ACS Education 8th

Cryptography

KISA AKCF asean

ACS

Index

- Cryptography overview
- Symmetric key Encryption
- Public-key Encryption
- Integrity
- Lab

01

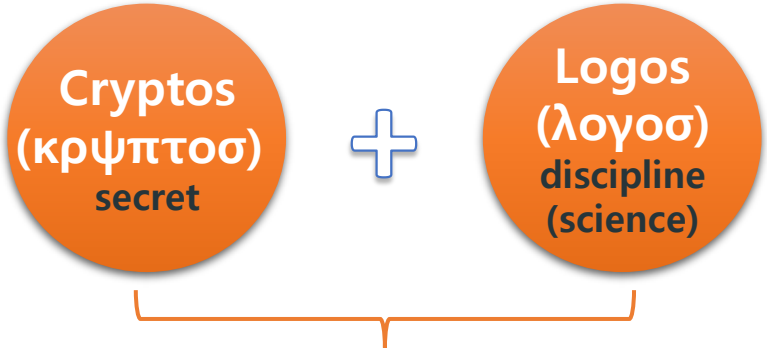
Cryptography overview

- Overview
- Fundamental theories of Cryptography
- Mathematical Basics for Understanding modern Cryptography
- Security of Cryptographic algorithms

Overview

Academically speaking, cryptography includes cryptography, which creates various encryption and decryption methods, and cryptanalysis, which interprets and analyzes them.

- Cryptology

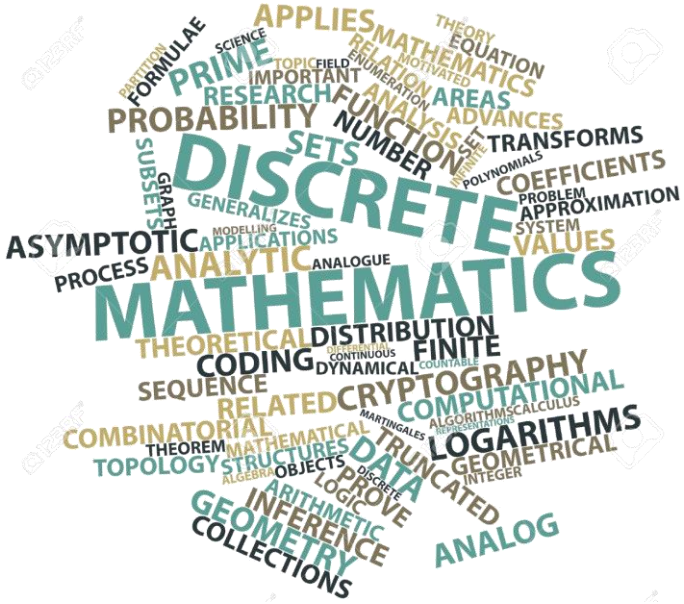


Cryptography	Cryptanalysis
Research cryptographic algorithms to protect plaintext.	Study the encryption process and ciphertext in order to decrypt plaintext.

Overview

Discrete means scattered apart. In the same vein, discrete mathematics means that the values are scattered rather than continuous.

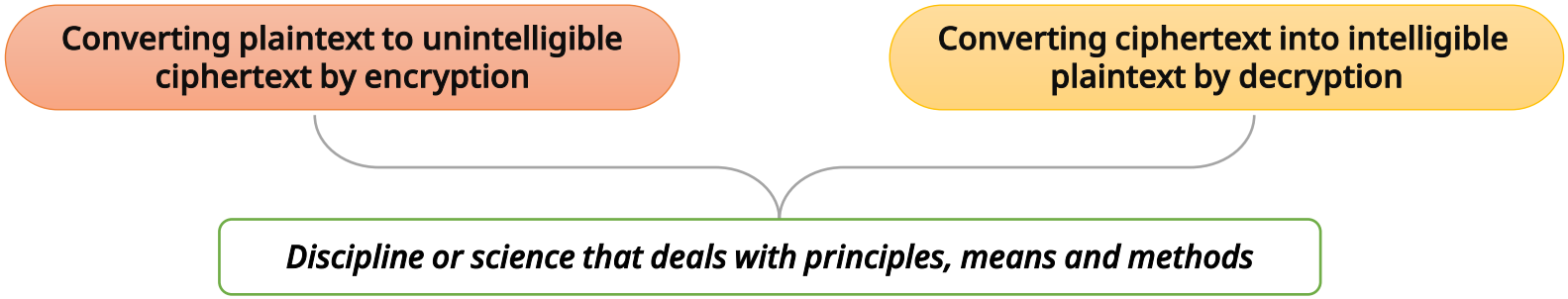
- Discrete mathematics
  - Means performing operations on numbers that are discrete rather than continuous.
  - Excludes subjects such as calculus and Euclidean geometry, which deal with continuity in mathematics.
  - Mostly works with integers.
  - The development of digital computers has stimulated the study of discrete mathematics.
    - Computer algorithms, programming languages, cryptography, software fields, etc.



Fundamental theories of cryptography

Academically speaking, cryptography includes cryptography, which creates various encryption and decryption methods, and cryptanalysis, which interprets and analyzes them.

- Cryptography
  - References : American National Standards Institute (ANSI) X9.31-1998*
    US National Institute of Standards and Technology (NIST) SP 800-2-1991**
    - "The discipline which embodies principles, means and methods"* "for the transformation of ordinary text (plaintext) into coded form (ciphertext) by encryption, and transformation of ciphertext into plaintext by decryption."**

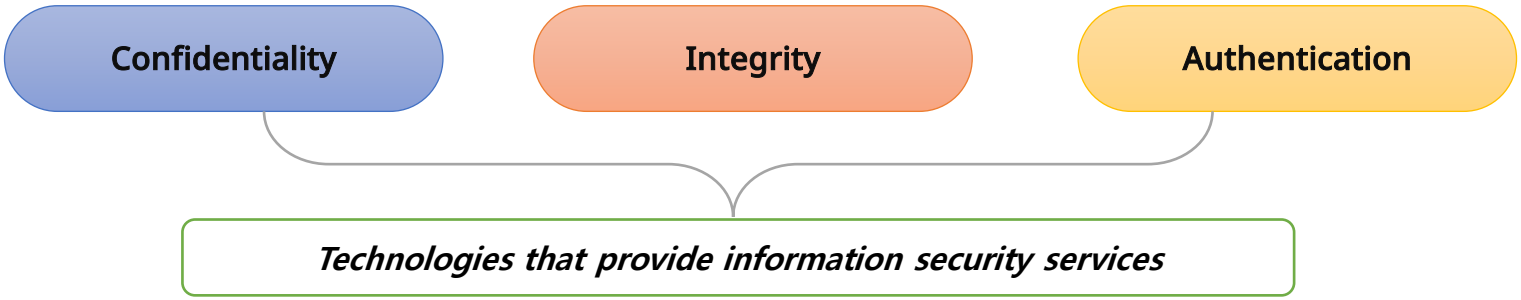


Source : https://csrc.nist.rip/projects/cryptographic-module-validation-program

Fundamental theories of cryptography

Academically speaking, cryptography includes cryptography, which creates various encryption and decryption methods, and cryptanalysis, which interprets and analyzes them.

- Information security and cryptography
- Modern cryptography extends to the study of various problems related to secret communication and their solutions, encompassing the field of information security.

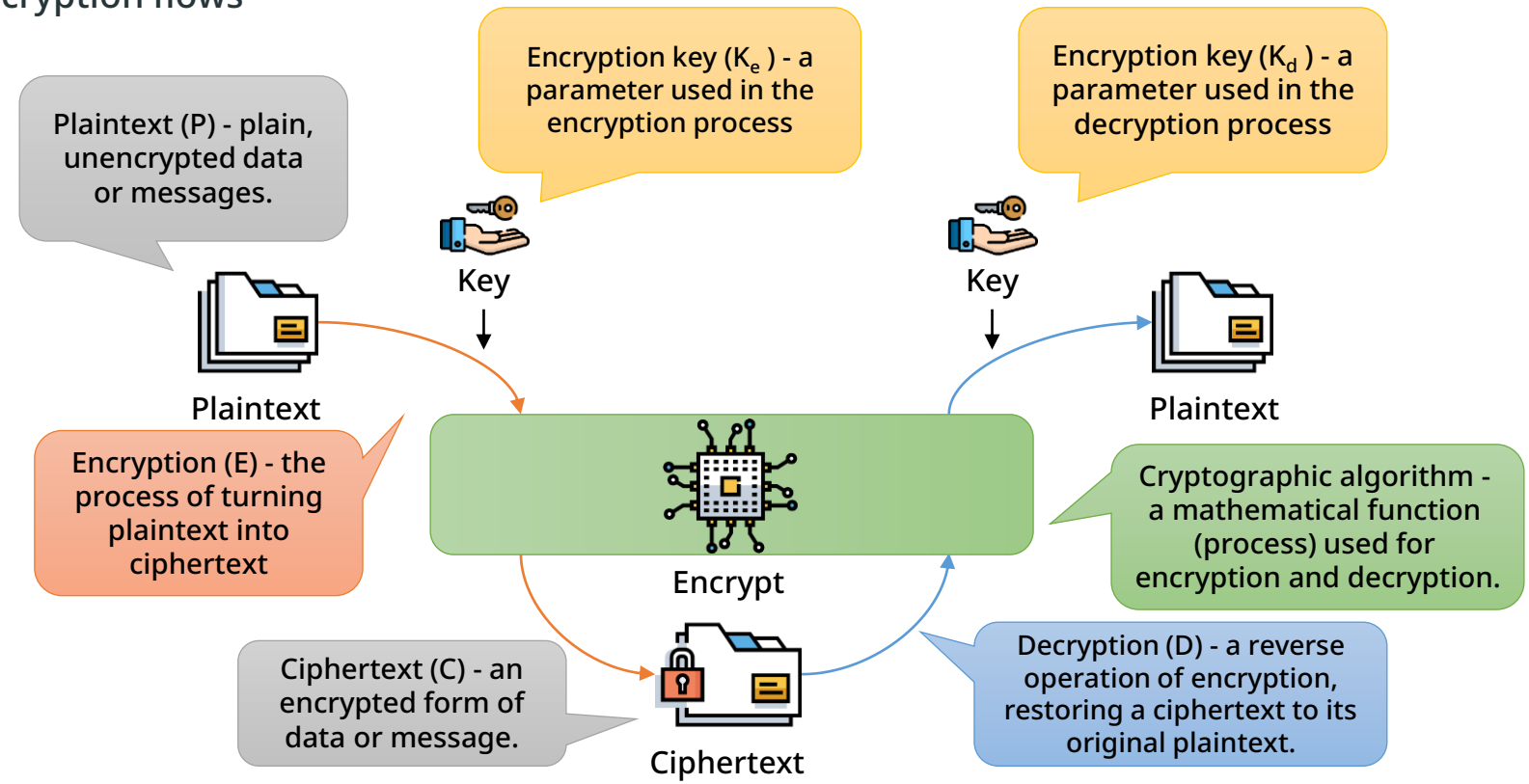


Fundamental theories of cryptography

Understanding cryptographic terminology

Traditionally, cryptography consists of transforming a human-readable plaintext into an unrecognizable ciphertext using a specific method, and then transforming it back into a human-readable plaintext.

- Encryption flows



Fundamental theories of cryptography

Understanding cryptographic terminology

Traditionally, cryptography consists of transforming a human-readable plaintext into an unrecognizable ciphertext using a specific method, and then transforming it back into a human-readable plaintext.

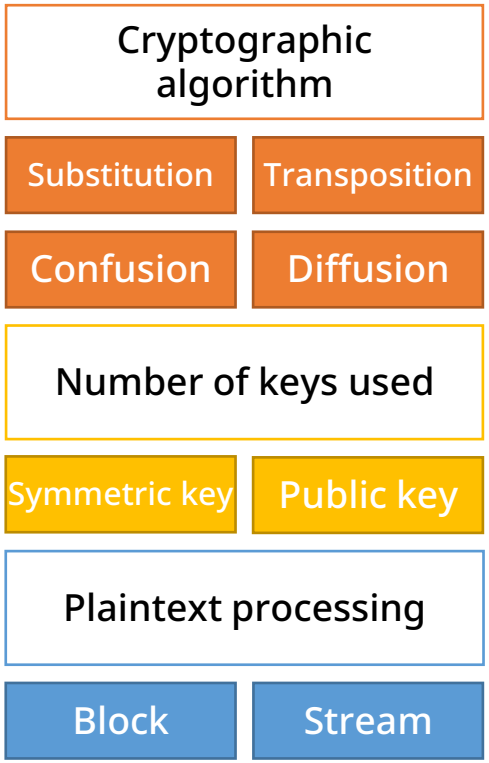
- Other cryptography terms
  - Cryptanalysis - the process of obtaining the original plaintext or key without having the key.
  - Cryptosystem - a set of processes for securing information, including the encryption and decryption processes, the encryption and decryption keys used, and key management.
  - Attacker - a third party who attempts to decrypt the ciphertext into plaintext.
  - Entity - a person who can send, receive, and modify information.

Fundamental theories of cryptography

Categorizing cryptographic techniques

One way to categorize cryptographic techniques is to group the four cryptographic principles into three eras.

- Cryptographic principles
  - Classic cryptography
    - Transposition cipher
    - Substitution cipher - simple substitution cipher
  - Modern cryptography
    - Substitution cipher - polyalphabetic substitution cipher
  - Contemporary cryptography
    - Confusion and diffusion



Fundamental theories of cryptography

Basic cryptography concepts

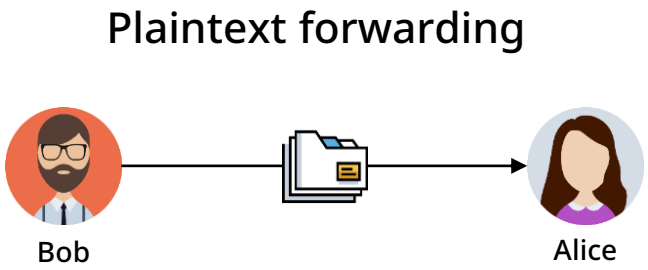
Fundamental concepts in such difficult cryptography can be understood in a storytelling way, using fictional characters. They are known as Alice and Bob and were first used in 1978 in the paper "A method for obtaining digital signatures and public-key cryptosystems."

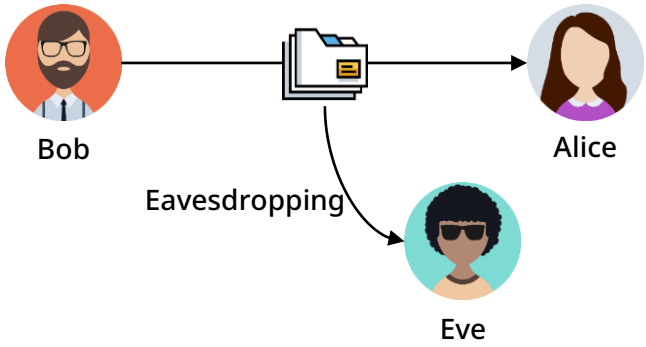
- Characters used in discussions of cryptography (how it's represented)
  - Alice and Bob
    - Alice : the person sending the message
    - Bob : the person receiving the message.
  - Eve and Mallory
    - Eve : eavesdropper
    - Mallory : malicious attacker
  - Trent and Victor
    - Trent : trusted arbitrator
    - Victor : verifier

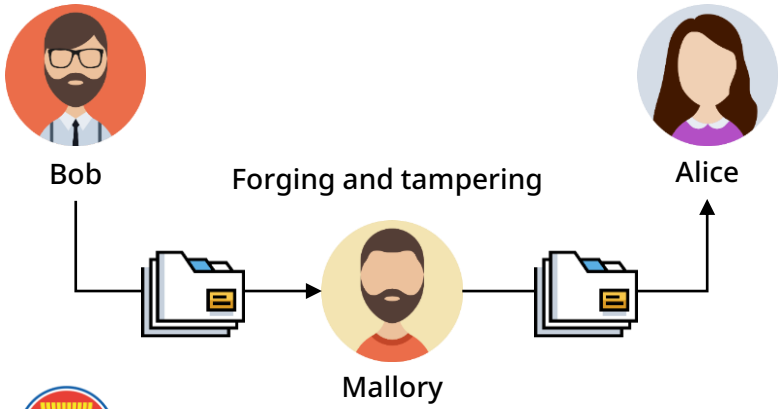
Fundamental theories of cryptography

Basic cryptography concepts

Fundamental concepts in such difficult cryptography can be understood in a storytelling way, using fictional characters. They are known as Alice and Bob and were first used in 1978 in the paper "A method for obtaining digital signatures and public-key cryptosystems."



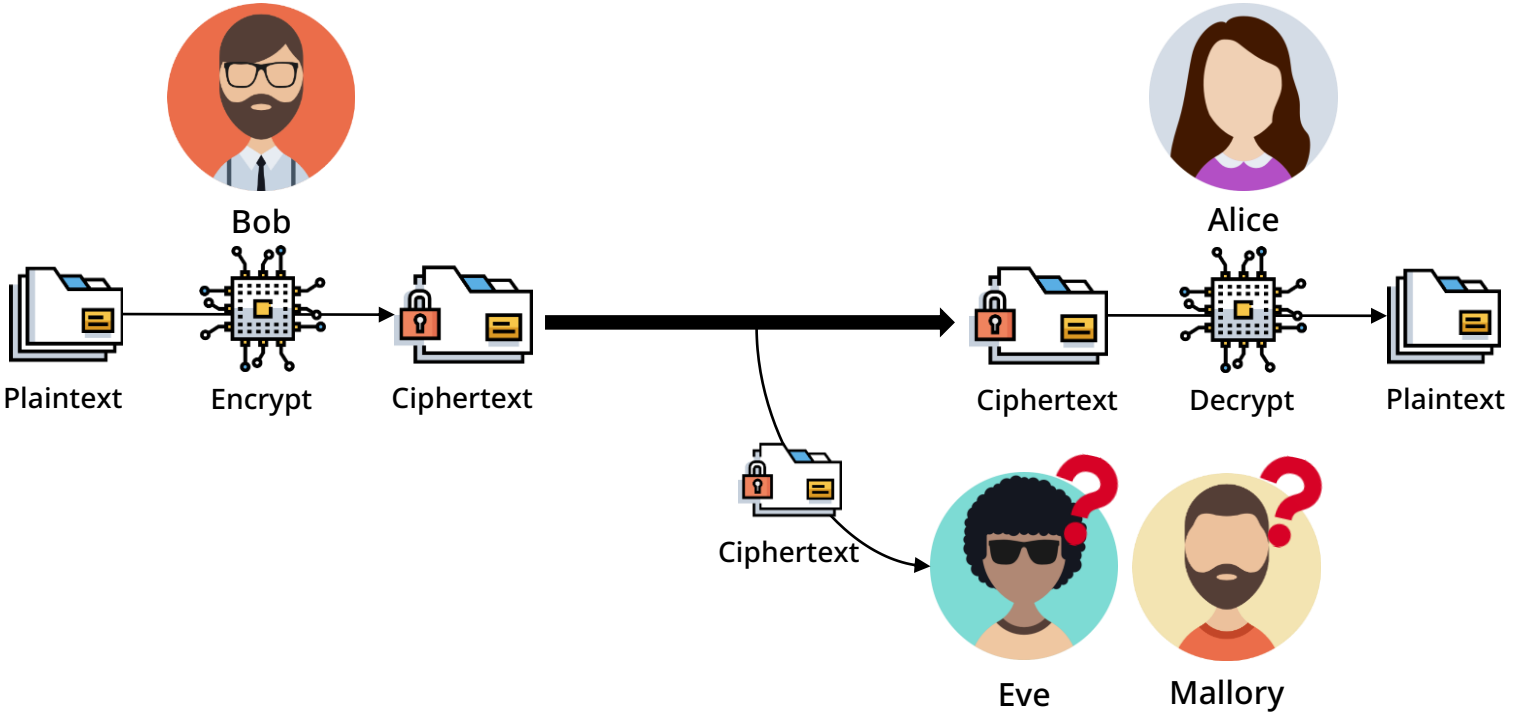




Fundamental theories of cryptography

Basic cryptography concepts

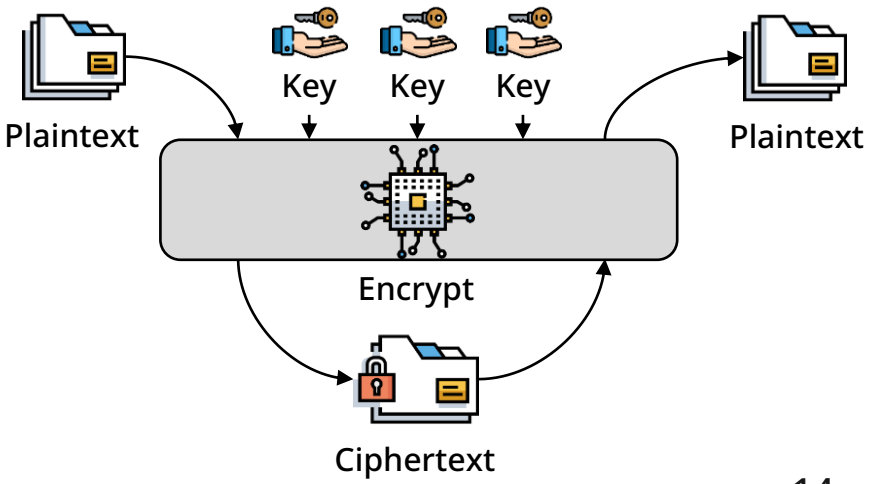
Fundamental concepts in such difficult cryptography can be understood in a storytelling way, using fictional characters. They are known as Alice and Bob and were first used in 1978 in the paper "A method for obtaining digital signatures and public-key cryptosystems."



Fundamental theories of cryptography

When implementing cryptographic algorithms in information security, cryptographic algorithms and key management are the most important aspects.

- Separate cryptographic algorithms and keys
  - Cryptographic algorithms
    - Using a new algorithm each time results in poor performance.
    - Designed to use one algorithm repeatedly
  - Keys
    - Designed to use different keys for different users
    - Build a more secure encryption system

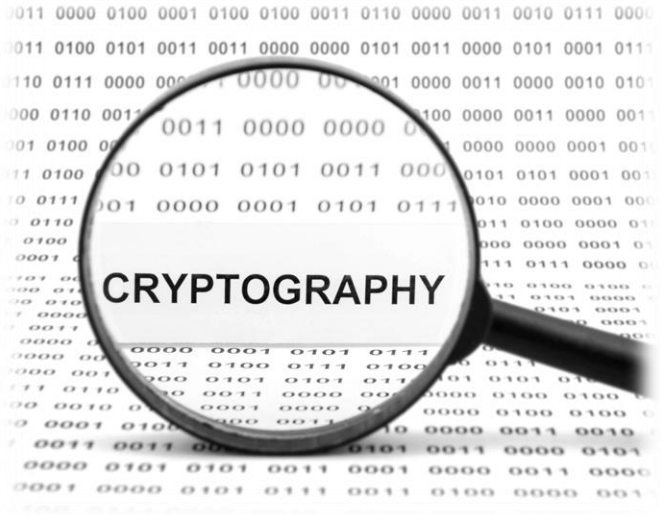


Fundamental theories of cryptography

Cryptography and security common sense

When implementing cryptographic algorithms in information security, cryptographic algorithms and key management are the most important aspects.

- Prohibit the use of secret cryptographic algorithms
  - Implementing the company's own cryptographic algorithms is a risky practice.
    - Keeping a cryptographic algorithm secret can only be secure to the extent that it is not exposed to create a larger problem.
    - This is called "security by obscurity."
- Encrypting weak cryptography is risky.
  - This is the feel of security that the word "cryptography" gives off, which is problematic.
- All encrypted cryptography can eventually be broken.
  - Should change your cryptography every 6 months under the Korean Personal Information Protection Act.
  - The Guide on Critical Information Infrastructure Protection in South Korea recommends changing it every 3 months.



Fundamental theories of cryptography

Classification of cryptographic algorithms

The main categories of cryptographic algorithms in use in the field of information security are the following.

- Classification by number of keys used for encryption and decryption
  - Secret Key Cryptography (SKC)
    - Use a single key for both encryption and decryption
    - Also called symmetric encryption
    - Used mainly for privacy and confidentiality

    DES AES
  - Public Key Cryptography (PKC)
    - Separate encryption and decryption keys
    - Also called asymmetric encryption
    - Used mainly for authentication, non-repudiation, and key exchange

    RSA PKCS
  - Hash Function
    - Use mathematical transformations for irreversible encryption
    - Used mainly for integrity verification (digital fingerprinting)

    MD5 SHA1

Fundamental theories of cryptography

Concepts of classical cryptography

Classical cryptography is based on ciphers that were used in the past but are rarely used today. Its focus was mainly on languages.

- Classic ciphers were developed based on existing languages.
  - Traditional methods of encrypting messages
- They were designed to be simple.
  - Some features of plaintext appear in ciphertext.
  - Cryptographic algorithms are highly vulnerable because ciphertext-only attacks and known-plaintext attacks are similar in type.
  - Not currently used by cryptographers using computers
- They are still useful in some way.
  - Classic ciphers are not used alone, but are used as one step or in combination with another method in modern cryptosystems.
  - The basic principles of classical cryptosystems are still used in modern cryptosystems, greatly aiding cryptographic research.

Fundamental theories of cryptography

Concepts of classical cryptography

Classic cryptography is based on ciphers that were used in the past but are rarely used today. Its focus was mainly on languages.

- Composition of classic cipher
  - Transposition
    - Method of creating ciphertext by changing the position of each character
  - Substitution (simple substitution)
    - Method of creating ciphertext by substituting different characters according to certain predetermined criteria.
  - Number of keys used
    - Symmetric keys
  - Plaintext processing
    - In blocks

Fundamental theories of cryptography

Concepts of modern cryptography

Modern cryptography evolved from simple ancient ciphers as mathematics advanced. Multiple combinatorial ciphers were developed into machine ciphers during World War II.

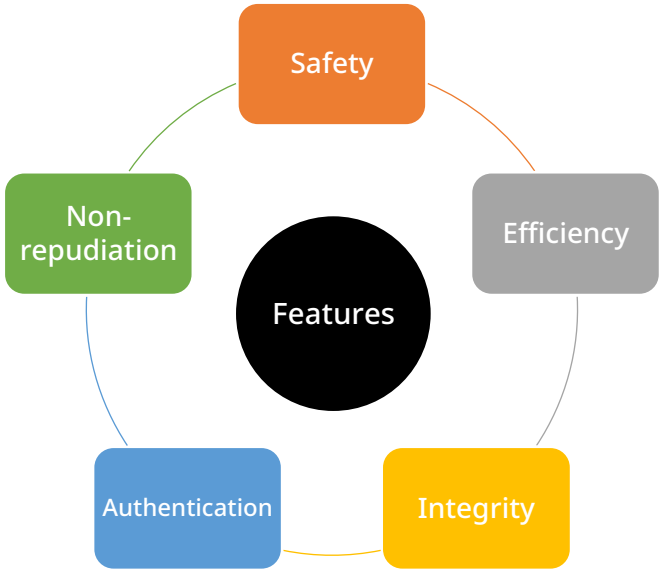
- Modern ciphers were developed based on existing languages as well.
  - Traditional methods for encrypting messages
  - Have more complex forms comparing with classic ciphers
- Modern cryptography began to be the subject of proofs in journal articles.
  - William F. Friedman, 1920, The Index of Coincidence and Its Applications in Cryptography.
    - Index of coincidence is a probabilistic method of calculating how closely two messages match (part of cryptanalysis)
    - It helped deciphering the Japanese PURPLE (Type B Cipher Machine) code in World War II.
  - Claude E. Shannon, 1949, Communication Theory of Secrecy Systems.
    - Prove the security of cryptographic schemes, presenting theories of confusion and diffusion.

Fundamental theories of cryptography

Contemporary cryptography

Contemporary cryptography was advanced by Stanford Univ. and MIT. In 1976, W. Diffie and M. E. Hellman of Stanford Univ. published the concept of public-key cryptography in their paper, New Directions in Cryptography. In 1978, R. Rivest, A. Shamir, and L. Adleman of MIT developed the RSA public-key cryptosystem based on the prime factorization method.

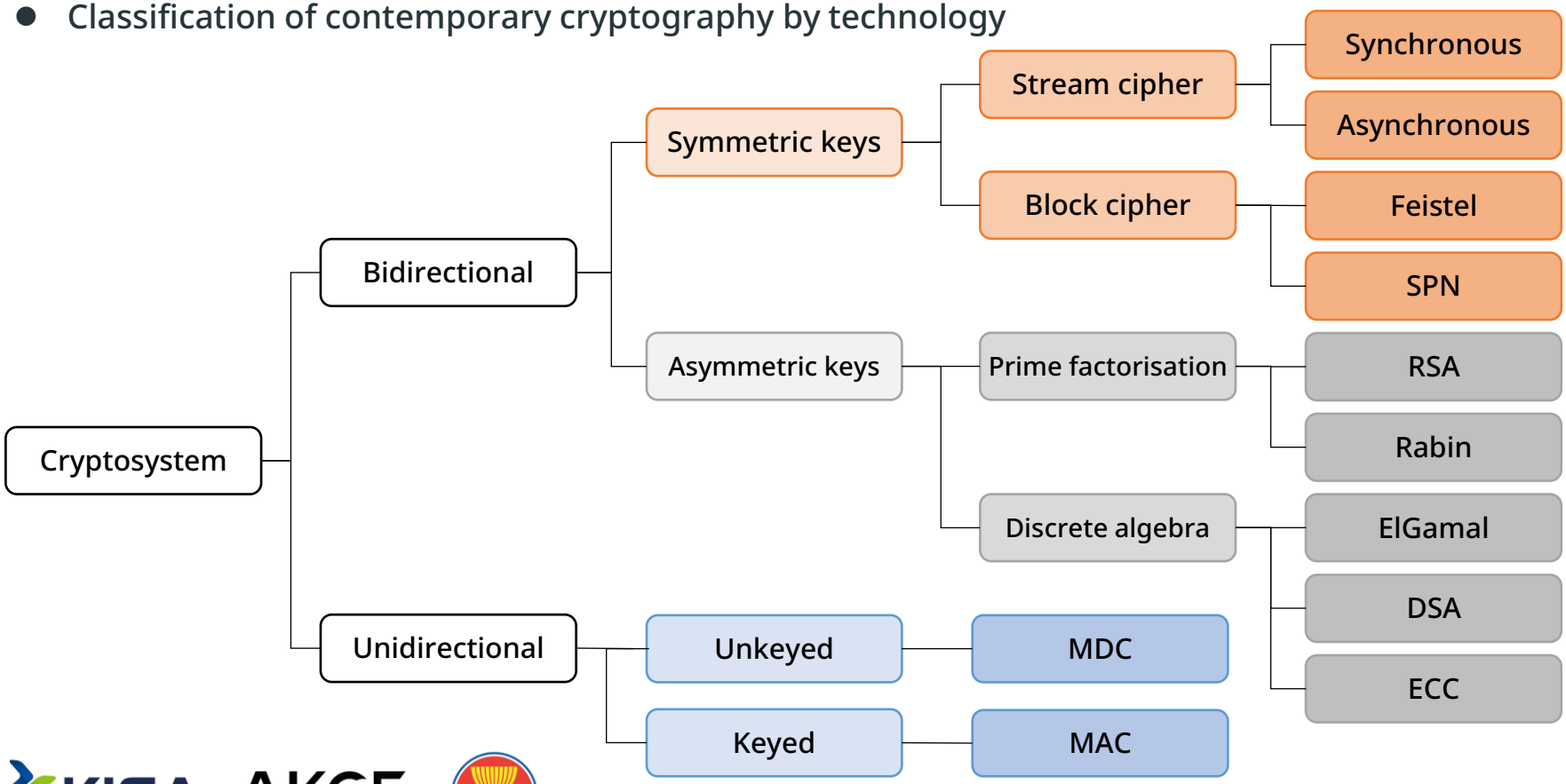
- Applied advanced mathematical theory
- The rise of civilian cryptography
  - Civilian use refers to the transfer of technology, products, etc. from military to civilian use.
- Using computer bits
  - As computing power increased, so did the difficulty of cryptography.
  - Cryptography began to have a close relationship with the computer information security.
    - Bitwise operations



Fundamental theories of cryptography

Contemporary cryptography was advanced by Stanford Univ. and MIT. In 1976, W. Diffie and M. E. Hellman of Stanford Univ. published the concept of public-key cryptography in their paper, New Directions in Cryptography. In 1978, R. Rivest, A. Shamir, and L. Adleman of MIT developed the RSA public-key cryptosystem based on the prime factorization method.

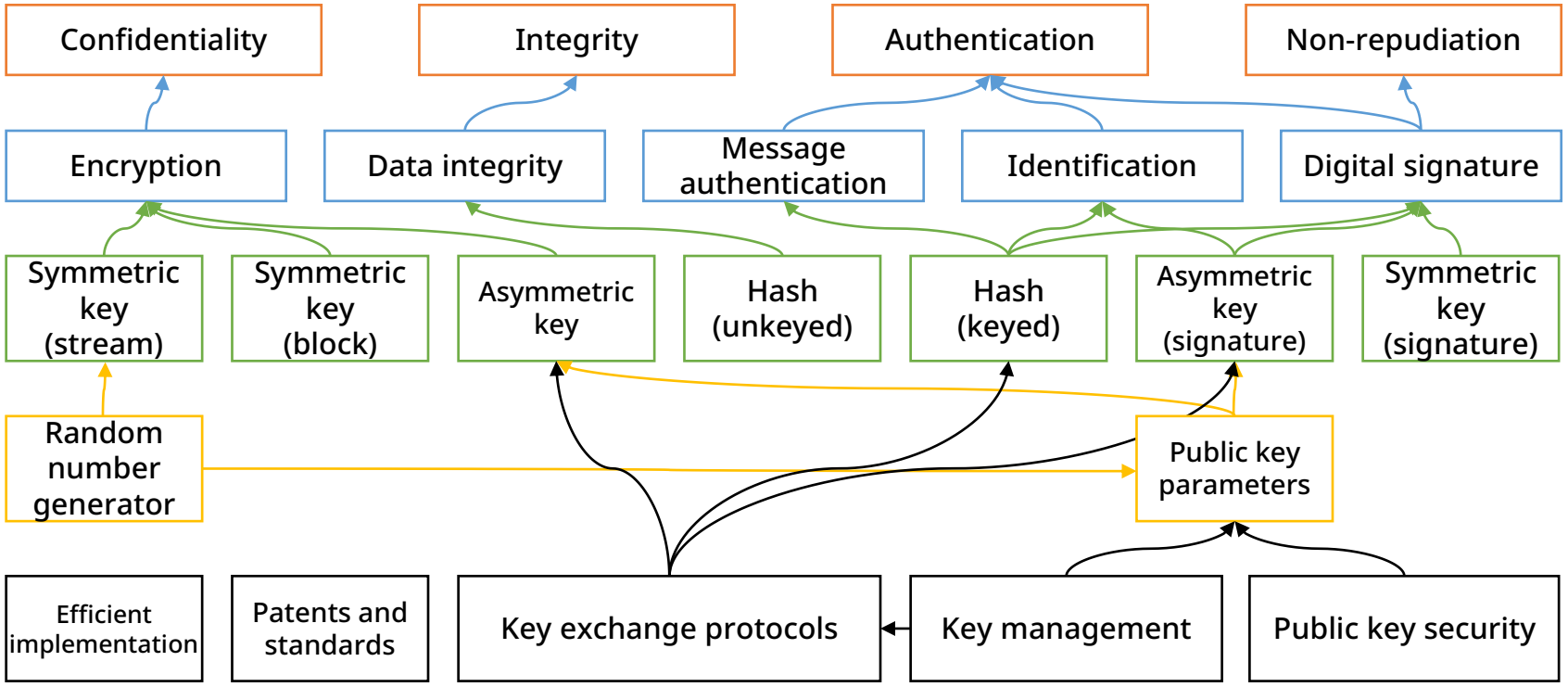
- Classification of contemporary cryptography by technology



Fundamental theories of cryptography

Contemporary cryptography was advanced by Stanford Univ. and MIT. In 1976, W. Diffie and M. E. Hellman of Stanford Univ. published the concept of public-key cryptography in their paper, New Directions in Cryptography. In 1978, R. Rivest, A. Shamir, and L. Adleman of MIT developed the RSA public-key cryptosystem based on the prime factorization method.

- Classification of contemporary cryptography by functionality



Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Proposition
  - A statement or mathematical expression that can be clearly determined as either true or false.
- Truth value
  - The value that a proposition represents as true or false
- Examples
  - The sum of the interior angles of a triangle is 360 degrees.
    - The sum of the interior angles of a triangle is 180 degrees, so the truth value is false (F).
  - 9 is a multiple of 3.
    - The truth value is true (T).
  - x is an integer, then if $x + 1 = 5$, x is equal to 4 .
    - The truth value is true (T).

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Negation
  - That which negates a proposition
  - Denoted by $\neg P$, $\sim P$, or $!P$
    - Also called a unary operator because it operates on a single term.

P	$\sim P$
T	F
F	T

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Conjunction
  - If the statements P and Q are propositions, $P\ AND\ Q$
  - Denoted by $P \wedge Q$
    - Can also be written as $P\ AND\ Q$ or P & Q

P	Q	$P\ AND\ Q$
T	T	T
T	F	F
F	T	F
F	F	F

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Disjunction
  - If the statements P and Q are propositions, $P\ AND\ Q$
  - Denoted by $P \vee Q$
    - Can also be written as $P\ OR\ Q$

P	Q	$P\ OR\ Q$
T	T	T
T	F	T
F	T	T
F	F	F

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Exclusive-or
  - If the statements P and Q are propositions, $P\ exclusive-or\ Q$
  - Denoted by $P \oplus Q$
    - Can also be written as $P\ XOR\ Q$

P	Q	$P\ XOR\ Q$
T	T	F
T	F	T
F	T	T
F	F	F

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Implication
  - If the statements P and Q are propositions, $P\ implies\ Q$
    - A proposition where P is a cause and Q is an effect
  - Denoted by $P \rightarrow Q$

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Biconditional
  - If the statements P and Q are propositions, $P\ if\ and\ only\ if\ Q$
    - A proposition where both P and Q are causes as well as effects
  - Denoted by $P \leftrightarrow Q$

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

Mathematical Basics for Understanding modern Cryptography

Converse, inverse, and contraposition

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Converse
  - For two propositions P and Q, if $P \rightarrow Q$, then $Q \rightarrow P$
- Inverse
  - For two propositions P and Q, if $P \rightarrow Q$, then $\sim P \rightarrow \sim Q$
- Contraposition
  - For two propositions P and Q, if $P \rightarrow Q$, then $\sim Q \rightarrow \sim P$

P	Q	$P \rightarrow Q$	$Q \rightarrow P$ $(\boldsymbol{converse})$	$\sim P \rightarrow \sim Q$ $(inverse)$	$\sim P \rightarrow \sim Q$ $(contraposition)$
T	T	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	F	T	T	T	T

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Tautology
  - A proposition whose compound proposition is always true, regardless of the truth value of each proposition.
- Contradiction
  - A proposition whose compound proposition is always false, regardless of the truth value of each proposition.

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Logical equivalence
  - When different compound propositions have the same truth value
  - Denoted by the symbol $\equiv$ or the symbol $\Leftrightarrow$

$p \wedge T \equiv p$	$p \vee F \equiv p$	Identity laws
$p \wedge F \equiv F$	$p \vee T \equiv T$	Domination laws
$p \wedge \neg p \equiv F$	$p \vee \neg p \equiv T$	Negation laws
$\neg(\neg p) \equiv p$		Double negation law
$p \wedge p \equiv p$	$p \vee p \equiv p$	Idempotent laws
$p \wedge q \equiv q \wedge p$	$p \vee q \equiv q \vee p$	Commutative laws
$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Logical equivalence
  - When different compound propositions have the same truth value
  - Denoted by the symbol ≡

$\neg(p \wedge q) \equiv \neg p \vee \neg q$	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's law
$p \wedge (p \vee q) \equiv p$	$p \vee (p \wedge q) \equiv p$	Absorption laws
$p \rightarrow q \equiv \neg p \vee q$		Implication law
$p \rightarrow q \equiv \neg p \rightarrow \neg q$		Contraposition law
$p \rightarrow q \equiv (p \wedge \neg q) \rightarrow F$		Reductio ad absurdum law

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Quantifier
  - Refers to words that indicate quantity in a statement
    - Examples
      - Some, at least one
  - Universe of discourse
    - Specify a certain scope to clarify a sentence, usually represented as D
  - Propositional function
    - A proposition $P(x)$ about the variable x contained in the universe of discourse D
    - Example
      - $P(x) = x^2$ is even. (D : a set of positive integers)

Mathematical Basics for Understanding modern Cryptography

A proposition is a statement or mathematical expression that can be clearly distinguished as either true or false. The essence of logic is to determine whether a proposition is true or false by using negation, conjunction, disjunction, exclusive-or, etc. in such a statement or mathematical expression.

