School of Electrical Engineering and Computing

SENG2250/6250
SYSTEM AND NETWORK SECURITY
(S2, 2020)

Cryptographic Techniques

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Outline

- Cryptology
  - *Cryptography*
  - *Cryptanalysis*
- Symmetric cryptosystems
  - *Classical Cipher*
  - *Block cipher and stream cipher*
  - *Modes and operations*
- Cryptographic hash functions
- Asymmetric cryptosystems
- Digital signatures

Cryptology

- The word of “Cryptology”
  - *the art/science of secure communication*
  - *From the Greek words:*
    - *kryptos*: Hidden
    - *logos*: Word
- Cryptography: the study of transforming a plaintext into a ciphertext and then transforming the ciphertext back into the plaintext
- Cryptanalysis: the study of transforming a ciphertext back into the original plaintext without knowledge of the key

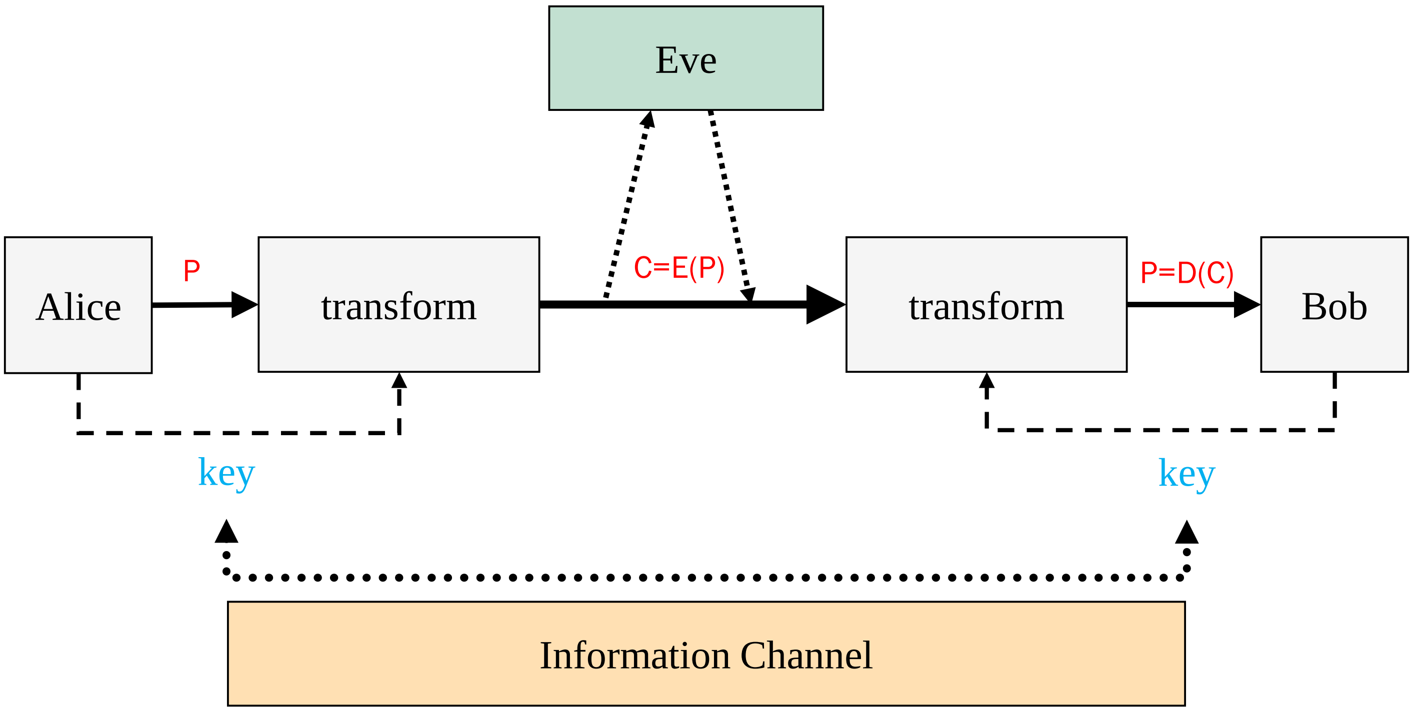
Basic Concepts

- Plaintext (P): the original clear message (M)
- Ciphertext (C): the transformed message
- Cipher: an algorithm for transforming or encrypting or ciphering a clear message into Ciphertext with which any unauthorized party cannot find the plaintext
- Key (K): a data unit used for encipher/deciphering or encryption/decryption.

Basic Concepts

- Encipher/encrypt (E_K): the process of converting plaintext to ciphertext using a cipher (E) and a key (K).
- Decipher/decrypt (D_K): the process of converting ciphertext back into plaintext using a cipher (D) and a key (K).
- Encryption and decryption are sometimes referred to as enciphering and deciphering, respectively.

The Basic Secrecy Channel



Key Dependence

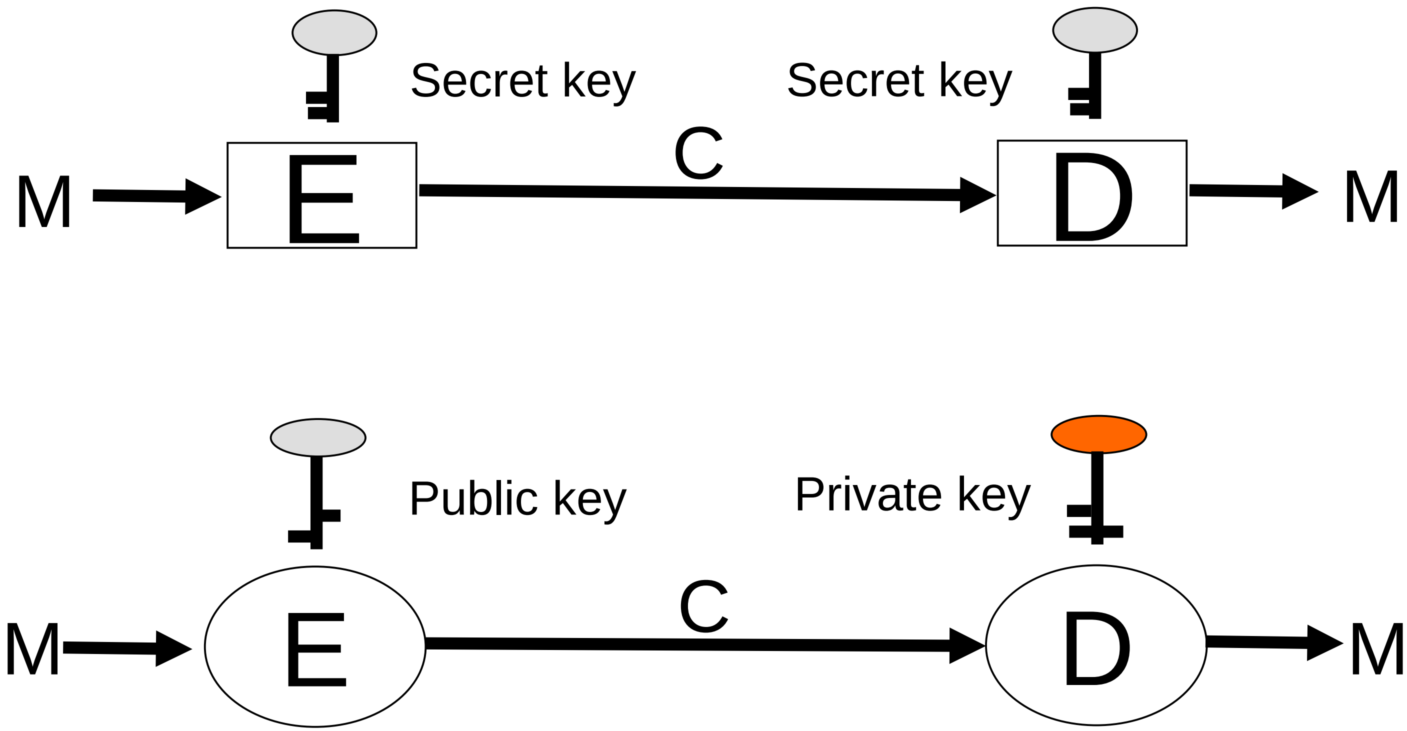
- The transformations are not universal, they are key dependent. The key K controls the transformation and is known only by Alice and Bob. The key is secret.
- If a transformation **does not** depend on a key, it is referred to as **encoding**, with the inverse transformation being referred to as **decoding**.
  - *Morse code.*
  - *ASCII code.*
  - *Base64.*
- Confusingly, if this follows this definition through, once we have chosen a key we can encode a message with a particular (now fixed) transformation.

