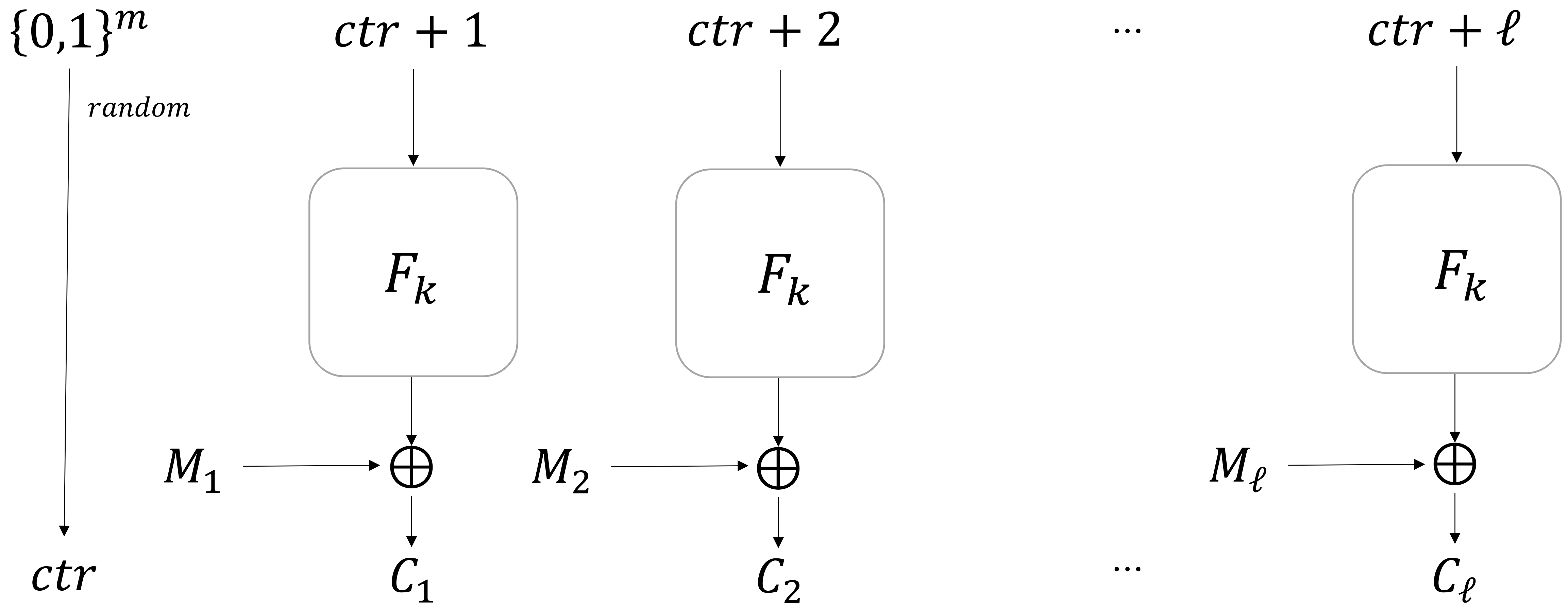
- The initialisation vector (IV) should be randomly chosen. Randomness ensures that repeated messages will be encrypted differently.
- Using a predictable IV can leak information to the adversary.

If F_k is a PRP, then the CBC mode of operation is IND-CPA secure.

- **Disadvantages**:
  - Encryption is sequential (no parallelisation) which affects performance.
  - Not tolerant to block losses; if a block is lost during transmission, then the subsequent block cannot be decrypted correctly.