- Types of quantifier
  - Universal quantifier
    - Denoted by the symbol $\forall x$ and written as for all x
    - Examples
      - $\forall x(P(x))$
      - The propositional function $P(x)$ is true for all domains x in the universe of discourse D
  - Existential quantifier
    - Denoted by the symbol $\exists x$ and written as for some x
    - Example
      - $\exists x(P(x))$
        - For each domain x belonging to D, there exists at least one such that the propositional function $P(x)$ is true.

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that have a common characteristic. Sets allow us to compute and represent sets in various forms, such as union, co-union, difference, and union.

- Set
  - A collection of objects (elements) that share common characteristics.
  - If element x belongs to set A, then,
    - Written as x is an element of the set A.
    - Denoted by $x \in A$
  - If the element is not an element x of the set A, then,
    - Denoted by $x \notin A$

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that share a common characteristic. Sets allow us to compute and express membership in various forms, such as universal set, empty set, relative complement, and union.

- Types of sets
  - Universal set
    - A set that contains all the elements of a given set.
    - Denoted by U
  - Empty set
    - A set that does not contain a single element
    - Denoted by $\{\}$ or $\emptyset$

▶ Practice question

- Explain what the set $A = \{ x \mid x + 1 < 0, x \text{ is a positive integer} \}$ is.

▶ Answer

- It is an empty set because there are no positive integers.

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that share a common characteristic. Sets allow us to compute and express membership in various forms, such as universal set, empty set, relative complement, and union.

- Types of sets
  - Subset
    - Define A to be a subset of B if A and B are sets and all elements of A are contained in B.
    - Denoted by $A \subseteq B$
    - $A \subseteq B \equiv (a \in A \rightarrow a \in B), \forall a$
  - Proper subset
    - A is a subset of B, but A and B are not the same.
    - Denoted by $A \subset B$

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that share a common characteristic. Sets allow us to compute and express membership in various forms, such as universal set, empty set, relative complement, and union.

- Types of sets
  - Finite set
    - A set of a finite number of elements
  - Infinite set
    - That which is not a finite set.
    - Well-known infinite sets
      - $R = \{x \mid x \text{ is a real number}\}$
      - $Q = \{x \mid x \text{ is a rational number}\}$
      - $Z = \{x \mid x \text{ is an integer}\}$
      - $N = \{x \mid x \text{ is a natural number}\}$
      - $R+ = \{x \in R \mid x > 0\}$
      - $I = \{x \in R \mid 0 \leq x \leq 1\}$

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that share a common characteristic. Sets allow us to compute and express membership in various forms, such as universal set, empty set, relative complement, and union.

- Element
  - Roster notation
    - A way of denoting a set by listing all the elements in a set within { } using commas.
  - Set-builder notation
    - A way of denoting a set by specifying the condition(s) of common properties of elements in a set.

▶ Practice question

- Denote the set containing 2, 4, 6, 8, and 10 in roster notation and set-builder notation.

▶ Answer

- Roster notation
  - Set $A = \{2, 4, 6, 8, 10\}$
- Set-builder notation
  - Set $A = \{ a \mid 2 \leq a \leq 10, a \text{ is an even number}\}$

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that share a common characteristic. Sets allow us to compute and express membership in various forms, such as universal set, empty set, relative complement, and union.

- Equality (equal sets)
  - Two sets A and B have the same elements
  - Sets are equal.
  - Denoted by $A = B$
  - While $a \in A$ is $a \in B$, $a \in B$ is $a \in A$, then $A = B$.
    - $\mathrm{A} = \mathrm{B} \equiv (\mathrm{a} \in \mathrm{A} \leftrightarrow \mathrm{a} \in \mathrm{B})$

Mathematical Basics for Understanding modern Cryptography

A set is a collection of elements that share a common characteristic. Sets allow us to compute and express membership in various forms, such as universal set, empty set, relative complement, and union.

- Cardinality
  - Number of elements in the finite set A
  - Denoted by $|A|$

▶ Practice question

- Find the cardinality(ies) of the given set $A = \{ a \mid a < 10, a \text{ is a positive integer} \}$.

▶ Answer

- Since the set is A={1, 2, 3, 4, 5, 6, 7, 8, 9}, the cardinality $|A|$=9.

Mathematical Basics for Understanding modern Cryptography

A relation is a structure for expressing associations between elements of a set. These structures can be represented using schema such as arrow diagrams and coordinate diagrams. To understand the properties of relations, we study types of relations such as reflection relations, symmetry relations.

- Basic relation
  - The representation of a relationality between elements is called a relation.
    - Denotes this relation as the symbol R
    - a has a relationship of R to b
    - sometimes denoted by ${}_aR_b$
    - Example
      - Two sets A, B, are in a binary relation where a subset of $A \times B$ is
        - in $(a, b) \in R$ in which $a \in A$ and $b \in B$.
    - Domain
      - The set of all first elements in the order pairs of the relation R : $dom(R)$
    - Range
      - The set of all second elements : $ran(R)$

Mathematical Basics for Understanding modern Cryptography

Relation

A relation is a structure for expressing associations between elements of a set. These structures can be represented using schema such as arrow diagrams and coordinate diagrams. To understand the properties of relations, we study types of relations such as reflection relations, symmetry relations.

- An n-ary relation
  - A relation between elements in two or more sets
  - Often used to express databases
    - Database
      - A set of data that is integrated, stored, and operated so that multiple application systems in an organization can share it.
        - Relational database model
        - Developed based on the concept of n-ary relation in a database.

▶ Practice question

- Find the number of relations that can be created from $A \times B$ in $A = \{a, b\}$, $B = \{1, 2\}$.

▶ Answer

- $A \times B = \{(a, 1), (a, 2), (b, 1), (b, 2)\}$
- $2^4 = 16$

Mathematical Basics for Understanding modern Cryptography

A relation is a structure for expressing associations between elements of a set. These structures can be represented using schema such as arrow diagrams and coordinate diagrams. To understand the properties of relations, we study types of relations such as reflection relations, symmetry relations.

- Inverse relation
  - A relation consisting of the inverses of elements of two or more sets.
    - Denoted by R^{-1}
    - $R^{-1} = \{(b, a) \in B \times A \mid (a, b) \in R\}$

▶ Practice question

- Find , the inverse relation R^{-1} of the relation $R = \{(1,4), (2,3), (2,5), (3,3)\}$.

▶ Answer

- $R^{-1} = \{(4,1), (3,2), (5,2), (3,3)\}$

Mathematical Basics for Understanding modern Cryptography

We will understand matrices expressed in a form of arrays and familiarize ourselves with the different types of matrices: zero, square, unit, and transposed.

- Matrix
  - A rectangular array of numbers
  - The matrix with m rows and n columns if m and n are positive integers :
    - $\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$
  - The element in the i-th row and the element in the j-th column : a_{ij}
  - The matrix with elements a_{ij} : $[a_{ij}]$

Mathematical Basics for Understanding modern Cryptography

We will understand matrices expressed in a form of arrays and familiarize ourselves with the different types of matrices: zero, square, unit, and transposed.

- Matrix
  - Equality
    - Given an $m \times n$ matrix $A = [a_{ij}]$ and an $r \times s$ matrix $B = [b_{ij}]$, for all i, j, if $m = r$, $n = s$ and $1 \leq i \leq m$, $1 \leq j \leq n$, then A and B are said to be equal.
      - Denoted by $A = B$
  - Matrix addition and subtraction
    - Given $m \times n$ matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, the two matrices can be added or subtracted :
      - $A + B = [a_{ij} + b_{ij}]$
      - $A - B = [a_{ij} - b_{ij}]$
  - Scalar multiplication
    - Given a $m \times n$ matrix $A = [a_{ij}]$ and a real (scalar) number k, the scalar multiplication of A and k :
      - $Ak = kA = [ka_{ij}]$

Mathematical Basics for Understanding modern Cryptography

We will understand matrices expressed in a form of arrays and familiarize ourselves with the different types of matrices: zero, square, unit, and transposed.

- Matrix
  - Properties of scalar addition and multiplication (O is a matrix where all elements are zero.)
    - $A + B = B + A$
    - $A + (B + C) = (A + B) + C$
    - $A + O = A = O + A$
    - $A + (-A) = O = (-A) + A$
    - $(-1)A = -A$
    - $c(A + B) = cA + cB$
    - $(c + d)A = cA + dA$
    - $(cd)A = c(dA)$

Mathematical Basics for Understanding modern Cryptography

We will understand matrices expressed in a form of arrays and familiarize ourselves with the different types of matrices: zero, square, unit, and transposed.

- Matrix
  - Practice question
    - Find $A+B$, $A+C$, $A+2B$, given that the matrices A, B, and C are as follows :
    - $A=\begin{bmatrix}1 & 2\\ -2 & 5\end{bmatrix}$, $B=\begin{bmatrix}3 & 4\\ 6 & 1\end{bmatrix}$, $C=\begin{bmatrix}-1 & 8 & 0\\ 2 & 0 & 4\end{bmatrix}$
  - Answer
    - $A+B=\begin{bmatrix}1 & 2\\ -2 & 5\end{bmatrix}+\begin{bmatrix}3 & 4\\ 6 & 1\end{bmatrix}=\begin{bmatrix}4 & 6\\ 4 & 6\end{bmatrix}$
    - $A+C$ are not computable because they are matrices of different sizes.
    - $A+2B=\begin{bmatrix}1 & 2\\ -2 & 5\end{bmatrix}+2\begin{bmatrix}3 & 4\\ 6 & 1\end{bmatrix}=\begin{bmatrix}1 & 2\\ -2 & 5\end{bmatrix}+\begin{bmatrix}6 & 8\\ 12 & 2\end{bmatrix}=\begin{bmatrix}7 & 10\\ 10 & 7\end{bmatrix}$

Mathematical Basics for Understanding modern Cryptography

We will understand matrices expressed in a form of arrays and familiarize ourselves with the different types of matrices: zero, square, unit, and transposed.

- Matrix
  - Matrix multiplication
    - Given an $m \times n$ matrix $A = [a_{ij}]$ and an $r \times s$ matrix $B = [b_{ij}]$, if $n = r$, then the multiplication of the matrices can be expressed as :
      - $m \times s$ matrix $AB = [c_{ij}]$
      - $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj}$
    - Properties of products and scalars
      - $(AB)C = A(BC)$
      - $A(B + C) = AB + AC$
      - $(B + C)A = BA + CA$
      - $k(AB) = (kA)B = A(kB)$

Mathematical Basics for Understanding modern Cryptography

We will understand matrices expressed in a form of arrays and familiarize ourselves with the different types of matrices: zero, square, unit, and transposed.

- Matrix
  - Matrix multiplication

▶ Practice question

- Find the product of the following two matrices AB.
- $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$

▶ Answer

- $AB = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 3 \\ 2 & 5 \end{bmatrix}$

Security of cryptographic algorithms

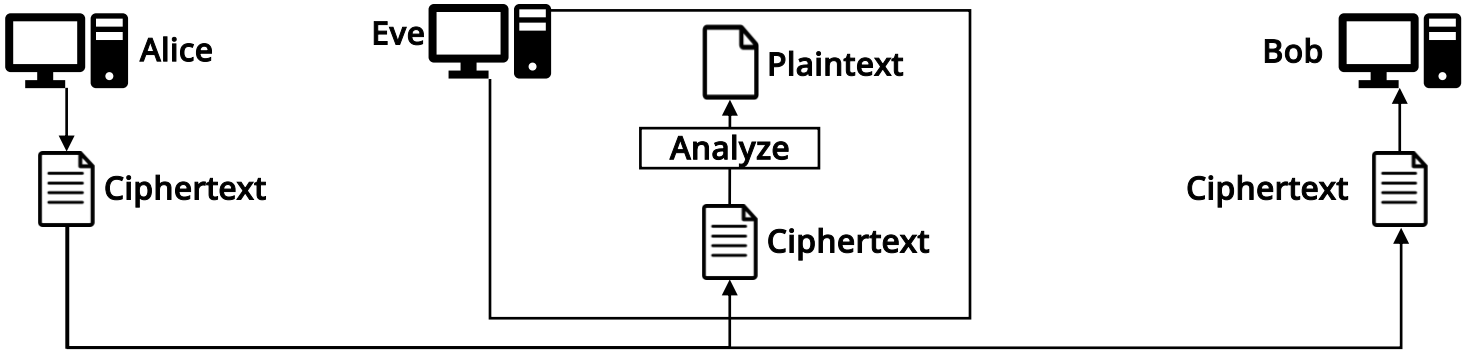
Just as cryptography is the science and art of creating a secret code, cryptanalysis is the science and art of breaking codes. Cryptanalytic techniques are needed to measure the vulnerability of a system, not to hack someone else's code.

- Cryptanalysis attacks
  - Ciphertext-only attacks
    - Exhaustive search attacks
    - Statistical attacks
    - Pattern attacks
  - Known-plaintext attacks
  - Chosen-plaintext attacks
  - Chosen-ciphertext attacks
  - Chosen text attacks

Security of cryptographic algorithms

Just as cryptography is the science and art of creating a secret code, cryptanalysis is the science and art of breaking codes. Cryptanalytic techniques are needed to measure the vulnerability of a system, not to hack someone else's code.

- Ciphertext-only attacks

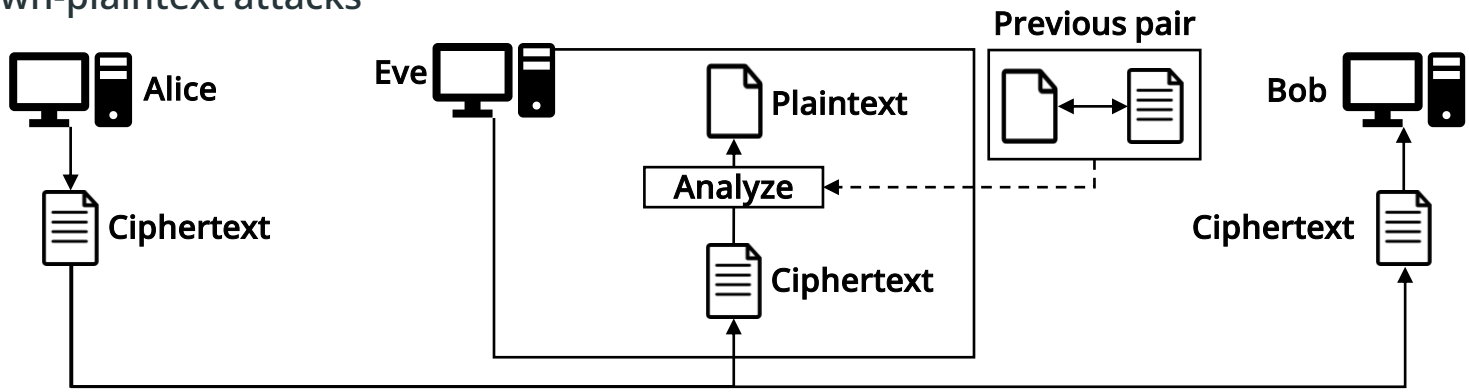


- Eve obtains a ciphertext and finds the corresponding plaintext and key.
  - Assuming Eve knows the encryption algorithm and can intercept the ciphertext.
- This is the easiest attack to apply because it only attempts to crack a ciphertext.
  - Exhaustive search attacks : repeats the attack until a meaningful plaintext is obtained.
  - Statistical attacks : exploits the frequency of alphabet usage.
  - Pattern attacks : exploits any pattern that may be present in a ciphertext.

Security of cryptographic algorithms

Just as cryptography is the science and art of creating a secret code, cryptanalysis is the science and art of breaking codes. Cryptanalytic techniques are needed to measure the vulnerability of a system, not to hack someone else's code.

- Known-plaintext attacks

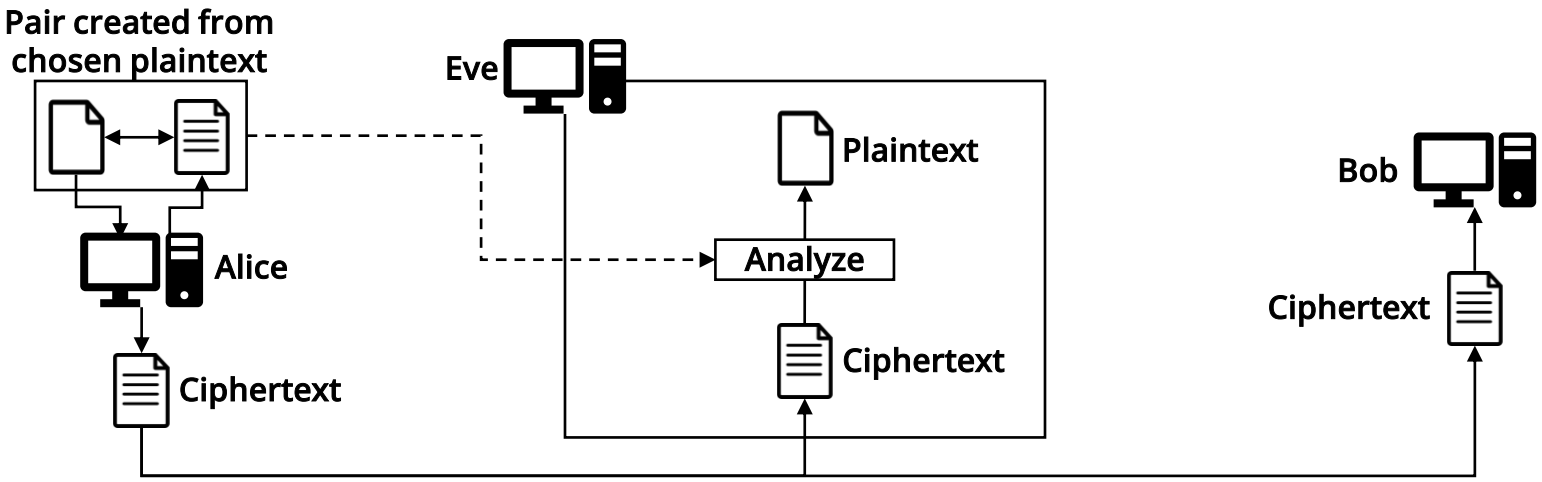


  - Known-plaintext attacks use boilerplate or common phrases.
    - Common phrases in emails and text messages, such as "hello" and "thank you"
  - Used to determine a secret key based on a known plaintext/ciphertext pair.

Security of cryptographic algorithms

Just as cryptography is the science and art of creating a secret code, cryptanalysis is the science and art of breaking codes. Cryptanalytic techniques are needed to measure the vulnerability of a system, not to hack someone else's code.

- Chosen-plaintext attacks

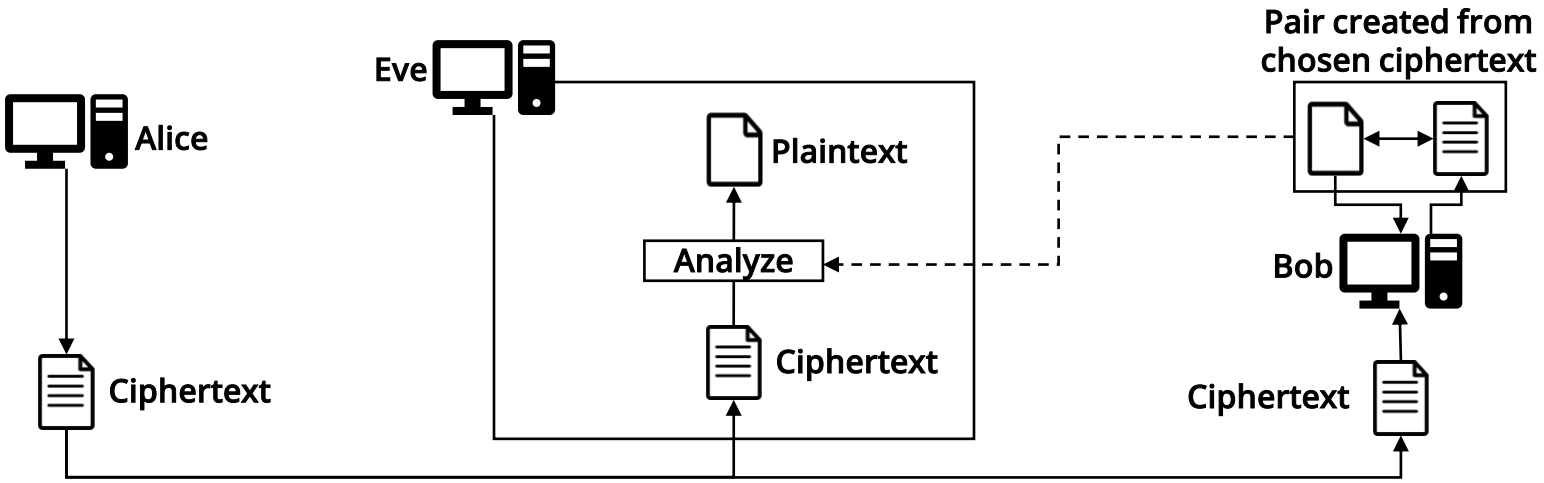


  - Similar to known-plaintext attacks
  - An attacker selects a plaintext/ciphertext pair.
  - Used when an attacker has access to the encryption module.

Security of cryptographic algorithms

Just as cryptography is the science and art of creating a secret code, cryptanalysis is the science and art of breaking codes. Cryptanalytic techniques are needed to measure the vulnerability of a system, not to hack someone else's code.

- Chosen-ciphertext attacks

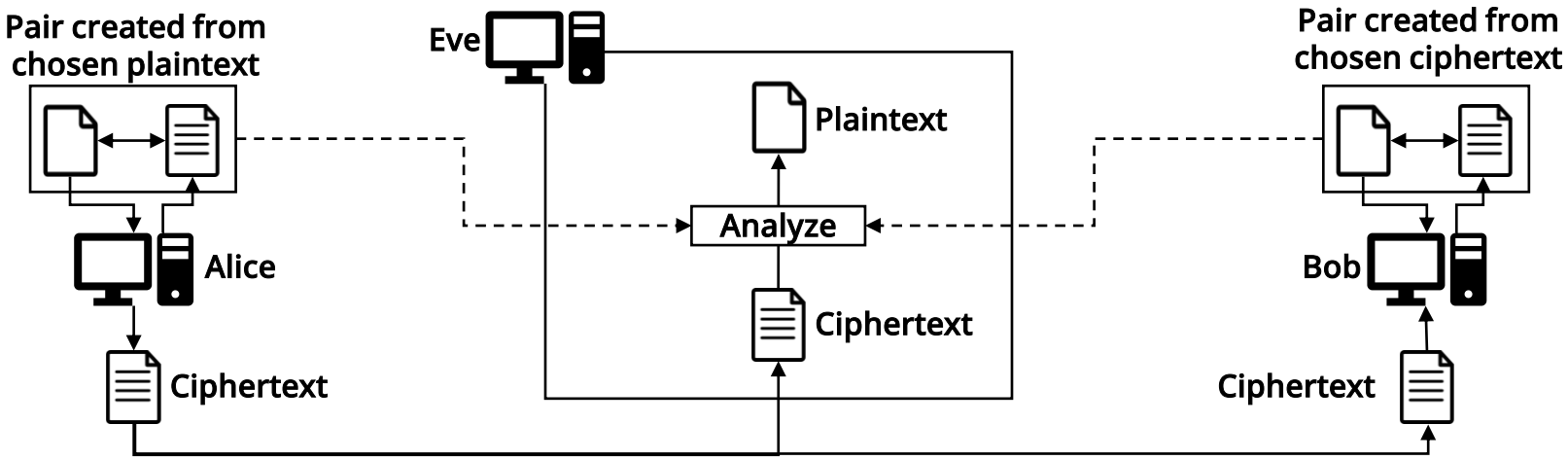


- Similar to chosen-plaintext attacks
- An attacker selects a ciphertext and obtains the corresponding plaintext.
- Used when an attacker has access to the decryption module.

Security of cryptographic algorithms

Just as cryptography is the science and art of creating a secret code, cryptanalysis is the science and art of breaking codes. Cryptanalytic techniques are needed to measure the vulnerability of a system, not to hack someone else's code.

- Chosen text attacks

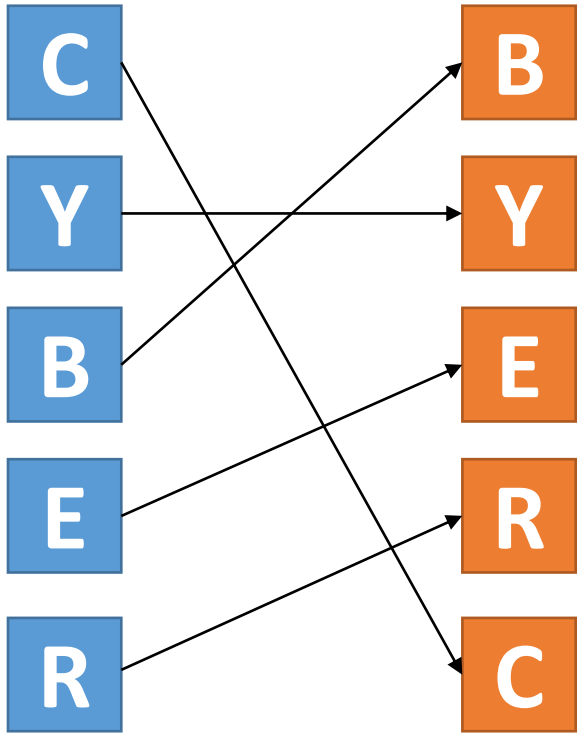


- An attacker selects a ciphertext/plaintext or plaintext/ciphertext pair.
- Used when an attacker has access to both the encryption and decryption modules.

Security of cryptographic algorithms

A transposition cipher is a method of repositioning characters on a character-by-character basis. It was used in ancient battles to covertly convey messages, but is no longer used directly, but rather in conceptual applications.

- Transposition ciphers
  - Rearranges text character by character for encryption
    - Permutation – an attacker picks a few specific characters and lists them in order.
    - Performs a permutation to obtain plaintext.
  - Types of transposition ciphers
    - Rail fence cipher
    - Route cipher
    - Columnar transposition cipher

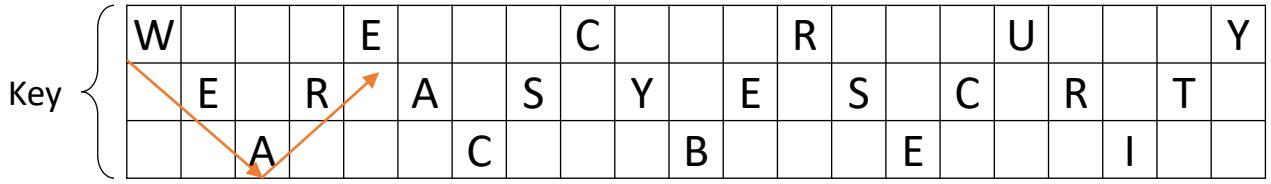


Security of cryptographic algorithms

Transposition cipher

A transposition cipher is a method of repositioning characters on a character-by-character basis. It was used in ancient battles to covertly convey messages, but is no longer used directly, but rather in conceptual applications.

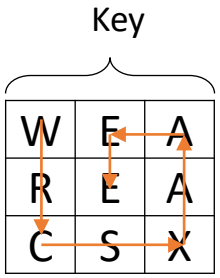
- Transposition ciphers
  - Rail fence cipher
    - What feels like organizing characters in a rail fence to encrypt them.
  - Example
    - Plaintext - WE ARE ACS CYBER SECURITY
    - Key - 3
    - Ciphertext - WECRUYERASYESCRTACBEI



Security of cryptographic algorithms

A transposition cipher is a method of repositioning characters on a character-by-character basis. It was used in ancient battles to covertly convey messages, but is no longer used directly, but rather in conceptual applications.

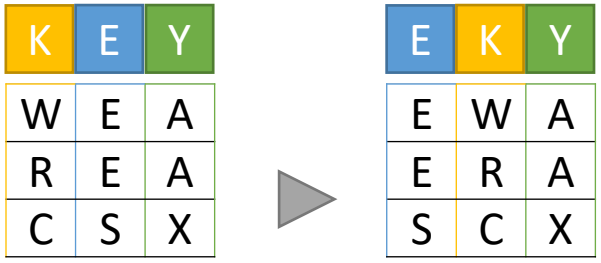
- Transposition ciphers
  - Route cipher
    - Transpose characters along a specified path, as indicated by the name “route.”
    - Specify path rules in addition to keys.
    - Fill in the X for spaces.
  - Example
    - Plaintext - WE ARE ACS
    - Key - 3
    - Path – counter-clockwise from top left
    - Passphrase - WRCSXAAEE



Security of cryptographic algorithms

A transposition cipher is a method of repositioning characters on a character-by-character basis. It was used in ancient battles to covertly convey messages, but is no longer used directly, but rather in conceptual applications.

- Transposition ciphers
  - Columnar transposition cipher
    - Use columnar transposition for encryption.
    - Assign a specific character to each column and use it as the key
  - Example
    - Plaintext - WE ARE ACS
    - Key - KEY
    - Ciphertext - EWAERASCX
  - This can be mixed with the route cipher :
    - Path – counter-clockwise from top left
    - Ciphertext - EESCXAAWR



Security of cryptographic algorithms

A transposition cipher is a method of repositioning characters on a character-by-character basis. It was used in ancient battles to covertly convey messages, but is no longer used directly, but rather in conceptual applications.

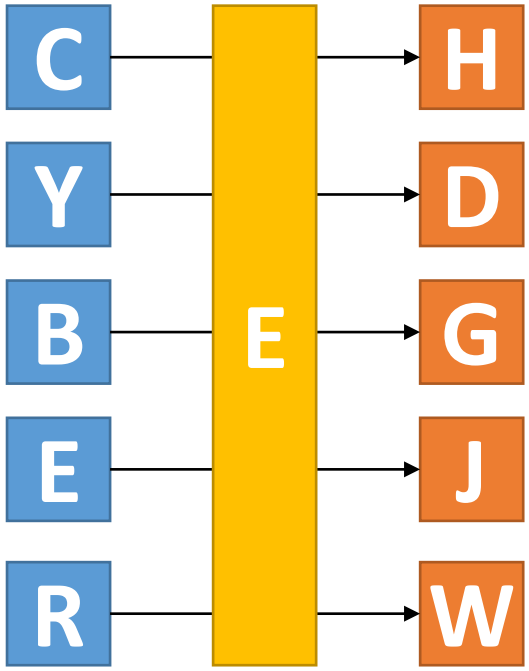
- Transposition ciphers
  - Other types of transposition ciphers
    - Double transposition
    - Myszkowski transposition
    - Disrupted Transposition
    - Grille
    - Scytale

Scytale ciphertext

Security of cryptographic algorithms

A simple substitution cipher is one in which characters are substituted according to a certain rule. Some elements are currently in use, but as in the case of transposition ciphers, the concept is applied indirectly rather than directly.

- Simple substitution ciphers
  - Character-by-character substitutions based on a given rule

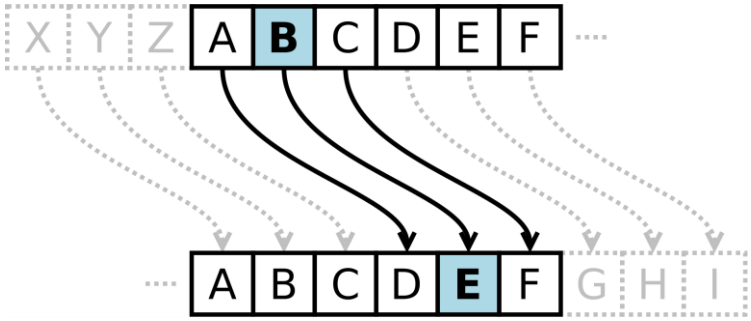


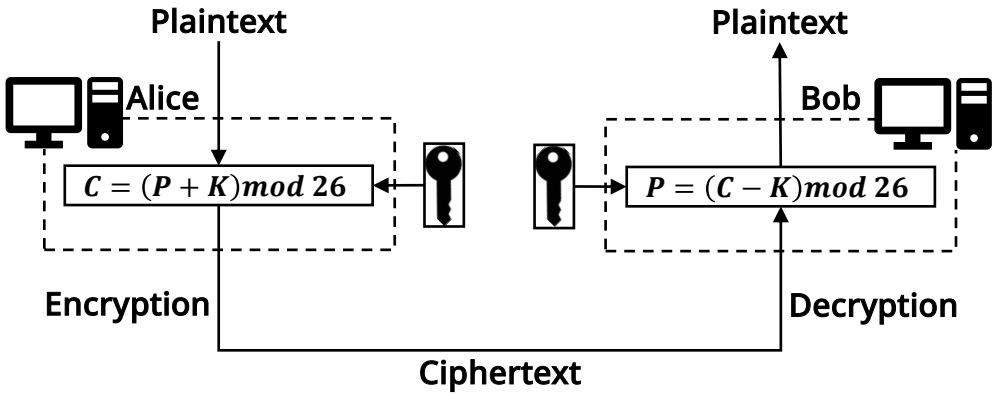
Security of cryptographic algorithms

Simple substitution cipher

A simple substitution cipher is one in which characters are substituted according to a certain rule. Some elements are currently in use, but as in the case of transposition ciphers, the concept is applied indirectly rather than directly.

- Caesar cipher
  - Swipe an alphabet a certain distance to replace it with another alphabet
  - Also known as shift cipher, Caesar shift, and additive cipher
  - All operations are performed within Z_{26}, assuming that plaintext is lowercase and ciphertext is uppercase.
  - The encryption algorithm is the key plus the plaintext character.
  - The decryption algorithm is the key minus the ciphertext character.



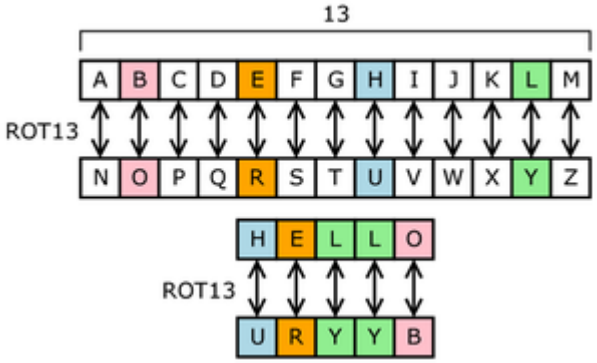


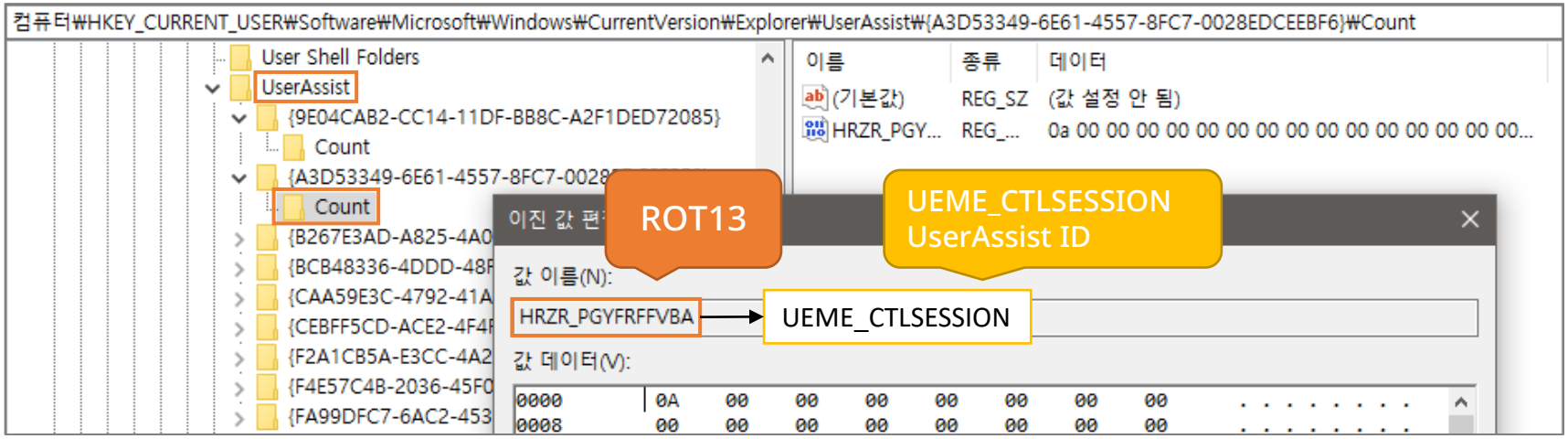
Security of cryptographic algorithms

Simple substitution cipher

A simple substitution cipher is one in which characters are substituted according to a certain rule. Some elements are currently in use, but as in the case of transposition ciphers, the concept is applied indirectly rather than directly.

- ROT13 (Rotate by 13)
  - Used in information security
  - This is a Caesar cipher with the key 13.
  - In the registry, the UserAssist key is configured as ROT13.

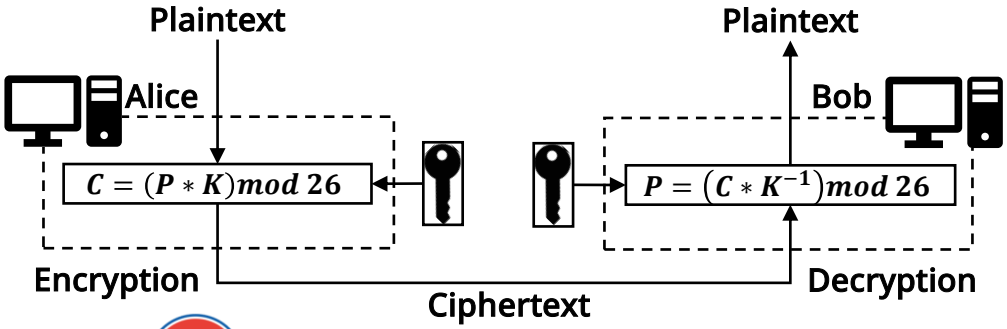




Security of cryptographic algorithms

A simple substitution cipher is one in which characters are substituted according to a certain rule. Some elements are currently in use, but as in the case of transposition ciphers, the concept is applied indirectly rather than directly.

- Multiplication cipher
  - The encryption algorithm performs modulo operations by multiplying the plaintext by the key.
  - The decryption algorithm performs modulo operations by multiplying the ciphertext by the inverse of the key.
  - The key must be an element of $Z_{26}{}^{*}$ to ensure that the encryption/decryption are inversely related.
  - Contains only the numbers for which $\gcd(26, x) \equiv 1$ holds when Euclidean algorism is applied.
    - Key space : 12 from $Z_{26}{}^{*} = \{1,3,5,7,9,11,15,17,19,21,23,25\}$
    - 13 is a prime number, but excluded from the keyspace since $\gcd(26,13) \equiv 13$ comes out.