Models of Encryption and Decryption

- Symmetric key encryption: Encryption key and decryption key are the same

- Asymmetric key encryption: Encryption key and decryption key are different.

Symmetric Encryption vs Asymmetric Encryption



Classical Ciphers

- Principle to a cipher
  - *Substitution (replace): leads to confusion.*
  - *Permutation (reorder): leads to diffusion.*
- Some classical ciphers
  - *Caesar cipher*
  - *Vigenere cipher – (lab discovery)*

Monoalphabetic Substitution Cipher

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
S	Y	G	L	Q	Z	T	C	H	M	U	A	D	I	N	V	R	E	J	O	X	W	F	K	P	B

```
Plaintext:   HI THIS IS ALICE
Ciphertext:  CH OCHJ HJ SAHGQ
```

Security of Monoalphabetic Substitution Ciphers

- To decipher, substitution alphabet must be known.
- To find the substitution table, exhaustive key search (*brute force*) can be used: try each key to decipher the ciphertext and accept the one that produces a meaningful plaintext.
- For an alphabet of size N, the number of possible keys is

$$N! \approx \sqrt{2\pi N}\,(N/e)^N \qquad 26! \approx 9^{26}$$

- In this case the algorithm is using a substitution alphabet and the key is the specific substitution used

Cryptanalysis

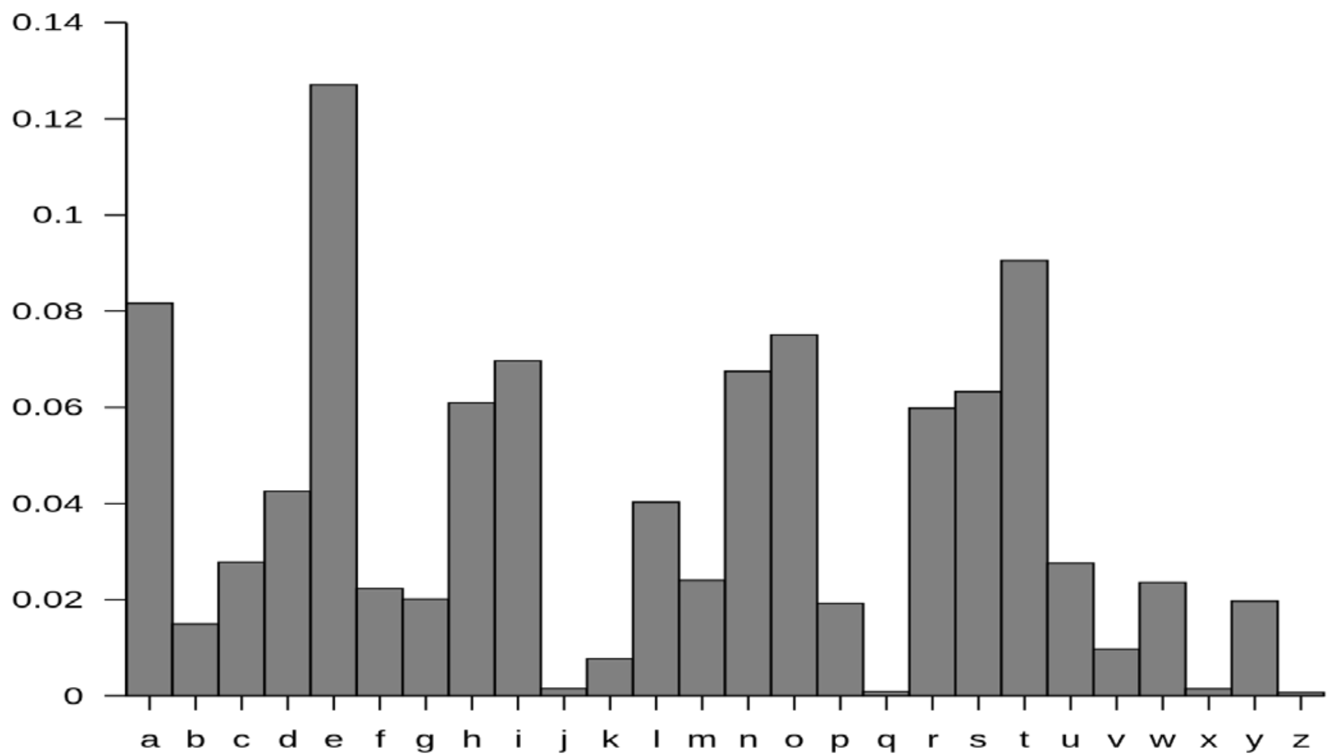
- Decrypt ciphertext to reveal the plaintext without key.
- Ciphertext only attack
  - *Known cipher text only*
- Known plaintext attack
  - *Known substantial amount of (plaintext, ciphertext) pairs.*
- Chosen ciphertext attack
  - *Have access to the cipher under an unknown Key.*
  - *Can choose special plaintext and/or ciphertext.*

Statistical Analysis

- No classical ciphers are secure against cryptanalysis
- Statistical properties of plaintext language can be used to cancel many keys in one step and enable the cryptanalyst to find the key without trying all of them.
- Frequency grouping:

I	e
II	t, a, o, i, n, s, h, r
III	d, l
IV	c, u, m, w, f, g, y, p, b
V	v, k, j, x, q, z

English Letter Frequency Distribution



Statistical Analysis

- Frequency of characters, bigrams (pair of consecutive letters) and trigrams (triple of consecutive letters) are important clues to cryptanalyst.
- Frequent bigrams:
  - *th, he, in, er, an, re, ed, on, es, st, en, at, to*
- Frequent trigrams:
  - *the, ing, and, her, ere, ent, tha, nth, was, eth, for, dth.*
- Note : Frequency counts are only 'clues' to the actual key used. They depend on the sample text considered.

Cryptanalysis - Example

- Ciphertext
  - *PHHW PH DIWHU WKH SDUWB*
- Ciphertext frequency
  - *H – 5, P – 2, W - 4, D - 2, I - 1, U - 2, K - 1, S - 1, B – 1*
  - *H → E, W → T,*
- Plaintext
  - *?EET ?E ??TE? T?E ???T?*
  - *P → A or D → A or U → A*
  - *D → A: ?EET ?E A?TE? T?E ?A?T?*
- Plaintext: MEET ME AFTER THE PARTY

Perfect Secrecy

- Using the knowledge of the plaintext languages, a set of possible plaintexts are determined with certain probability.
- In a system with perfect secrecy, knowledge of the cryptogram does not help the enemy.

$$P(X = x) = P(X = x | Y = y), \forall\, x, y$$

- They are just as likely to guess the plaintext associated with a ciphertext after they see the ciphertext as they are before they see it.

Theorem (Shannon)

In a system with perfect secrecy the number of keys is at least equal to the number of messages.

- This tells us that to achieve perfect secrecy in practice, many key bits must be exchanged. This is not practical.
- How can we measure security then if we know it probably isn't perfect?
- Shannon proposed *unicity distance* as the measure of security.

Unicity Distance

- N_0 is the least number of ciphertext characters needed to determine the key uniquely. If there are *E* keys and they are chosen with uniform probability, unicity distance is given by:

$$N_0 = \frac{\log_2 E}{d}$$

where *d* is the *redundancy* of the plaintext language.

Redundancy and Rates

- Redundancy of a language is defined in terms of the *rates* of the language.

$$d = R - r \text{ bits}$$

- The **absolute rate R** of a language is the minimum number of bits to represent each character, assuming characters are equally likely and emitted independently. For an alphabet of size A,

$$R = \log_2 A.$$

- The **true rate r** of a language is the average number of bits required to represent characters of that language. This uses the real probability distribution of characters.

Redundancy and Rates

- For English;

  R $\approx$ 4.7 bits, r $\approx$ 1-1.5 bits, d $\approx$ 3.2 bits.

- True rate is always smaller than absolute rate, and the difference is the redundancy.
- All natural languages are redundant, for example:

  Bb invitd Alic fr dinr, bt sh rfusd.