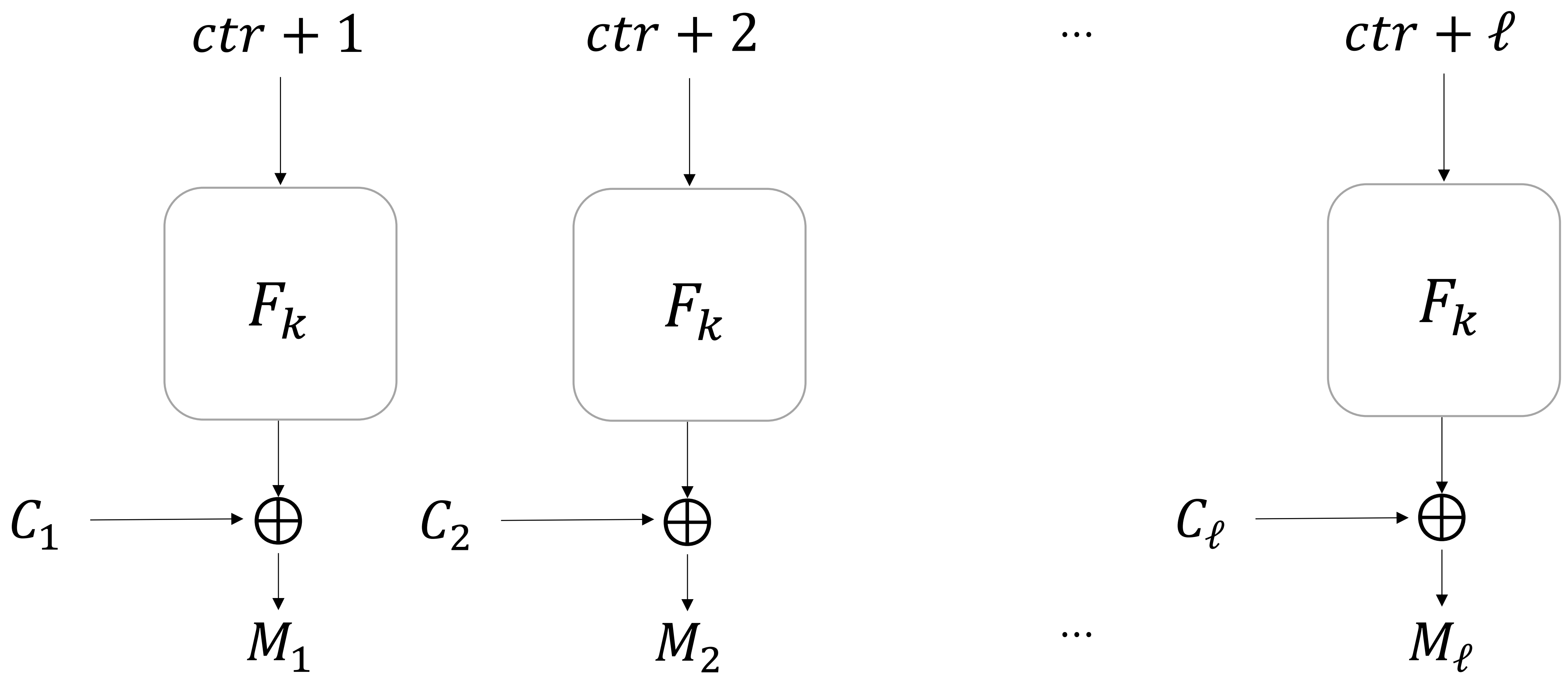
Counter (CTR) mode

- Encryption of message $M = M_1 \mid\mid M_2 \mid\mid \cdots \mid\mid M_\ell$ under k:



Counter (CTR) mode

- Decryption of ciphertext $C = ctr \mid\mid C_1 \mid\mid C_2 \cdots \mid\mid C_\ell$ under k:



Counter (CTR) mode

- The counter $ctr + i$ should be unique (e.g., nonce ctr may be randomly chosen). Uniqueness ensures that repeated messages will be encrypted differently.