Security of cryptographic algorithms

- Multiplication cipher
  - Euclidean algorithm
    - Euclidean algorithm for finding the greatest common divisor of two natural numbers.
    - Denoted by $\gcd(x, y)$
    - E.g., $\gcd(26,7)$ $\quad X = A - B * R \quad \therefore X = 26 - 7 * 3 = 5$

R	A	B	X
3	26	7	5
1	7	5	2
2	5	2	1
	2	1	

  - Extended Euclidean algorithm
    - In Z_{26}, find the inverse element of 7.
      - Apply the extended algorithm, to the part where 1 appears by the Euclidean algorithm.
      - The number at the position of 1 is the inverse element of 7.
      - In Z_{26}, the number ranges from 0 to 25, so -11mod26
      - Therefore, the inverse element of 7 is 15
      - Validation : $15 * 7 mod 26 \equiv 1$

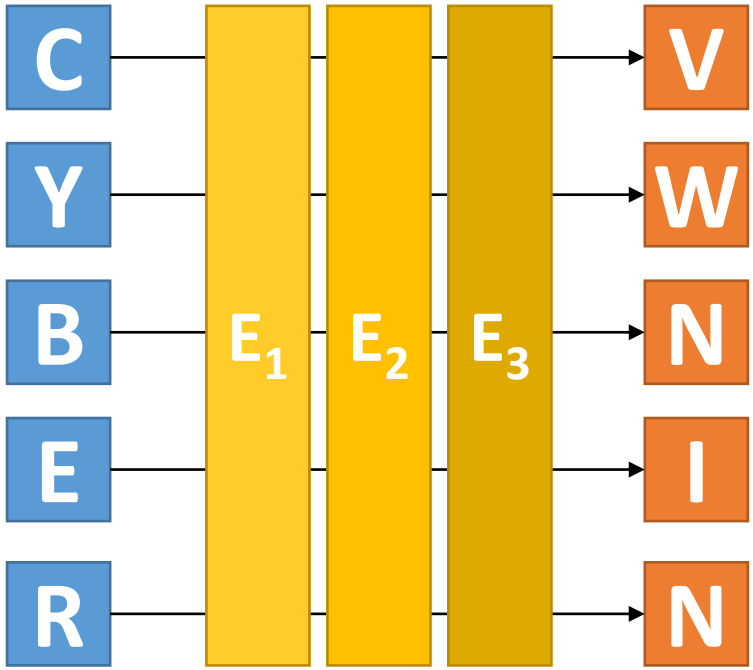
R				A	B	X
3	26	7	5	0	1	-3
1	7	5	2	1	-3	4
2	5	2	1	-3	4	-11
	2	1		4	-11	

Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Polyalphabetic substitution cipher (PSC)
  - Character-by-character substitution based on multiple types of set rules



Security of cryptographic algorithms

Polyalphabetic substitution cipher

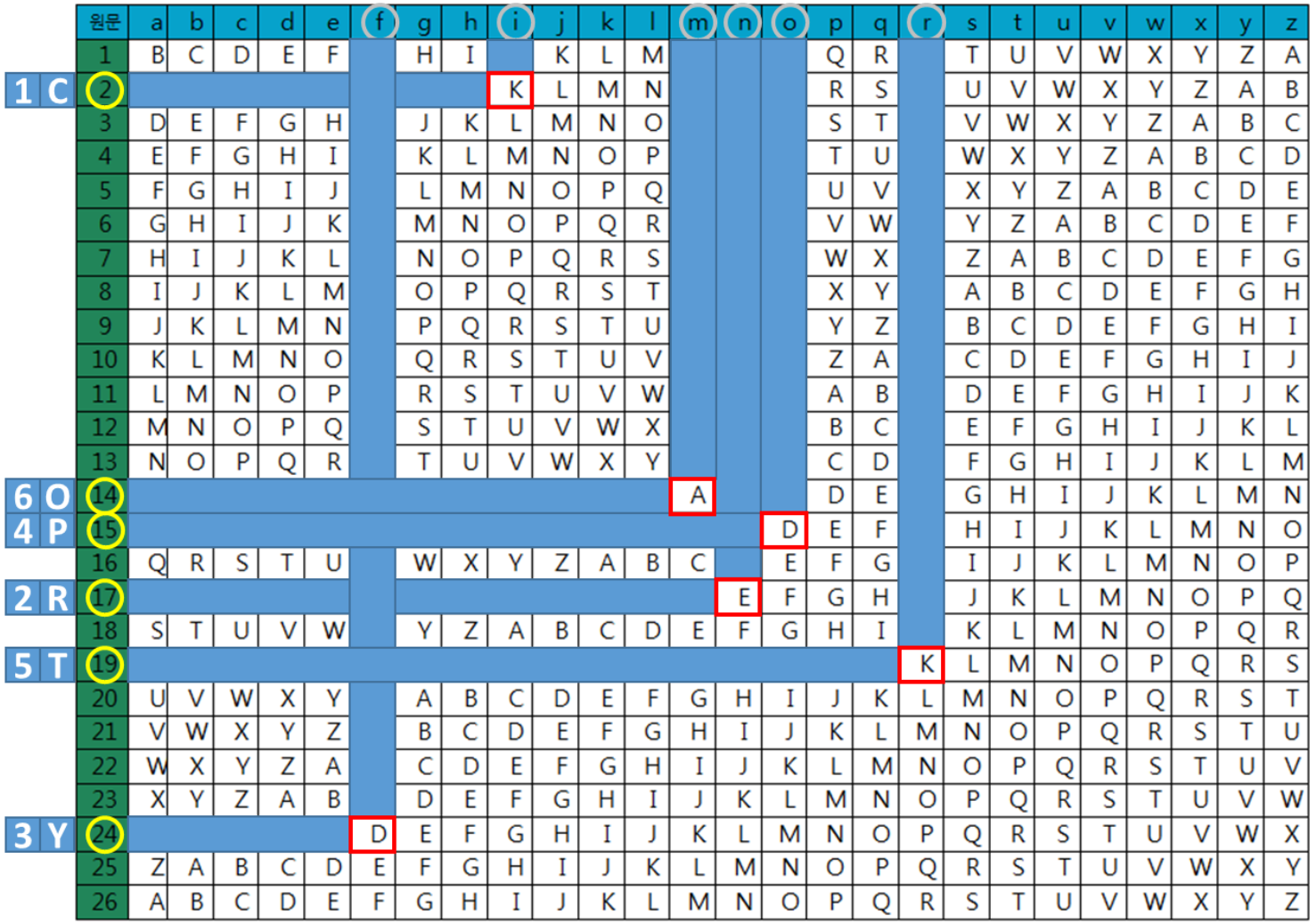
Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Vigenère cipher
  - There are key, plaintext, and Vigenère tables.
  - Key and Vigenère tables are freely configurable, but require fixed principles.
  - Can be decrypted to plaintext with key, ciphertext, and Vigenère tables together.
    - E.g., key - CRYPTO, plaintext - INFORMATION SECURITY

Key	C	R	Y	P	T	O	C	R	Y	P	T		O	C	R	Y	P	T	O	C
P	I	N	F	O	R	M	A	T	I	O	N		S	E	C	U	R	I	T	Y
C	K	E	D	D	K	A	C	K	G	D	G		G	G	T	S	G	B	H	A

Security of cryptographic algorithms

Polyalphabetic substitution cipher



원문	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
1	B	C	D	E	F		H	I		K	L	M				Q	R		T	U	V	W	X	Y	Z	A
1 C 2									K	L	M	N				R	S		U	V	W	X	Y	Z	A	B
3	D	E	F	G	H		J	K	L	M	N	O				S	T		V	W	X	Y	Z	A	B	C
4	E	F	G	H	I		K	L	M	N	O	P				T	U		W	X	Y	Z	A	B	C	D
5	F	G	H	I	J		L	M	N	O	P	Q				U	V		X	Y	Z	A	B	C	D	E
6	G	H	I	J	K		M	N	O	P	Q	R				V	W		Y	Z	A	B	C	D	E	F
7	H	I	J	K	L		N	O	P	Q	R	S				W	X		Z	A	B	C	D	E	F	G
8	I	J	K	L	M		O	P	Q	R	S	T				X	Y		A	B	C	D	E	F	G	H
9	J	K	L	M	N		P	Q	R	S	T	U				Y	Z		B	C	D	E	F	G	H	I
10	K	L	M	N	O		Q	R	S	T	U	V				Z	A		C	D	E	F	G	H	I	J
11	L	M	N	O	P		R	S	T	U	V	W				A	B		D	E	F	G	H	I	J	K
12	M	N	O	P	Q		S	T	U	V	W	X				B	C		E	F	G	H	I	J	K	L
13	N	O	P	Q	R		T	U	V	W	X	Y				C	D		F	G	H	I	J	K	L	M
6 O 14													A			D	E		G	H	I	J	K	L	M	N
4 P 15															D	E	F		H	I	J	K	L	M	N	O
16	Q	R	S	T	U		W	X	Y	Z	A	B	C		E	F	G		I	J	K	L	M	N	O	P
2 R 17														E	F	G	H		J	K	L	M	N	O	P	Q
18	S	T	U	V	W		Y	Z	A	B	C	D	E	F	G	H	I		K	L	M	N	O	P	Q	R
5 T 19																		K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z		B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A		C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B		D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
3 Y 24						D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	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
26	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

Vigenère Table

Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Playfair cipher
  - Developed by the British physicist Charles Wheatstone and the mathematician and geologist John Playfair.
  - Features
    - Key - a 5x5 alphabetical matrix containing certain words
    - Matrix organization
      - Represent all spellings, with keys, in a matrix of 25 without duplicates.
      - In 26 alphabet letters, I/J or Q/Z count as one letter.
      - The order is up to the array creator, except for certain words.
    - Example
      - Matrix containing a certain word (CYBER)

C	Y	B	E	R
A	D	F	G	H
I/J	K	L	M	N
O	P	Q	S	T
U	V	W	X	Z

Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Playfair cipher
  - Encryption
    - Plaintext - HELLO EVERYONE
      - Pair two letters of the alphabet and insert a random alphabet (usually X) between the repeated letters.
      - HE LX LO EV ER YO NE
    - Key - matrix

C	Y	B	E	R
A	D	F	G	H
I/J	K	L	M	N
O	P	Q	S	T
U	V	W	X	Z

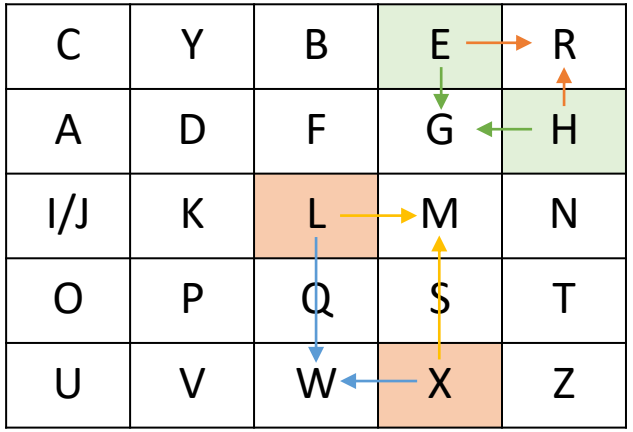
Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Playfair cipher
  - Encryption
    - Ciphertext 1
      - HE are diagonal to each other, so choose words on opposite diagonals of the same square size.
      - H to E is down → up
        - The move to the same direction creates a GR pair.
    - Ciphertext 2
      - LX creates MW in the same way.

Plaintext - HELLO EVERYONE
Variation - HE LX LO EVER YO NE



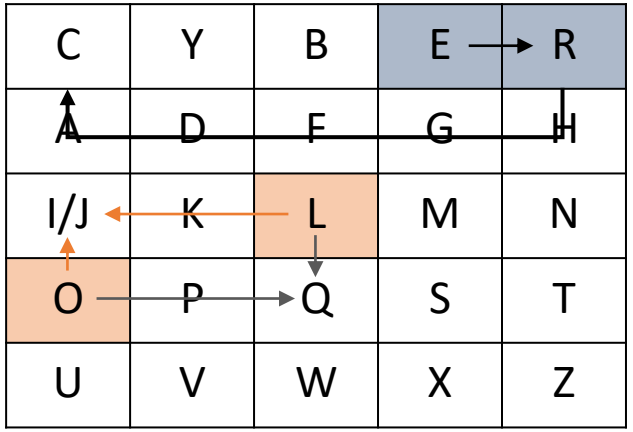
Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Playfair cipher
  - Encryption
    - Ciphertext 3
      - LO creates IQ.
    - Ciphertext 4
      - EV creates YX.
    - Ciphertext 5
      - ER is on the same line and in this case move one space to the right.
      - For the right end, move to the opposite side of the same row.
      - Since the direction is from E to R, ER creates RC.

Plaintext - HELLO EVERYONE
Variation - HE LX LO EVER YO NE



Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Playfair cipher
  - Encryption
    - Ciphertext 6
      - YO creates CP.
    - Ciphertext 7
      - NE creates MR.
    - Final ciphertext :
      - GR MW IQ YX RC CP MR
    - Although this is not present in the current example, if a pair of two letters exists in the same column,
      - Replace them with the characters in the set direction.
      - E.g., EM would create GS.

Plaintext - HELLO EVERYONE
Variation - HE LX LO EVER YO NE

C	Y	B	E	R
A	D	F	G	H
I/J	K	L	M	N
O	P	Q	S	T
U	V	W	X	Z

Security of cryptographic algorithms

Polyalphabetic substitution cipher

Polyalphabetic substitution ciphers were developed to address the shortcomings of simple substitution ciphers. It uses multiple iterations of the simplex algorithm, which increased in complexity as machines were built and advanced during the war.

- Playfair cipher
  - Decryption
    - Final ciphertext - GR MW IQ YX RC CP MR
      - GR is diagonal to each other, so this pair is replaced by HE.
      - MW is replaced by LX.
      - RC moves one place to the left, i.e., to the opposite side of the algorithm, which creates ER.
      - ...

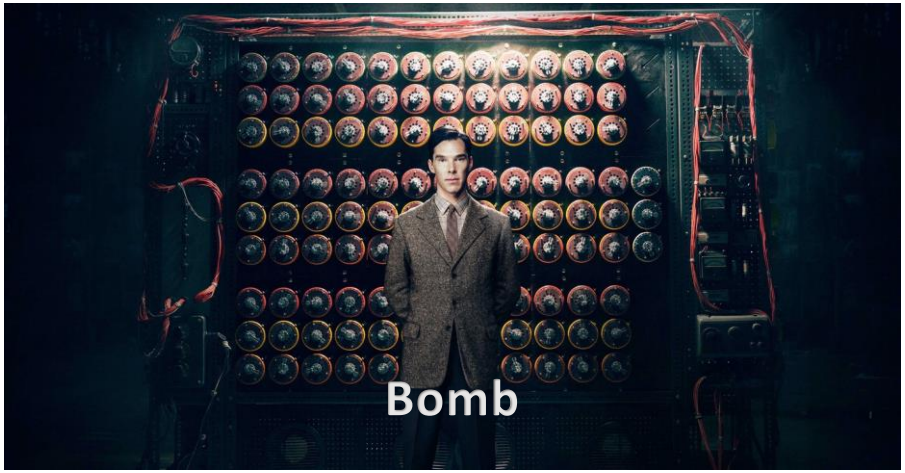
C	Y	B	E	R
A	D	F	G	H
I/J	K	L	M	N
O	P	Q	S	T
U	V	W	X	Z

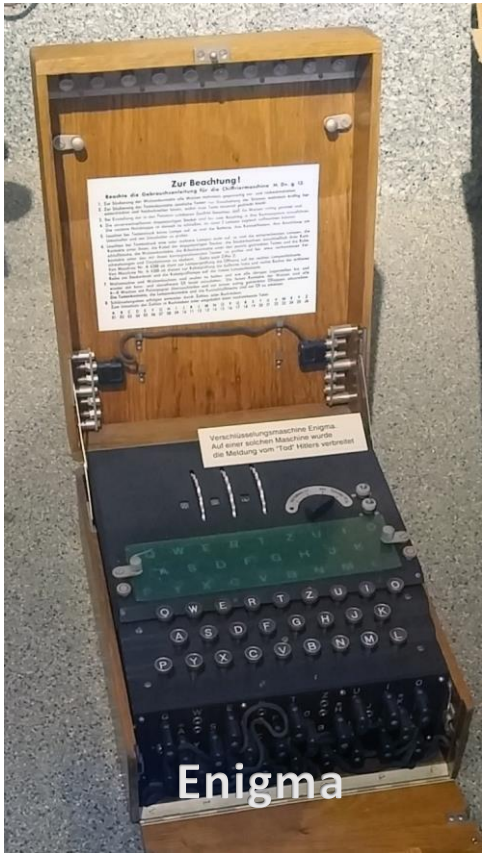
Security of cryptographic algorithms

Mechanical substitution cipher

A machine substitution cipher is a mathematically based cryptosystem invented during World War II to encrypt military and strategic messages that needed to be kept secret. Most of these machines are complex implementations of polyalphabetic substitution ciphers.

- Mechanical substitution ciphers
  - Enigma
    - German cryptographic system invented during World War II
    - Enigma cryptanalysis system
      - The Bombe in the UK





Security of cryptographic algorithms

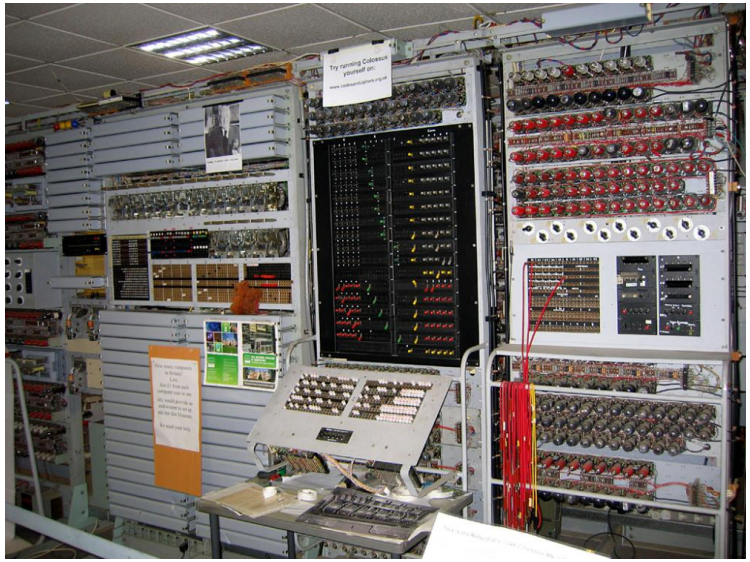
Mechanical substitution cipher

A machine substitution cipher is a mathematically based cryptosystem invented during World War II to encrypt military and strategic messages that needed to be kept secret. Most of these machines are complex implementations of polyalphabetic substitution ciphers.

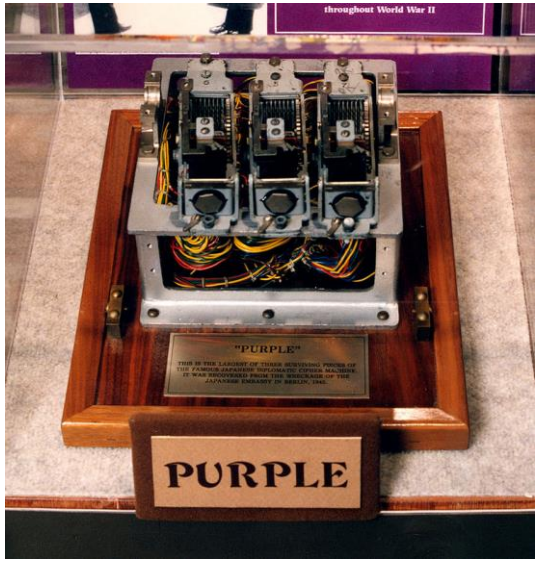
- Mechanical substitution ciphers
  - There were a variety of mechanical substitution ciphers other than the examples listed here.
  - These became obsolete with the advent of computers.

Boris Hagelin in Switzerland

Colossus in UK



Purple (Type B Cipher Machine) in Japan



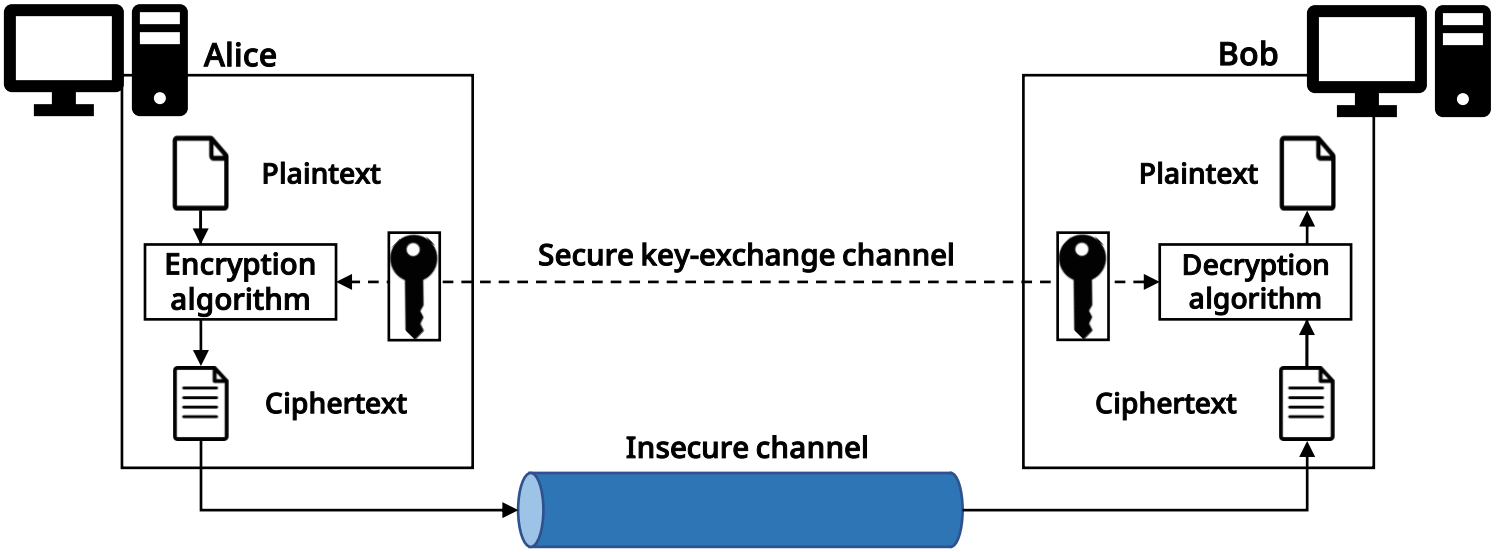
02

Symmetric Key Encryption

- Block cipher overview
- Block ciphers : operation modes
- Block ciphers : DES, 3DES, AES
- Stream cipher

Block cipher overview

- Symmetric-key cipher : an algorithm that uses the same secret key for encryption and decryption.
  - Advantages : fast encryption and decryption speeds
  - Disadvantages : requires sharing of the same symmetric key for encrypted communication



  - Plaintext : an original message sent by Alice to Bob
  - Ciphertext : an encrypted message sent over a channel
  - Alice and Bob use an encryption and decryption algorithm and a shared secret key.

Block cipher overview

- Alice's encryption : $C = E_k(P)$
- Bob's decryption : $P = D_k(C), P = D_k(C) = D_k\big(E_k(P)\big) = P$
- Assuming that encryption/decryption are inversely related, $D_k\big(E_k(x)\big) = E_k\big(D_k(x)\big) = x$ is established.
- Claude Shannon's information theory
  - Diffusion : the property that changes in the plaintext cannot affect changes in the ciphertext - Non-linear function
    - Hide the relationship between plaintext and ciphertext
    - Make ciphertexts unbreakable by the frequency of occurrence of the language
  - Chaos : the property of breaking a plaintext into multiple ciphertexts - Linear function
    - Hide the relationship between ciphertext and key
    - Make it impossible to find the key by using ciphertexts
  - Kerckhoff's principle
    - A cryptosystem should be secure even if an attacker knows the encryption/decryption algorithm.
    - This means that the cryptosystem should be secure based on the complexity of the key alone.

Block cipher overview

A symmetric-key cipher means that the keys used for encryption and decryption are the same. Common types include block ciphers, which encrypt messages in blocks, and stream ciphers, which are used to encrypt real-time communications.

- Well-known block cipher algorithms
  - Data Encryption Standard (DES)
    - A cryptographic algorithm established by the US National Institute of Standards and Technology (NIST) in 1977.
    - Encrypted 64-bit blocks with 56-bit keys
    - Small key lengths make it vulnerable to exhaustive key search attacks
    - Perform 3-DES with triple encryption as a temporary alternative
  - Advanced Encryption Standard (AES)
    - A cryptographic algorithm established by the US NIST in 2001.
    - The Rijndael cryptographic algorithm proposed in the contest became AES.
    - Encrypted in 128-bit blocks with 128-bit or larger keys.

Block cipher overview

A symmetric-key cipher means that the keys used for encryption and decryption are the same. Common types include block ciphers, which encrypt messages in blocks, and stream ciphers, which are used to encrypt real-time communications.

- Stream cipher is a type of symmetric-key cryptography.
- Faster encryption than block ciphers
- Generate a combined ciphertext with a logical exclusive-or (XOR) by generating a keystream where the ciphertext is the same length as the plaintext.
- Both the ciphertext generator and the receiver must share the same secret key and the same initial state of the random number generator.
- Usage
  - Commonly used in wireless communications
- Used with Linear Feedback Shift Register (LFSR) designs
  - Used to generate random numbers to make keys more secure when generating keystreams of the same length as plaintext.

Block cipher overview

A symmetric-key cipher means that the keys used for encryption and decryption are the same. Common types include block ciphers, which encrypt messages in blocks, and stream ciphers, which are used to encrypt real-time communications.

- Types
  - Synchronous stream ciphers
    - When decrypting a ciphertext to find the plaintext, there must be a synchronization between the keystream and the ciphertext.
    - The keystream is generated independently of the plaintext, so the ciphertext and the keystream in the ciphertext are independent, reducing the chance of information leakage.
    - High speed cryptographic processing and low error propagation rate are advantages, but low diffusion effect and lack of self-motivation are disadvantages
  - Self-synchronous stream ciphers
    - Keystreams are generated by function relations from plaintext or ciphertext
    - Even if bits of ciphertext are lost or altered during transmission, the effect of the error is finite.
    - May include the ability to correct errors
    - Easy to break due to dependencies between keystream and ciphertext

Block cipher overview

A block cipher algorithm is a fully developed algorithm that uses block-based cipher design and operation. Well-known block cipher algorithms include DES, AES, and Blowfish.

- An algorithm used as a cryptosystem based on a block cipher design.
  - Often referred to as a symmetric key cipher
- Well-known block cipher algorithms
  - Lucifer / DES
  - Rijndael / AES
  - Blowfish

	DES	AES
Year of development	1976	1999
Block size	64	128
Key length	56	128, 192, 256
Number of rounds	16	9, 11, 13
Password primitive	Substitution, permutation, chaos, and diffusion	Substitution, shift, bit-mixing, chaos, diffusion
Design	Public	Public
Design theory	Private	Public

Block cipher overview

There are three key elements to constructing a block cipher : whitening and tweakability to increase key security and performance, ciphertext complexity requirements, and the structure of the repeated block cipher.

- Block cipher design
  - Refers to the design approach underlying the block cipher algorithm.
  - Basic block cipher design
    - Key security and performance
      - Whitening – performs round key XOR followed by encryption, and the final encryption with the round key before generating the ciphertext.
      - Tweakable - tweaks the encryption key one more time using a tweakable key that is separate from the round key.
    - Complexity requirements
      - Avalanche - an avalanche-like increase in the size of a ciphertext compared to the size of the plaintext.
      - Nonlinearity - complexity increased by configuring the spread to be irregular (non-linear).

Block cipher overview

There are three key elements to constructing a block cipher : whitening and tweakability to increase key security and performance, ciphertext complexity requirements, and the structure of the repeated block cipher.

- Block cipher design
  - Refers to the design approach underlying the block cipher algorithm.
  - Basic block cipher design
    - Iterated block cipher structure
      - Feistel
      - Substitution-Permutation Network (SPN)
      - Lai-Massey - rarely used

Block cipher overview

There are three key elements to constructing a block cipher : whitening and tweakability to increase key security and performance, ciphertext complexity requirements, and the structure of the repeated block cipher.

- Iterated block cipher
  - Features
    - Basic structure shared by most block ciphers
    - Repeated use to create cryptographically strong structures
    - Apply chaos and diffusion in each round
    - As the number of rounds increases, so does the height.
  - Requires a key scheduling process where a key is entered to generate a round key
    - Primary key - the original encryption key
    - Round key - an independent encryption key created by splitting the primary key
  - Advantages - security improves as the number of rounds increases.
  - Disadvantages - less practical as the number of rounds increases.

Block cipher overview

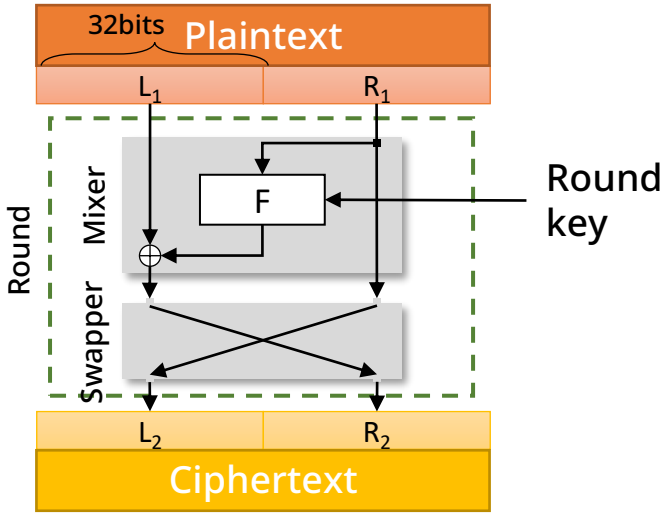
There are three key elements to constructing a block cipher : whitening and tweakability to increase key security and performance, ciphertext complexity requirements, and the structure of the repeated block cipher.

- Iterated block cipher
  - Feistel structure
    - Structure for which the inverse function does not exist
    - Decryption algorithm uses the same components.
    - Flexible in designing compared to SPN
      - SPN changes its entire value in one round, but only half of the Feistel structure changes.
        - Structures designed with different sizes or numbers of partitions are called unbalanced Feistel structures.
    - Have a strong cipher design by iterating over weak rounds
    - Signature algorithm using Feistel structure – DES
      - DES executes 16 rounds of the Feistel structure.

Block cipher overview

There are three key elements to constructing a block cipher : whitening and tweakability to increase key security and performance, ciphertext complexity requirements, and the structure of the repeated block cipher.

- Iterated block cipher
  - Feistel structure
    - Divide a 64-bit plaintext block back into 32 bits
    - XOR the left 32-bit block with the encrypted right 32-bit block
    - Write the result to the right 32 bits
    - Store existing right 32 bits in left 32 bits



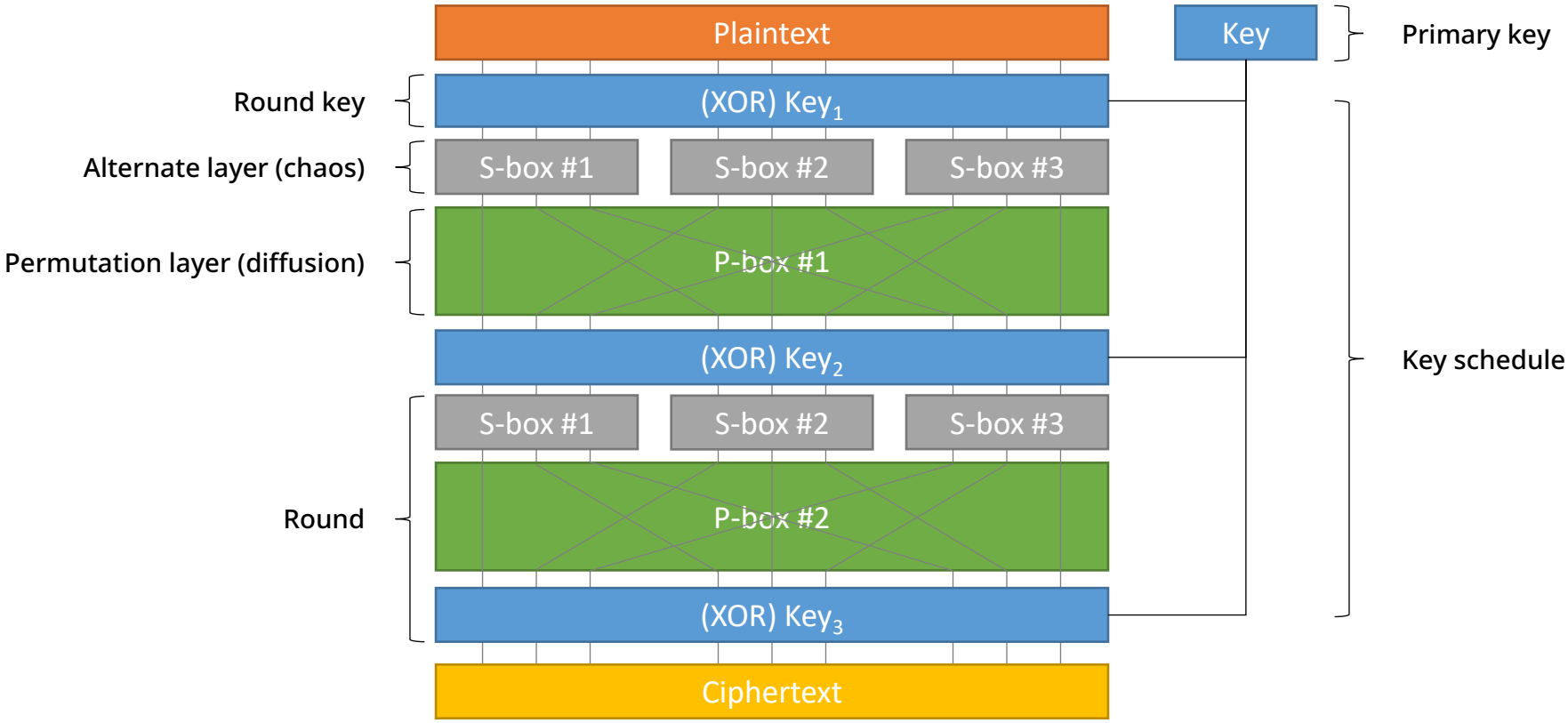
Block cipher overview

There are three key elements to constructing a block cipher : whitening and tweakability to increase key security and performance, ciphertext complexity requirements, and the structure of the repeated block cipher.

- Iterated block cipher
  - Substitution-Permutation Network (SPN) structure
    - Designed by Claude Shannon, the architect of chaos and diffusion
    - Uses two layers in each round
    - If the substitution layer causes chaos, the permutation layer diffuses
      - Substitution layer = S-boxes, permutation layer = P-boxes
      - Replace with another value by S-box round key
      - P-boxes mix S-boxes coming in as input with S-boxes going out as output to create a diffusion
    - Signature algorithm using SPN structure - AES

Block cipher overview

- Iterated block cipher
  - Substitution-Permutation Network (SPN) structure (example - Round 2)



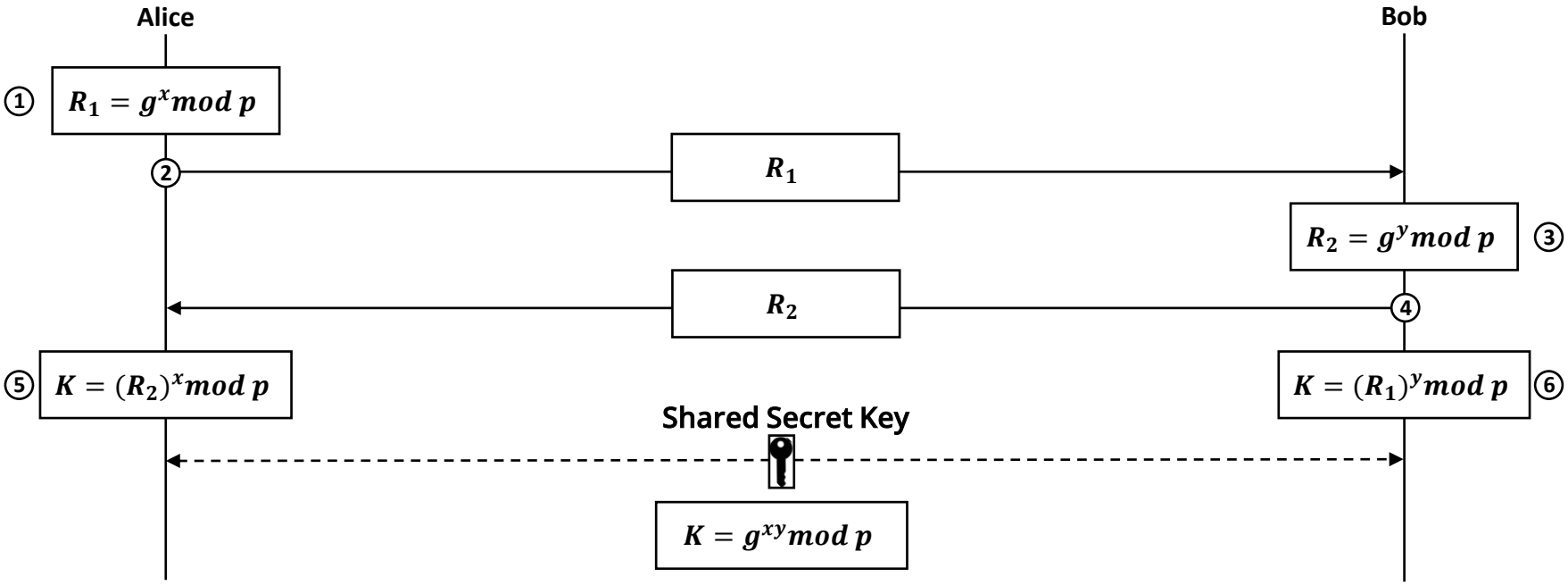
Block cipher overview

- Diffie-Hellman key exchange consensus
  - Theory that computing someone else's public key and your private key yields a secret key.
  - Used for symmetric key exchange, not encryption or signing.
  - Use discrete algebra problems
    - When $y = g^x mod\ p$, it's easy to get y if you know g, x, p, but hard to get x if you know g, y, p.
  - Both communicating parties generate symmetric keys without a KDC.
  - Procedure
    - Alice chooses a random large number x inside $0 \leq x \leq p-1$ and calculates $R_1 = g^x mod\ p$.
    - Bob chooses another random large number y inside $0 \leq y \leq p-1$ and calculates $R_2 = g^y mod\ p$.
    - Alice sends to Bob R_1. Here Alice is not sending the value x, but only R_1.
    - Bob sends to Alice R_2. Here Bob does not send y, but only R_2
    - Alice calculates $K = (R_2)^x mod\ p$.
    - Bob calculates $K = (R_1)^y mod\ p$.

$$\boldsymbol{K = (g^x\ mod\ p)^y\ mod\ p = (g^y\ mod\ p)^x\ mod\ p = g^{xy} mod\ p}$$

Block cipher overview

- Diffie-Hellman key exchange consensus



- Bob's calculation result : $K = (R_1)^y mod\ p = (g^x) mod\ p)^y\ mod\ p = g^{xy} mod\ p$
- Alice's calculation result : $K = (R_2)^x mod\ p = (g^y) mod\ p)^x\ mod\ p = g^{xy} mod\ p$
- Alice doesn't know the value of y and Bob of x, but they both get the same K.

In the Diffie-Hellman method, the symmetric key is $K = g^{xy} mod\ p$

Block cipher overview

- Diffie-Hellman key exchange consensus
  - For example, find K when $g = 7, p = 23$ (calculate with small numbers).
    - Alice selects $x = 3$ and calculates $R_1 = 7^3 mod\ 23 = 21$.
    - Bob selects $y = 6$ and calculates $R_2 = 7^6 mod\ 23 = 4$.
    - Alice sends to Bob 21.
    - Bob sends to Alice 4.
    - Alice computes the symmetric key $K = 4^3 mod\ 23 = 18$.
    - Bob computes the symmetric key $K = 21^6 mod\ 23 = 18$.
    - Alice and Bob each receive K and see the same value of 18 for both Alice and Bob.
      - $g^{xy} mod\ p = 7^{18} mod\ 23 = 18$

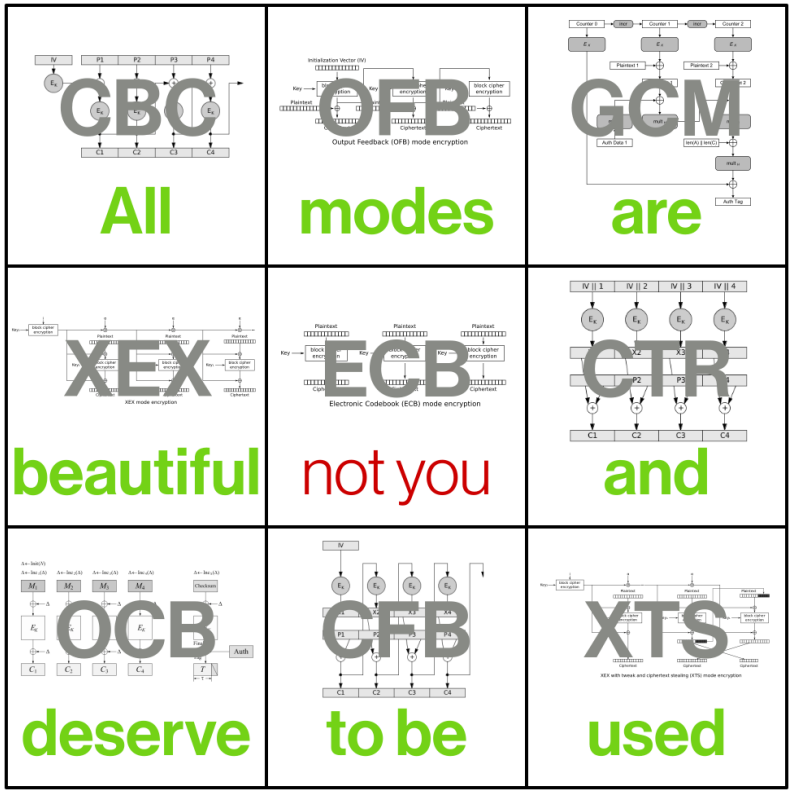
Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

- Features

  Method for solving the problem of how to enforce a cipher when the length of the plaintext is greater than the block size.