- This sentence is readable because we can fill all the missing characters: that is, all the missing characters are redundant.

Redundancy and Rates

- Redundancy is related to structure.
- A truly random source has no redundancy.

mmhfsdacxnvfdvvdfpnfuipawedka

- Every character in this string is necessary: if one of them is omitted the information it carries is lost and cannot be recovered.
- Redundancy occurs because of the non-uniform letter frequencies, bigram (trigram …) frequencies and other (e.g. grammatical) structures of the language.

Measuring security

- We can use unicity distance as a measure to compare the security of various ciphers.

- Recall that we presented security as being about protecting *assets* against possible *threats*.

Block Cipher

- In a block cipher algorithm, plaintext bits are grouped into blocks and then processed.
- Fixed length input → fixed length output
- Block size and key size are two important parameters to the security of such algorithms.
- Some attacks
  - *Dictionary attack*
  - *Meet-in-the-middle attack*

Data Encryption Standard (DES)

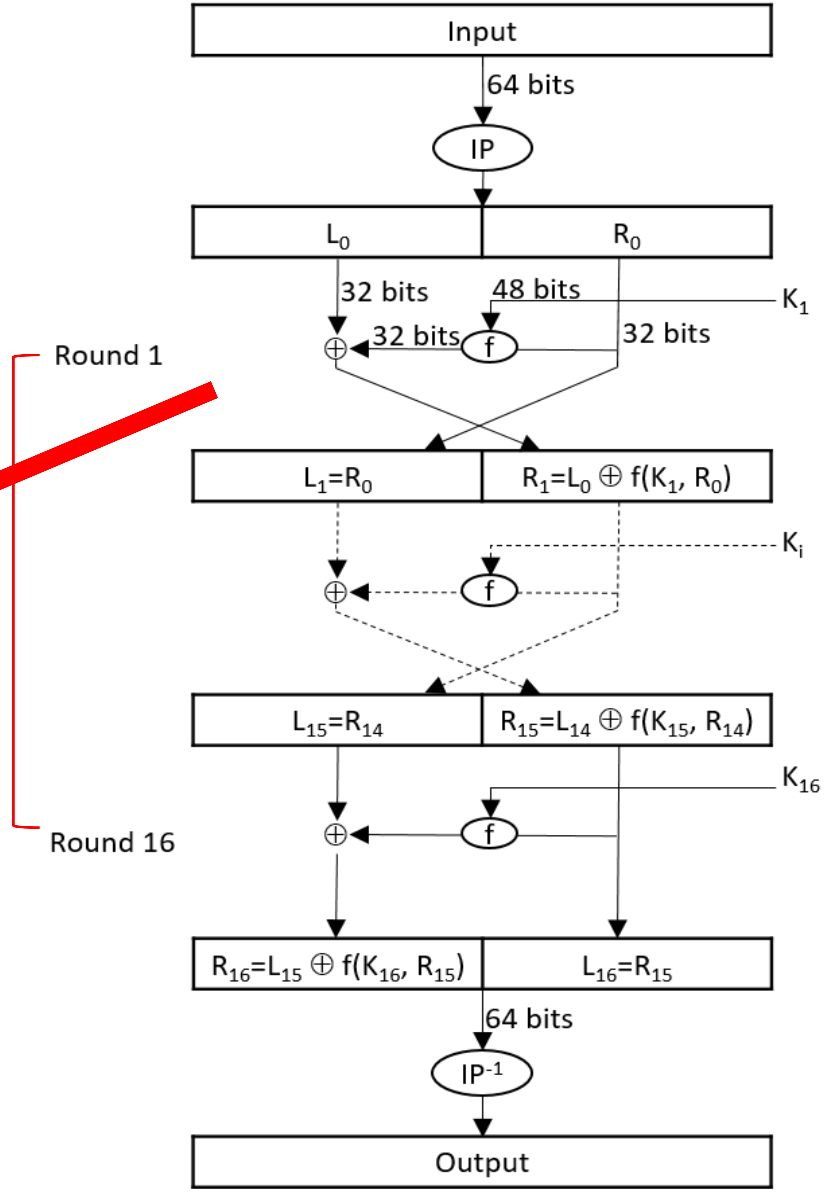
- Data Encryption Standard (DES) developed by IBM and adopted by NIST with NSA approval for US government unclassified information
- Block cipher
- Key size: 56 bits
- Block size: 64 bits
- Key space: 2^{56} (can be reduced by attacks ☹)
- Substitution (S-box) and Permutation (P-box)
- Feistel structure

DES Structure

Step 1. Do XOR of R0 and K1.

Step 2. Take the output of XOR into S-boxes.

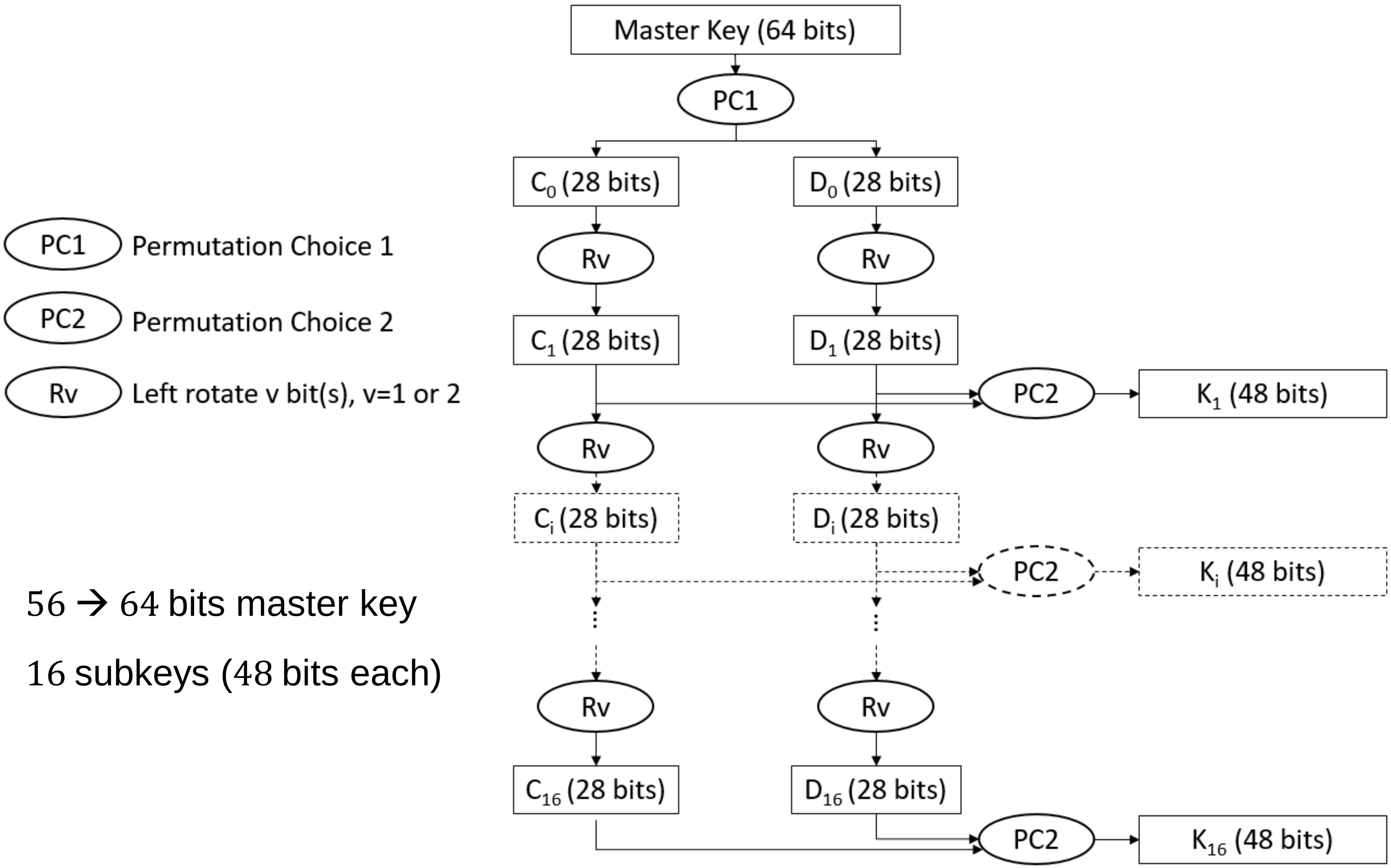
Step 3. Do XOR of L0 and f's output



DES Structure

- The function *f* is a non-linear transformation, and is the source of the cryptographic strength of DES.
- IP is the initial permutation, and has no cryptographic significance.
- IP is used to facilitate getting bits onto the chip in VLSI DES.