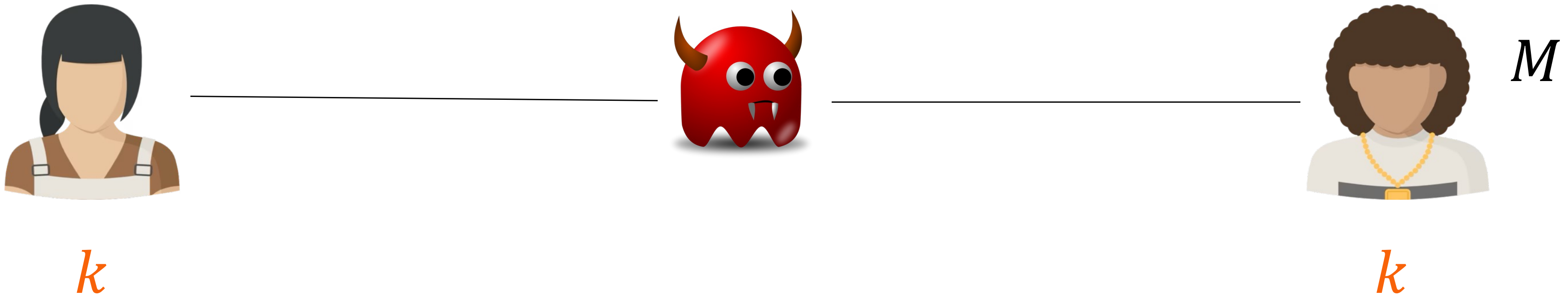
If F_k is a PRP, then the CTR mode of operation is IND-CPA secure.

- **Advantages**:
  - Encryption and decryption can be parallelised.
  - Requires only the implementation of the underlying cipher's encryption algorithm (F_k^{-1} is never used).
  - Preprocessing of the cryptographic operations (F_k evaluations) is possible; when the message blocks it is XORed with the pre-computed cryptographic output.

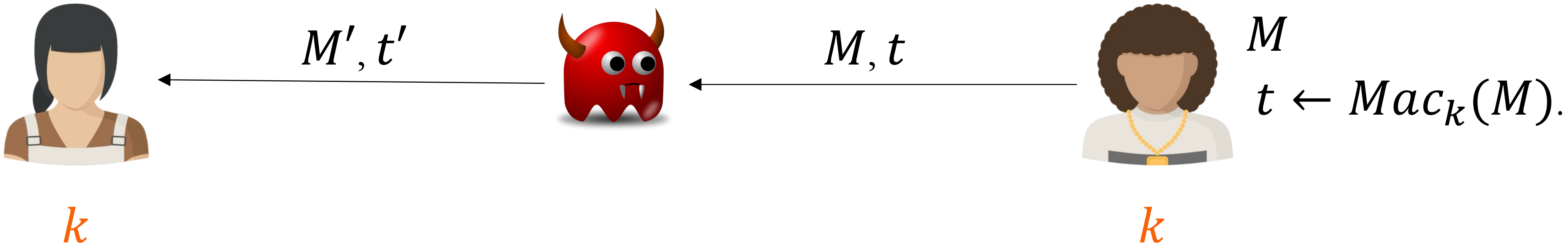
Message integrity

- Secure modes of operation achieve IND-CPA security, i.e., confidentiality against a passive attacker that does not attempt to modify the data.
- They do not provide defence against active attackers that may tamper the message contents (e.g., flipping the bits of the ciphertext will change the result of the decryption without this being detected).
- In Cybersecurity, message integrity is a key objective!
- Symmetric Cryptography introduces message authentication codes (MACs) as means to preserve message integrity.