  - Use both the block operating mode and the cryptographic algorithm.
    - E.g., implementing IPSec encapsulated secure payloads using AES-CTR RFC : https://tools.ietf.org/html/rfc3686
  - Block operating mode is not only used for block ciphers.
    - It is used for block ciphers, stream ciphers, hash functions, and more.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

- Understanding padding
  - In cryptography, padding is the addition of data at the beginning, middle, or end of plaintext when encrypting it.
    - It is used to correct incorrectly sized blocks when plaintext is divided into blocks.
  - Types of padding
    - Bit padding
      - Start with bit 1 and fill it with bit 0 (e.g., 100...)
    - Byte padding
      - Fill specific bytes according to a specified rule
        - ANSI x9.23, ISO 10126, PKCS#5 and PKCS#7, ISO/IEC 7816-4
    - Zero padding
      - Fill with bit 0 (e.g., 000...)

Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

- Error propagation
  - Refers to whether or not one block affects other blocks when it fails.
    - Claimed to be manipulable by malicious users through error multiplication
  - This property was discussed until it evolved into message authentication codes and authenticated encryption.
    - Because it validates whether the message is corrupt or not.
- Parallel implementation
  - Refers to whether or not each block can be decrypted in isolation.
    - Has a significant impact on performance metrics

Block ciphers : operation modes

Types of block operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

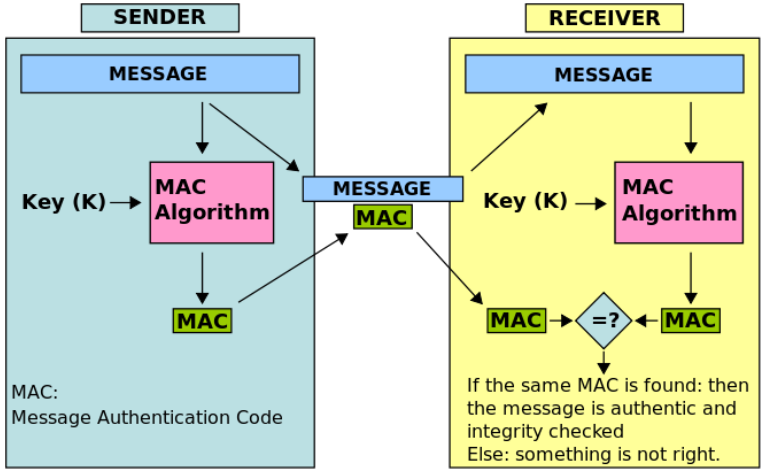
- Block operation modes
  - Common operating modes
    - Electronic Codebook (ECB)
    - Cipher Block Chaining (CBC)
    - Propagating Cipher Block Chaining (PCBC)
  - Operation modes converted to stream ciphers
    - Cipher Feedback (CFB)
    - Output Feedback (OFB)
    - Counter (CTR)

Block ciphers : operation modes

Types of block operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

- Authentication uses the following message block.
  - Message Authentication Code (MAC)
  - Authenticated Encryption (AE)
    - Authenticated Encryption with Associated Data (AEAD)
    - Encrypt-then-MAC (EtM)
    - Encrypt-and-MAC (E&M)
    - MAC-then-Encrypt (MtE)

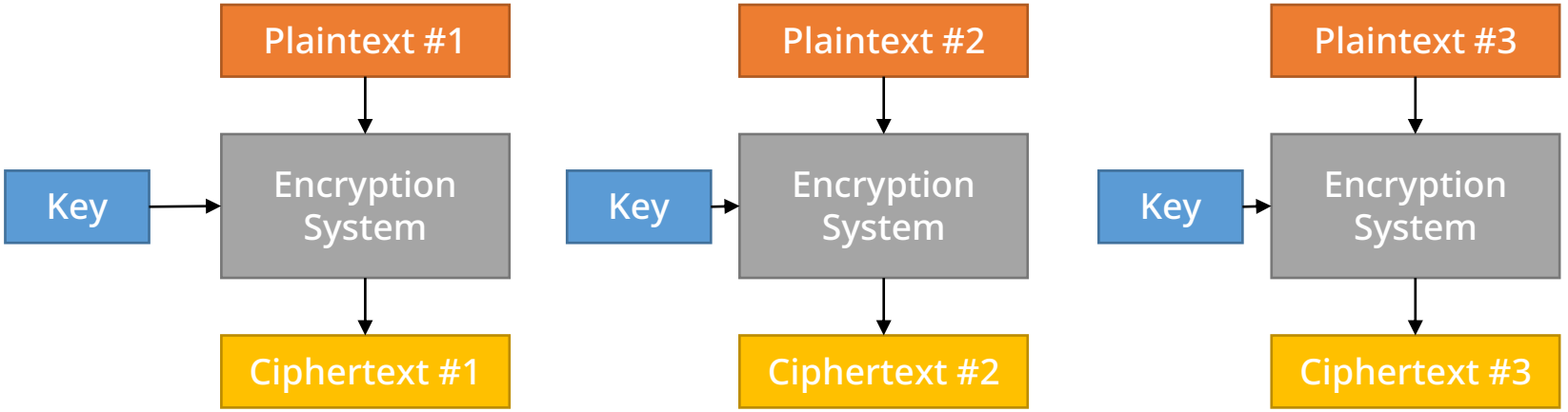


Block ciphers : operation modes

Electronic codebook

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

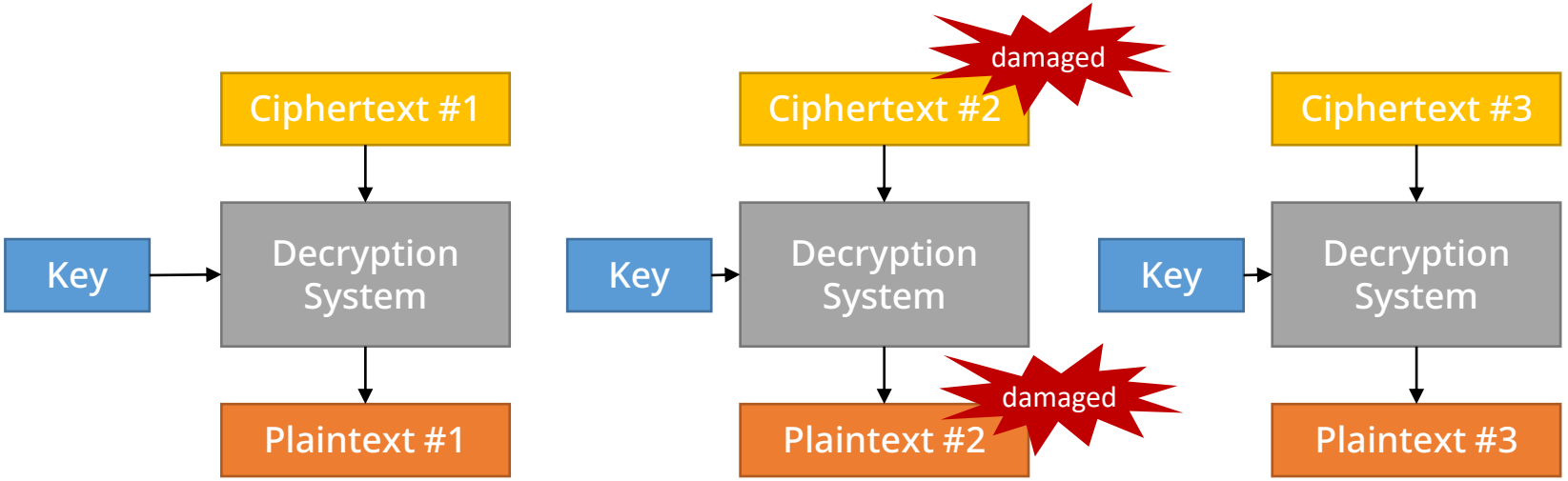
- Electronic Codebook (ECB)
  - Encrypt plaintext blocks without any algorithm
  - Not used because it's simple and has weaknesses
  - Relationship between plaintext and ciphertext
    - Encryption : $C_i = E_k(P_i)$, Decryption : $P_i = D_k(C_i)$



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

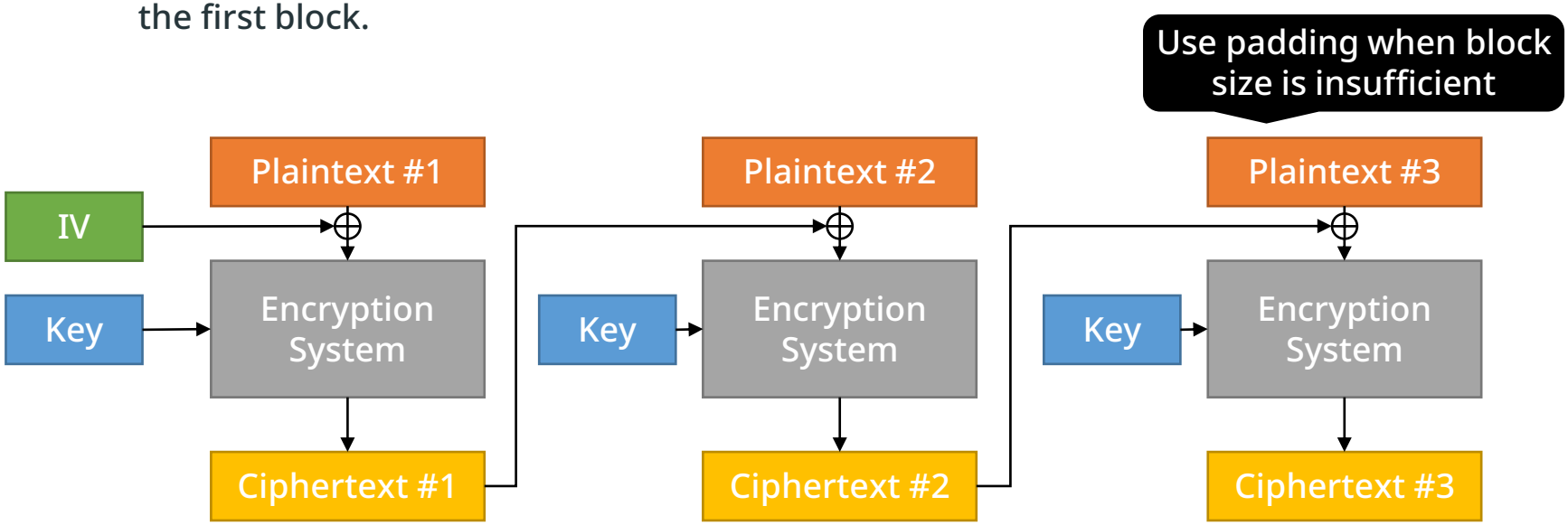
- Electronic Codebook (ECB)
  - Decryption is done in reverse order because it uses a symmetric key.
  - Error propagation
    - An error in one block will only propagate to related blocks because each operates separately.
    - However, even if Ciphertext #2 is partially damaged, Plaintext #2 will be completely damaged.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

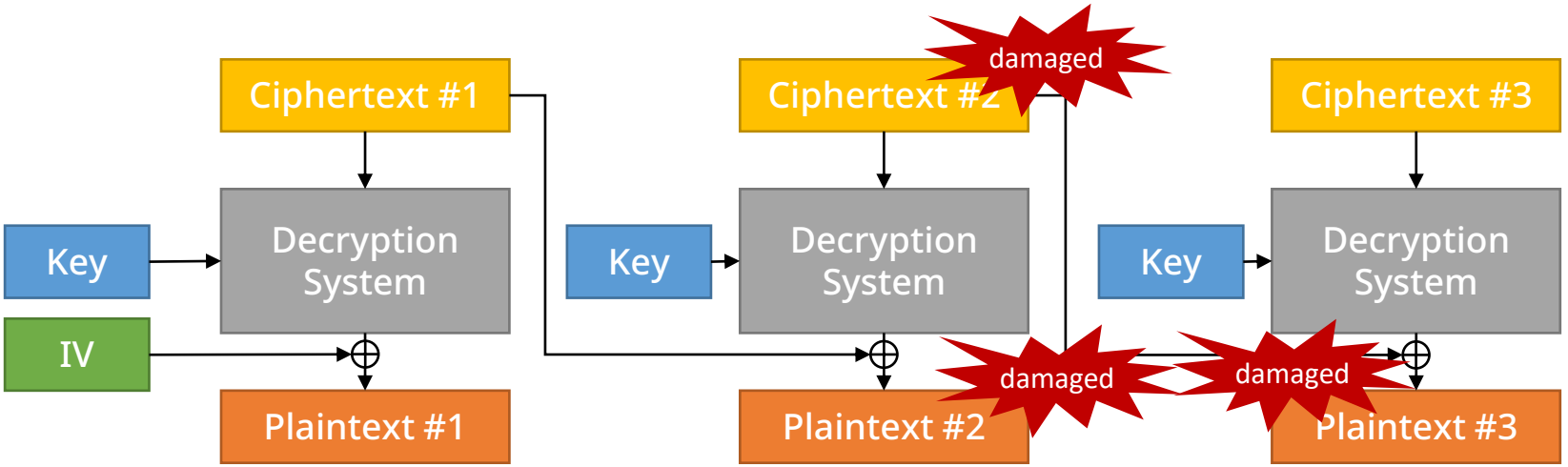
- Cipher-Block Chaining (CBC)
  - XOR the Initialization Vector (IV) in the first block and the ciphertext in the next block.
    - IVs can be predefined and kept secret, or they can be made public
      - However, the IV must not be modulated, and modulation will change the bit values in the first block.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

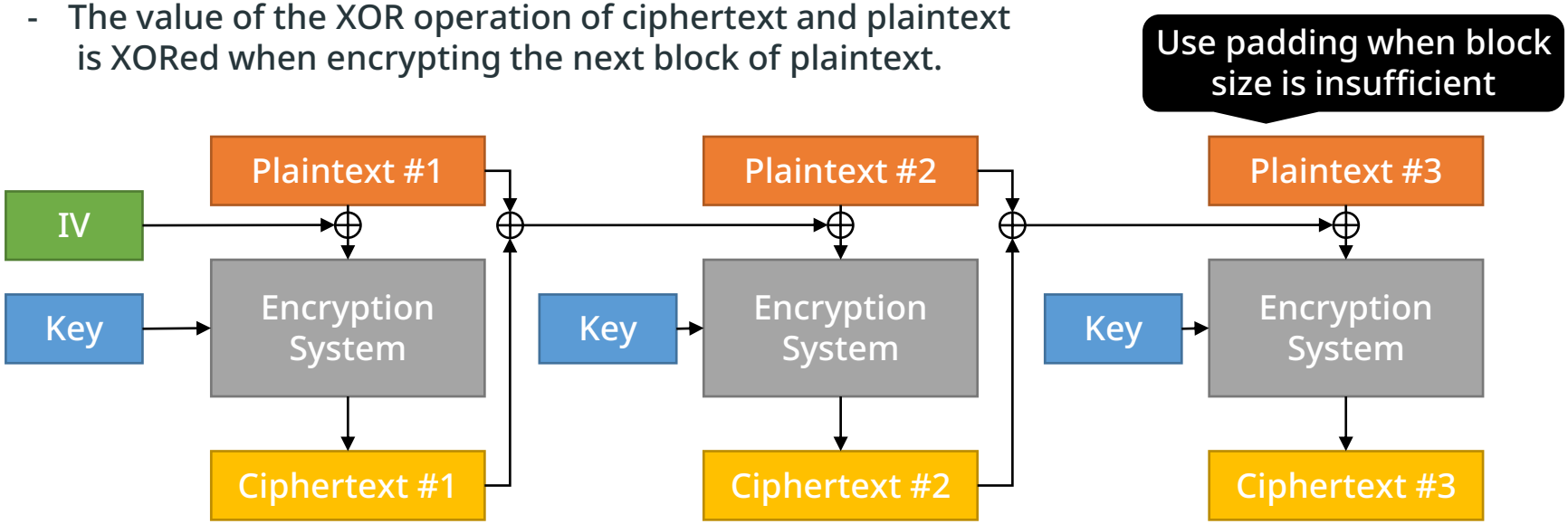
- Cipher-Block Chaining (CBC)
  - Decryption is done in reverse order because it uses a symmetric key.
  - Error propagation : an error propagates to the corresponding block in the broken ciphertext and the even plaintext of the next block.
    - However, if Ciphertext #2 is partially damaged, Plaintext #2 will be completely damaged, but Plaintext #3, which performs the XOR operation, will be partially damaged.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

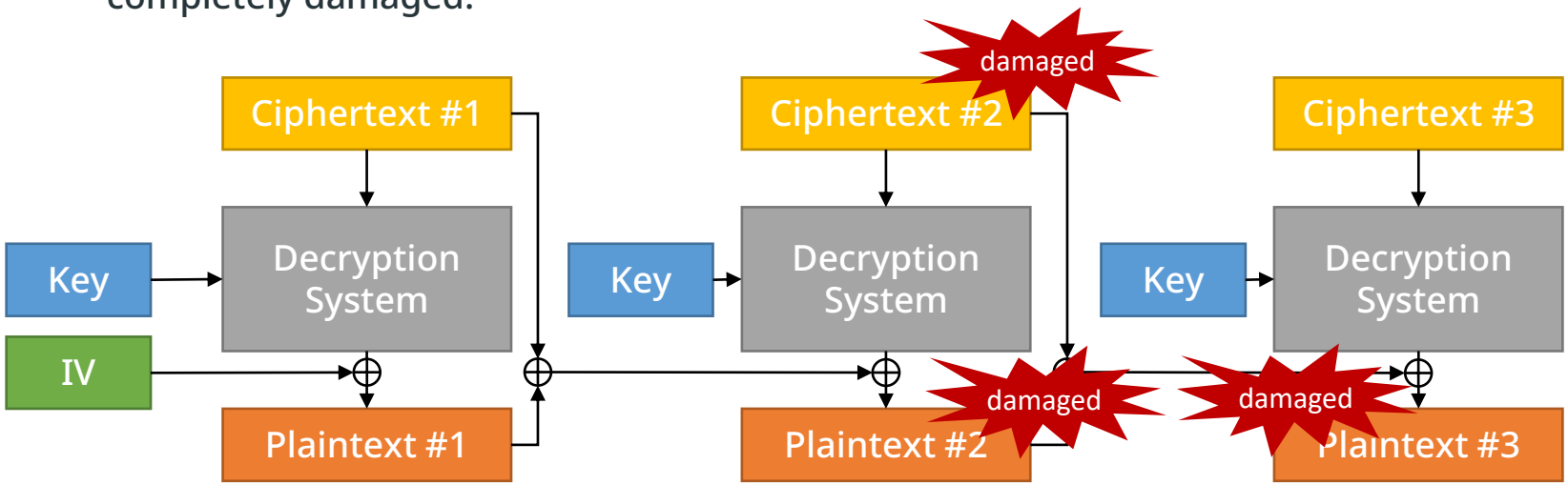
- Propagating Cipher Block Chaining (PCBC)
  - The method of using the initial vector for the first block is the same as in CBC mode.
  - Blocks from the second on are used to create the next ciphertext, which includes not only ciphertext but also plaintext.
  - The value of the XOR operation of ciphertext and plaintext is XORed when encrypting the next block of plaintext.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

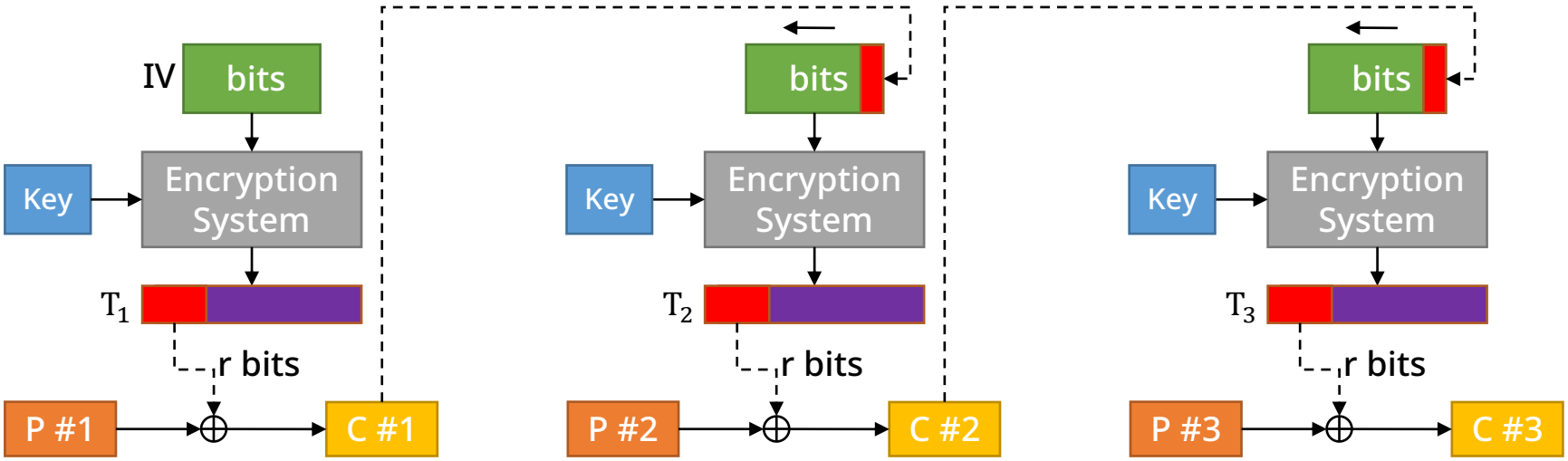
- Propagating Cipher Block Chaining (PCBC)
  - Decryption is done in reverse order because it uses a symmetric key.
  - Error propagation : an error propagates to the corresponding block in the broken ciphertext and the even plaintext of the next block.
    - However, even if Ciphertext #2 is partially damaged, Plaintext #2 and Plaintext #3 will be completely damaged.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

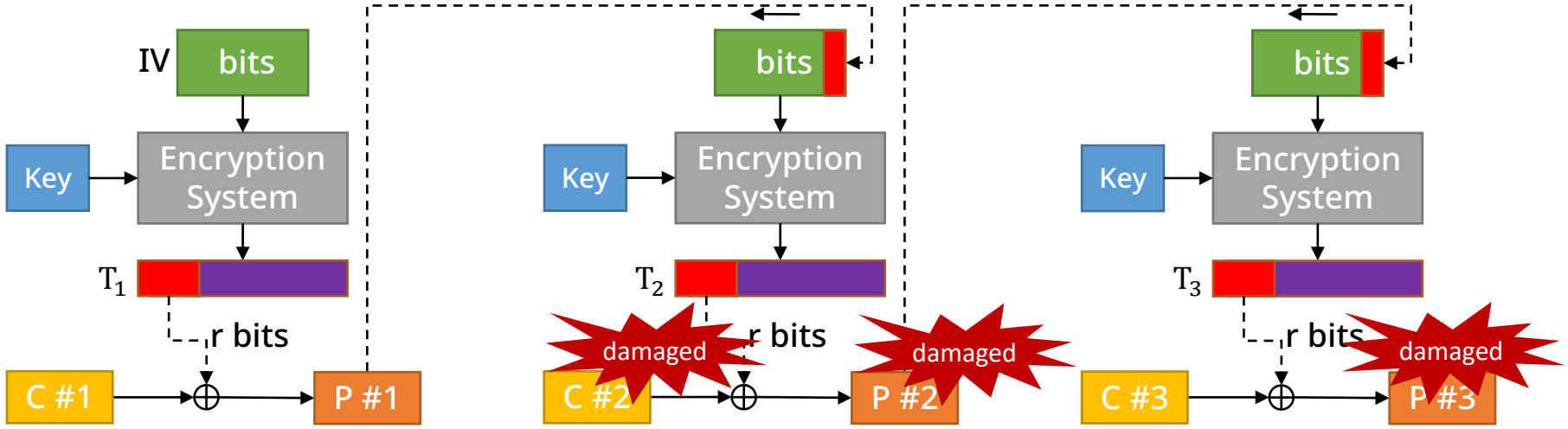
- Cipher Feedback (CFB)
  - CBC variant that classifies block ciphers as self-synchronous stream ciphers
  - Characterized by feeding the generated cipher into the next cryptosystem
    - Consider input to a cryptosystem as feedback (cipher feedback from feeding back a cipher)
  - Encrypted use of initialization vectors (IVs)



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

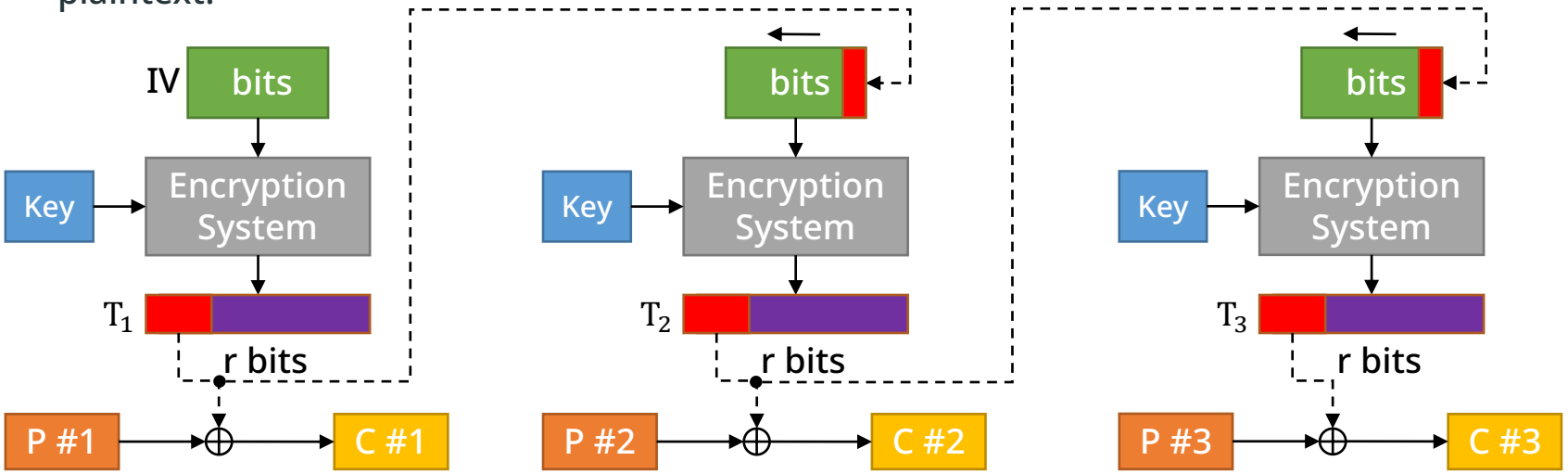
- Cipher Feedback (CFB)
  - Decryption is done in reverse order because it uses a symmetric key.
  - Error propagation : an error propagates to the corresponding block in the broken ciphertext and the even plaintext of the next block.
    - Yet, if part of Ciphertext #2 is damaged, it will perform an XOR operation with Plaintext #2, resulting in partial damaged of Plaintext #3.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

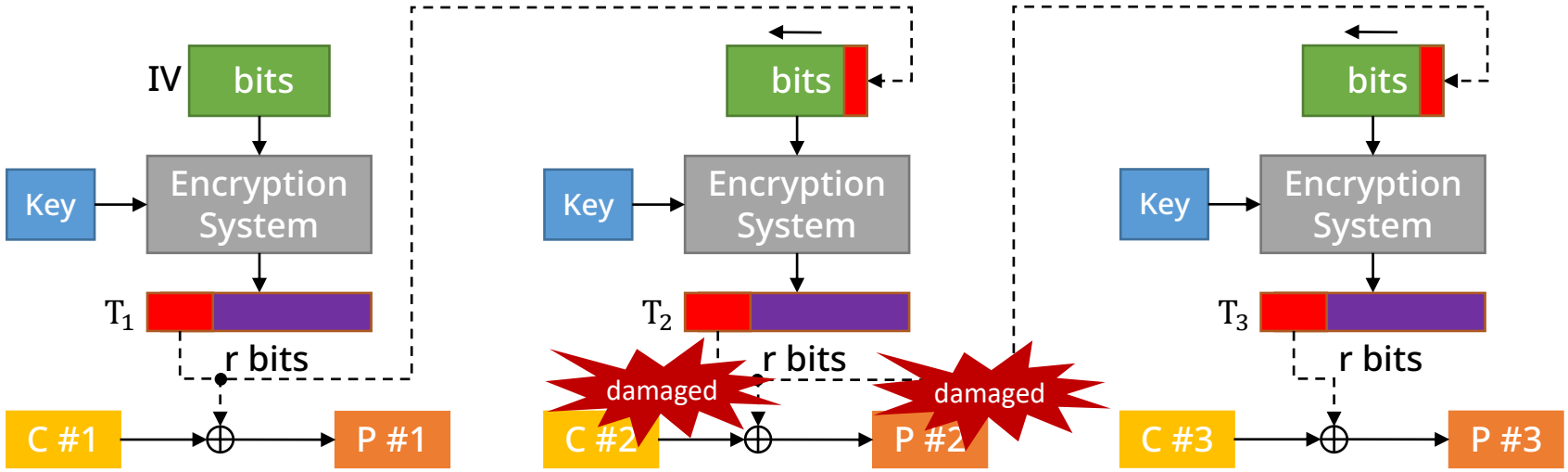
- Output Feedback (OFB)
  - Similar to CFB mode, but different input to the cryptographic algorithm
    - Consider the input to a cryptosystem as feedback (output feedback from feeding the output)
  - Use the output of the cryptosystem to generate another block cipher before XORing it with plaintext.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

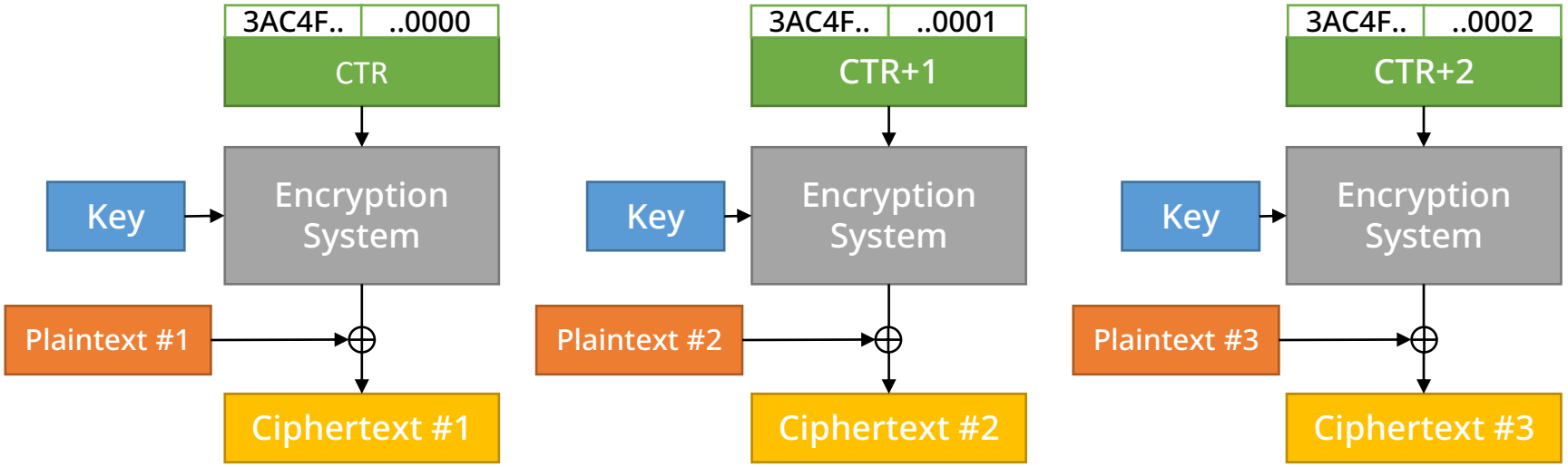
- Output Feedback (OFB)
  - Decryption is done in reverse order because it uses a symmetric key.
  - Error propagation: an error propagates to a block that is related to its own block, because each block operates independently.
    - However, if Ciphertext #2 is partially damaged, Plaintext #2 will also be partially damaged.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

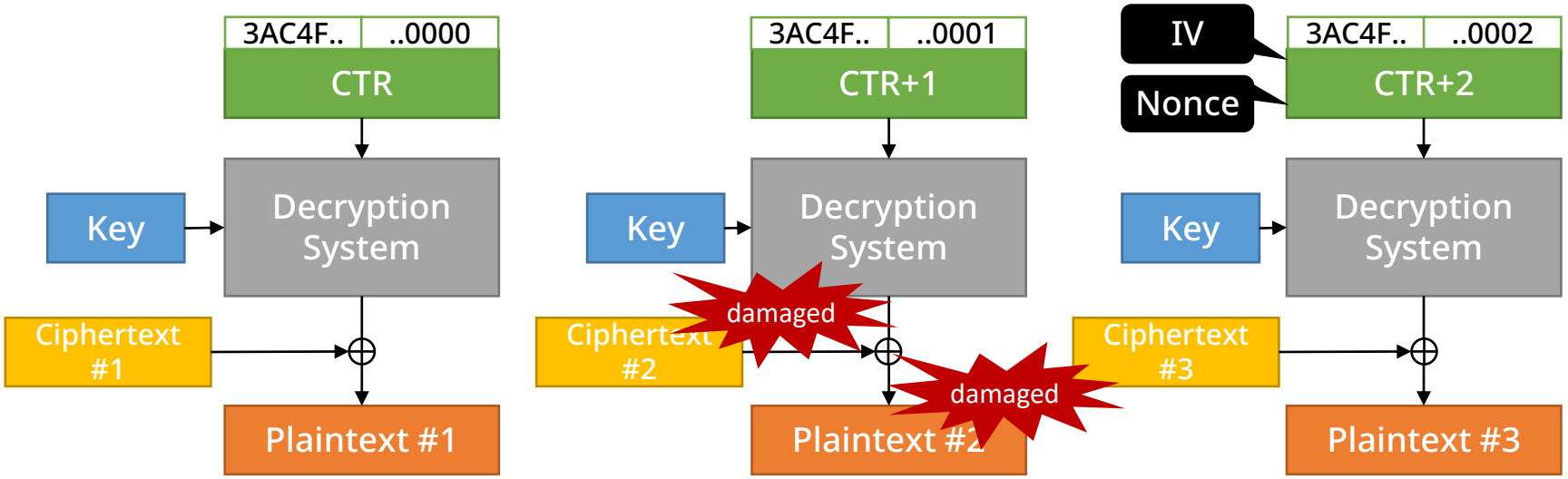
- CTR (Counter)
  - Applying the CTR operation mode to a block cipher algorithm converts it to a stream cipher.
  - Proposed by Diffie and Hellman in 1979
  - Encrypt a counter with IV incremented by 1.



Block ciphers : operation modes

A block cipher is a method of encrypting a long plaintext by dividing it into blocks of a certain length and encrypting them block by block. The problem of how to apply the cipher when the length of the plaintext is greater than the block size gives rise to various block operating modes.

- CTR (Counter)
  - A combination of a random value IV that can only be used once and a nonce used as a counting
  - Error propagation: an error propagates to a block that is related to its own block, because each block operates independently.
    - Yet, if Ciphertext #2 is partially damaged, Plaintext #2 will also be partially damaged.



Block ciphers : operation modes

- Comparing block operation modes

Mode	Advantage	Disadvantage	Application	Remark
ECB	• Simple and fast processing • Enables parallel encryption/decryption • Resistant to error propagation	• Plaintexts produce identical ciphertexts. • Can manipulate ciphertext	• Encrypted transmission of short phrases, such as encryption keys	Not recommended for use
CBC	• Repetition in plaintext is not reflected in ciphertext. • Only decryption can be parallelized. • Can decrypt random blocks of ciphertext	• Encryption cannot be parallelized. • Vulnerable to error propagation	• Universal block cipher • Authentication	Recommended
CFB	• No padding required • Only decryption can be parallelized. • Can decrypt random blocks of ciphertext	• Encryption cannot be parallelized. • Replaying attacks is possible • Vulnerable to error propagation	• Universal block cipher • Authentication	Replaced to CTR

Block ciphers : operation modes

- Comparing block operation modes

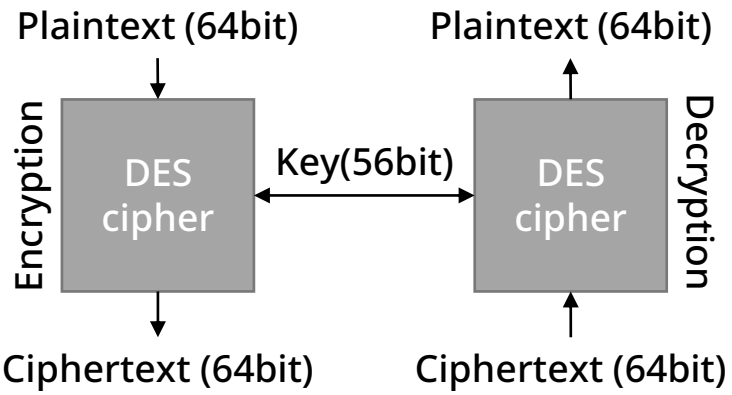
Mode	Advantage	Disadvantage	Application	Remark
OFB	• No padding required • Can be pre-configured for encryption/decryption • Encryption/decryption are identical. • Resistant to error propagation	• Unable to parallelize • If an active attacker bit-reverses a block of ciphertext, the corresponding plaintext block is bit-reversed.	• Transmitting noisy streams (e.g., satellite communications)	Replaced to CTR
CTR	• No padding required • Encryption/decryption can be prepared in advance. • Encryption/decryption are identical. • Resistant to error propagation • Enables parallel encryption/decryption	• If an active attacker bit-reverses a block of ciphertext, the corresponding plaintext block is bit-reversed.	• Universal block-oriented transport • High-speed encryption processing	Recommended

Block ciphers : DES, 3DES, AES

DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years. However, it has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, 3-DES, which uses DES three times, is recommended.