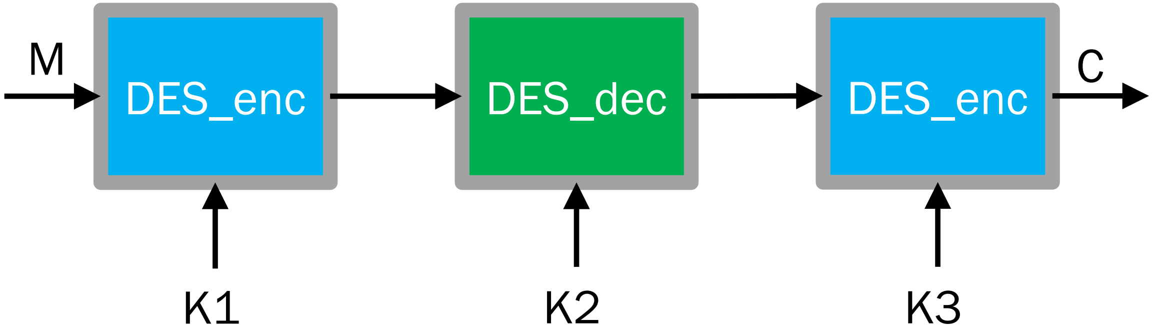
Key Schedule



- 56 → 64 bits master key
- 16 subkeys (48 bits each)

Triple DES

- DES is no longer secure that we need to find a solution.
- Three keys, K1, K2 and K3.
- If K1=K2=K3, it is a single DES!
- 112-bit security level (k1, k2 and k3 are independent).

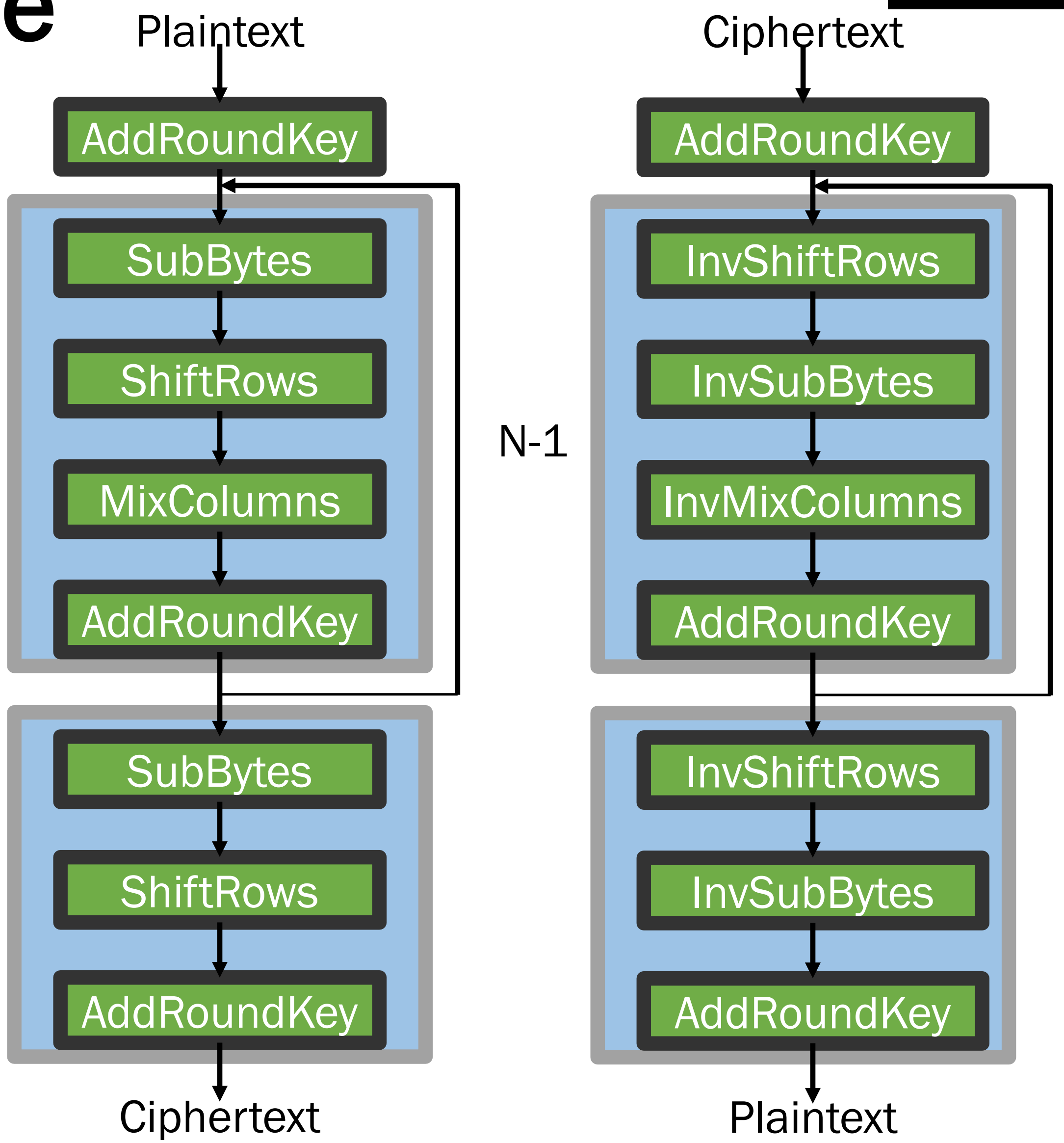


Advance Encryption Standard (AES)

- NIST Requirements
  - *Block size 128 bits*
  - *Cipher should offer variable key lengths of 128, 192 and 256 bits*
  - *Cipher should be more efficient than Triple DES and operate faster than Triple DES across a range of platforms*
  - *Selection process public and the selected algorithm should be available royalty-free worldwide*
- One of the submissions Rijandael selected to be AES.
  - *Block Cipher with a variable block size and key size*
  - *Key size : 128, 192 or 256 bits*
  - *For detailed information, see www.nist.gov*

AES Structure

Key Size (bits)	Rounds (N)
128	10
192	12
256	14



AES Operations

- Substitution
  - *S-box, 8-bit to 8-bit*
  - *Non-linear*
- Shift row
  - *Rotate order of bytes in each row*
- Mix column
  - *Linear mixing of a word column*
- Add round key
  - *Addition*
  - *Provide secret randomness*

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

Attacks Against AES

- The full AES is not broken.
- Reduced round versions are broken (theoretically not practically): There are attacks as follows.

  7 rounds for 128-bit keys. (Chosen-plaintext)

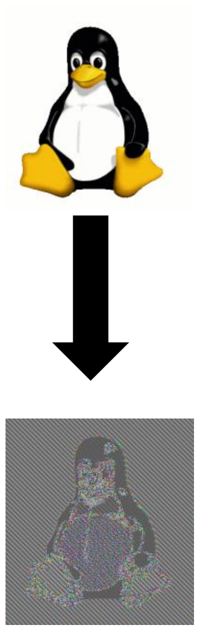
  8 rounds for 192-bit keys. (Chosen-plaintext)

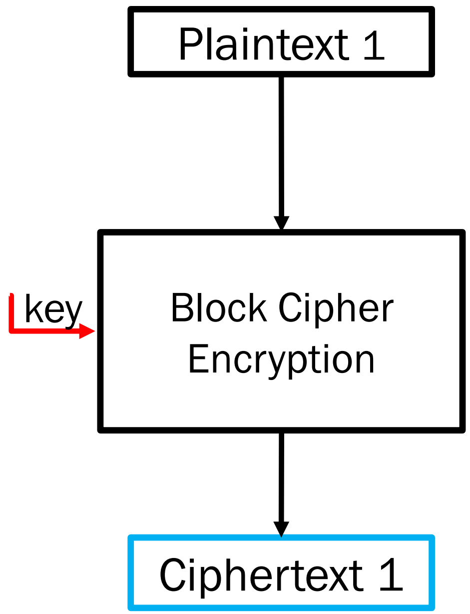
  9 rounds for 256-bit keys. (Related key)