Message integrity using a MAC

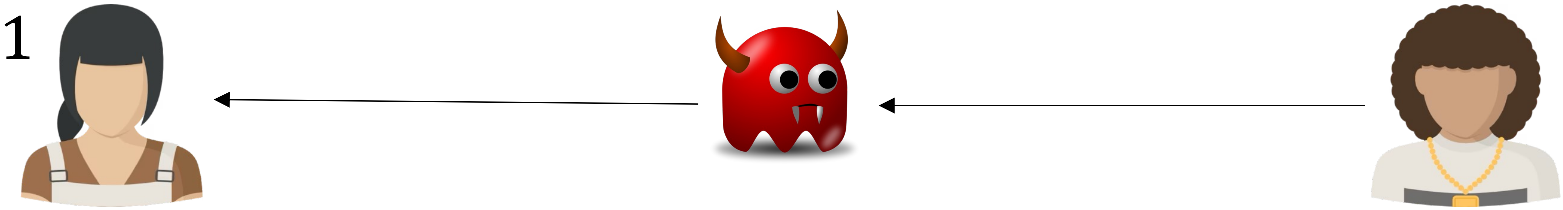


Message integrity using a MAC



Message integrity using a MAC

$Vrfy_k(M', t') \stackrel{?}{=} 1$



k k

Definition of MACs: Syntax, Correctness, and Security

- A message authentication code is a triple of algorithms as follows:
  - A key-generation algorithm Gen that outputs a key k.
  - A tag-generation algorithm Mac that takes as input a key k and a message (plaintext) M and outputs a tag t. We write $t \leftarrow Mac_k(M)$.
  - A verification algorithm $Vrfy$ that takes as input a key k, a message M, and a tag t, and outputs a bit b. We write $b \coloneqq Vrfy_k(M, t)$.

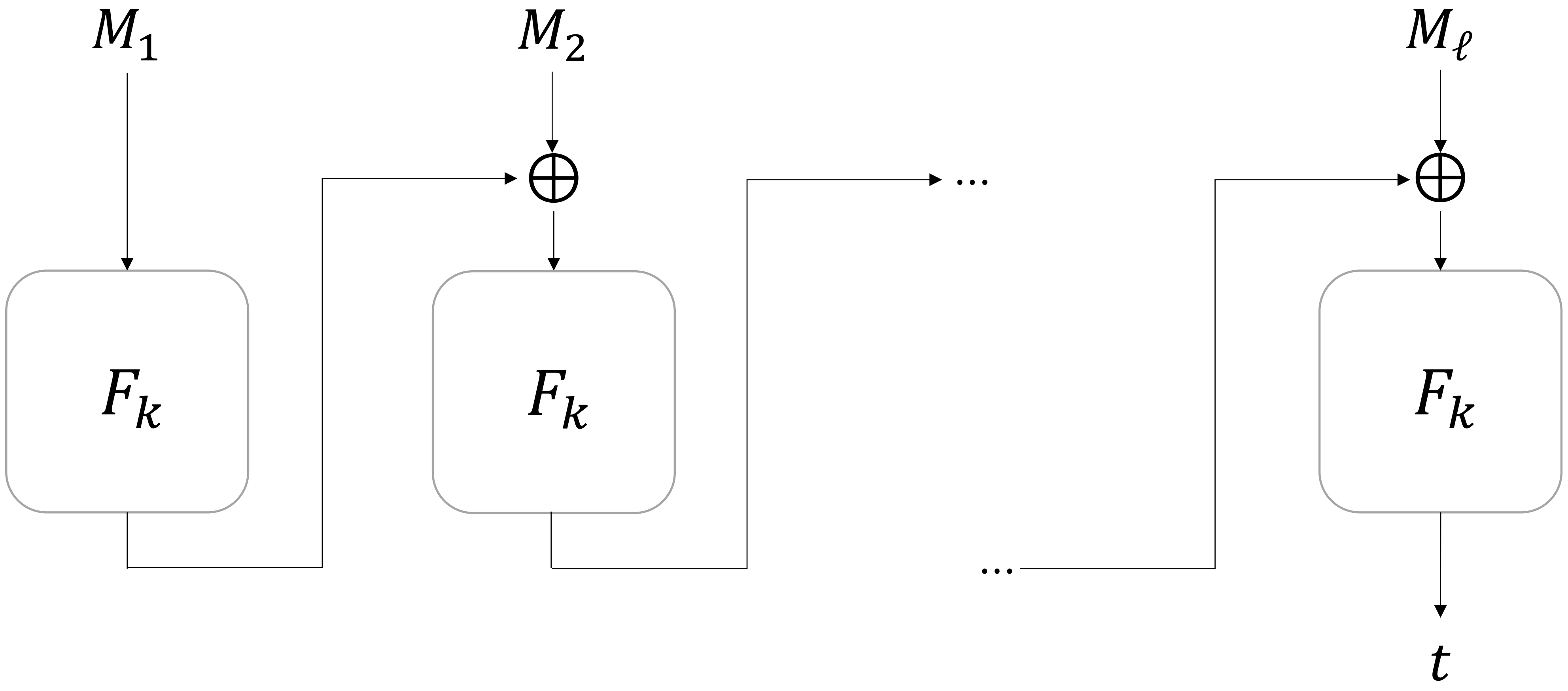
- **Correctness**: for every key k and every message M, it holds that

$$Vrfy_k\big(M, Mac_k(M)\big) = 1$$

- **Security (Existential Unforgeability)**: no adversary can generate a valid tag on a new message that was not previously sent (and authenticated) by one of the communicating parties.