- Lucifer / DES
  - Developed by IBM
    - Symmetric-key cipher, a modification of the Lucifer cipher system
  - Adopted as a US federal standard in 1977 (published as a draft in 1975)
    - Used as the standard until 1997
  - Use 64-bit blocks, 56-bit keys
    - Split and use a 64-bit block in 32 bits
    - 8 bits of the 64-bit key are used for parity checking
      - Use 56 bits to generate the 48-bit key in actual use
      - 48-bit keys are called round keys
  - Provided a starting point for block cipher research



Block ciphers : DES, 3DES, AES

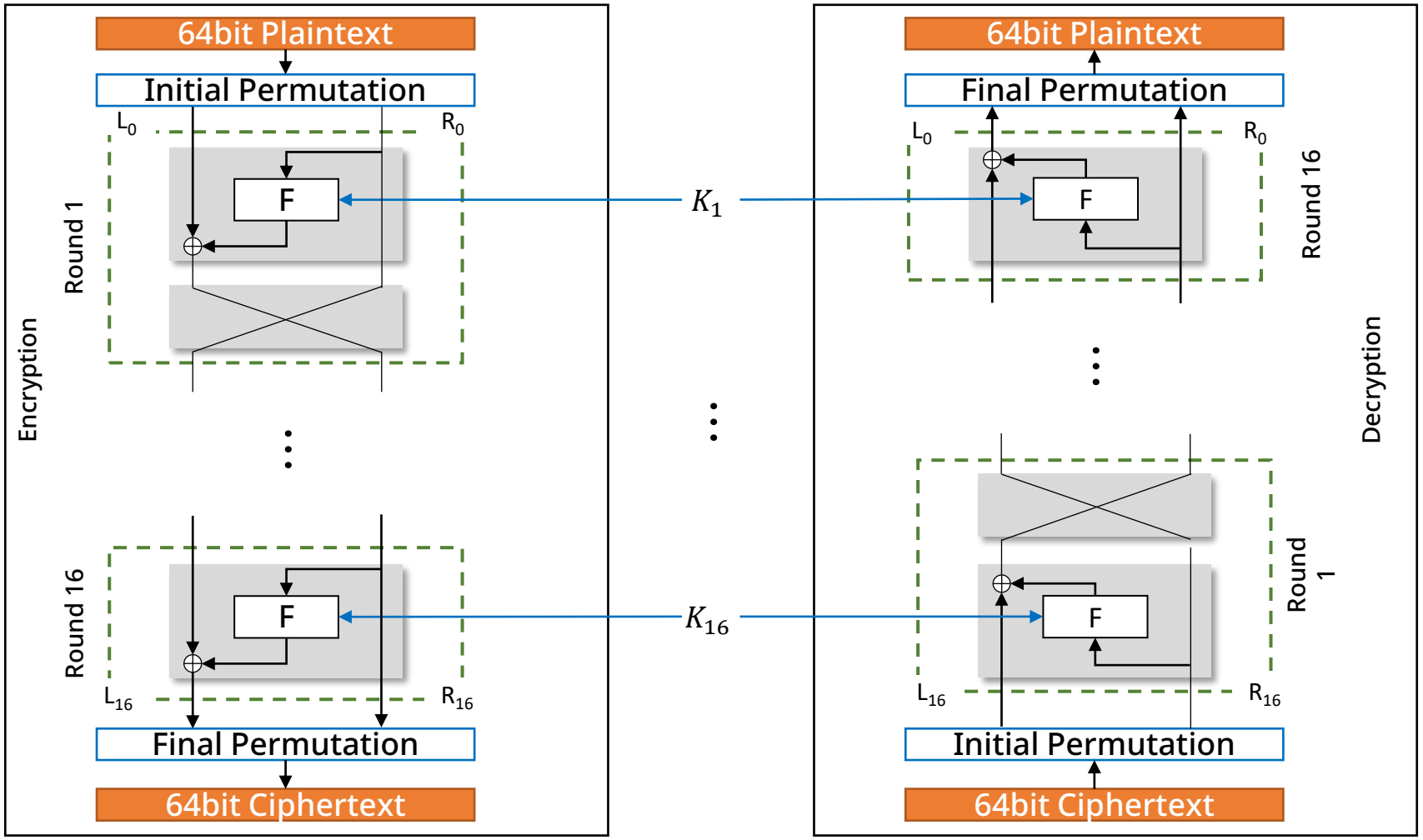
DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years. However, it has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, 3-DES, which uses DES three times, is recommended.

- Lucifer / DES
  - Since DES was created, there has been a security debate about two things.
    - The key size is too small at 56 bits, which is a security risk.
    - There has been a debate over the existence of trapdoors in S-boxes.
      - Trapdoor - an intentional implementation of a feature that allows users to look through plaintext
  - Differential cryptanalysis was published in 1990, proving that attacks are possible.
    - S-boxes were configured for differential attack at the time of DES design.
  - Proposed 56-bit key exhaustive enumeration research by RSA company
    - Decryption engine discovered in 1998.
  - Uses triple DES (3-DES) with extended key length for security
    - Promoted gradual migration to AES ciphers

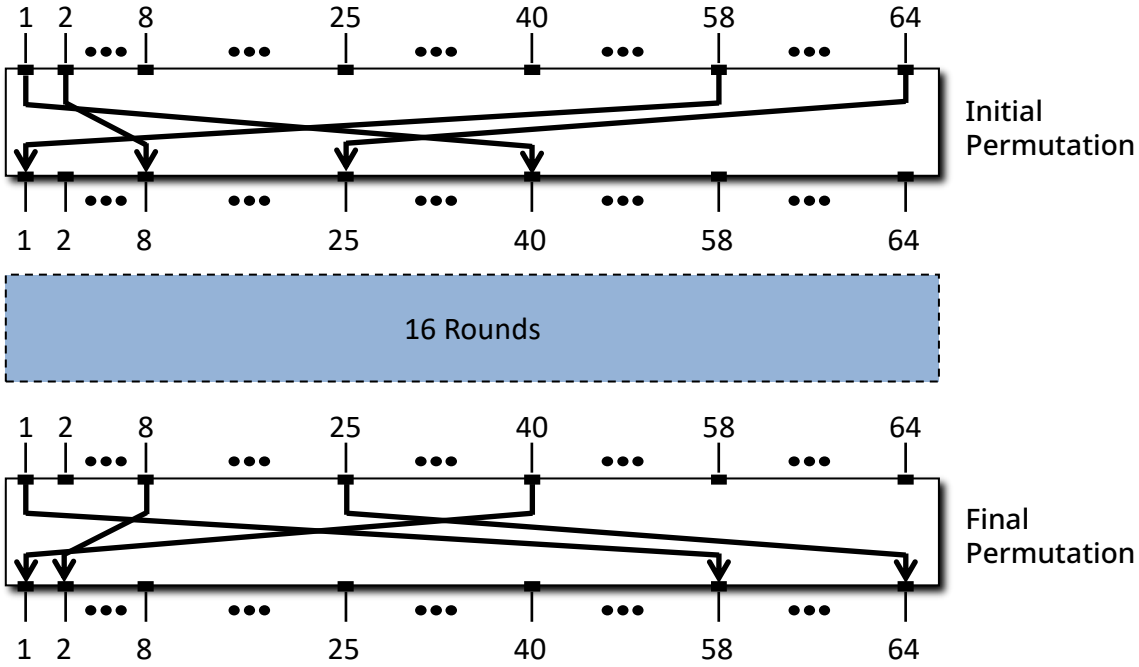
Block ciphers : DES, 3DES, AES

- DES overall structure



Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Initial and final substitutions
    - Each substitution takes 64 bits of input and rearranges them according to predefined rules.
    - Has 64 input ports and corresponding output ports
    - Initial and final substitutions are inversely related to each other.
    - Keyless simple substitution



Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Initial and final substitutions
    - Initial/final substitution table

Initial Permutation							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Final Permutation							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

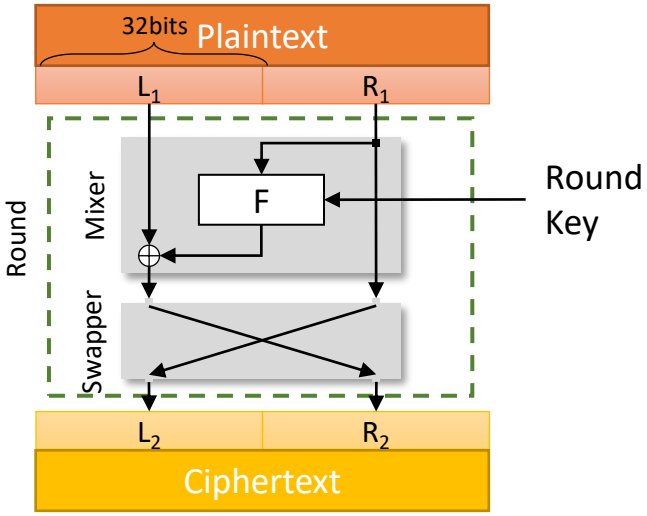
      - The 1st input of the initial substitution is the 40th output, and the 58th input of the final substitution is the 55th output.
    - The initial substitution and the final substitution are fixed functions that are independent of the value of the key.
    - It is not clear why these two substitutions are included in DES, and the design logic is not publicly available.

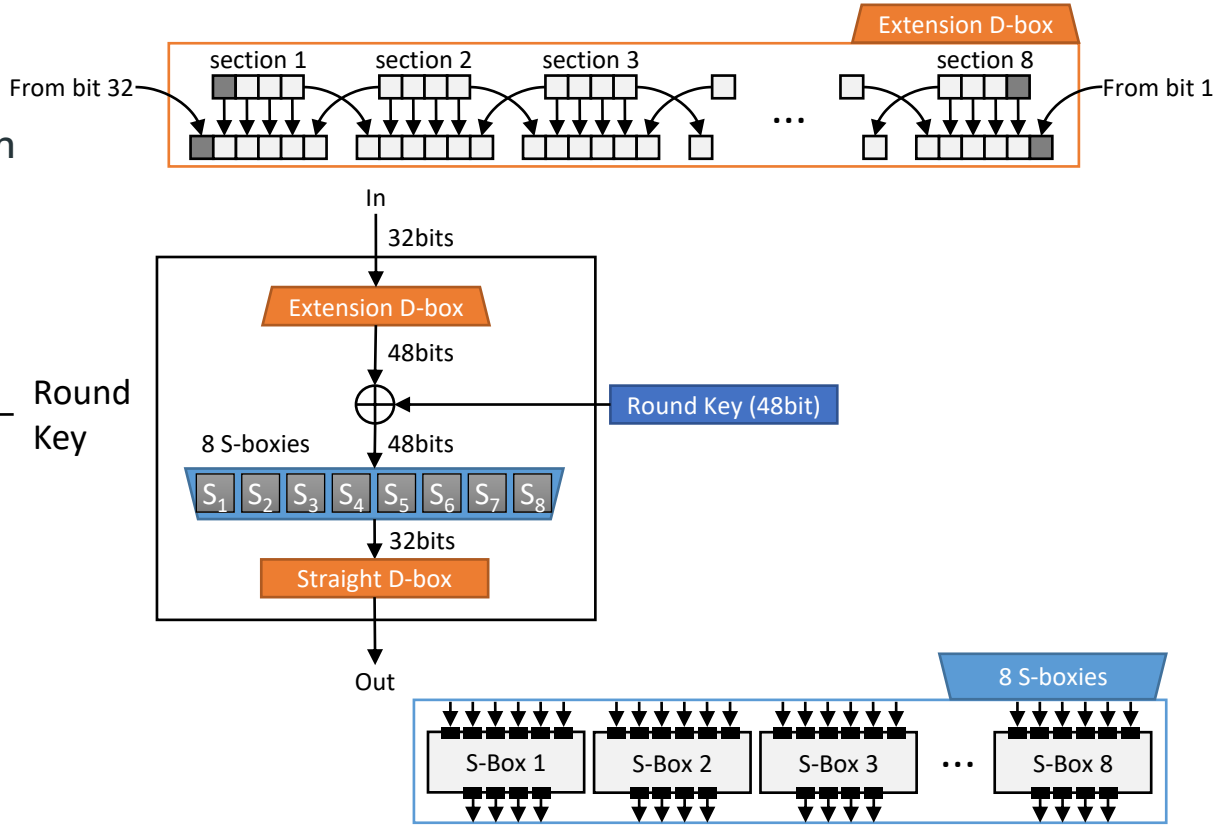
Block ciphers : DES, 3DES, AES

DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Feistel structure
    - Principle of the F function

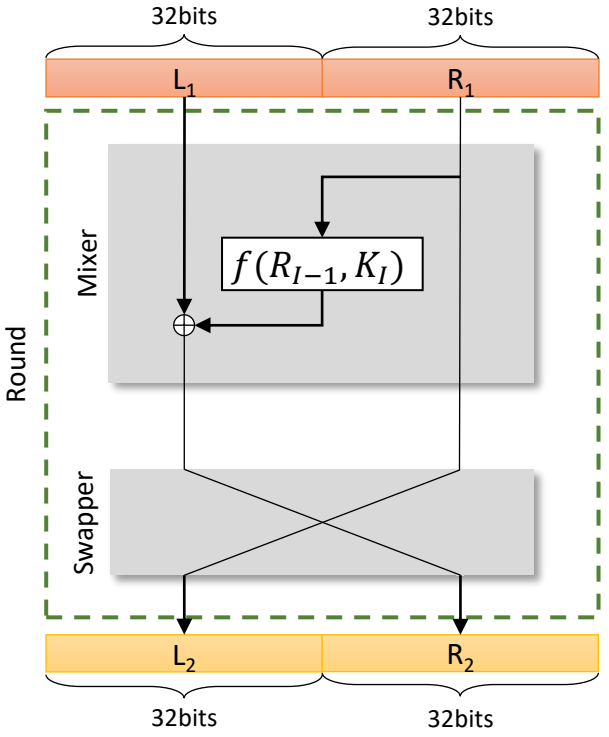




Block ciphers : DES, 3DES, AES

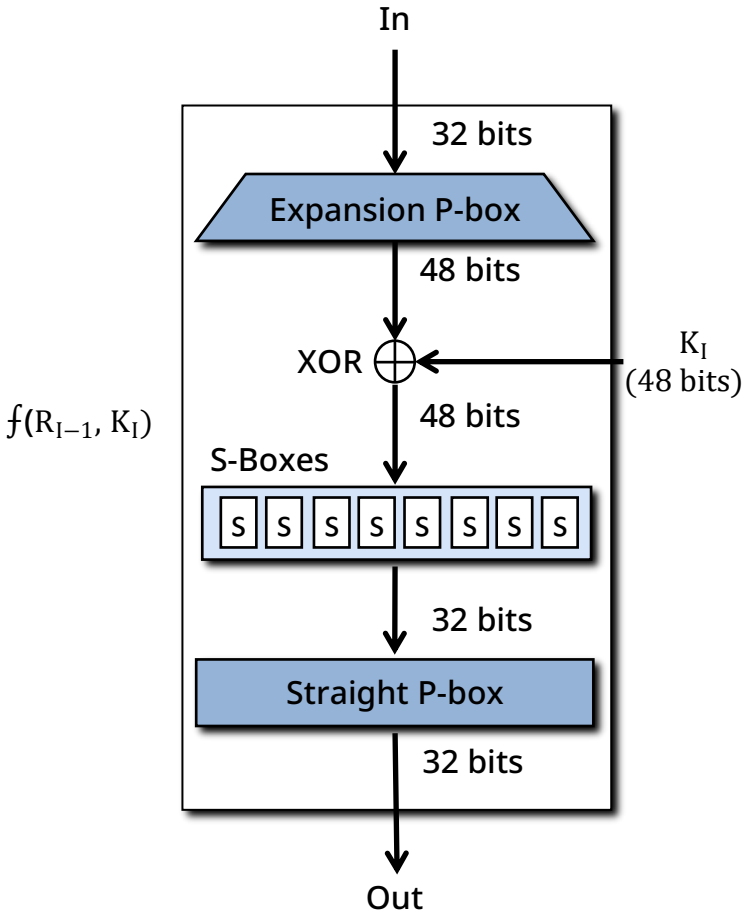
The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - DES uses 16 rounds.
    - Each round is a Feistel cipher.
    - Each round contains two cipher elements.
      - Mixer
        - 32 bits on the right are XORed with the key.
      - Swapper
        - The left 32 bits swap places with the right 32 bits.



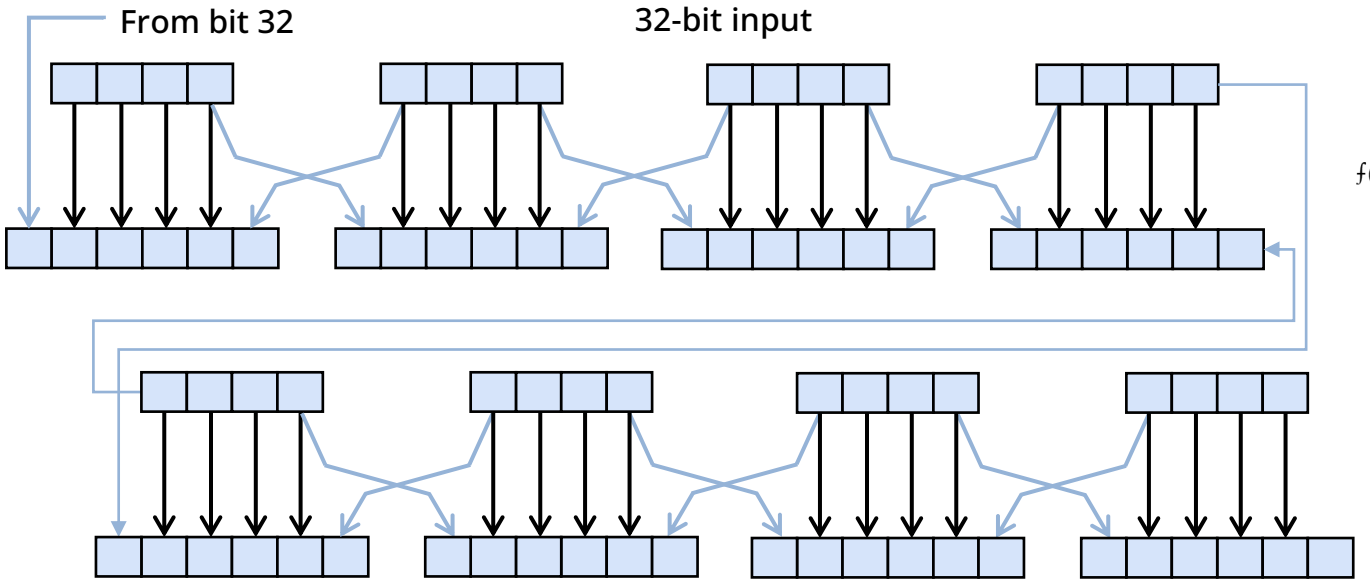
Block ciphers : DES, 3DES, AES

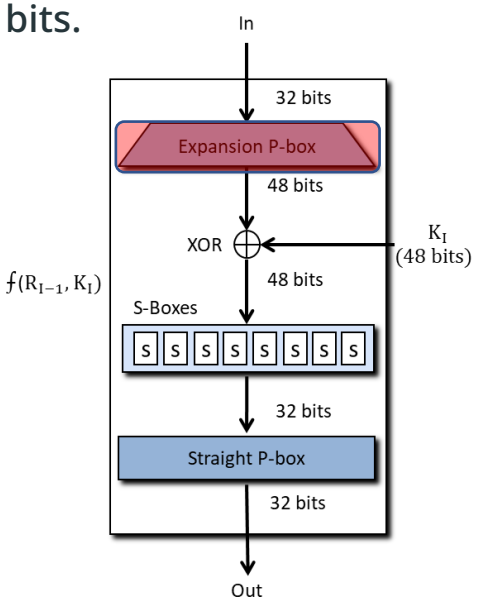
- Lucifer / DES
  - Apply a 48-bit key to the 32 bits on the right to yield a 32-bit output value
  - Organization of DES functions
    - Expansion P-box
    - Key XOR
    - 8 S-boxes
    - Simple P-box



Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Expansion P-box
    - The input value, which is 32 bits, but because the key is 48 bits, is expanded to 48 bits.
    - The number of output ports is 48, but their value can be any value from 1 to 32.
    - Some bits of the input value affect the value of one or more output bits.





Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Expansion P-box table

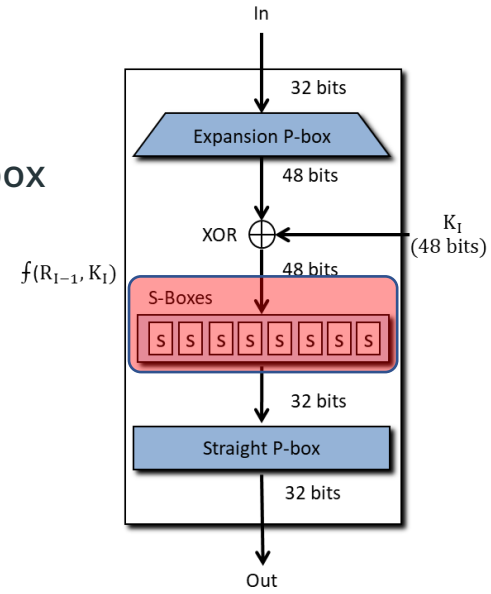
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

  - XOR
    - XOR round-keys with the expanded right 48 bits after expansion and substitution.
    - Round keys are used in this operation.

Block ciphers : DES, 3DES, AES

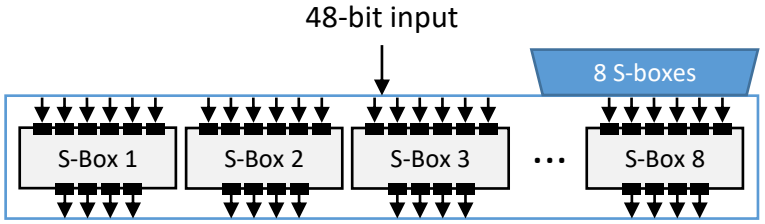
The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - S-box calculations
    - Perform 1 S-box operation of **110110** - different table values per box
      - Find rows : extract first 1 bit (1) and last bit (0) -> $10_{(2)} = 2$
      - Find columns : convert the remaining 4 bits to decimal -> $1011_{(2)} = 11$
  - 8 S-boxes each have a different table.



S-Box 1 table

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13



Block ciphers : DES, 3DES, AES

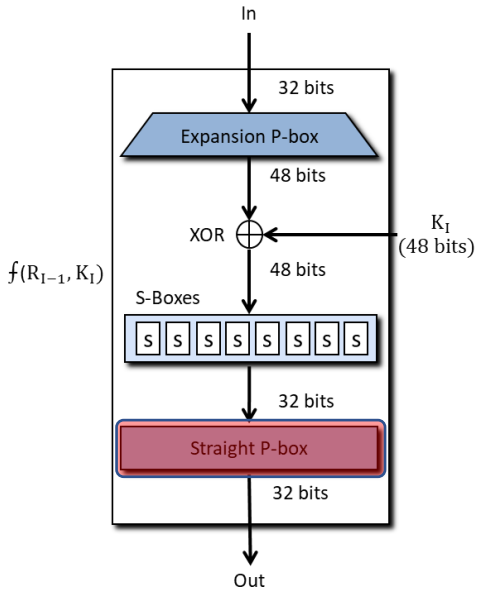
DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Straight P-box (simple substitution)
    - Simple substitution table

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

    - Last operation in DES
    - Has 32 bits of input and 32 bits of output
    - Follows the same rules as the previous initial/final substitution table

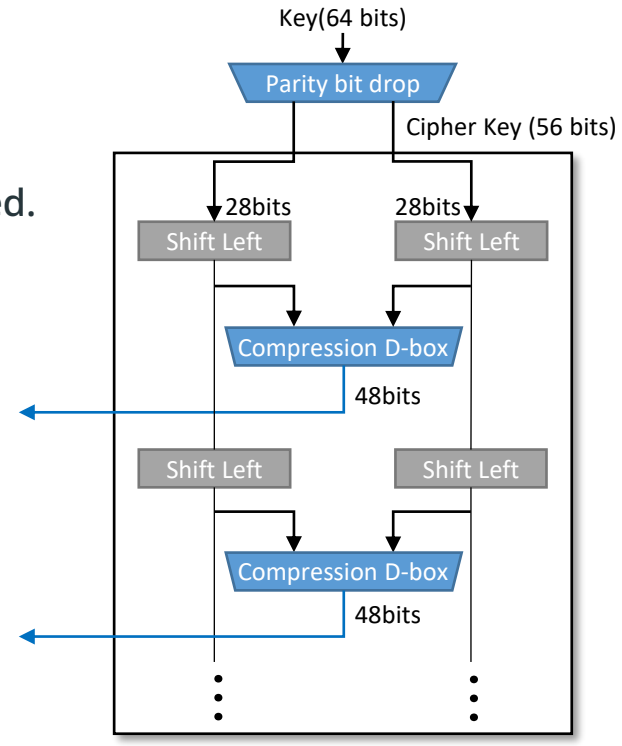


Block ciphers : DES, 3DES, AES

DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Key generation algorithm
    - Round key generator generates 16 48-bit round keys from a 56-bit encryption key.
    - 64 bits become 56 bits when the first 8-bit parity is removed.
    - In DES, rounds 1, 2, 9, and 16 proceed with a 1-bit shift left
    - The other rounds proceed with a 2-bit shift left.



Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Key generation algorithm
    - The pre-processing before key expansion is a shrink substitution that removes the parity bit.
    - Drop parity bits (8, 16, ..., 64) in a 64-bit key.
    - Remove parity and replace the remaining bits.
    - Table with parity bits removed

57	49	41	33	25	17	9	1
58	50	42	34	26	18	10	2
59	51	43	35	27	19	11	3
60	52	44	36	63	55	46	39
31	23	15	7	62	54	46	38
30	22	14	6	61	53	45	37
29	21	13	5	28	20	12	4

Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Key generation algorithm
    - Cyclic left move
      - Key is split into two parts of 28 bits each after simple substitution.
      - Each part is cyclically shifted 1 or 2 bits to the left.
        - Rounds 1, 2, 9, and 16 have a cyclic shift bit amount of 1 bit, and the rest have 2 bits.
      - The divided parts are combined to form 56 bits.
      - It is unclear why only a fraction of the shifted bits are 1 bit.
      - The cyclic shift bit amount

Rounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Block ciphers : DES, 3DES, AES

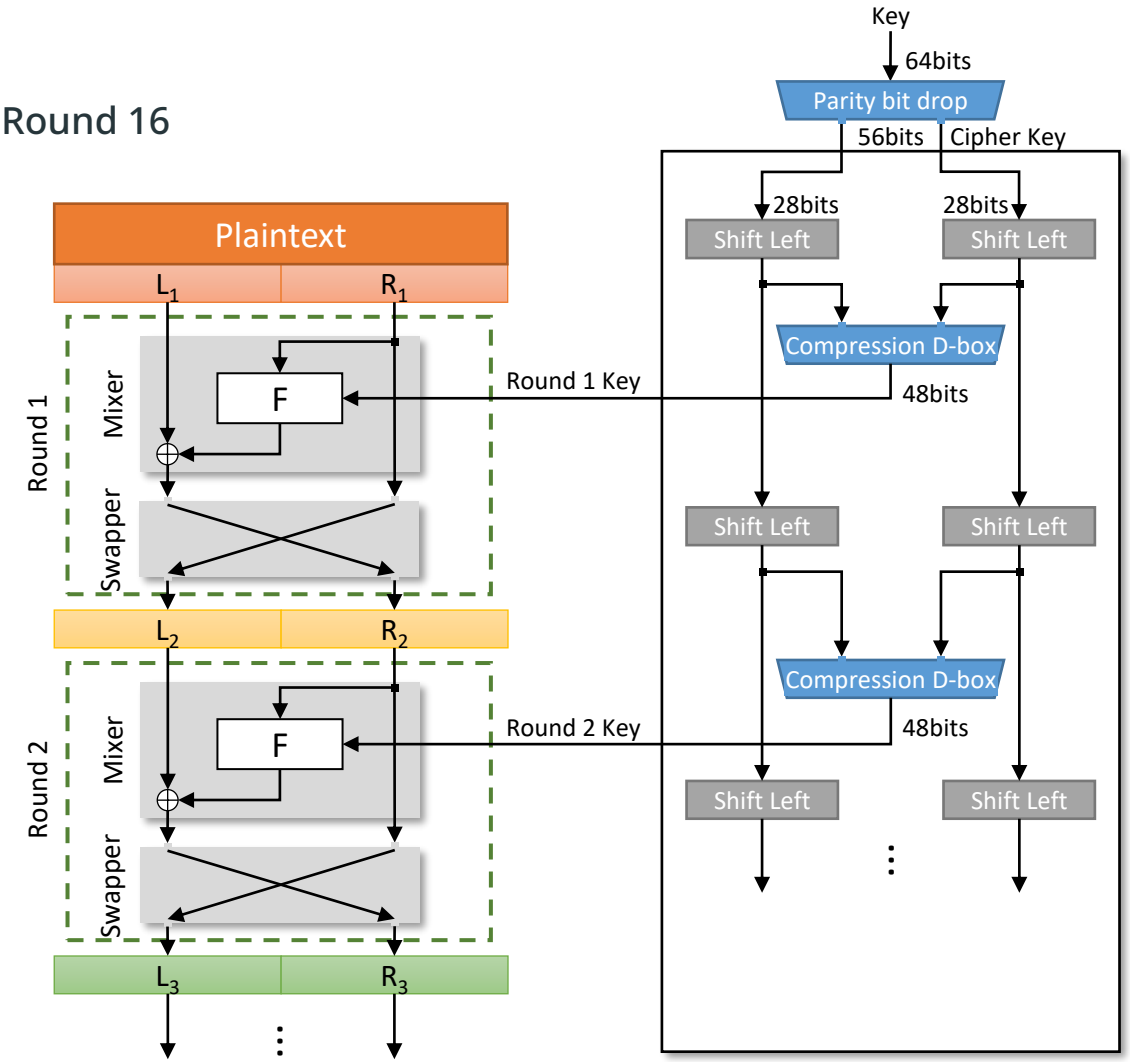
The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Key generation algorithm
    - Collapsed substitution
      - Used to convert 56 bits to 48 bits
      - 48-bit output value is used as round key for one round.

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Repeat until Round 16



Block ciphers : DES, 3DES, AES

DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Properties
    - No statistical correlation exists between plaintext, key, and ciphertext
    - All bits of plaintext and key participate in determining every bit of ciphertext
    - Small changes in plaintext or key cause large changes in ciphertext
      - Avalanche effect - also known as the landslide effect
    - Completeness effect
      - Each bit of the ciphertext means that it needs to rely on many bits of the plaintext.
      - Diffusion and chaos caused by P-boxes and S-boxes in DES show very strong completeness effects.

Block ciphers : DES, 3DES, AES

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Properties
    - Complementation property
      - Let C be the ciphertext corresponding to a plaintext P and a key K. If P and K are complementary, then the corresponding ciphertext C is also complementary.
        - $\bar{C} = E_{K(P)}$, $\bar{C} = E_{\bar{K}(\bar{P})}$
        - Since the complementation property is XORed with the key and plaintext in the rounding function, if both the key and plaintext are complementary, they are canceled out by the XOR operation and output the same value as if they were not complementary in the first place.
        - The complementation property can optimize the operation of 2^{55} exhaustive key searches using two known plaintexts (known-plaintext attack).

Block ciphers : DES, 3DES, AES

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Properties
    - Weak keys, semi-weak keys, and possible weak keys
      - Weak keys
        - Keys in a cipher that can be easily decrypted by certain operations.
        - The type and nature of weak keys depend on the structure of the cipher, and if a weak key exists in the cipher, it should be avoided.
      - Weak keys in DES
        - Assuming that the 16 round keys are K1, K2, ..., K16, where K1 = K16, K2 = K15 , ..., K8 = K9 , the encryption process and decryption process match
        - $E_K\left(E_K(P)\right) = P$

Block ciphers : DES, 3DES, AES

DES block cipher algorithm

The DES block cipher algorithm was adopted as a standard in 1977 and has been in use for more than 20 years, but has been gradually replaced by the AES block cipher algorithm due to bit size limitations caused by advances in computer performance. As a workaround, we recommend 3-DES, which uses DES three times.

- Lucifer / DES
  - Properties
    - Weak keys, semi-weak keys, and possible weak keys
      - Semi-weak keys
        - If there are weak-key-like properties between two keys K and K'.
        - DES does not become a group (if it did, it would be decrypted by a birthday attack with a computation of 2^{28})
        - $E_K\left(E_{K'}(P)\right) = P$
      - Possible weak keys
        - There are 48 possible weak keys that generate only four different round keys.

Block ciphers : DES, 3DES, AES

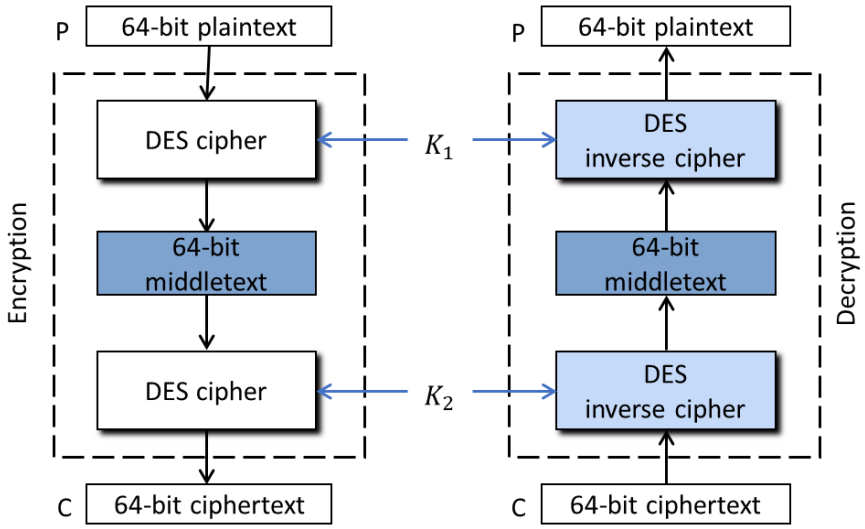
- Lucifer / DES
  - Multiple DES
    - The main criticism of DES is its short key length.
      - The first solution
        - Abandon DES and design a new algorithm. E.g., AES
      - The second solution
        - Operate multiple encryptions of DES with multiple keys
        - If DES is assumed to be a group, then using 2 keys k_1, k_2 makes no sense.
        - For DES to be a group, it must satisfy $\log_2(2^{64}!) \approx 2^{70}$, but the key length of DES is only 56 bits, which makes it impossible to be a group.
        - Since DES is not a group, it is very difficult to find one that satisfies the following:

$$E_{k_2}\left(E_{k_1}(P)\right) = E_{k_3}(P)$$

        - This means that double or triple DES can be used.

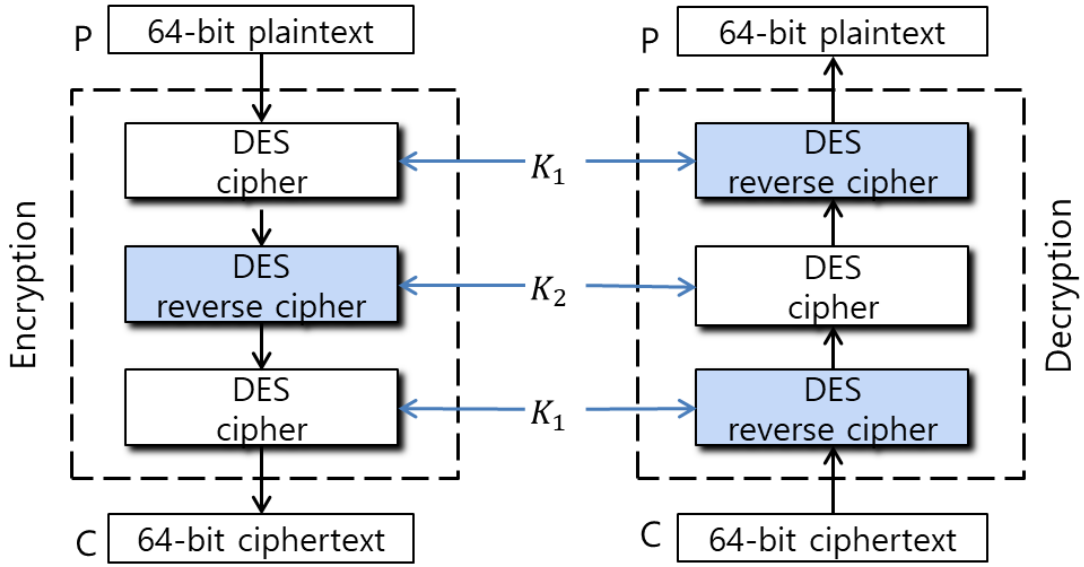
Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Double DES
    - Use two DES encryption algorithms for encryption and two decryption algorithms for decryption
    - Use different keys for each DES
    - Double the key size to 112 bits, but still vulnerable to known plaintext attacks
  - Meet-in-the-Middle Attack
    - Double DES seems to increase the number of key searches from 2^{56} to 2^{112}, but the meet-in-the-middle attack keeps it at 2^{57}, with only a slight improvement.
    - In double DES, the middle text values of the first encryption and the description match.



Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Triple DES
    - Triple DES with two keys
      - Use k_1 for the first and third and k_2 for the second
      - Use decryption algorithm in the middle of the encryption process
      - Triple DES with two keys is vulnerable to known-plaintext attacks, but stronger than double DES.



Block ciphers : DES, 3DES, AES

- Lucifer / DES
  - Triple DES
    - Triple DES with three keys
      - Because of the potential for known-plaintext attacks against two-key triple DES, some programs use three-key Triple DES.
      - Use the DES cryptographic algorithm three times for encryption and three times for decryption
      - But must be compatible with DES using the algorithm once
        - The encryption process follows the EDE (Encryption/Decryption/Encryption) sequence.
        - The decryption process follows the DED (Decryption/Encryption/Decryption) sequence.

Block ciphers : DES, 3DES, AES

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - 1997-01-02 : basic requirements for AES presented.
    - Use block cipher algorithm.
    - Key length is 128, 192, or 256 bits.
    - Block size is 128 bits.
    - Must be available for smart cards.
    - Must be patent-free (royalty-free).
  - 1997-09-12 : the US NIST officially called for AES candidates.
  - 1998-08-20 : 15 first round AES candidate algorithms announced.
  - 1999-08-09 : 5 second round AES candidate algorithms announced (Rijndael, MARRS, RC6, SERPENT, and TWOFISH).
  - 2000-10-02 : Rijndael was selected as AES.
  - 2001-12-06 : officially registered and released to the public.

Block ciphers : DES, 3DES, AES

AES block cipher algorithm

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - Features
    - Use SPN structure instead of Feistel
    - Smart card use was the requirement, but designed to include software, hardware, etc.
    - Support for block sizes of 192 and 256 bits as well as 128 bits
    - Number of rounds (performance and reliability) depends on key length.

	Key length	Block size	Number of rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

    - 128 bits = 2128 = 3.4 x 1038
      - A computer computing 2^55 keys (56 bits) per second would take 149 billion years to break Rijndael using a brute force attack.

Block ciphers : DES, 3DES, AES

AES block cipher algorithm

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - Features
    - Each round (except the last) operates uniformly with the following elements.
      - These elements are called layers.
        - SubBytes (byte-by-byte substitution using an S-box)
        - ShiftRows (a permutation that cyclically shifts the last three rows in the state)
        - MixColumns (substitution using Galois Fields, corps de Galois, GF(28) arithmetic)
        - Add round key (bit-by-bit XOR with an expanded key)
      - Rounds except the last : ByteSub → ShiftRow → MixColumn → Key XOR
      - Last round : ByteSub → ShiftRow → Key XOR

Source: https://www.lri.fr/~fmartignon/documenti/systemesecurite/5-AES.pdf

Block ciphers : DES, 3DES, AES

AES block cipher algorithm

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - SubBytes (byte-by-byte substitution using an S-box)
    - Unlike DES, AES uses the same S-box
      - Responsible for the cipher's chaotic nature
    - Use 28 as a Galois Field
      - Finite field - a set with a finite number of elements.
    - Split 8 bits into 4 bits
      - Leading 4 bits are rows, trailing 4 bits are columns
      - E.g., 0x8A → row 8 and column A → 0x7E

AES Sub Bytes Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	3C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A5	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	03	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

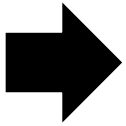
Block ciphers : DES, 3DES, AES

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - ShiftRows (a permutation that cyclically shifts the last three rows in the state)
    - Organized in a 4x4 array because the block is 128 bits.
    - Row 1 does not perform any shift operation.
    - Shift to the left by the number of rows starting with the next row.

Block

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}



Block with shifted rows

a_{00}	a_{01}	a_{02}	a_{03}
a_{11}	a_{12}	a_{13}	a_{10}
a_{22}	a_{23}	a_{20}	a_{21}
a_{33}	a_{30}	a_{31}	a_{32}

Block ciphers : DES, 3DES, AES

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - MixColumns (substitution that uses Galois Fields, corps de Galois, GF(2^8) arithmetic)
    - Responsible for spreading ciphers
    - Proceed to array operations

MixColumn

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Block

$$\times \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

MixedColumn block

$$= \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ b_{20} & b_{21} & b_{22} & b_{23} \\ b_{30} & b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Block ciphers : DES, 3DES, AES

A symmetric key, also known as a secret key, means that the encryption key is the same as the decryption key.