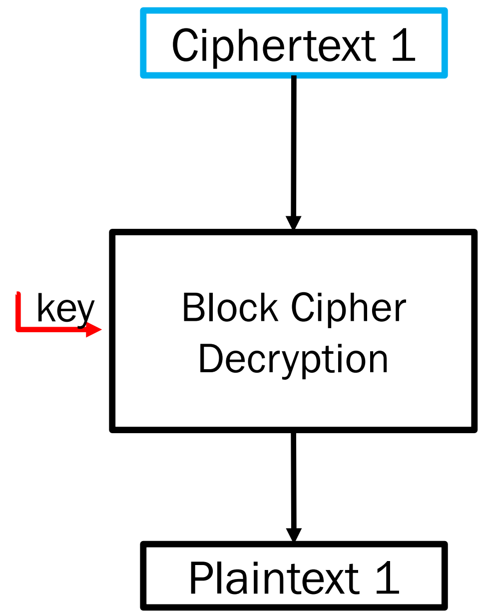
- The best theoretic attack breaks up to 8 rounds with over 2^{120} complexity for 128-bit keys and 2^{204} for 256-bit keys.
- Side-channel attacks have been shown to be successful:
  - *Effectively these are against particular implementations, not the algorithm itself.*
  - *They may not be practical anyway.*

Operation Modes

- Electronic Codebook (ECB) encryption and decryption

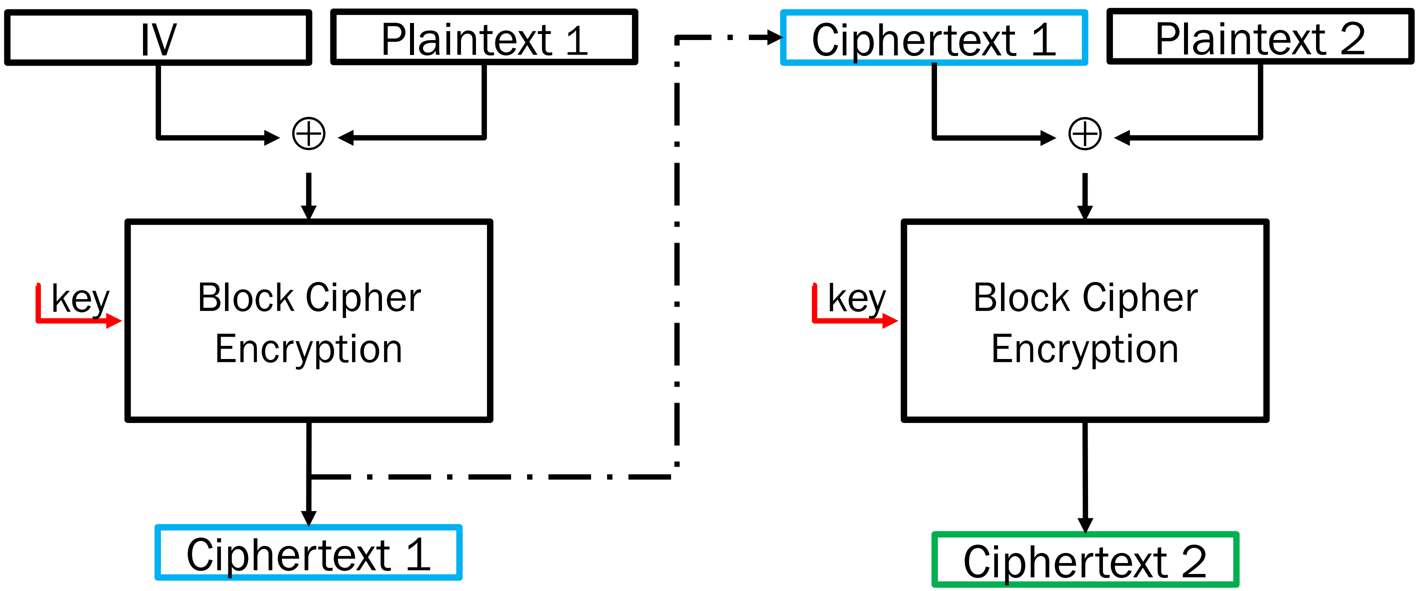






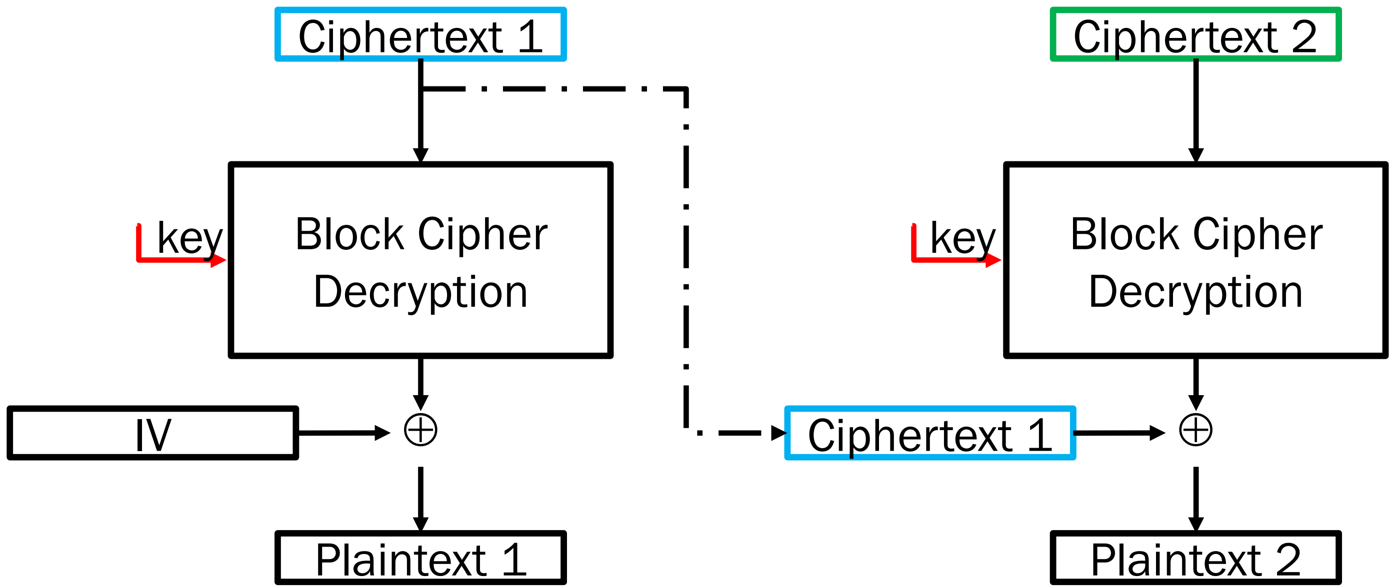
Cannot hide the pattern of information, identical plaintext results in the same ciphertext.