A MAC construction: CBC-MAC

- Tag-generation for message $M = M_1 \mid\mid M_2 \mid\mid \cdots \mid\mid M_\ell$ under k:



A MAC construction: CBC-MAC

- No initialisation vector is needed, the algorithm is deterministic.

If F_k is a PRP, then CBC-MAC achieves existential unforgeability for messages of fixed length $\ell \cdot m$.

- (Basic) CBC-MAC is not secure for variable-length messages.
  - There are ways to modify the basic construction so that it is secure for variable-length messages. E.g., encrypt the value t using a different key $\hat{k}$ and output the tag $\hat{t} = F_{\hat{k}}(t)$.
  - Alternatively, other constructions such as CMAC or HMAC can be used to provide message integrity for variable-length messages.

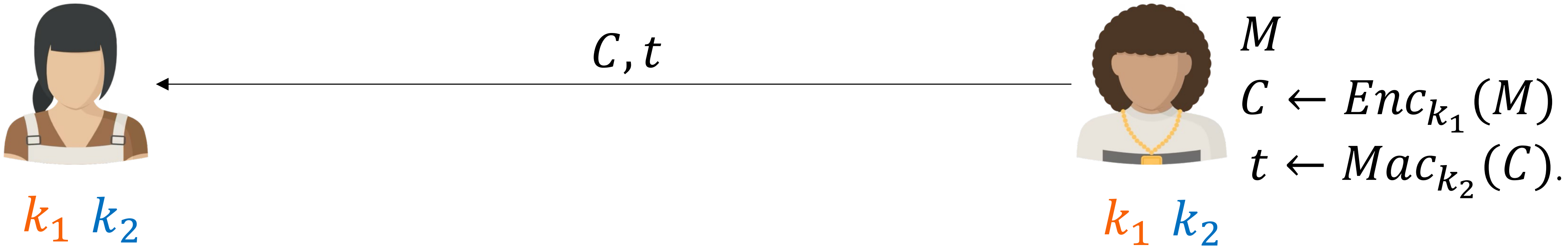
Confidentiality and Integrity

- When designing MACs, confidentiality is not a concern, only the integrity of the message is guaranteed.
- Modes of operation protect the confidentiality of the message.
- **Challenge**: Can we combine the two primitives, so that both confidentiality and integrity are achieved?
- **Solution**: Encrypt-then-Authenticate!

Encrypt-then-Authenticate

- Alice and Bob share a symmetric encryption key k_1 and a MAC key k_2.
- Bob wans to send a message M through an insecure channel.

Encrypt-then-Authenticate



Encrypt-then-Authenticate

$$Vrfy_{k_2}(C, t) \stackrel{?}{=} 1$$

$$\Downarrow$$

$$M \leftarrow Dec_{k_1}(C)$$

k_1 k_2

k_1 k_2

Encrypt-then-Authenticate

If the underlying encryption scheme is IND-CPA secure and the underlying MAC achieves existential unforgeability,
then the Encrypt-then-Authenticate combination achieves confidentiality and integrity.

Authenticated encryption

- Authenticated Encryption schemes are encryption schemes that ensure both confidentiality and integrity.
- Encrypt-then-Authenticate is a method of constructing Authenticated Encryption via a black-box use of an IND-CPA encryption scheme and a secure MAC.
- Alternatively, more efficient Authenticated Encryption constructions exist. E.g.,
  - Galois Counter Mode (GCM).
  - Offset Codebook Mode (OCB).

End of Lecture 6

The slides content is related to Sections 2.1, 2.2, 20.1, 20.3, and 20.5 of "Computer Security Principles and Practice (3rd Edition)" by Stallings and Brown