- Rijndael / AES
  - Add round key (bit-by-bit XOR with an expanded key)
    - Perform an XOR operation on a block of data and a key.
    - This operation is also performed on a matrix basis.

Round key

$$\begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix}$$

Block

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

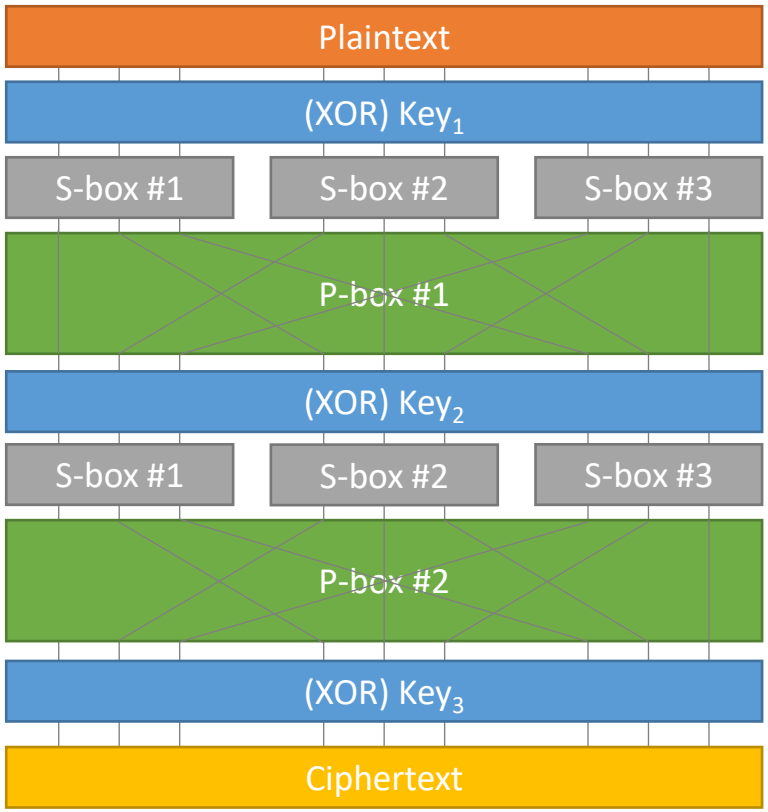
Cipher block

$$\begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix}$$

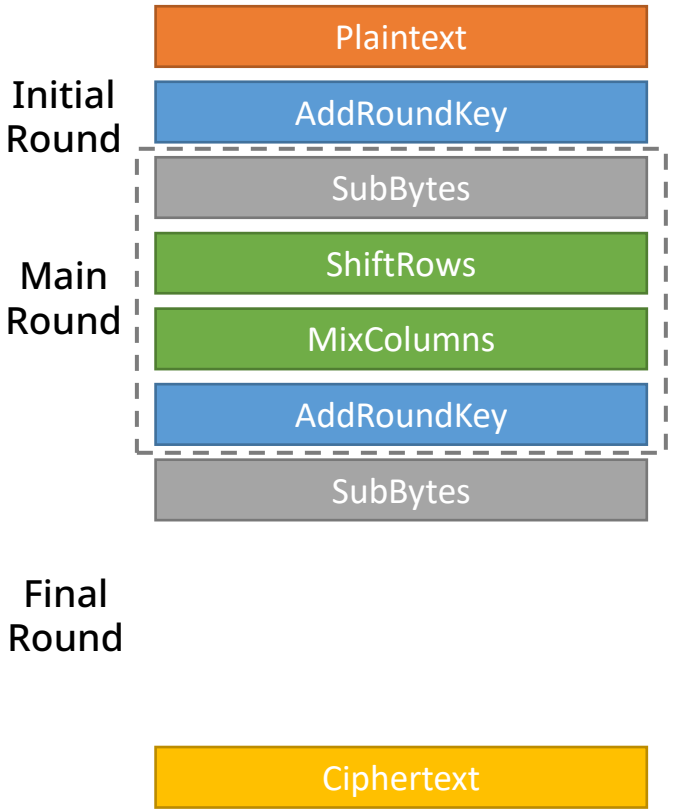
Block ciphers : DES, 3DES, AES

- Rijndael / AES

SPN block cipher structure

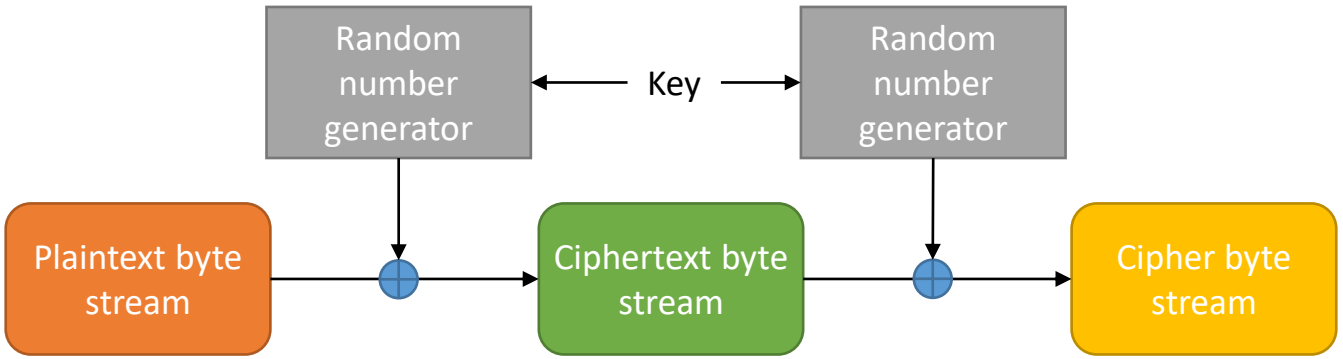


AES algorithm



Stream cipher

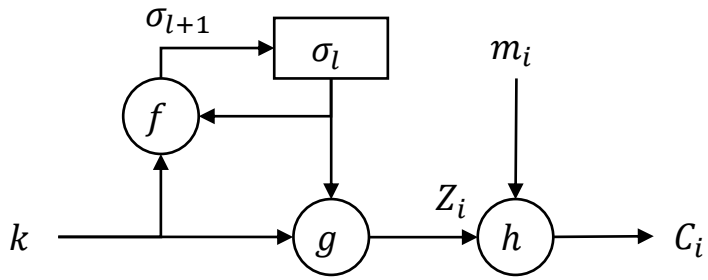
Stream ciphers generate a key stream of the same length as the plaintext, which is then combined with the plaintext and the key binary sequence in a bitwise logical exclusive-or (XOR) operation and proceed encryption.



Stream cipher

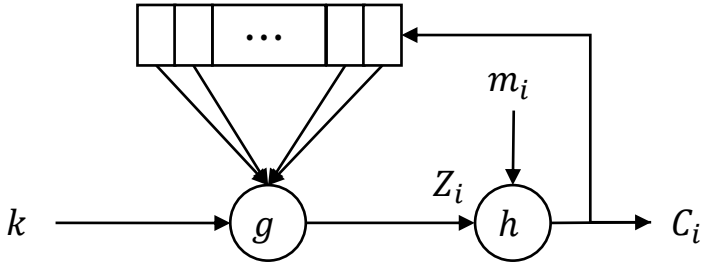
Stream ciphers generate a key stream of the same length as the plaintext, which is then combined with the plaintext and the key binary sequence in a bitwise logical exclusive-or (XOR) operation and proceed encryption.

- Synchronous



- Easy to create
- No error propagation
- Can detect insertions and deletions
- Synchronization required
- Require data authentication & integrity check
- Require strong sequence

→ Self-synchronous



→ Limited error propagation

→ Difficult to detect insertions and deletions

→ Plaintext is diffused over ciphertext

→ High resistance to eavesdropping

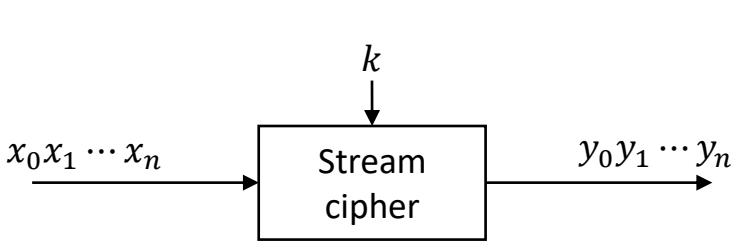
→ Difficult to create

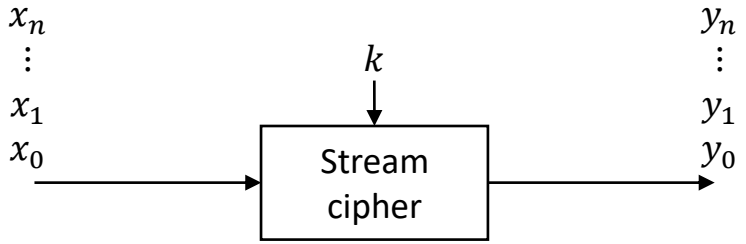
Stream cipher

Stream ciphers generate a key stream of the same length as the plaintext, which is then combined with the plaintext and the key binary sequence in a bitwise logical exclusive-or (XOR) operation and proceed encryption.

- Comparing stream ciphers with block ciphers

Stream cipher	Block cipher
Can work with small chunks of plaintext	Behave on large blocks of data
Faster than block ciphers	Slower than stream ciphers
Implement with less code	Implemented with a lot of code
Use secret key only once	Repeatedly using passkeys
Example : One time pad	Example : Data Encryption Standard (DES)
Well-known application : SSL	Well-known application : databases, files
Better suited for hardware implementation	Easy to implement in software





Stream cipher

Stream ciphers generate a key stream of the same length as the plaintext, which is then combined with the plaintext and the key binary sequence in a bitwise logical exclusive-or (XOR) operation and proceed encryption.

- Advantages of stream ciphers
  - Use a random number generator
    - Generate long-cycle binary sequences from short-length keys at low cost and high speed
  - Enable real-time encryption
    - Ideal for media and telecommunications environments
  - Manipulating one bit of the ciphertext will only affect the decryption of that bit.
    - Less vulnerable to communication errors than block ciphers
      - Block ciphers can't be fully decrypted if one bit is tampered with.
  - Allow mathematically rigorous analysis of the security of cryptosystems
  - Ideal for protecting communication data, including telecommunications

Stream cipher

Stream ciphers generate a key stream of the same length as the plaintext, which is then combined with the plaintext and the key binary sequence in a bitwise logical exclusive-or (XOR) operation and proceed encryption.

- Security requirements for stream ciphers
  - The security of a stream cipher depends on how resistant the key sequence is to various types of cryptographic attacks.
  - In general, Beker, Siegenthaler, and Golic meet the criteria listed below.
    - Period : the output key sequence must have a guaranteed minimum value for the period
    - Randomness : the output key sequence should have good randomness properties
    - Linear complexity : the output key sequence must have a large linear complexity
    - Correlation immunity : the output key sequence must have a high correlation immunity.
    - Number of key stream cycles : the output key sequence must occur in at least one key stream cycle

Stream cipher

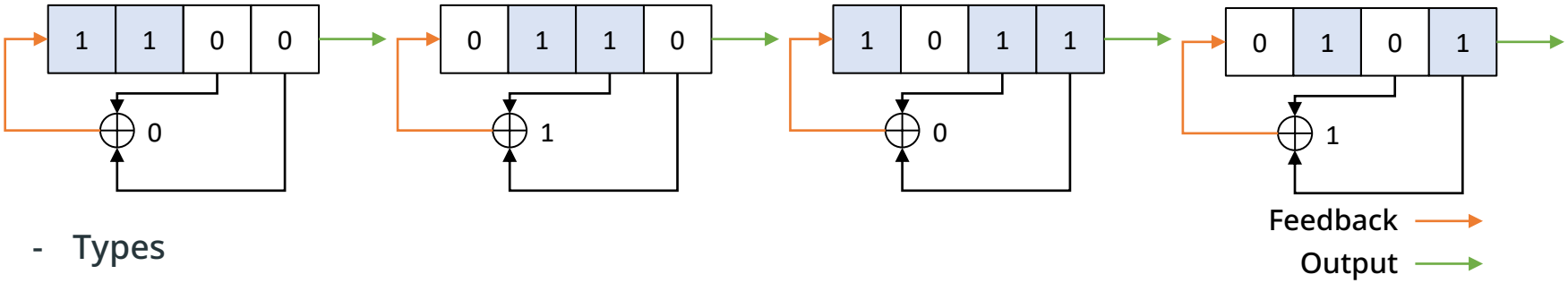
Just as there are different types of block ciphers depending on their design, LFSR is a representative design for stream ciphers.

- Linear feedback shift registers
  - Pseudorandom number generation function
    - The use of LFSRs in stream ciphers is common.
    - Random numbers generated by feedback
      - Bits that influence the feedback are called tabs.
    - Output is the bit that is discarded.
    - Represent the underlying form as a seed
  - Use linear functions
    - Repeating operations at a given interval
    - Use polynomials
    - Typically operate as a logical exclusive-or (XOR)

Stream cipher

Just as there are different types of block ciphers depending on their design, LFSR is a representative design for stream ciphers.

- Understanding how random number generation works with LFSR (4-bit)
  - The number of all cases that can be generated with 4 bits minus one case (0000) is generated ($2^n - 1$).

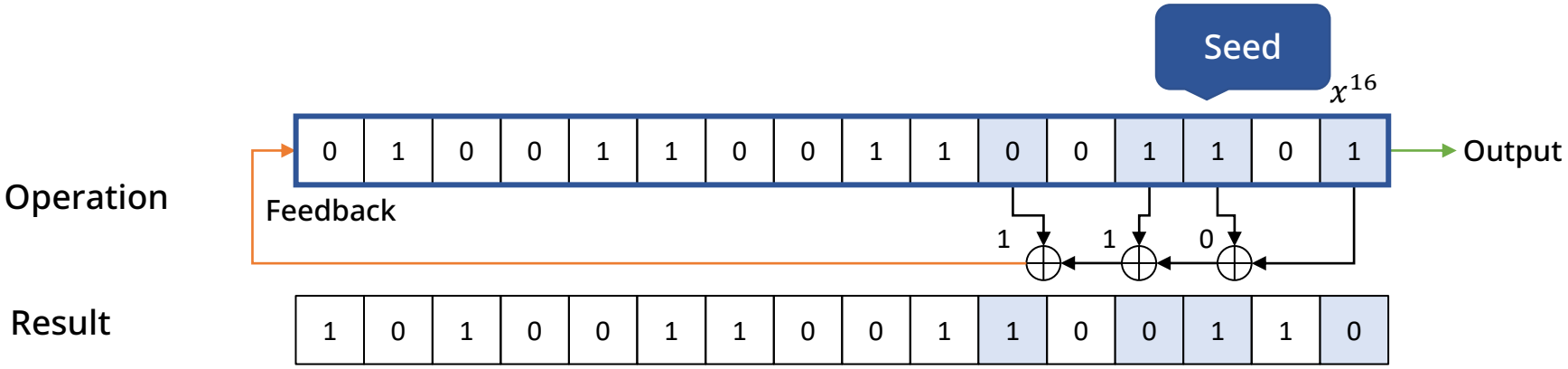


- Types
  - Fibonacci LFSR
  - Galois LFSR
  - Xorshift LFSR

Stream cipher

Just as there are different types of block ciphers depending on their design, LFSR is a representative design for stream ciphers.

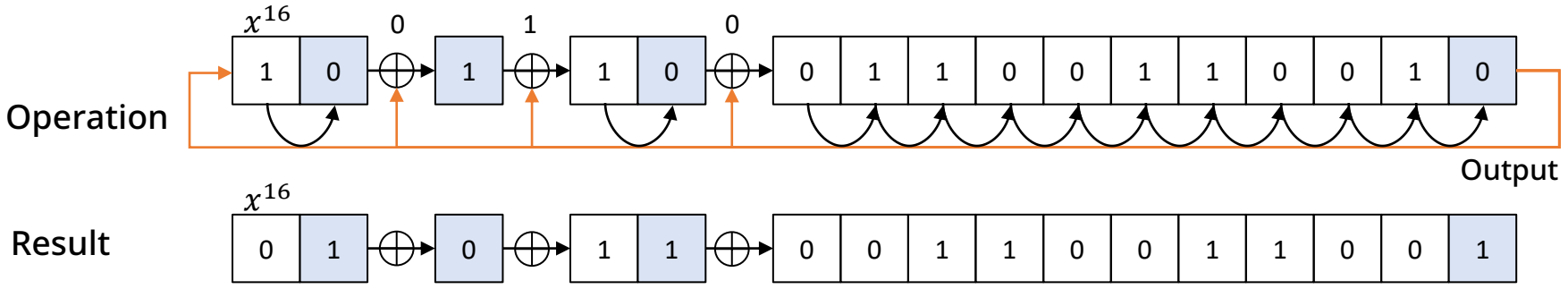
- Fibonacci LFSR
  - Also called external LFSR because the value is computed externally on the tab.
  - In its most basic form
  - E.g., as a tap, the role is responsible for bits 16, 14, 13, and 11.
    - $x^{16} + x^{14} + x^{13} + x^{11}$



Stream cipher

Just as there are different types of block ciphers depending on their design, LFSR is a representative design for stream ciphers.

- Galois LFSR
  - Also called Internal LSFR because the value is computed internally on the tab.
  - Designed to replace the existing Fibonacci LFSR – operates in reverse order.
  - E.g., as a tap, the role is responsible for bits 16, 14, 13, and 11.
    - $x^{15} + x^{14} + x^{12} + 1$



03

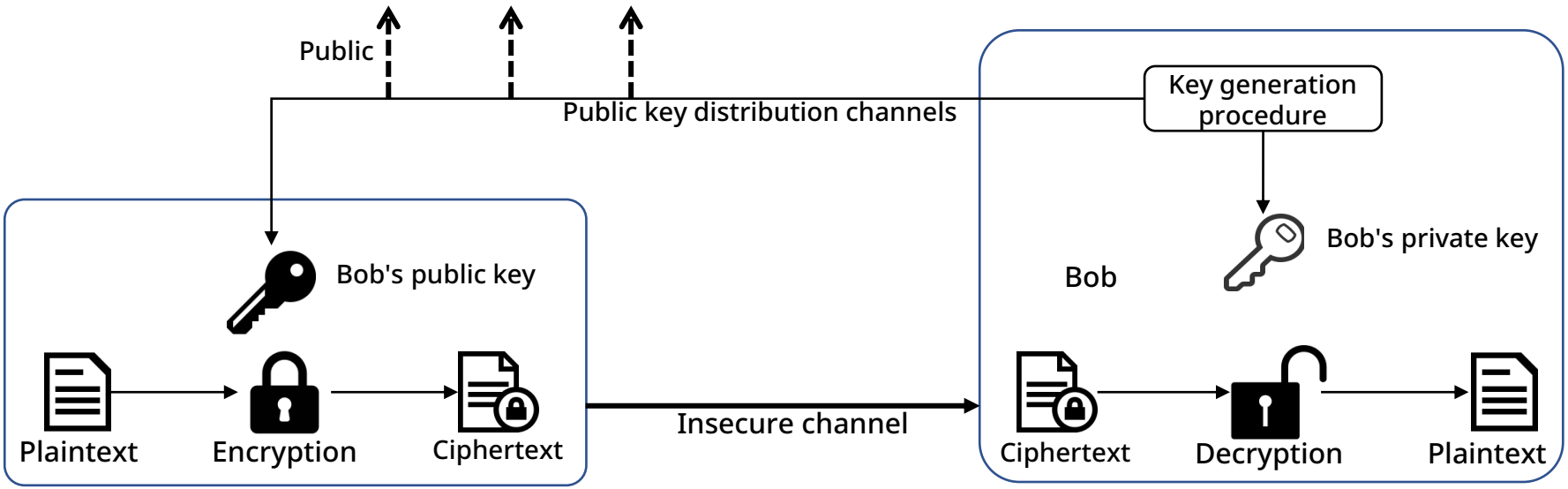
Public-key encryption

- Public-key encryption overview
- RSA
- ElGamal
- ECC

Public-key encryption overview

An asymmetric-key cryptosystem means that the keys used for encryption and decryption are different. These ciphers are created using diverse mathematical challenges.

- The encryption and decryption keys are different and is also known as public-key encryption because some keys are public.
  - Public key ≠ private key
  - Use mathematical challenges to implement cryptosystems



Public-key encryption overview

An asymmetric-key cryptosystem means that the keys used for encryption and decryption are different. These ciphers are created using diverse mathematical challenges.

- Types
  - Cryptosystems using a prime factorization problem
    - A challenge that exploits the fact that the product of two given primes is easy, but finding the two primes in the multiplied value is extremely difficult.
    - Types - RSA, Rabin
    - E.g., $89 * 97 = X(8633)$ $\quad\quad$ $8633 = X * Y$
  - Cryptosystems using a discrete algebra problem
    - Discrete algebra problem : given a finite group G and constructors g and g^x,

      find the exponential product x
    - When $y = g^x mod\ p$, it's easy to calculate y if you know g,x,p, but it's harder to find x if you know y,g,p.
    - That is, the difficulty of computing logarithms, which is the inverse of exponential computation

      ($x = log_g y$)
    - Types - ElGamal, DSA, ECC

Public-key encryption overview

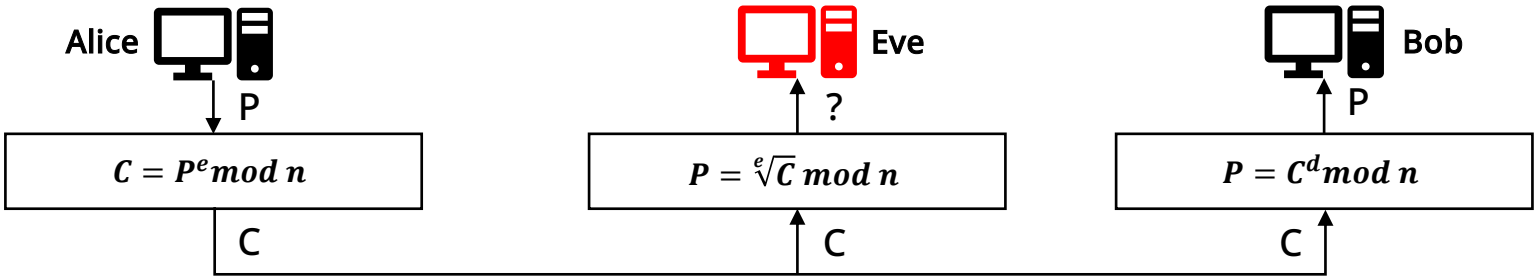
Symmetric-key vs. asymmetric-key ciphers

- Symmetric-key and asymmetric-key ciphers are different. These differences are what differentiate their purposes.

Division	Symmetric-key cryptography	Asymmetric-key cryptography
Key	Symmetric key (secret key)	Asymmetric key (public and private)
Encryption/decryption key relationship	Encryption key = decryption key	Encryption key ≠ decryption key
Number of keys	$N \times (N-1)/2$	$2 \times N$
Cipher method	Symbol (character, bit) substitution	Apply mathematical functions
Advantage	Fast computation, multiple algorithms	No need to share private keys, easy to add more communication destinations (public key distribution)
Disadvantage	Difficult in distributing and managing keys	Slow (exponential operations)
Well-known example	DES, AES	RSA
Cipher algorithm	Secret/Public	Public
Sending a secret key	Required	Not required
Security certification	Hard	Easy
Electronic signatures	Complex	Simple

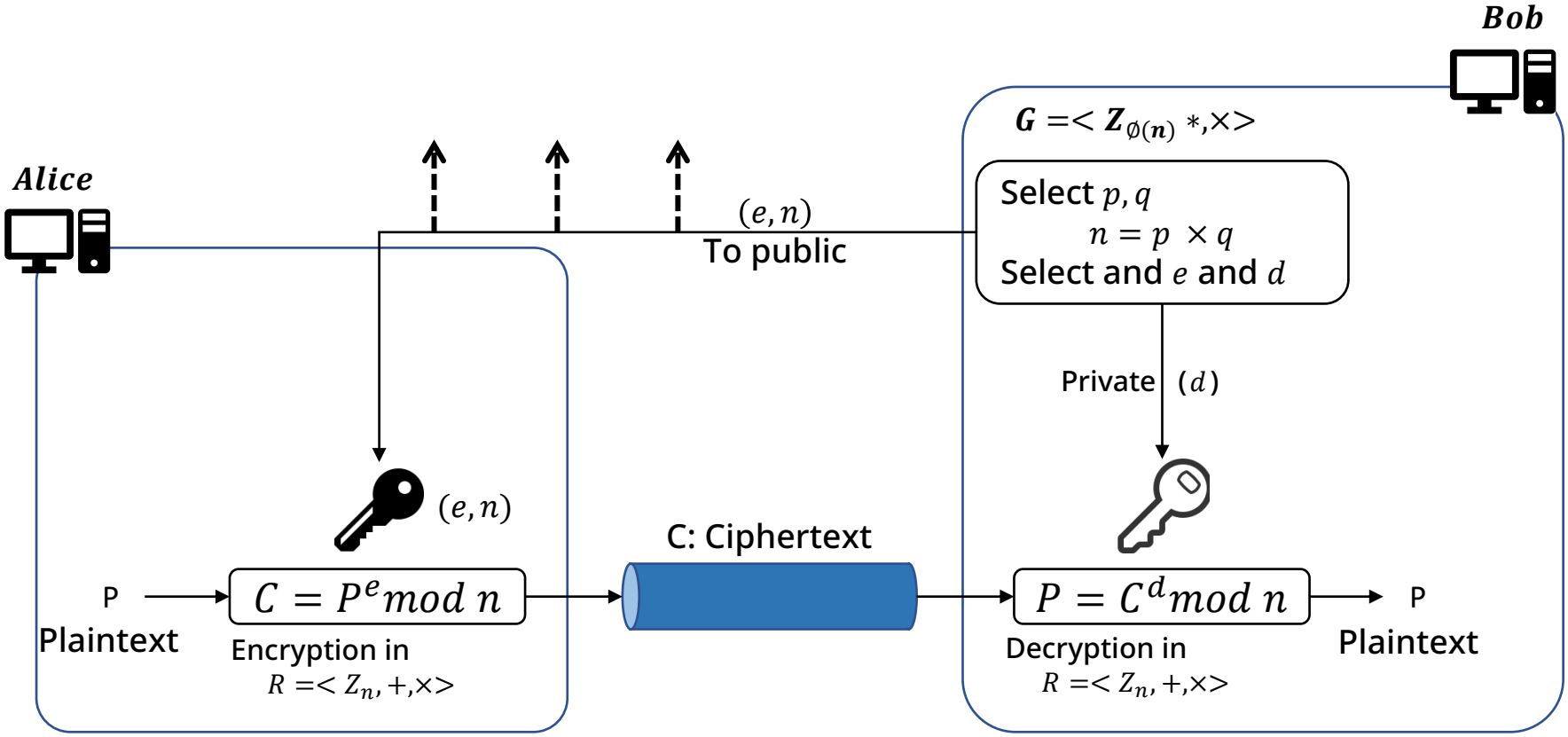
RSA

- Public-key cryptosystem developed in the US in 1978 by Rivest, Shamir and Adleman.
  - The most popular public-key cryptosystems on the market today
  - Use one-way trapdoor functions
    - Given large prime numbers p and q, it is easy to compute $N = p \times q$, but difficult to find large prime numbers p and q from the given N - a mathematical challenge
  - Use two exponents e (public key) and d (private key)
  - Alice generates ciphertext C from plaintext P using $C = P^e mod\ n$.
  - Bob obtains plaintext P from ciphertext C using $P = C^d mod\ n$.
  - Modulo n is generated by a very large number of key generation processes.
  - The attacker must obtain $\sqrt[e]{C}\ mod\ n$ for the attack
  - Alic and Bob have polynomial complexity, while Eve faces exponential complexity.



RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.



RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.

- Two algebraic structures
  - RSA uses two algebraic structures
    - Encryption/decryption : a public circle
      - Everyone knows the structure of this circle because N is public.
      - Anyone can use this circle to encrypt and send to Bob.

$$R = < Z_n, +, \times >$$

    - Key generation : a private group
      - Perform multiplication and division only
      - Use it to generate private and public keys
      - Keep the private group secret

$$R = < Z_{\emptyset(n)}\ *, \times >$$

RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.

- How to generate keys
  - Bob generates his public key (e) and private key (d) and declares the sequence pair (e, n) as his public key.
    - $n = p * q$
    - $\emptyset(n) = (p-1)(q-1)$
    - Choose e that satisfies the condition that $1 < e < \emptyset(n), d$ and $\emptyset(n)$ are disjoint.
    - $d = e^{-1} mod \emptyset(n), d$ is an inverse element of $e \; mod \; \emptyset(n)$.
  - Select two prime numbers p, q with different values (each size is recommended to be 1024 bits for security).
    - Each prime number has about 309 digits in decimal notation.
  - Modulo n is 2048 bits, which is about 618 digits in decimal notation.
  - Discard $p, q, \emptyset(n)$ after generating the keys.

RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.

- RSA proof
  - If $n = p * q, a < n$ and k is an integer, then $a^{k*\emptyset(n)+1} \equiv a(mod\ n)$ is true.
  - Plaintext when sent P
  - Decrypted plain text P_1

$$\boldsymbol{P_1 = C^d mod\ n = (P^e mod\ n)^d mod\ n = P^{ed} mod\ n}$$

$$\boldsymbol{ed = k * \emptyset(n) + 1}$$

$$\boldsymbol{P_1 = P^{ed} mod\ n \rightarrow P_1 = P^{\emptyset(n)+1} mod\ n}$$

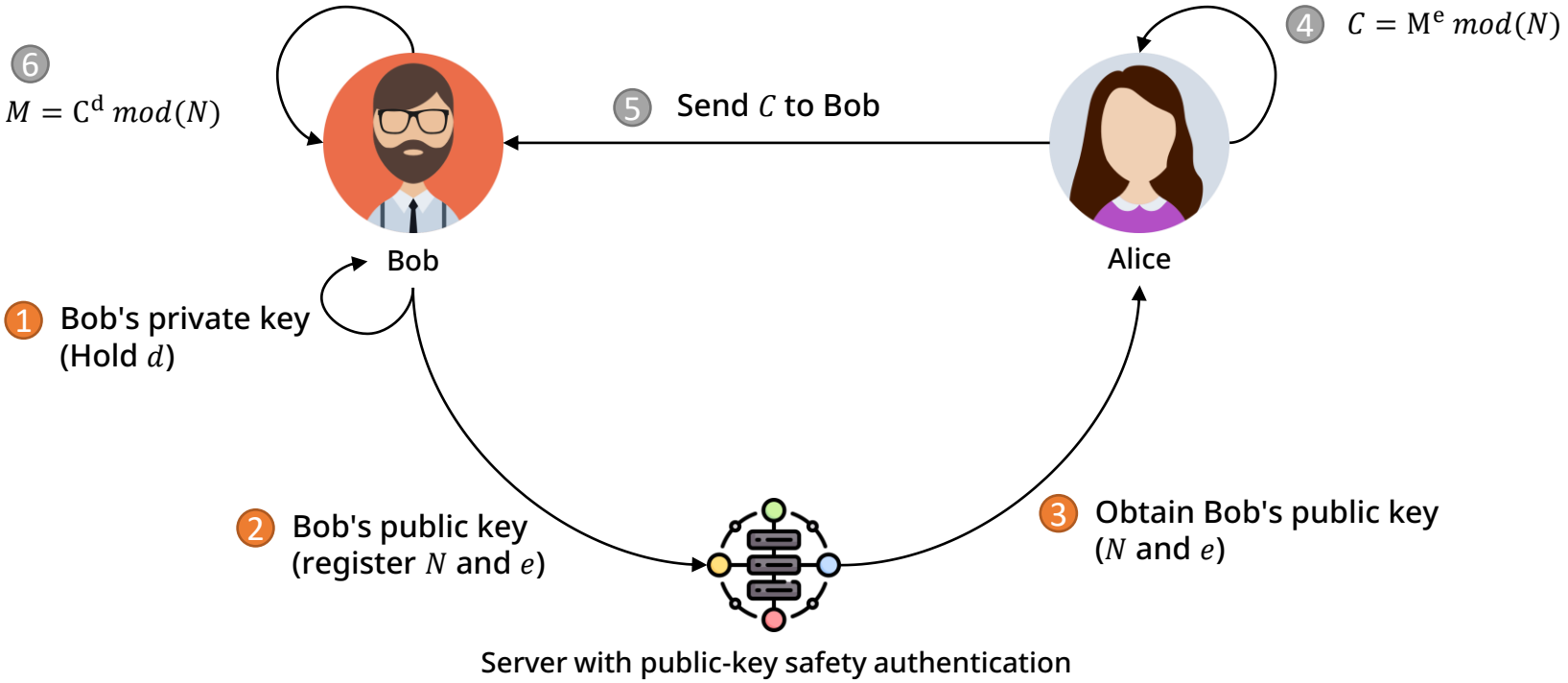
$$\boldsymbol{P_1 = P^{k*\emptyset(n)+1} mod\ n = p\ mod\ n}$$

RSA

- RSA example
  - Bob selects as $p = 7, q = 11$, calculating with $n = q * p = 7 * 11 = 77$.
  - $\emptyset(n) = (7 - 1)(11 - 1) = 60$
  - Bob chooses two exponent e and d that belongs to ${Z_{60}}^*$ (select $e = 13$).
  - If e is 13, then d becomes 37, which is an inverse element of e $(e^{-1} mod\ 60 = 37)$.
  - Alice (sender)
    - Plaintext(P) = 5
    - $C = 5^{13} mod\ 77 \equiv 26\ mod\ 77$
    - Ciphertext(C) = 26
  - Bob (recipient)
    - Ciphertext(C) = 26
    - $P = 26^{37} mod\ 77\ \equiv 5\ mod\ 77$
    - Plaintext(P) = 5

RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.



RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.

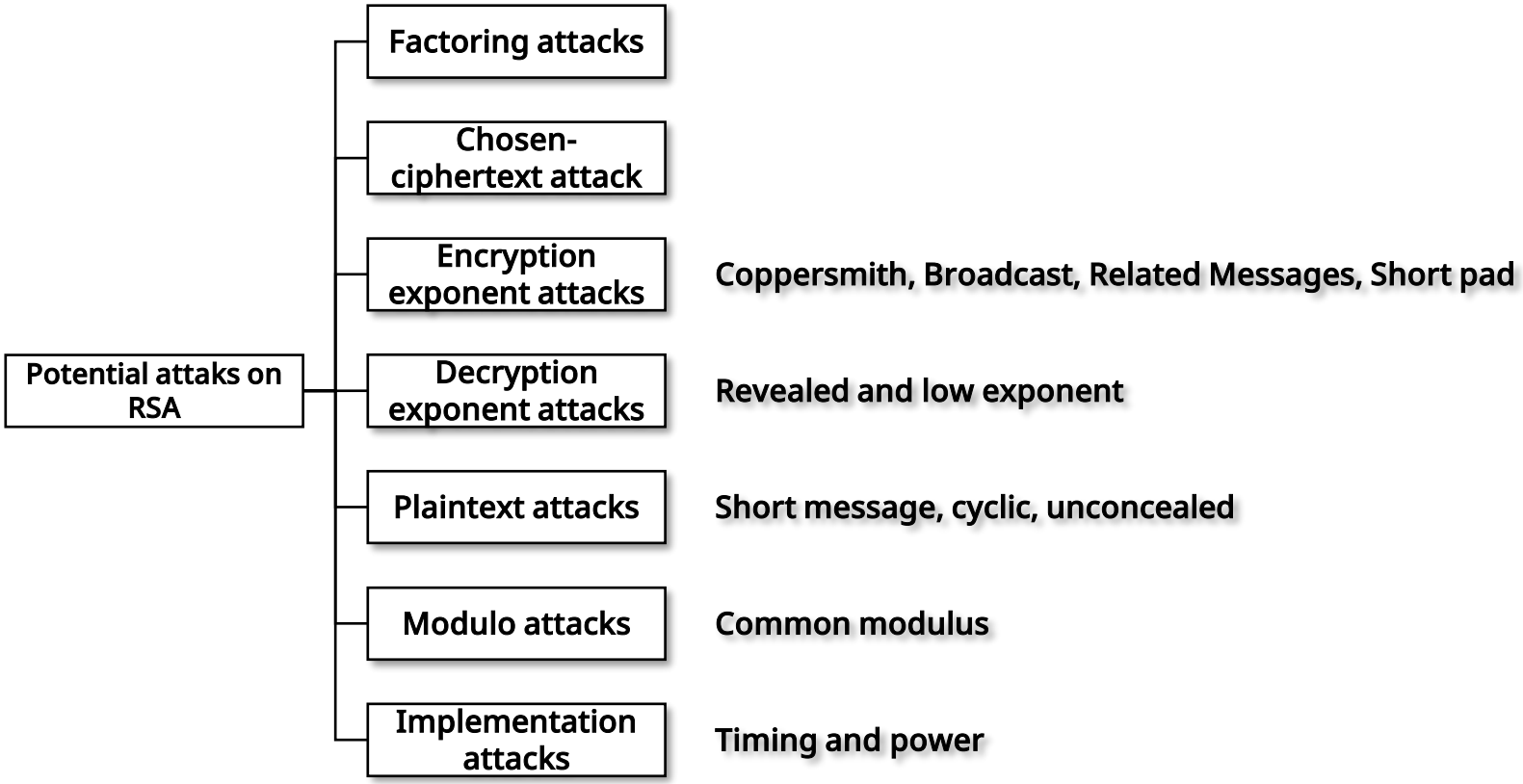
- Conditions for making RSA ciphers more secure
  - Must not be factorizable by Fermat's method, Pollard Rho method, etc.
    - p and q are not the same and have approximately the same size digits.
    - $p-1$ and $q-1$ take large prime arguments, respectively.
    - $p-1$ and $q-1$ must have a small greatest common divisor.
    - p and q must be large enough (currently 2048 bits or larger is required).

RSA

RSA is a cryptosystem implemented using a mathematical challenge called prime factorization. Block and stream ciphers are called symmetric-key ciphers because they encrypt and decrypt using the same secret key, while RSA is called an asymmetric-key cipher because it encrypts and decrypts using different secret keys.

- Conditions for making RSA ciphers more secure
  - Conditions for other parameters
    - The private key d should be a moderately large number.
      - It is usually chosen with $\max\{p, q\} + 1 < d < n - 1$.
    - For efficiency reasons, choose the public key e in 3, 17, $2r + 1$ (small r), etc.
      - Because it's an exponential operation, it uses a small number for efficiency.
      - However, for reliability, 3, 5, 17, 257, and 65537 are often used.
        - $2^1 + 1$, $2^2 + 1$, $2^4 + 1$, $2^8 + 1$, $2^{16} + 1$
      - rfc4871 "Domain Keys Identified Mail (DKIM) Signatures"
        - The e used in mail signatures is at least 65537 ($2^{16} + 1$).
        - Use larger values in special cases.
    - Decryption is slower than encryption because d is typically larger.

RSA


Potential attaks on RSA
Factoring attacks
Chosen-ciphertext attack
Encryption exponent attacks
Coppersmith, Broadcast, Related Messages, Short pad
Decryption exponent attacks
Revealed and low exponent
Plaintext attacks
Short message, cyclic, unconcealed
Modulo attacks
Common modulus
Implementation attacks
Timing and power

RSA

- Factoring attacks
  - RSA has a large modulo value, making it impossible to perform prime factorization in a reasonable amount of time.
  - Bob chooses p and q.
    - He computes $n = p \times q$ and makes n public and p, q private.
  - If Eve can prime factorize n to get p and q,
    - She can compute $\emptyset(n) = (p-1)(q-1)$, which allows her to compute $d = e^{-1} mod\ \emptyset(n)$ from the public value e.
  - Safety
    - RSA requires that n be at least 300 decimal digits long.
      - The modulo value must be at least 1024 bits.
    - The most recently recommended value is 2048 bits or higher (as of 2019).