Cipher Block Chaining (CBC) - Encryption

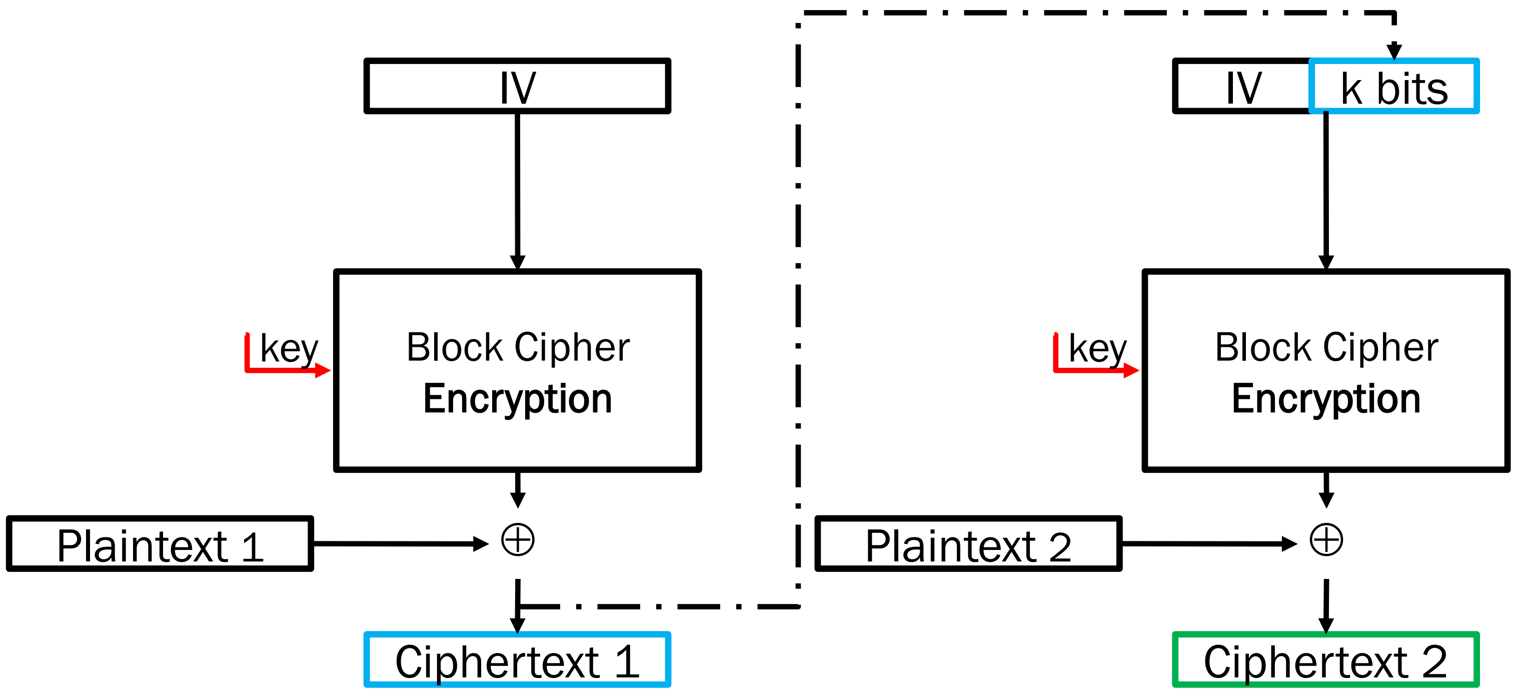


Initialisation Vector (IV): is a (random) value which has the same length as the plaintext block.

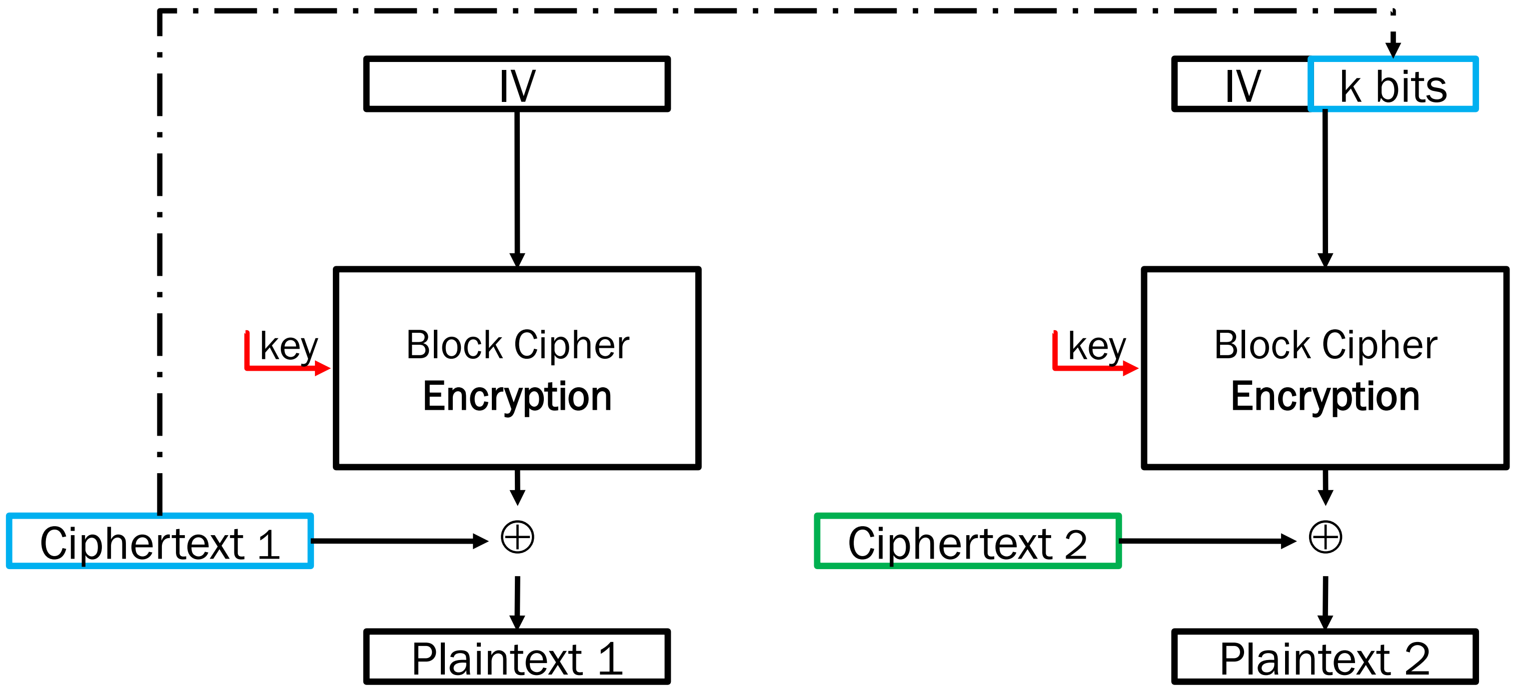
Cipher Block Chaining (CBC) - Decryption



Cipher Feedback (CFB) - Encryption

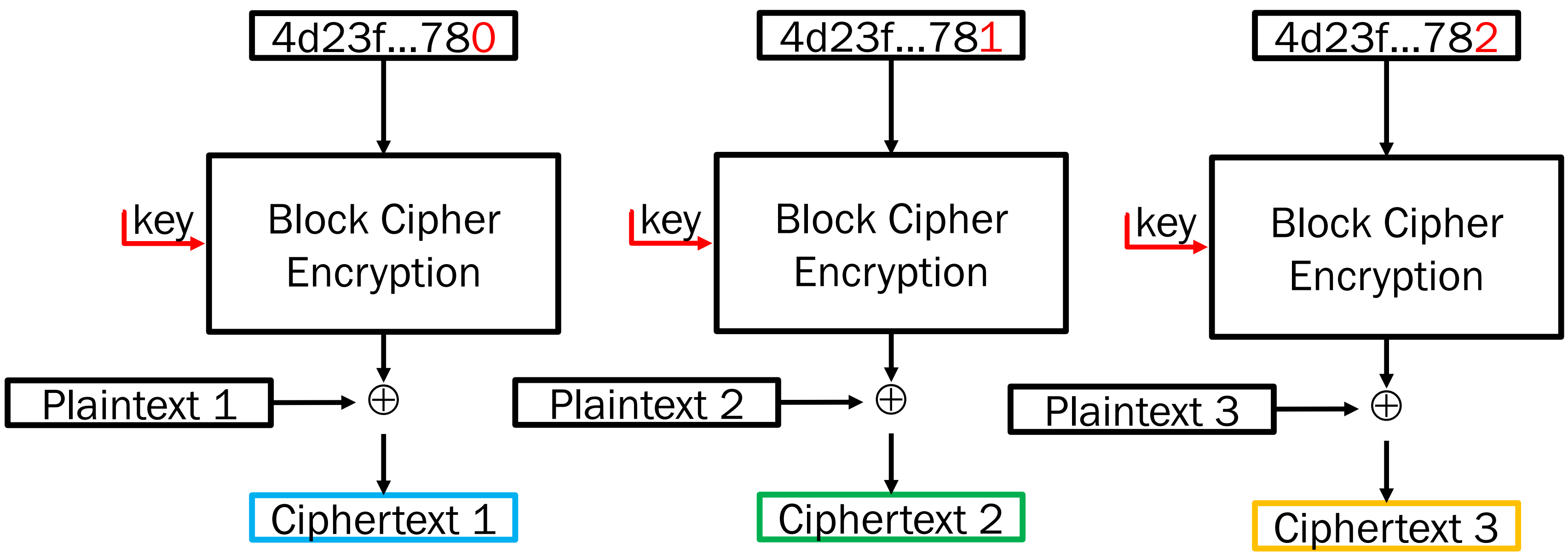


Cipher Feedback (CFB) - Decryption



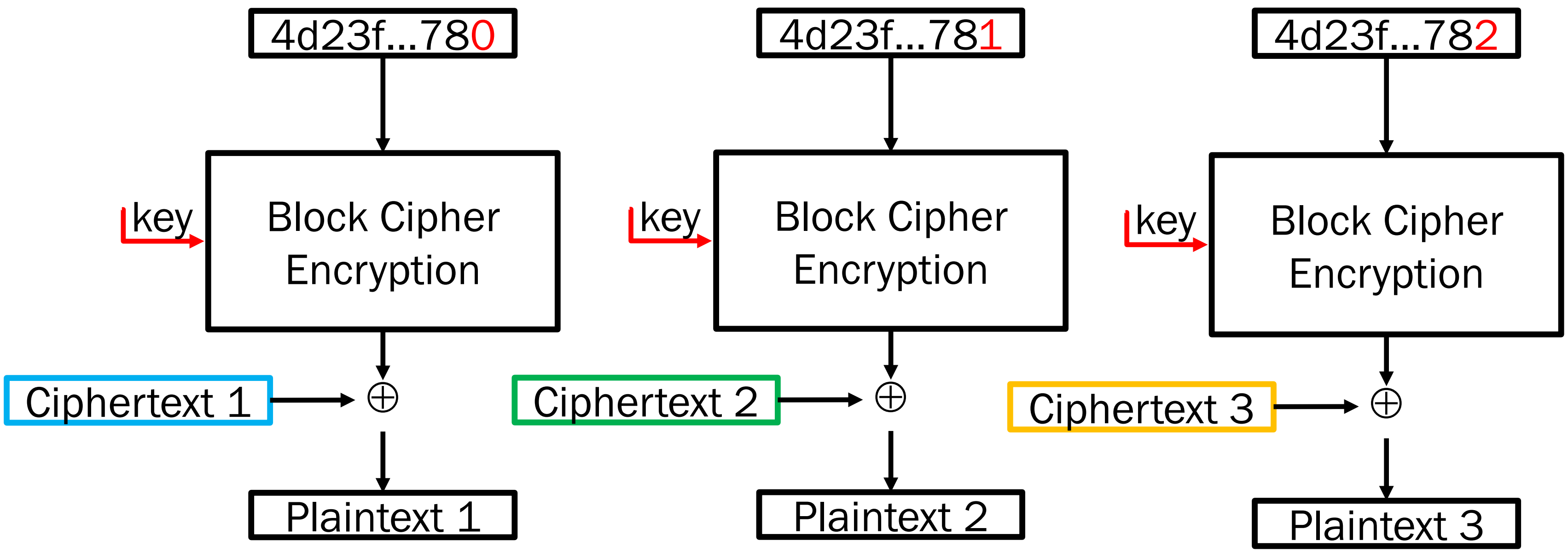
Counter Mode - Encryption

IV = Nonce
E.g., IV = Nonce = 4d23f…780



Counter Mode - Decryption

IV = Nonce
E.g., IV = Nonce = 4d23f...780



Stream Cipher

- In block ciphers plaintext characters are grouped in blocks and then each block is encrypted.

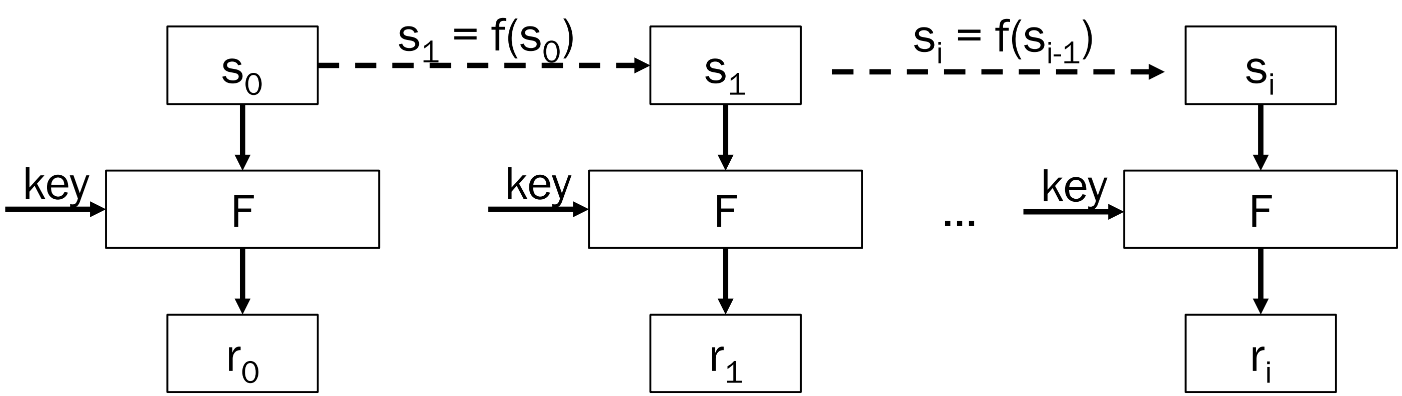
- In stream ciphers, characters are encrypted one at a time.
  - *For example the Vigenère cipher.*
  - *Note that characters could themselves be a block of bits (in the sense the algorithm operates byte by byte say).*

Stream Cipher - Encryption

	K	k_0	k_1	k_2	k_3	k_4	…
+	P	p_0	p_1	p_2	p_3	p_4	…
=	C	c_0	c_1	c_2	c_3	c_4	…

$$\begin{array}{rl} 1\,0\,1 & \leftarrow P \\ \oplus\ \ 1\,1\,0 & \leftarrow K \\ \hline 0\,1\,1 & \leftarrow C \end{array}$$

- Keystream Generation – PRNG

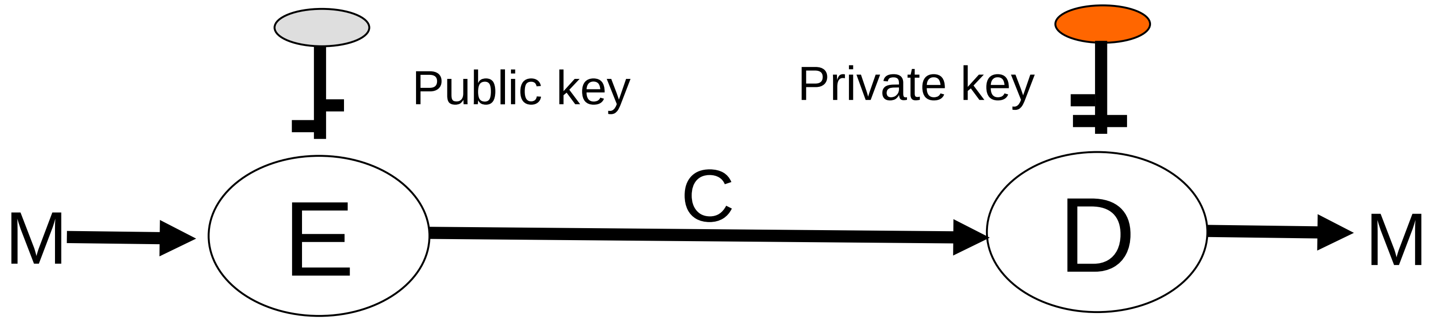


Key Stream

- Key stream is pseudorandom.
- Generated by non-linear procedure
- The same master key (initial vector) result in the same key stream if the same key stream generation algorithm applied.

Public Key Cryptography (PKC)

- Symmetric key cryptosystem
  - *all parties share/use the same secret key*
- Public (Asymmetric) key cryptosystem
  - *Public key: a publicly known key to everyone.*
  - *Private key: is known to one owner only.*
  - *Encryption and digital signatures*



One-way Trapdoor Functions

- A function $f: X \rightarrow Y$ is called one-way if
  - *For all $x \in X$, find $f(x)$ is easy.*
  - *Given $y \in Y$, find x, s.t. $f(x) = y$, is hard.*
- An example based on the factorisation problem
  - *Given n primes $p_1, p_2, p_3, \ldots, p_n$, it is easy to find their product*
    $$N = p_1 p_2 p_3 \ldots p_n.$$
  - *Given a large number N it is computationally hard to find its prime factors.*

One-way Trapdoor Functions

- A trapdoor is a piece of knowledge (say, private key) which makes it easier to find X from Y.