RSA

- Chosen-ciphertext attacks
  - Use the multiplicative properties of RSA.
  - Suppose Bob decrypts a random ciphertext that is not the C requested by Eve.
  - Eve can intercept C and obtain P using the following procedure.
    - Eve chooses a random integer$X$ $(X \in {Z_n}^*)$
    - She computes $Y = C \times X^e mod\ n$.
    - She then sends Y to Bob to decrypt and get $Z = Y^d mod\ n$.

$$
\begin{aligned}
\boldsymbol{Z} &= \boldsymbol{Y^d mod\ n} \\
&= \boldsymbol{(C \times X^e)^d\ mod\ n} \\
&= \boldsymbol{\left(C^d \times X^{ed}\right)\ mod\ n} \\
&= \boldsymbol{\left(C^d \times X\right) mod\ n} \\
&= \boldsymbol{(P\ \times X) mod\ n}
\end{aligned}
$$

$$
\begin{aligned}
&\boldsymbol{Z = (P\ \times X) mod\ n} \\
&\therefore \boldsymbol{P = Z\ \times\ X^{-1}\ mod\ n}
\end{aligned}
$$

RSA

- Chosen-ciphertext attacks
  - An example using the RSA formula from the previous page.
  - $p = 7, q = 11, e = 13, d = 37, P = 5, C = 26, n = 77, \emptyset(n) = 60$
  - Eve selects a random integer X that belongs to $Z_{77}{}^{*}$ (where $X = 17$, $X^{-1} = 68$).
  - Eve computes $Y = C * X^{e} \; mod \; n$ $(Y = 26 * 17^{13} \; mod \; 77 = 50, \therefore Y = 50)$.
  - Eve decrypts to get $Z = Y^{d} \; mod \; n$ $(Z = 50^{37} \; mod \; 77 = 8, \therefore Z = 8)$.
  - She calculates P using $P = Z \times X^{-1} \; mod \; n$.
    - $P = 8 * 17^{-1} \; mod \; 77$
    - $P = 8 * 68 \; mod \; 77 = 5$
    - $\therefore P = 5$

RSA

- Encryption exponent attacks
  - Use a cryptographic exponent of e, usually 3, to save encryption time.
  - They do not break the cryptosystem itself, but we must be prepared for these attacks.

  - Coppersmith's (theorem) attacks
    - When a polynomial $f(x)$ with modulo n and exponent e has one root less than or equal to $n^{1/e}$, then
    - Complexity can be found in polynomial time for $\log n$.
    - Reduced time compared to traditional complexity $\sqrt[e]{C} \; mod \; n$.
    - Applicable to $C = f(P) - P^e \; mod \; n$.
    - $e = 3$ and if 2⁄3 of the bits in the plaintext P are known, then all the remaining bits can be determined.

RSA

- Encryption exponent attacks
  - Broadcast attacks
    - An attack is possible when one sender uses the same e to send messages to members of a group of people.
    - Create and send ciphertext to 3 people using the same $e = 3$ and different moduli n_1, n_2, n_3 for each.

$$\boldsymbol{C_1 = P^3 \bmod n_1} \qquad \boldsymbol{C_2 = P^3 \bmod n_2} \qquad \boldsymbol{C_3 = P^3 \bmod n_3}$$

    - Compute $M = n_1 \times n_2 \times n_3$ as a common modulo using Chinese remainder theorem.
    - The attacker gets $C' = P^3 \bmod M$.
    - $P^3 < M$, and therefore $C' = P^3$
    - The attacker can compute a cubic equation to get P.

RSA

- Encryption exponent attacks
  - Related message attacks
    - Encrypt two plaintexts P_1, P_2, with $e = 3$.
    - When sending encrypted C_1 and C_2, if P_1 and P_2 are in a linear relation, it is possible to compute plaintext P_1, P_2 in a short period of time.
  - Short pad attacks
    - Alice pads r_1 with the message in transit and then sends an encrypted C_1.
    - Eve intercepts C_1 to prevent Bob from receiving the message, and Bob notifies Alice.
    - Alice creates a new pad r_2 and then sends an encrypted C_2.
    - Eve intercepts C_2 as well.
    - Eve receives C_1 and C_2 and knows that they both encrypt the same plaintext.
    - Since r_1 and r_2 are short, Eve can obtain the source plaintext message M.

RSA

- Encryption exponent attacks
  - Public decryption exponent attacks
    - If the attacker knows the private key d, the ciphertext can be decrypted.
    - Prime factorize n and determine the values of p and q using a probabilistic algorithm
    - If the recipient simply replaces the compromised encryption exponent and uses the same modulo value,
      - The attacker can also decrypt ciphertexts generated with new encryption exponents.
    - This means that if the private key d is compromised, p, q, n, e and d everything has to be regenerated.
  - Small decryption exponent attacks
    - $q < p < 2q$ and if the private key d is $d < (1/3)n^{1/4}$, then the security is threatened by a special attack based on consecutive fractions, a topic in number theory.
    - To prevent decryption exponent attacks, d must be used that satisfies $d \geq (1/3)n^{1/4}$.

RSA

- Plaintext attacks
  - In RSA, plaintext and ciphertext are integers between 0 and n -1.
  - The attacker would have information about the plaintext.
  - Short message attacks
    - Encrypt all messages until a ciphertext appears that is identical to the intercepted message.
    - When encrypting short messages, pad the message with additional random bits at the beginning and end of the message before encrypting it.
  - Cyclic attacks
    - Substitution of plaintext for ciphertext
    - Based on the fact that successive encryptions of a ciphertext will eventually result in a plaintext.
    - When $C_k = C$ is reached, the value obtained in the previous step is returned as a plaintext.

$C_1 = C^e \ mod \ n$ //C is the intercepted ciphertext.
$C_2 = {C_1}^e \ mod \ n$
…
$C_k = {C_{k-1}}^e \ mod \ n \ \rightarrow$ If $C_k = C$, stop here.
$P = C_{k-1}$

RSA

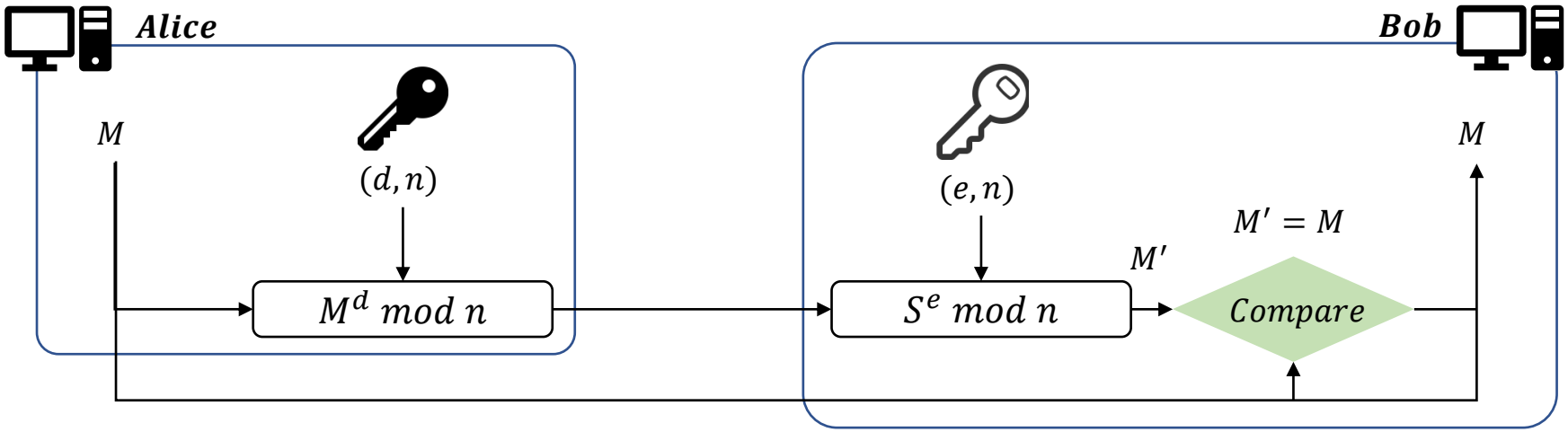
- Plaintext attacks
  - Unconcealed attacks
    - Based on the fact that ciphertext and plaintext are commutative.
    - Messages that don't hide who you are when encrypted.
    - Usually the encryption exponent is odd, so messages such as P=0 or P=1 are encrypted with themselves.
    - Cryptographic programs should always check that the computed ciphertext is the same as the plaintext.

RSA

- Modulo attacks
  - Generic modulo attacks
    - Possible when certain populations use the same modulo value n.
    - A group of trusted third parties select p and q.
    - Compute n and $\emptyset(n)$ to generate and serve (e_i, d_i) to all members of the group.
    - Alice sends the ciphertext $C = P^{eB} \, mod \, n$ to Bob.
    - Bob decrypts the received ciphertext $P = C^{dB} \, mod \, n$ using the exponent d_B, which is used as his private key.
    - Eve can decrypt Alice's message if she is also a member of this group and the exponent pair (e_E, d_E) is provided by a trusted third party.
    - Eve's own exponent (e_E, d_E) allows for a probabilistic attack to prime factorize n.
    - Bob's private key d_B can be obtained.
    - Each object must choose its own modulo value.

RSA

- How to generate keys
  - Same key generation process as traditional RSA cryptosystems
  - The signer chooses two prime numbers p, q and computes $n = p * q$.
  - The signer computes $\emptyset(\mathrm{n}) = (\mathrm{p} - 1)(\mathrm{q} - 1)$.
  - Choose a public key e and use d as a secret key that satisfies $e * d = 1 \; mod \; \emptyset(n)$.
  - Then make n and e public.

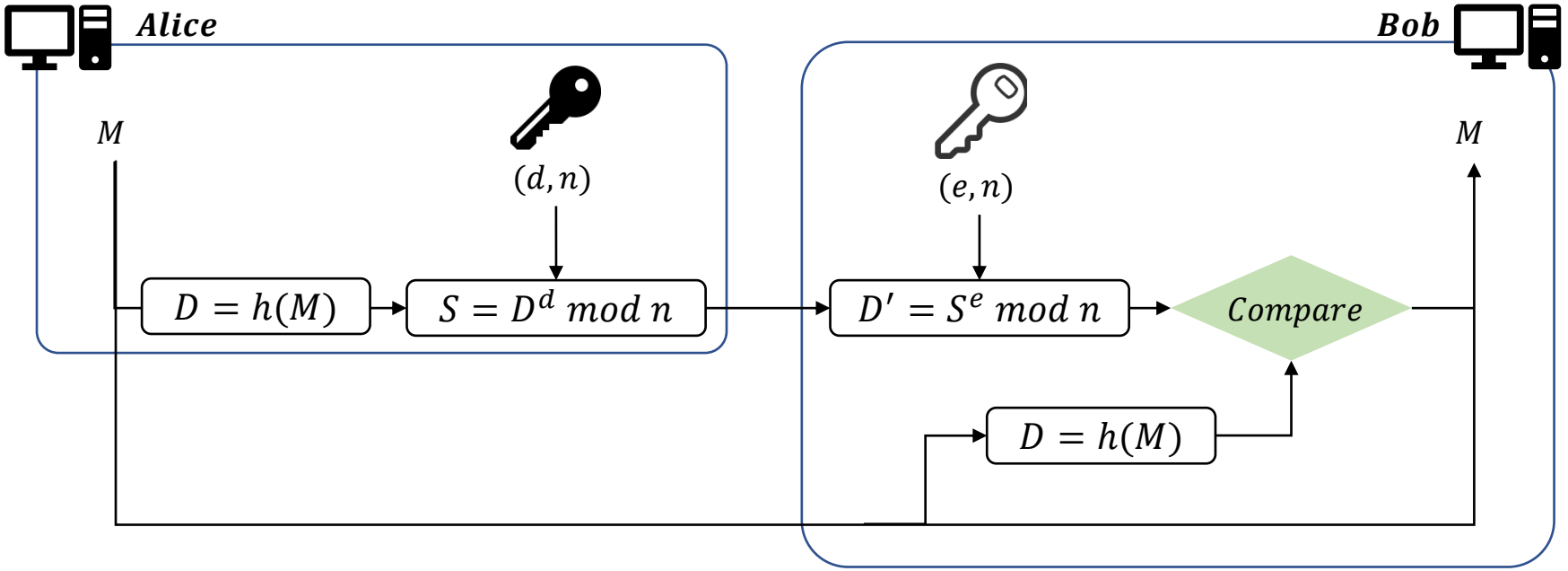


$$M' \equiv M(mod\ n) \rightarrow S^e \equiv M(mod\ n) \rightarrow M^{d*e} \equiv M(mod\ n)$$

RSA

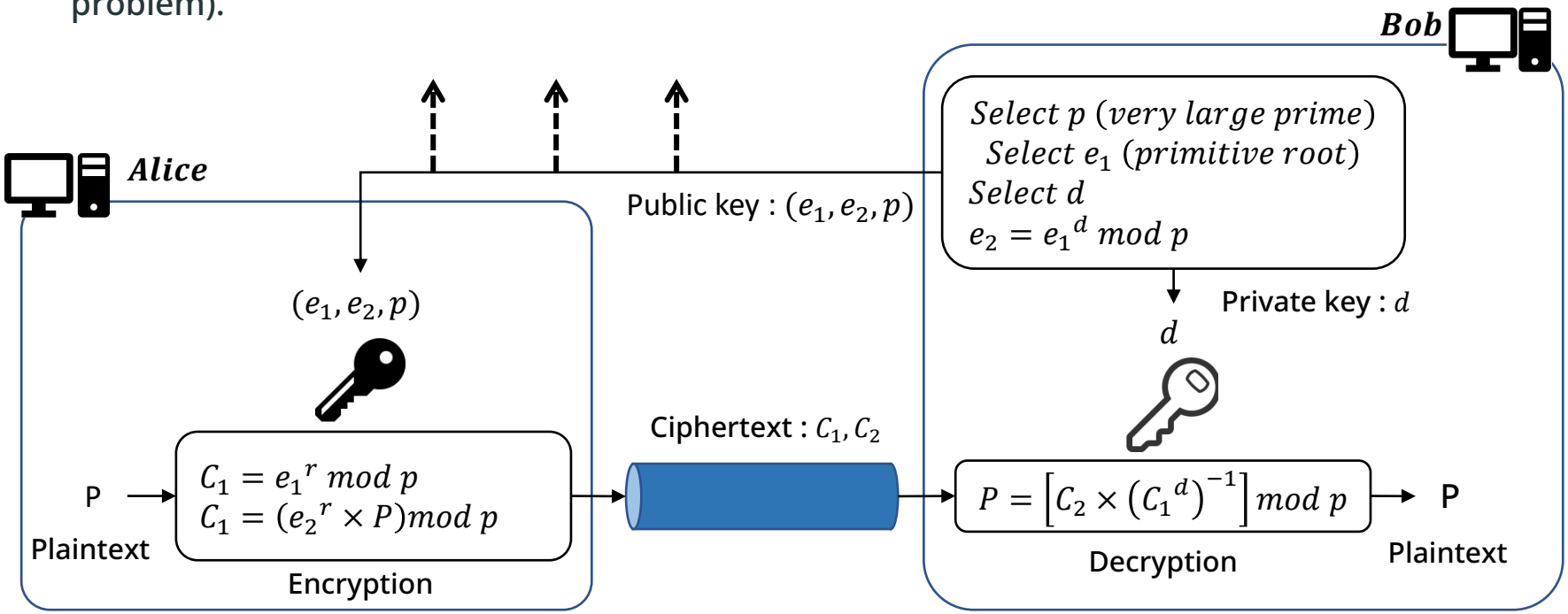
RSA signatures for message digests

- RSA signature structures are slow.
- Signing a message digest speeds up the signing and verification process.
- Using strong cryptographic hash functions makes signatures harder to attack.
- Create and use the message digest $D = h(M)$.



ElGamal

- Proposed cipher using a discrete algebra problem on a finite body
- p is a very large prime number, e_1 is one primitive root of $G =< Z_p{}^*, \times >$, r is an integer,
- Fast exponential algorithm (square-squared method) can be used to easily compute $e_2 = {e_1}^r \ mod \ p$.
- If e_1, e_2 and p are known, it is impractical to compute $r = \log_{e_1} e2 \ mod \ p$ (discrete algebra problem).



ElGamal

- How to generate a key
  - Choose a sufficiently large prime number p.
  - Select d from $G =< {Z_p}^*, \times >$ that satisfies $1 \leq d \leq p-2$.
  - Select the primitive root e_1 of $G =< {Z_p}^*, \times >$.
  - Compute ${e_2} = {e_1}^d \ mod \ p$.
  - What's public are e_1, e_2, p
  - What's private is d.

- Encryption
  - Select r on $G =< {Z_p}^*, \times >$.
  - ${C_1} = {e_1}^r \ mod \ p$
  - ${C_2} = (P * {e_2}^r) mod \ p$

ElGamal

- Decryption
  - $P = \left[C_2 \left(C_1{}^d\right)^{-1}\right] mod\ p$

- Proof
  - The ElGamal decryption representation $C_2 \times (C_1{}^d)^{-1}$ becomes P, which can be seen through substitution.

$$\left[C_2 \times \left(C_1{}^d\right)^{-1}\right] mod\ p = \left[\left(e_2{}^r \times P\right) \times \left(e_1{}^{rd}\right)^{-1}\right] mod\ p = \left(e_1{}^{dr}\right) \times P \times \left(e_1{}^{rd}\right)^{-1} = P$$

ElGamal

- Example
  - Bob selects 11 for p and 2 for e_1 (where 2 is the primitive root of ${Z_{11}}^*$).
  - He computes 3 for the value d and $e_2 = {e_1}^d = 8$.
  - Here, the public key is (2, 8, 11) and the private key is (3)
  - Alice chooses $r = 4, P = 7$.

  - Encryption
    - $C_1 = {e_1}^r \ mod \ p = 16 \ mod \ 11 = 5 \ mod \ 11$
    - $C_2 = (P * {e_2}^r) mod \ p = (7 * 8^4) mod \ 11 (7 * 4096) mod \ 11 = 6 \ mod \ 11$
    - $\therefore C_1 = 5, C_2 = 6$
  - Decryption
    - $\left[C_2 * ({C_1}^d)^{-1}\right] mod \ p = 6 * (5^3)^{-1} \ mod \ 11 = 6 * 3 \ mod \ 11 = 7 \ mod \ 11$
    - $\therefore P = 7$

ElGamal

- Alice sends $C_2 = [{e_2}^r * P] mod\ p = \left[\left({e_1}^{rd}\right) * P\right] mod\ p$.
  - $({e_1}^{rd})$ acts as a mask to hide the value of P and must be removed to obtain the value of P.
- Because modulo operations are used, Bob can create a duplicate of the mask.
  - Using the inverse element for multiplication to remove the effect of masks.
- Alice sends Bob a potion of the mask, which is $C_1 = {e_1}^r$.
  - Bob needs to compute ${C_1}^d$ to create a duplicate of the mask,
    - Because ${C_1}^d = ({e_1}^r)^d = ({e_1}^{rd})$.
    - Bob obtains a duplicate of the mask, calculates its inverse, and multiplies it by C_2 to remove the mask.
- Bob helps Alice create the mask $({e_1}^{rd})$ without exposing the value d.
- Alice helps Bob create the mask $({e_1}^{rd})$ without exposing the value r.

ElGamal

- Small modulo attacks
  - If p is a small number, Eve can easily compute $d = \log_{e_1} e2 \ mod \ p$ and store it to decrypt all messages sent to Bob.
  - Eve uses C_1 to find out the random number used in that Alice sends $r = \log_{e1} C1 \ mod \ p$ to her.
  - Relying on the fact that discrete algebra problems with very large modulo are unsolvable,
    - For safety, p must be chosen to be greater than or equal to 2048 bits.
- Known-plaintext attacks
  - If Alice encrypts two plaintexts P and P' with the same randomized exponent r, then if Eve knows one of them, she also knows the other.
  - $C_2 = P \times ({e_2}^r) mod \ p$
  - ${C'}_2 = P' \times ({e_2}^r) mod \ p$

$$({e_2}^r) = C_2 \times P^{-1} \ mod \ p$$
$$P' = {C'}_2 \times ({e_2}^r)^{-1} mod \ p$$

ElGamal

- Example
  - Use the expression from the previous procedure as an example.

If $p = 11,\ e_1 = 2,\ d = 3,\ e_2 = 8,\ r = 4,\ P = 7,\ C_1 = 5,\ C_2 = 6$,

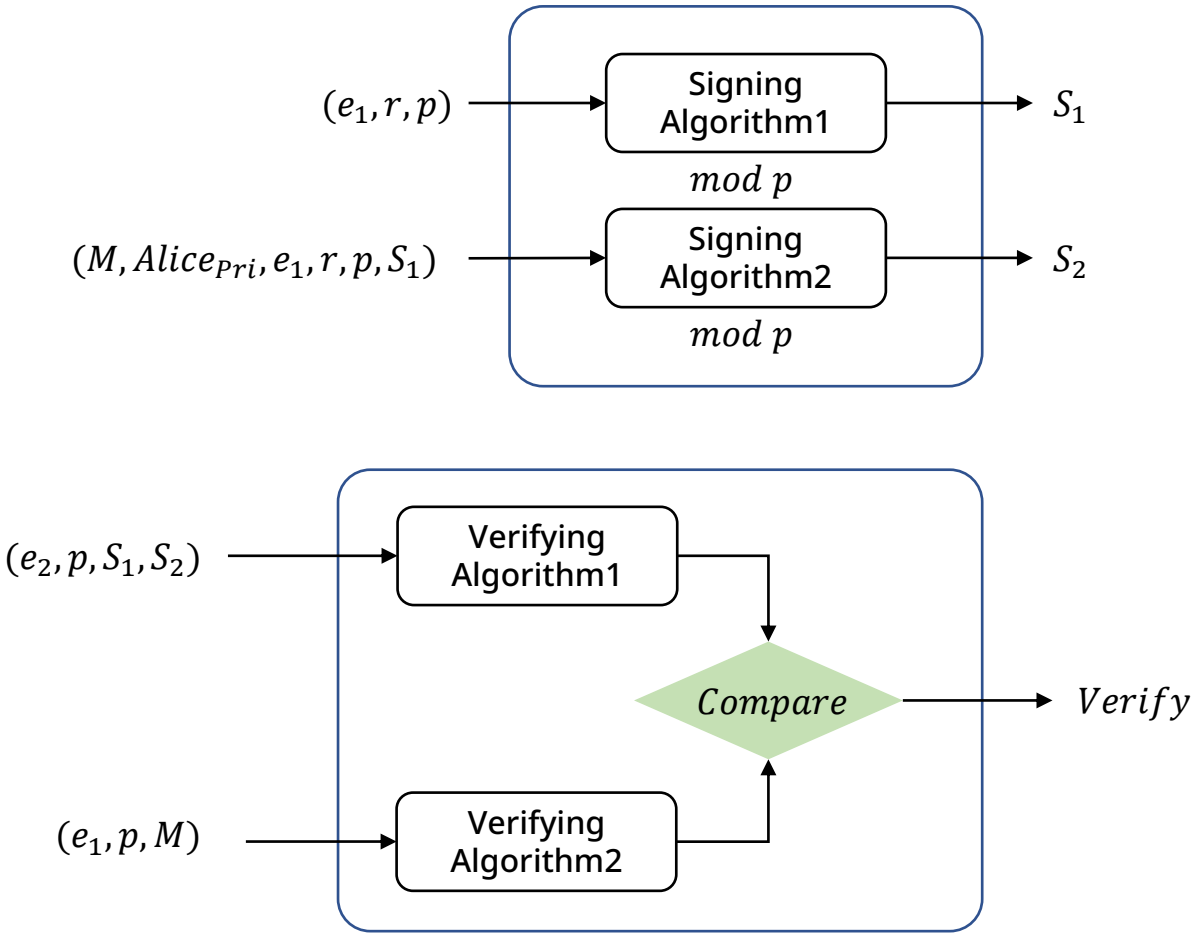
- It is assumed that we have calculated an additional $P' = 9,\ {C_1}' = 5,\ {C_2}' = 3$.
- Small modulo attacks
  - Convertible to $r = \log_{e1} C_1\ mod\ p \rightarrow\ C_1 = {e_1}^r\ mod\ p$.
  - $r = \log_2 5\ mod\ 11\ \rightarrow 5 = 2^r\ mod\ 11$
  - $\therefore r = 4$
- Known-plaintext attacks
  - $({e_2}^r) = C_2 \times P^{-1}\ mod\ p$
  - $({e_2}^r) = 6 \times 7^{-1}\ mod\ 11\ \rightarrow ({e_2}^r) = 6 \times 8\ mod\ 11\ \rightarrow ({e_2}^r) = 4\ mod\ 11$
  - $P' = {C_2}' \times ({e_2}^r)^{-1}\ mod\ p$
  - $P' = 3 \times (4)^{-1}\ mod\ 11\ \rightarrow\ P' = 3 \times 3\ mod\ 11\ \rightarrow\ P' = 9\ mod\ 11$

ElGamal

- Can sign and verify using the ElGamal cryptosystem
- Use the same key but different algorithms
- How to generate keys
  - Same as the key generation process in the ElGamal cryptosystem.
  - p is a sufficiently large prime number that a discrete algebra problem cannot be solved within ${Z_P}^*$.
  - e_1 is a primitive root in ${Z_P}^*$.
  - The sender chooses his private key d to be a number smaller than $p-1$.
  - Compute ${e_2 = {e_1}^d}$,
  - The signer's public key is (e_1, e_2, P), and the private key is d.

ElGamal

ElGamal digital signature



ElGamal

- Signature : signer signs the digest of the message.
  - Public and private keys are used repeatedly, but each time with a different secret random number.
  - The signer computes the first signature $S_1 = {e_1}^r \ mod\ p$.
  - Then calculates the second signature $S_2 = (M - d * S_1) * r^{-1} mod\ (p-1)$.
    - Here, r^{-1} is an inverse element of the multiplication of the modulo p.
  - The signer sends M, S_1, S_2 to the verifier.
- Validation : the validator performs the following steps after receiving M, S_1, S_2.
  - Check that $0 < S_1 < p$ and $0 < S_2 < p-1$.
  - Calculate $V_1 = {e_1}^M \ mod\ p$ and $V_2 = {e_2}^{S_1} * {S_1}^{S_2} \ mod\ p$.
  - Compare V_1 with V_2.
- Comparison procedure : use $e_2 = {e_1}^d$ and $S_1 = {e_1}^r$.

$$
\begin{aligned}
V_1 \equiv V_2 (mod\ p) \rightarrow\ {e_1}^M &\equiv {e_1}^{S_1} * {S_1}^{S_2} (mod\ p) \\
&\equiv \left({e_1}^d\right)^{S_1} ({e_1}^r)^{S_2} (mod\ p) \\
&\equiv \left({e_1}^{dS_1 + rS_2}\right) (mod\ p)
\end{aligned}
$$

ElGamal

- Example
  - Signer (Alice) selects $p = 3119, e_1 = 2, d = 127$, calculates $e_2 = 2^{127} mod\ 3119 = 1702$.
  - She selects $r = 307$, $M = 320$.

  - Signer (Alice)

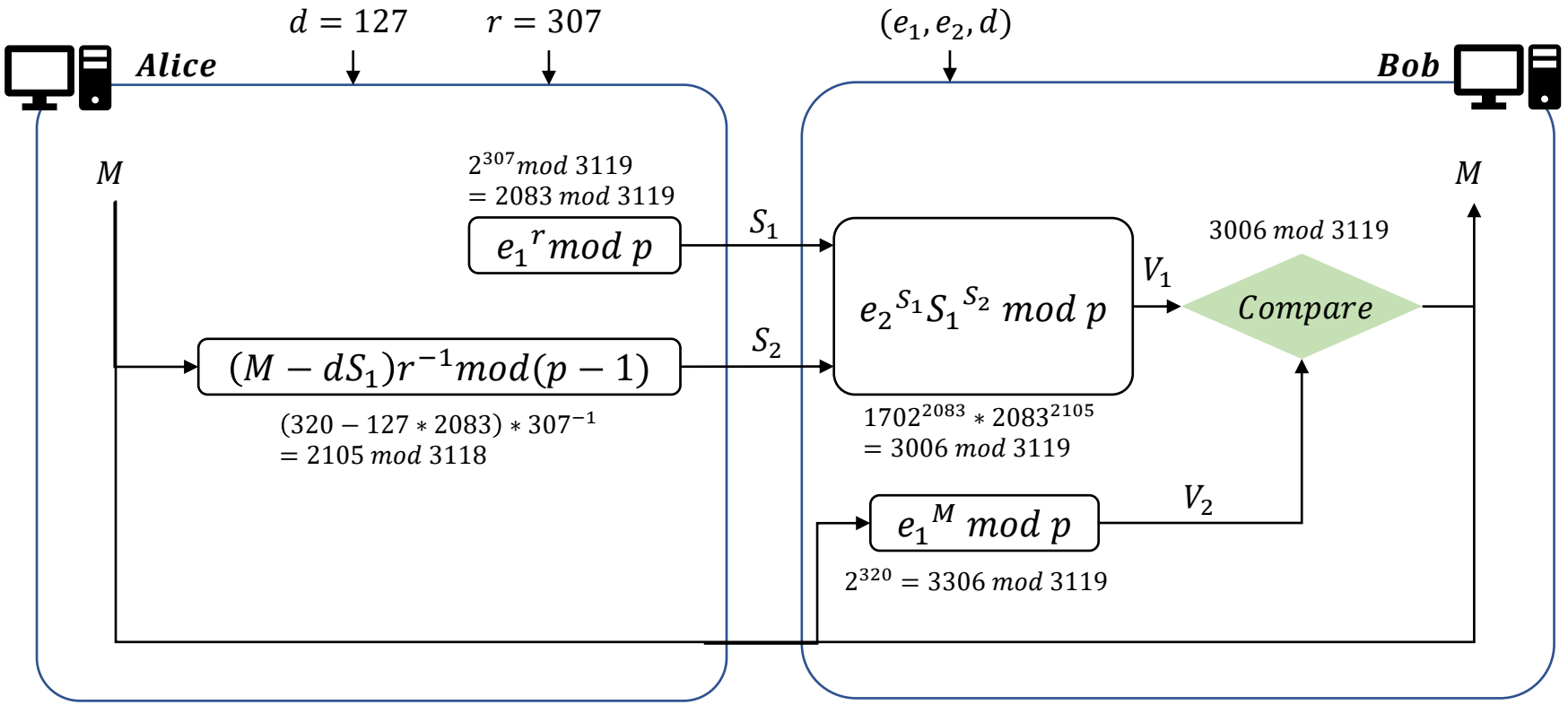
$$S_1 = {e_1}^r = 2^{307} = 2083\ mod\ 3119$$
$$S_2 = (M - d * S_1) * r^{-1} = (320 - 127 * 2083) * 307^{-1} = 2105\ mod\ 3118$$

  - Validator (Bob)

$$V_1 = {e_1}^M = 3006\ mod\ 3119$$
$$V_2 = d^{S_1} * {S_1}^{S_2} = 1702^{2083} * 2083^{2105} = 3006\ mod\ 3119$$

ElGamal

- Example
  - Signer (Alice) selects $p = 3119, e_1 = 2, d = 127$, calculates $e_2 = 2^{127} mod\ 3119 = 1702$.
  - She selects $r = 307$, $M = 320$.



ElGamal

- Key-only forgery : the attacker can only obtain the public key, the two kinds of forgery are possible.
  - The attacker has a pre-generated message M.
    - The attacker chooses two legitimate signatures S_1, S_2 for this message (selective forgery attack).
    - The attacker selects S_1 and calculates S_2.
    - ${S_1}^{S_2} \equiv {e_1}^{M} d^{-S_1} \ mod \ p$ or $S_2 \equiv \log_{S_1}({e_1}^{M} d^{-S_1}) \ (mod \ p)$ is required.
    - Conversely, when S_2 is chosen, it becomes more difficult to compute S_1.
  - The attacker has random M, S_1, S_2.
    - $M = xS_1 \ mod \ (p-1), S_1 = -yS_2 mod(p-1)$
    - If satisfactory x, y are obtained, the message can be forged.
    - In the end, it is a meaningless forgery.
  - Forging known messages
    - Suppose an attacker intercepts the message M and the signature $S_1 \ S_2$.
      - You can find a message $M'$with the signatures S_1, S_2 that has the signature.
      - This is a useless forgery and not a useful attack.

ECC

Understanding elliptic curves

Elliptic Curve Cryptography (ECC) was proposed independently by Neil Koblitz and Victor Miller in 1985. The main advantage of elliptic curve ciphers over traditional public key cryptography, such as RSA or ElGamal ciphers, is that they provide a similar level of security while using shorter keys.

- Elliptic curve
  - A curve in the plane, a group of points with coordinates x and y.
  - The equation of a curve defines all the points on that curve.
    - $y = 3$ is a horizontal line with vertical coordinates of 3. A curve of the form $y = ax + b$ is a straight line with a fixed number a and b.
    - $x^2 + y^2 = 1$ is a circle of radius 1 centered on the origin, and points on any curve are all pairs of (x, y) that satisfy the equation of that curve.
    - Elliptic curves used in cryptography are curves whose equation typically looks like this : $y^2 = x^3 + ax + b$, where the constants a and b define the shape of the curve.

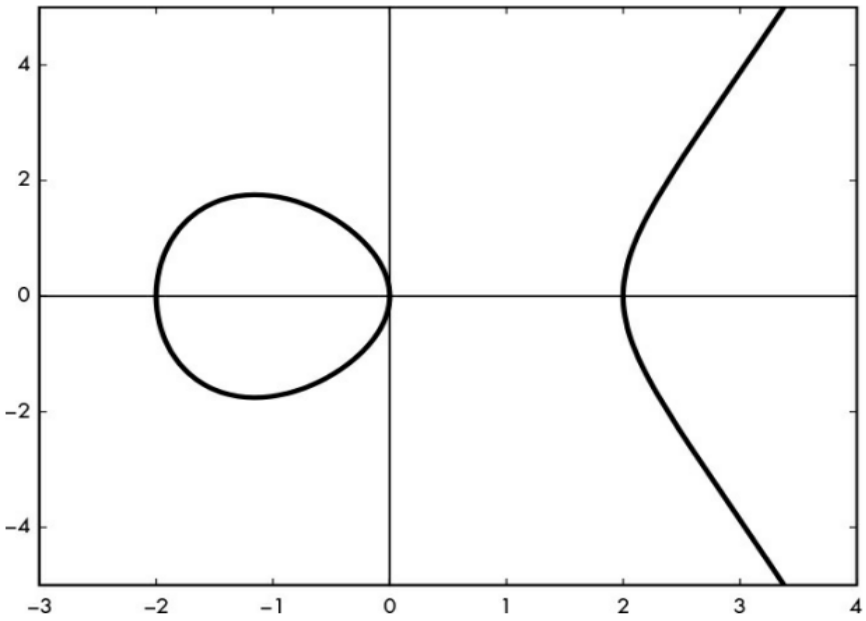
Source : https://en.wikipedia.org/wiki/Elliptic_curve

ECC

Elliptic curve principle

Elliptic Curve Cryptography (ECC) was proposed independently by Neil Koblitz and Victor Miller in 1985. The main advantage of elliptic curve ciphers over traditional public key cryptography, such as RSA or ElGamal ciphers, is that they provide a similar level of security while using shorter keys.

- Elliptic curve example
  - The figure below shows an elliptic curve that satisfies $y^2 = x^3 - 4x$.



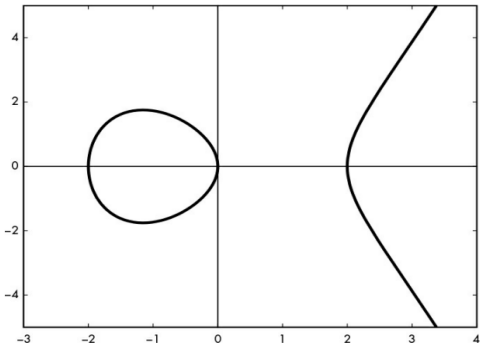
<Elliptic curve with equation $y^2 = x^3 - 4x$ shown above the real numbers>

ECC

Elliptic curve principle

Elliptic Curve Cryptography (ECC) was proposed independently by Neil Koblitz and Victor Miller in 1985. The main advantage of elliptic curve ciphers over traditional public key cryptography, such as RSA or ElGamal ciphers, is that they provide a similar level of security while using shorter keys.

- Elliptic curve example
  - All points that make up a curve that falls in the range where x is between -3 and 4.
  - These are points on the left part of the curve that look like a circle, or points on the right part of the curve that represent a parabola.
  - All of these points have (x, y) coordinates that satisfy the equation $y^2 = x^3 - 4x$ on the curve.
  - For example, for $x = 0$, $y^2 = x^3 - 4x = 0^3 - 4 \times 0 = 0$. Therefore, $y = 0$ is a solution, and the point $(0, 0)$ belongs to the curve. Similarly, for $x = 2$, the solution of the equation is $y = 0$, which means that the point $(2, 0)$ belongs to the curve.



ECC

In ECC, it is important to distinguish between points on the curve and points off the curve. Points on the curve are used for secure operations from a security perspective, while points off the curve can pose a security threat.

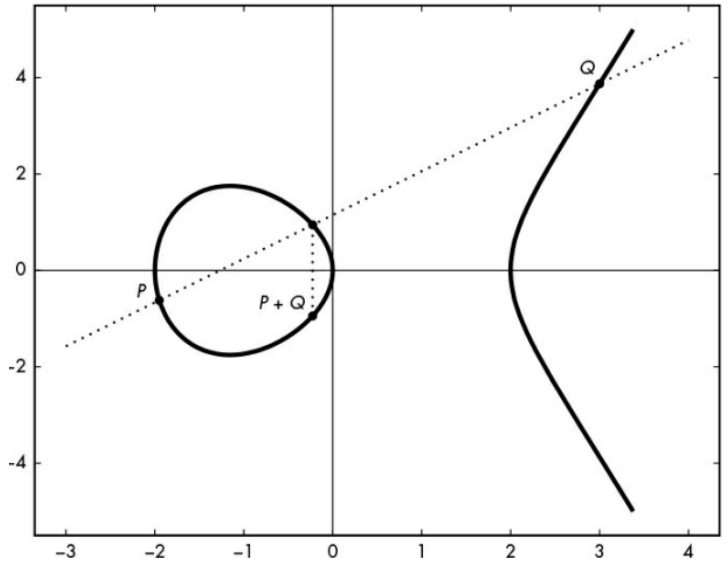
- Elliptic curve example
  - **Equations on a curve don't always have solutions**
    - Example 1
      - To find the point corresponding to the coordinate $x = 1$, use $y^2 = x^3 - 4x$ to find y^2.
      - When this equation is solved, the result is -3 with no corresponding solution $y^2 = -3$.
      - Since there is no solution to the curve equation for $x = 1$, there is no point on the curve on the x axis at that location, as shown in the figure.
    - Example 2
      - To find a solution for x = -1, we have the equation $y2$ = -1 + 4 = 3.
      - This equation has two solutions ($y = \sqrt{3}$, $y = -\sqrt{3}$), the square root of 3 and its negative value.
      - Since squaring always yields a positive number, $y^2 = (-y)2$ holds for all real numbers y.
      - The curve in the figure is symmetric about the x axis for all points that solve this equation (as are all elliptic curves of the form $y^2 = x^3 + ax + b$).

ECC

Add two points

We have seen that the points on an elliptic curve are all coordinates (x, y) that satisfy the equation of the curve such as $y^2 = x^3 + ax + b$. We will review the "rule of addition," which is how we add points on an elliptic curve.

- Geometrically understanding the location of $R = P + Q$ relative to points P and Q on the curve
  - Suppose you want to add two points, P and Q, on an elliptic curve to get a new point, R.
  - Draw a line connecting P and Q that is determined by geometric rules.
  - Find the intersection of that line with the curve, and at that intersection, find a point that is symmetric about the x axis, which is $R = P + Q$.
    - In the figure, the line connecting P and Q intersects a point between P and Q.
    - Point R is the point that is symmetric about the intersection and the x axis, with the same x component of the coordinates and opposite signs of the y component.