- A trapdoor one-way function is a function which looks like a one-way function but is equipped with a secret trapdoor. If this secret door is known, the inverse can be easily calculated.

RSA Public Key Cryptosystem

- Rivest, Shamir and Adleman (1978)
- It is the de facto standard for PKC.
- It supports secrecy (encryption) and authentication and can be used to produce digital signatures.
- Security is related to the factorisation problem
  - *RSA uses the knowledge that it is easy to find primes and multiply them together to construct composite numbers, but it is difficult to factor a composite number.*

Preliminaries for RSA

- Let $p \in \mathbb{N}$, then we denote by $\phi(p)$ the number of integers a with $1 \leq a \leq n$ which are relatively prime to p.
  - *If p is prime, then $\phi(p) = p - 1$*
  - *$\phi(p)$ is called the order of p.*
- Given $1 \leq a \leq n - 1$, we can find (using extended Euclidean algorithm) $b \in [1, n-1]$, s.t $ab = 1 \bmod n$, if $\gcd(a, n) = 1$.
- Given $n = pq$, where p, q are large primes, factorising n is computationally difficult.

RSA

- Key Generation
  - *Choose two large primes* p *and* q*, compute*
    $$n = pq, m = \phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$$
  - *Choose* $e \in [1, m-1]$*, such that* $\gcd(e, m) = 1.$
  - *Find* d*, such that* $ed = 1\ mod\ m$
  - *Public key:* (n, e)
  - *Private key:* (p, q, d)
- Encryption
  - X *– plaintext;* Y *– ciphertext*
    $$Y = X^e\ mod\ n$$
- Decryption Encryption and decryption use the same function!
  $$X = Y^d\ mod\ n$$

Example

- Choose two primes $p = 3, q = 11$
- Compute
  - $n = 3 * 11 = 33$
  - $\phi(n) = (p - 1)(q - 1) = 2 * 10 = 20$
- Choose e = 7, where $\gcd(7, 20) = 1$
- Find $d = 3$, where $ed = 1 \bmod 20$
- Public key $(e, n) = (7, 33)$
- Private key $(p, q, d) = (3, 11, 3)$
- Plaintext: M$=$ 6, must satisfy $0 \leq x \leq n - 1$
- Ciphertext: $C = M^e = 6^7 = 30\, mod\ 33$
- Decrypt: M$=$ $C^d = 30^3 = 6\, mod\ 33$

RSA: One-Way Trapdoor Function

- Given two large primes p, q, it is easy to calculate

$$n = pq, \;\; \phi(n) = (p-1)(q-1)$$

BUT

- Given n, it is hard to find
  - *p, q, such that $n = pq$ (hard problem)*
  - $\phi(\mathrm{n})$
- Given e, it is hard to find d, s.t. $ed = 1 \; mod \; \phi(n)$

UNLESS

- If the trapdoor $\phi(n)$ is known, it is easy to find d. How?

Is that possible to use the same e for all users?

Hash Functions

- A hash function h takes as input arbitrary size of message m, and outputs a fixed size block, named a message digest or hash value v.

$$h: \{0,1\}^* \rightarrow \{0,1\}^\ell$$

- Hash value is expected to be uniformly distributed.

- One bit different in m, all bits in v are likely to be different.

Hash Functions

- The space of v is much smaller than the message space.

- Collision
  - *It is possible that two distinct messages output the identical hash value.*
    $$h(m_1) = h(m_2), \qquad m_1 \neq m_2$$

  - *Example*
    $$h(m) = m \bmod 19, \qquad h(8) = h(27)$$

Cryptographic Hash Functions

- A cryptographic hash function is required to be
  - *A hash function*
  - *One-way (pre-image resistant): it is easy to calculate but hard to invert.*

$$h(m) \to v, v \nrightarrow m$$

  - *Second pre-image resistant: given a message m_1, it is hard to find another message m_2 such that*

$$h(m_1) = h(m_2), \qquad m_1 \neq m_2$$

  - *Collision resistant: it is hard to find messages $m_1 \neq m_2$, but $h(m_1) = h(m_2)$.*

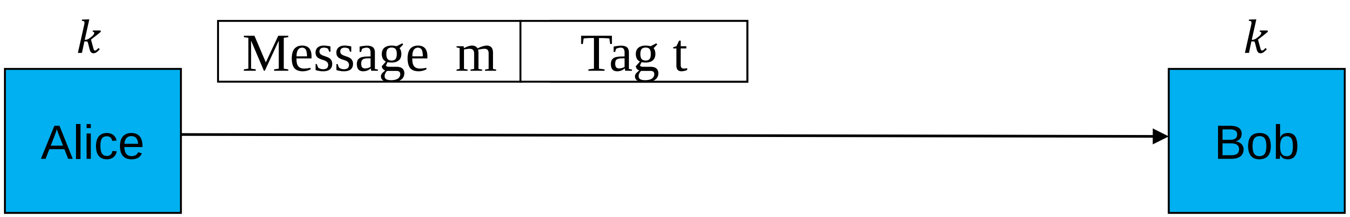
Examples

- MD5 (Broken)
  - *Rivest (1991)*
  - *128-bit digest*
  - *512-bit message block*
- Secure Hash Algorithms
  - *SHA-0 (broken)*
  - *SHA-1 (broken)*
  - *SHA-224, SHA-256, SHA-512 …*

Message Authentication Code (MAC)

- Shared key primitive for providing message integrity check.
- A MAC system is a pair of algorithms (Mac, Ver)
  - *K – secret key space*
  - *M – message space*
  - *T – tag space*

$$Mac(k, m) \rightarrow t$$
$$Ver(k, m', t) \rightarrow True/False$$



CBC-MAC

- Use a key and n-bit block cipher to generate an m-bit MAC.
  - *Input: message x, block cipher E, key k*
  - *Output: m-bit MAC on x (m is the block length of E, e.g, 128-bit of AES)*
- Procedure
  - *Padding – pad x if necessary for blocking.*
  - *Blocking – break padded text into n-bit block for E.*
  - *CBC processing.*
  - *Optional process to increase strength of MAC.*
  - *The MAC is the m-bit block t, which is a tag.*

HMAC

- A method of constructing a MAC from a cryptographic hash function.

$$HMAC(k,m) = h((k \oplus \mathrm{opad})||h((k \oplus ipad)||m))$$

- h - is an iterated hash function
- k - is the key.
- $||$ - concatenation
- $\oplus$ - XOR
- opad $=$ 0x5c5c5c…5c5c (one-block-long hexadecimal constants)
- ipad = 0x363636…3636 (one-block-long hexadecimal constants)

Digital Signatures

- A digital signature is the electronic analogy of handwritten signature.
  - *It ensures integrity of the message and authenticity of the sender.*

Properties

  - *Easy to generate – an authorised singer can generate a signed document.*
  - *Easy to verify – the signed document is publicly verifiable.*
  - *Unforgeability – it is hard to generate a signature for a particular message.*
  - *Non-repudiation – signer cannot deny the signed message.*

- A digital signature scheme consists of three algorithms
  - *Key Generation*
  - *Sign*
  - *Verification*

Digital Signature Schemes

- Key Generation
  - *Generate a pair* (pk, sk) *of public and private keys.*
  - *Public key cryptography*

- Sign: signature generation algorithm
  - *Given a message* m*, private (signing) key* sk*, a signature* $s = Sig(m)$ *is generated.*

Digital Signature Schemes

- Signature verification algorithm
  - *Given a message m, a signature s and a public key pk, the verification algorithm returns Ture or False.*
  - *True – the message m is authentic*
  - *False – the message m is not authentic.*

- Cryptographic hash functions
  - *Compute message digest.*
  - *Sign on the digest rather than the message.*

RSA Signature

- System parameters
  - *Public key:* (n, e)*, where* $n = pq$
  - *Private key:* (p, q, d)
  - *Hash function:* h
  - *Message:* m
- Signature generation

$$s = h(m)^d \ mod \ n$$

- Signature verification

$$\text{Check if, } h(m) = s^e \ mod \ n$$

  - *It returns True if and only if the above equation holds.*