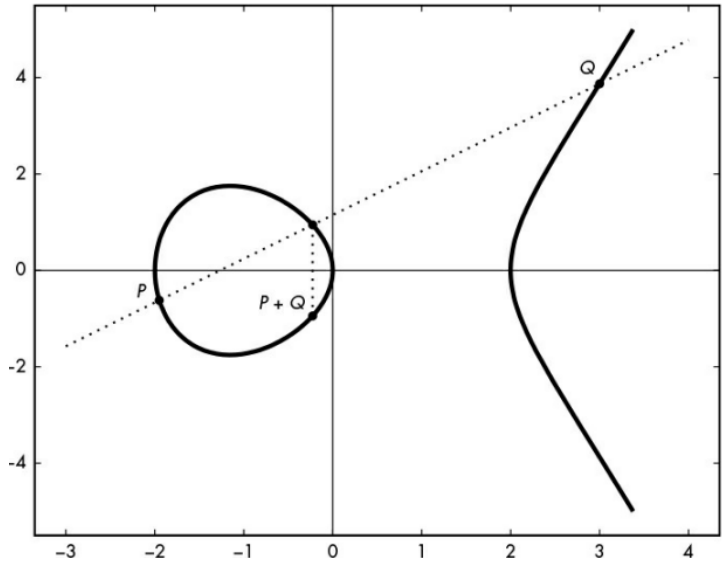


ECC

We have seen that the points on an elliptic curve are all coordinates (x, y) that satisfy the equation of the curve such as $y^2 = x^3 + ax + b$. We will review the "rule of addition," which is how we add points on an elliptic curve.

- Geometrically understanding the location of $R = P + Q$ relative to points P and Q on the curve
  - To calculate the coordinates (x_R, y_R) of point R, use the coordinates of point P (x_p, y_p) and the coordinates of point Q (x_Q, y_Q), using the following formula
    - $x_R = m^2 - x_p - x_Q$
    - $y_R = m(x_p - x_R) - y_p$
  - Here, m is the slope of the straight line connecting P and Q,

    $m = (y_Q - y_p) / (x_Q - x_p)$.



ECC

Elliptic Curve Diffie-Hellman (ECDH) allows two people who wish to communicate cryptographically to share a secret key, and enables symmetric-key cryptographic communication.

- Diffie-Hellman key exchange scheme using ECC
  - Premise
    - All should agree to use the 6-element tuple from (p, a, b, g, n, h) as a specific definition domain parameter
    - In some cases, you can also write a 7-element tuple with $p = \{m, f(x)\}$.
    - Each party must have the appropriate key pair (d_x, Q_x) on the ECC.
      - d is a randomly chosen integer between 1 and $n - 1$ that is the private key.
      - The public key Q is an integer satisfying $Q = dg$.

ECC

Elliptic Curve Diffie-Hellman (ECDH) allows two people who wish to communicate cryptographically to share a secret key, and enables symmetric-key cryptographic communication.

- Diffie-Hellman key exchange scheme using ECC
  - Procedure
    - Alice has a key pair (d_A, QA) and Bob has a key pair (d_B, QB), and they know each other's public keys by exchanging them.
    - Alice computes the point $(x_A, yA) = d_A Q_B$, and Bob computes the point $(x_B, y_B) = d_B Q_A$.
      - $d_A Q_B = d_A d_B * g = d_B d_A * g = d_B Q_A$
    - The shared secret value is xk.
      - Many modern Elliptic Curve Diffie-Hellman (ECDH)-based ciphers hash the shared secret value to generate a symmetric key.
  - It is ECDH when the public key is permanent (static) and ECDH Ephemeral(ECDHE) when it is ephemeral.

ECC

Elliptic Curve Digital Signature Algorithm (ECDSA)

The standard algorithm for signatures using ECC is the Elliptic Curve Digital Signature Algorithm (ECDSA). It has replaced RSA and DSA signatures in many applications, is the only signature algorithm used in Bitcoin, and is supported by many TLS and SSH implementations.

- E-signatures via elliptic curves
  - A verification algorithm in which a signer uses their private key to create a signature, and a verifier uses the signer's public key to check the accuracy of the signature.
    - The signer holds the number d as a private key, and the verifier holds $P = dG$ as a public key.
    - Both parties know in advance which elliptic curve to use, the order of the curve (n, the number of points on the curve), and the coordinates of the base point G in advance.

ECC

Elliptic Curve Digital Signature Algorithm (ECDSA)

The standard algorithm for signatures using ECC is the Elliptic Curve Digital Signature Algorithm (ECDSA). It has replaced RSA and DSA signatures in many applications, is the only signature algorithm used in Bitcoin, and is supported by many TLS and SSH implementations.

- **Signature generation**
  - The signer uses their private key to create a signature for the message.
    - The signer first uses a hash function, such as SHA-256 or BLAKE2, to generate the message's hash value h.
    - The signer chooses a random number k between 1 and $n-1$, and computes a point with coordinates (x, y), called kG.
    - Set $r = x \bmod n$, compute $s = (h + rd) / k \bmod n$, and use these values as the signature (r, s).
    - The length of the signature depends on the length of the coordinates used.
    - E.g., If you use a curve whose coordinates are 256-bit numbers, r and s are each 256 bits long, resulting in a 512-bit signature.

ECC

Elliptic Curve Digital Signature Algorithm (ECDSA)

The standard algorithm for signatures using ECC is the Elliptic Curve Digital Signature Algorithm (ECDSA). It has replaced RSA and DSA signatures in many applications, is the only signature algorithm used in Bitcoin, and is supported by many TLS and SSH implementations.

- **Signature verification**
  - The process of validating a signed message using a public key to verify its accuracy.

    1. The verifier computes the inverse of s of the signature.
       - The inverse of s is denoted by $w = 1 / s$, where s is defined as $s = (h + rd) / k$, so w is equal to $k / (h + rd) \; mod \; n$.
    2. The verifier calculates u by multiplying w and h.
       - $wh = hk \, (h + rd) = u$
    3. The verifier calculates v by multiplying w and r.
       - $wr = rk(h + rd) = v$
    4. Now, the verifier uses the following formula to calculate the point Q.
       - $Q = uG + vP$
         Where P is the signer's public key, defined as $P = dG$. To accept the signature as valid, the verifier checks that the coordinate x of Q is equal to the value r in the signature.

ECC

Elliptic Curve Digital Signature Algorithm (ECDSA)

The standard algorithm for signatures using ECC is the Elliptic Curve Digital Signature Algorithm (ECDSA). It has replaced RSA and DSA signatures in many applications, is the only signature algorithm used in Bitcoin, and is supported by many TLS and SSH implementations.

- **Signature verification**
  - The process of validating a signed message using a public key to verify its accuracy.

5. Replace the public key P with the actual value dG to compute the point Q.
   - $uG + vdG = (u + vd)G$
6. Replace u and v with their actual values.
   - $u + vd = hk\ (h + rd) + drk\ /\ (h + rd) = (hk + drk)\ /\ (h + rd) = k\ (h + dr)\ /\ (h + rd) = k$

This shows that $(u + vd)$ is equal to the value of k chosen during signature generation, and that the point kG is equal to $uG + vdG$. The verification algorithm succeeds in computing the same point kG that was computed during signature generation. The verifier checks for validity by verifying that the x coordinate of kG is the same as the received r value, otherwise the signature is rejected as invalid.

04

Integrity

- Hash function overview
- MD5
- SHA-1, SHA-2, SHA-3

Hash function overview

The concept of hash functions originated in the 1950s, but even before the term "hash" was coined, the same functions were developed to transform and compress data for efficient storage and retrieval.

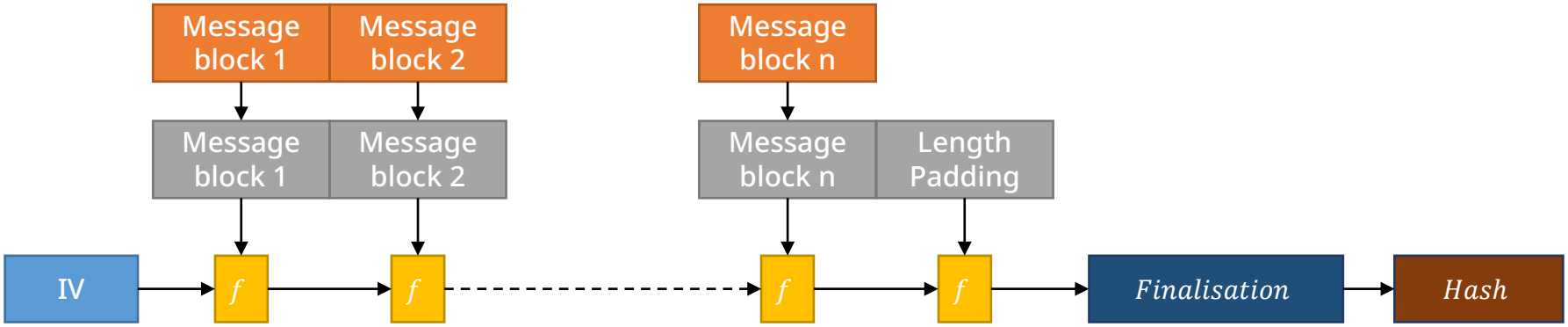
- A cryptographic hash function takes in a random message and outputs a fixed-length message digest.
  - The value obtained by a hash function is called a hash value, hash code, hash checksum, or hash.
- Used in computer software for very fast data retrieval
- Can speed up database searches and table searches
- Used to verify the integrity of transmitted data and as an HMAC block to prove the sender
- The hash function quality is determined by the probability of hash collisions in the input domain.
  - The higher the probability of collisions, the harder it is to distinguish between different data, and the more expensive it is to search.
- Distinction between cryptographic and non-cryptographic hash functions
  - Cryptographic hash functions : MD5, SHA series functions
    - Must be secure against reverse and secondary phases, and collision pairs; used for authentication
  - Non-cryptographic hash functions : CRC32

Hash function overview

Merkle-Damgard construction

The concept of hash functions originated in the 1950s, but even before the term "hash" was coined, the same functions were developed to transform and compress data for efficient storage and retrieval.

- Iterative hash functions
  - General structure
    - Framework for cryptographic hash function structures
    - Takes in bits and outputs bits
    - $f: \{0,1\}^m \times \{0,1\}^n = \{0,1\}^m$
    - However, this f should be such that it is difficult to get the two inputs from the output and to know the different m_1, m_2 that satisfy $f(m_1) = f(m_2)$.

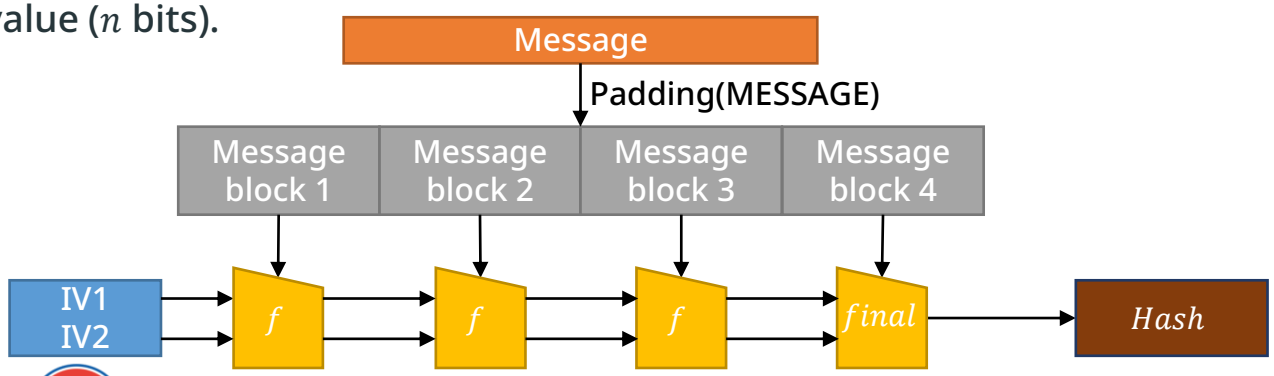


Hash function overview

Merkle-Damgard construction

The concept of hash functions originated in the 1950s, but even before the term "hash" was coined, the same functions were developed to transform and compress data for efficient storage and retrieval.

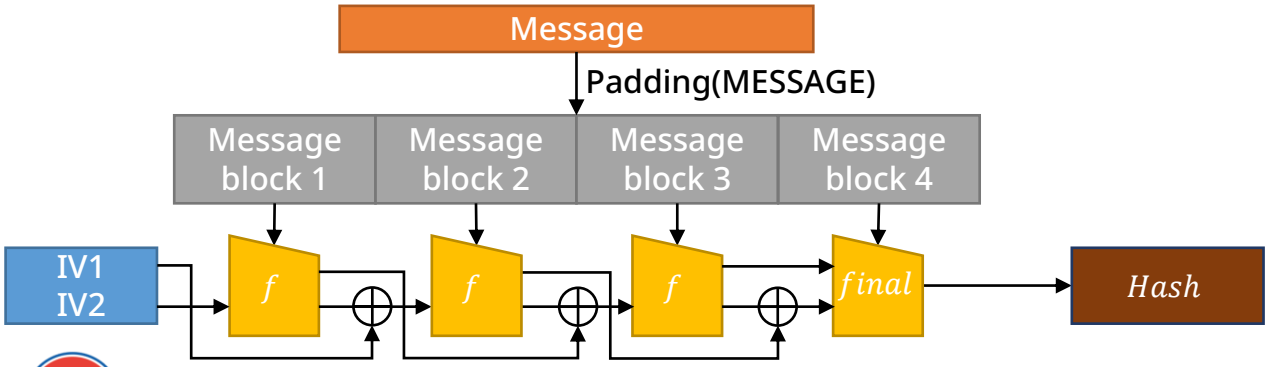
- Iterative hash functions
  - Wide pipe structure
    - Structural weaknesses and multiple collision attacks in the basic structure led to the development of the wide pipe hash structure.
    - Similar to the Merkle-Damgard configuration, but with a larger internal state.
      - This means that the bit length used internally is much larger than the output bits.
      - If n bits of hash are needed, f compresses $2n$ bits of concatenated values and m bits of messages into a $2n$-bit output.
      - In the last step, the second compression compresses the internal hash value ($2n$ bits) into a final hash value (n bits).



Hash function overview

The concept of hash functions originated in the 1950s, but even before the term "hash" was coined, the same functions were developed to transform and compress data for efficient storage and retrieval.

- Iterative hash functions
  - Quick wide pipe structure
    - Algorithm that roughly doubles the speed of wide-pipe hash functions
    - Half goes into the subsequent compression function, and the other half goes into the next compression function.
    - Combine with the output of the corresponding compression function
    - XOR half of the previous concatenation value and pass it as the output of the compression function
    - Use longer message blocks for each iteration

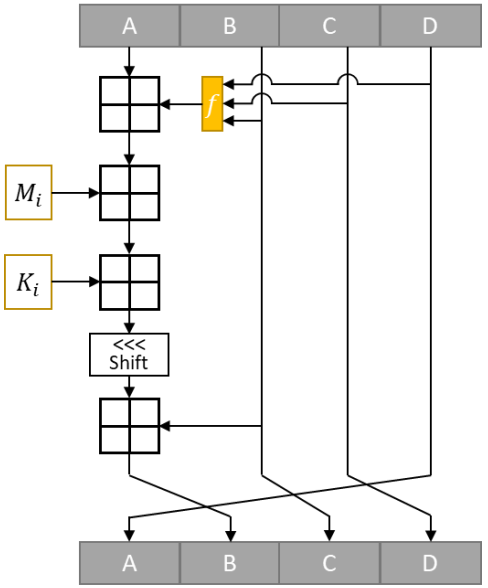


MD5

Overview

MD5 is a 128-bit cryptographic hash function specified as RFC1321 and used for integrity checks, such as verifying that a program or file is original. A design flaw was discovered in 1996, and a hash collision was discovered in 2006 using the computing power of a single laptop in less than a minute, making it an obsolete hash algorithm today.

- Take in a random-length message and output a 128-bit-length value
  - Input messages are split into 512-bit blocks
  - Pad messages first and divide by 512 so they fall apart
  - Add the first single bit, 1, to the end of the message
  - Fill with zeros up to 64 bits less than the length of a multiple of 512
  - Fill the remaining 64 bits with an integer equal to the original message length.
  - A single message block is processed in four steps
    - Each step is called a round
    - Nonlinear function f, modulo addition, and left rotation
    - There are four F-functions, and a different F is used for each round.
      - $F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge z)$ $\quad G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$
      - $H(X,Y,Z) = X \oplus Y \oplus Z$ $\quad I(X,Y,Z) = Y \oplus (X \vee \neg Z)$

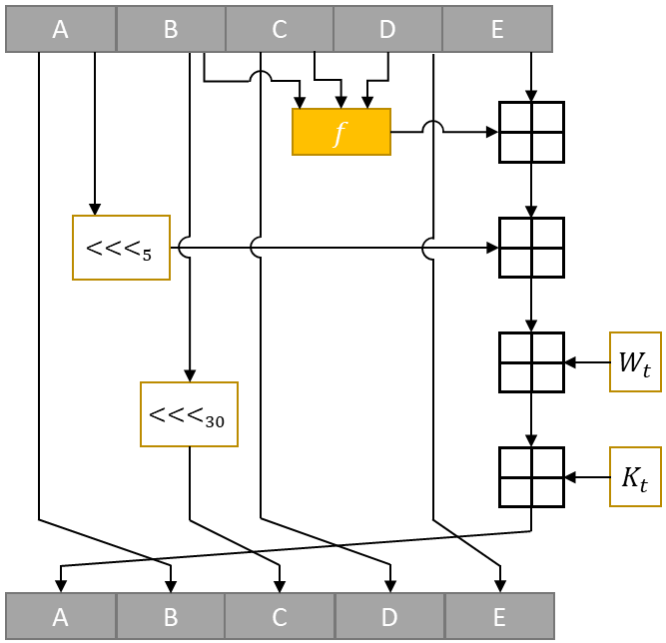


⊕ : XOR
∧ : logical conjunction
∨ : logical disjunction
¬ : NOT
⊞ : modulo 2^{32}

SHA-1, SHA-2, SHA-3

The Secure Hash Algorithm (SHA) is a standard developed by NIST based on MD5 and published as FIP 180. It was revised as FIP 180-1, which included SHA-1, and revised again as FIP 180-2, which included versions of SHA-224, SHA-256, SHA-384, and SHA-512.

- Recognizing that SHA-1's strong collision resistance was broken in 2005, NIST began the process of selecting a new one-way hash function in 2007.
  - Competitive bidding process, such as in the AES selection process.
  - In 2012, the selection was finalized and an algorithm called Keccack was chosen as the final standard.
    - This is the current SHA-3.
  - Why NIST selected Keccack as SHA-3
    - Totally different structure than SHA-2
    - Transparent design and easy to analyze
    - Work well on a variety of devices and in any combination
    - High performance when embedded in hardware
    - Better security than the last competing algorithm



SHA-1, SHA-2, SHA-3

The Secure Hash Algorithm (SHA) is a standard developed by NIST based on MD5 and published as FIP 180. It was revised as FIP 180-1, which included SHA-1, and revised again as FIP 180-2, which included versions of SHA-224, SHA-256, SHA-384, and SHA-512.

- NIST first published FIPS 180 as the secure hash standard, SHA, which is referred to as SHA-0 to distinguish it from other functions.
- After some time, FIPS 180 was abolished and FIPS 180-1 (SHA-1) was published
  - Add a one-bit rotation operation to the compression function of SHA-0
  - Address issues in the original algorithm that reduced cryptographic security
    - However, it did not disclose what the problem actually was.
  - SHA-1 is generally known to be more difficult to attack cryptographically than SHA-0.
  - SHA-0 and SHA-1 generate a 160-bit hash value from a message of up to 2^{64} bits.
    - Based on methods similar to those used in the MD4 and MD6 hash functions

SHA-1, SHA-2, SHA-3

The Secure Hash Algorithm (SHA) is a standard developed by NIST based on MD5 and published as FIP 180. It was revised as FIP 180-1, which included SHA-1, and revised again as FIP 180-2, which included versions of SHA-224, SHA-256, SHA-384, and SHA-512.

- NIST later published four variants with longer hash values (collectively referred to as SHA-2)
  - SHA-256, SHA-384, and SHA-512 first published as drafts in 2001
  - Designated as a formal standard along with SHA-1 in 2002 (FIPS 180-2)
  - Added to the SHA-224 standard in 2004 to match the hash length to the key length of triple DES.
  - SHA-256 and SHA-512 are hash functions that use 32-byte and 64-byte words, respectively.
    - Some constants are different, but the structure is exactly the same except for the number of rounds
  - SHA-224, SHA-384 are SHA-256 and SHA-512 hash values computed with different initial values and truncated to fit the final hash value length.
- Collision resistance of hashes
  - Weak collision resistance : for a given x, collision resistance is weak when it is difficult to find $y \neq x$ such that $H(x) = H(y)$.
  - Strong collision resistance : if x and y are found to have the same hash output value, such that $H(x)=H(y)$.

SHA-1, SHA-2, SHA-3

Algorithm		Size	Internal size	Block size	No. of courses	List of operations used	Security strength	Crash
MD5		128	128 (4*32)	512	64	+, and, xor, rot, add, or	<64	Found
SHA-0		160	160 (5*32)	512	80	+, and, or, xor, rotl	<80	Found
SHA-1		160	160 (5*32)	512	80	+, and, or, xor, rotl	<63	Found
SHA-2	SHA-224	224	256 (8*32)	512	64	+, and, or, xor, shr, rotr	112	
	SHA-256	256		512			128	
	SHA-384	384	512 (8*64)	1024	80	+, and, or, xor, shr, rotl	192	
	SHA-512	512					256	
	SHA-512/224	224					112	
	SHA-512/226	256					128	
SHA-3	SHA3-224	224	1600 (5*5*64)	1152	24	+, and, xor, rot, not	112	
	SHA3-256	256		1088			128	
	SHA3-384	384		832			192	
	SHA3-512	512		576			256	
	SHAKE128	d(variable)		1344			min(d|2,128)	
	SHAKE256	d(variable)		1088			min(d|2,256)	

05

Lab

- Break Caesar cipher
- Break DES
- Break hash with Hashcat

Break Caesar cipher

The Caesar cipher is a relatively easy ciphertext to crack. There are two main ways to break this ciphertext.

- Exhaustive search attack
  - Decrypt the ciphertext "P svcl Jyfwavnyhwof" when Eve intercepted it.

K	1	O rubk Ixevzumxgvne
	2	N qtaj Hwduytlwfumd
	3	M pszi Gvctxskvetlc
	4	L oryh Fubswrjudskb
	5	K nqxg Etarvqitcrja
	6	J mpwf Dszquphsbqiz
	7	I love Cryptography

  - "I love Cryptography" was obtained when running an exhaustive search attack.

Break Caesar cipher

The Caesar cipher is a relatively easy ciphertext to crack. There are two main ways to break this ciphertext.

- Statistical attack
  - In English, the alphabet is used in the following order of frequency : E (12.7%), T (9.1%), A (8.2%), O (7.5%), and N (7.0%).
  - Characters that occur frequently in ciphertext are more likely to be E, T, A, O, and N in plaintext.
    - Analyze frequency to infer plaintext counterparts to ciphertext.
    - It's a probability, so it's possible that it’s not.
  - Frequency of common plaintext
    - 2-character frequency : TH, HE, and IN are the most common.
    - 3-character frequency : THE, and ING are the most common.

Break Caesar cipher

The Caesar cipher is a relatively easy ciphertext to crack. There are two main ways to break this ciphertext.

- Statistical attack
  - Ciphertext
    - juhhwlqj hyhubrqh zhofrph wr dfv lqirupdwlrq vhfxulwb zruog
    - Frequency of occurrence of characters in the given ciphertext
      - h = 8
      - r = 6
      - w = 4
      - Substituting the most frequent character h for e in the plaintext indicates that the key is 3.
    - Decrypted plaintext
      - greeting everyone welcome to acs information security world

Break DES

- OpenSSL
  - Open source implementations of TLS and SSL, used for communicating data over networks.
    - The core library is written in C.
      - Implement basic encryption and various utility functions
    - Supported encryption algorithms
      - AES
      - Blowfish
      - DES/T-DES
      - IDEA
      - RC4
    - Supported hash functions
      - MD5
      - SHA-1
      - MDC-2

Break DES

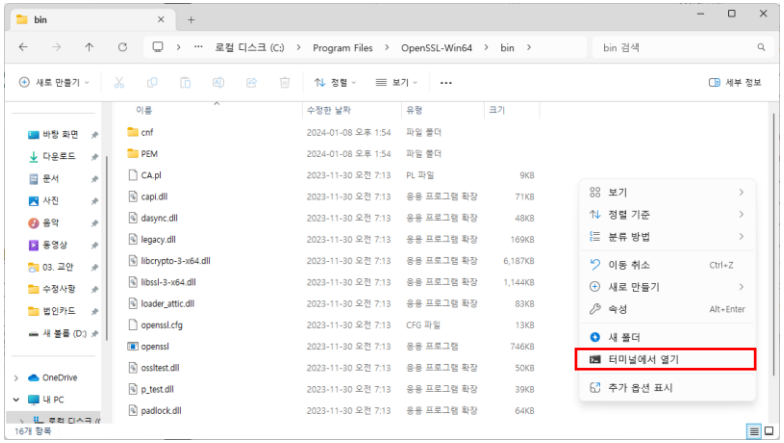
- OpenSSL
  - Installed by default on Linux/Mac, but requires installation on Windows
  - Access via the following link.
    - https://slproweb.com/products/Win32OpenSSL.html
    - Download "Win64 OpenSSL v3.2.0"-exe among several other versions.

Download Win32/Win64 OpenSSL

Download Win32/Win64 OpenSSL today using the links below!

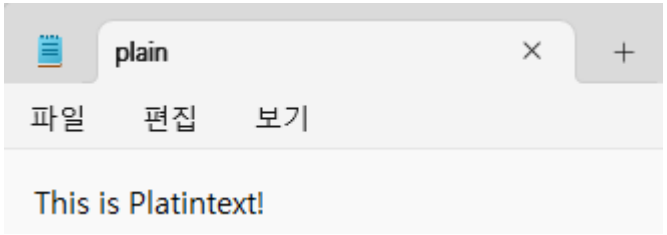
File	Type	Description
Win64 OpenSSL v3.2.0 Light EXE | MSI	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.2.0 (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.2.0 EXE | MSI	200MB Installer	Installs Win64 OpenSSL v3.2.0 (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.2.0 Light EXE | MSI	4MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v3.2.0 (Only install this if you need 32-bit OpenSSL for Windows). Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.

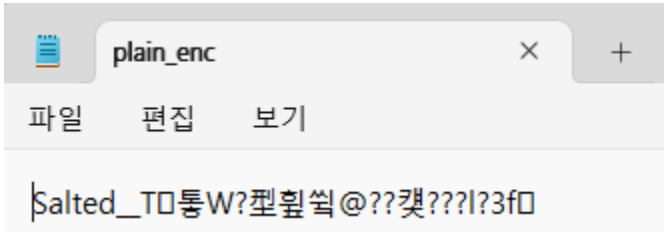
- Access the below path from the C drive after installation.
  - C:\Program Files\OpenSSL-Win64\bin
  - Right click to open in Terminal.



Break DES

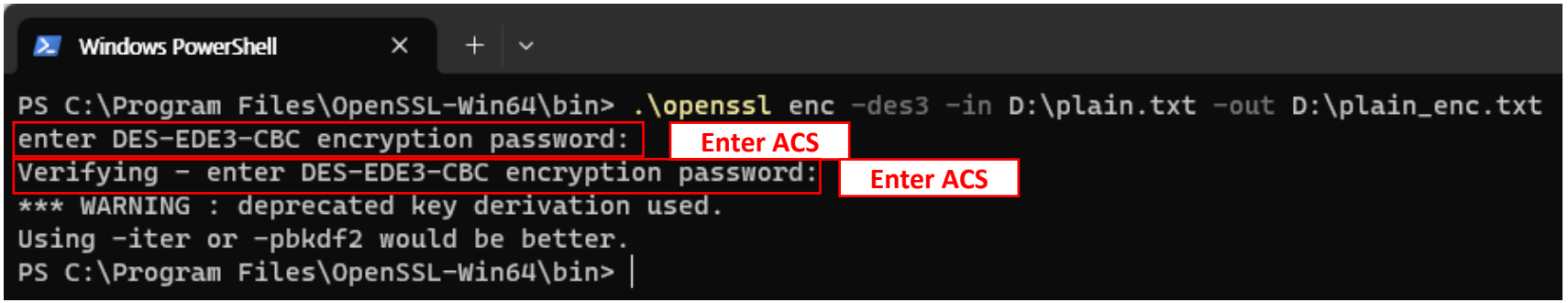
- OpenSSL-DES encryption/decryption
  - Write and save text in Notepad (save as plain.txt).
    - Type “It is Plaintext!”





  - Type commands in Terminal.

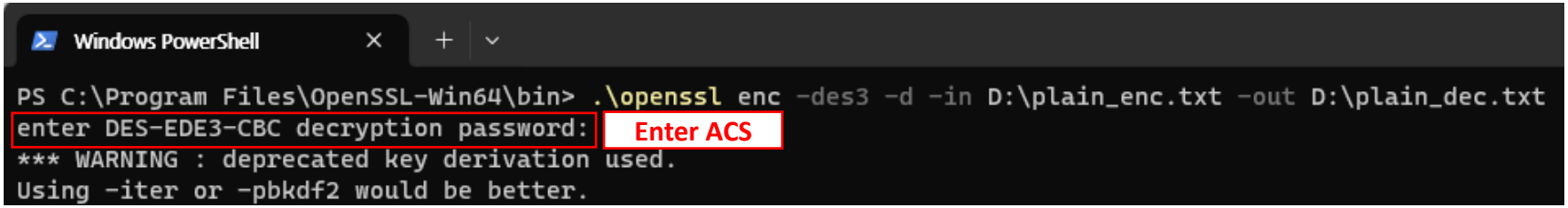
```
PS C:\> .\openssl enc -des3 -in D:\plain.txt -out D:\plain_enc.txt
```

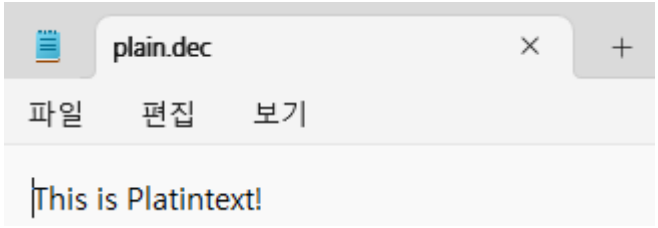


Break DES

- OpenSSL-DES encryption/decryption
  - Decrypt the encrypted Notepad file.

```
PS C:\> .\openssl enc -des3 -d -in D:\plain_enc.txt -out D:\plain_dec.txt
```





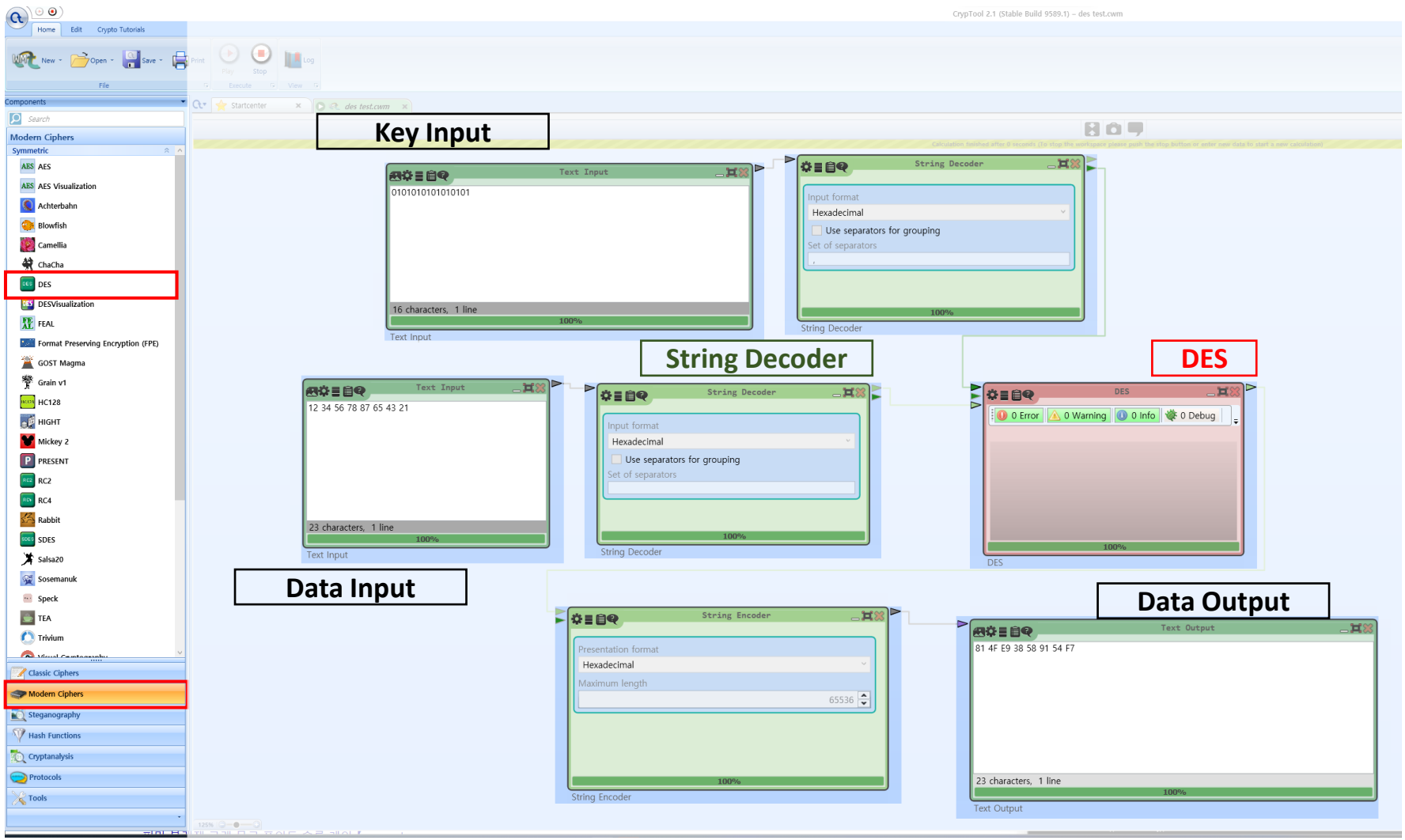
Break DES

Cryptool is an open source, free e-learning software for learning cryptography and cryptanalysis concepts. According to the IT security magazine Hakin9, Cryptool is a globally popular software in the field of cryptography.

- Cryptool
  - Offer e-learning program for visualization of cryptography and cryptanalysis
  - Cryptool implements more than 400 cryptographic algorithms
  - Provide contemporary symmetric and asymmetric ciphers, including RSA, ECC, digital signatures, and hybrid encryption, as well as classical ciphers
  - Classical ciphers include solvers (analyzers) in addition to algorithms.
  - Download link : https://www.cryptool.org/en/ct2/

Break DES

Cryptool2



Break DES

- Weak keys in DES
  - The vulnerability is that the operation after parity stripping is either all 0s, all 1s, or half 0s and half 1s.

Keys before parities drop (64 *bits*)	*Actual Key* (56 *bits*)
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 1F1F 1F1F	0000000 FFFFFFF
E0E0 E0E0 E1E1 E1E1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

  - If you encrypt a block with a weak key and encrypt the result with the same weak key,
    - You get the original block.
    - You get the same result even when the block is decrypted twice.

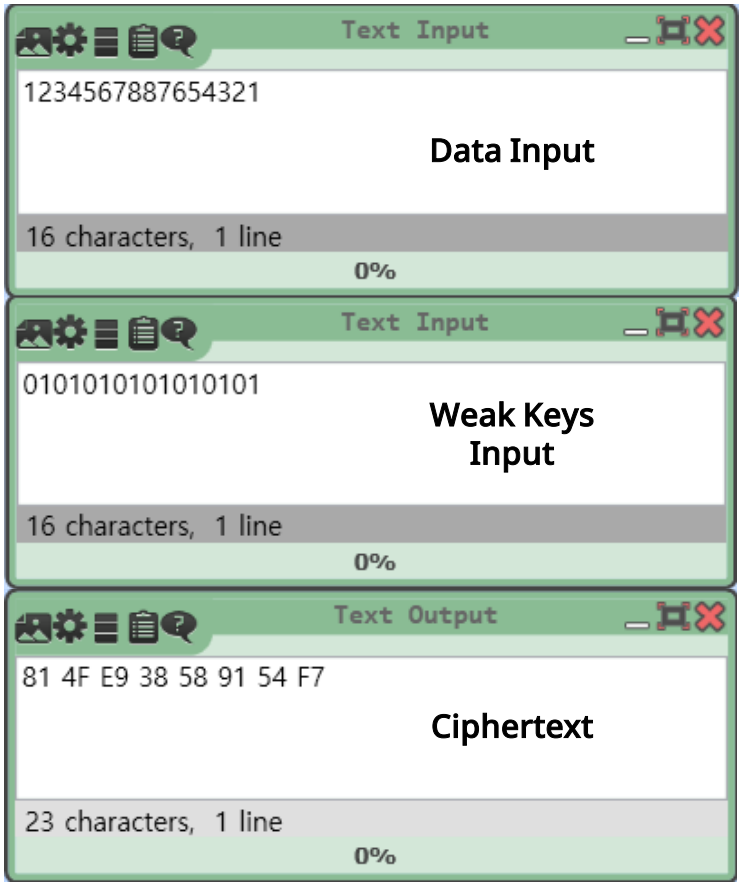
Break DES

- Create a ciphertext using the DES you created in the previous section.
  - Weak key encryption test
    - Data : 12 34 56 87 65 43 21
    - Key : 01 01 01 01 01 01 01 01
    - Ciphertext : 81 4F E9 38 58 91 54 F7
  - Paste the encryption result from here back into the input of the Encrypt command.
    - Data : 81 4F E9 38 58 91 54 F7
    - Key : 01 01 01 01 01 01 01 01
    - Ciphertext : 12 34 56 78 87 65 43 21
  - Using the weak key, we can see that the original message is output with only two encryptions.

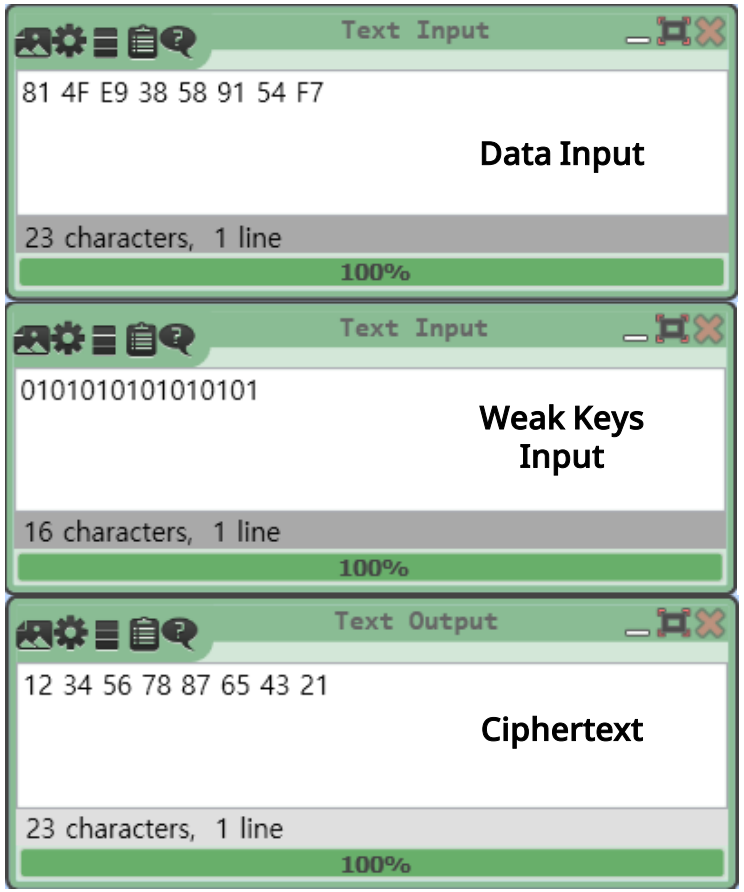
Break DES

DES weak-key encryption

First encryption



Second encryption



Break DES

- Semi-weak keys in DES
  - Six key pairs called semi-weak keys

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 1FE0 1FE0	E01F E01F E01F E01F
01E0 01E0 01E0 01E0	E001 E001 E001 E001
1FFE 1FFE 1FFE 1FFE	FE1F FE1F FE1F FE1F
011F 011F 011F 011F	1F01 1F01 1F01 1F01
E0FE E0FE E0FE E0FE	FEE0 FEE0 FEE0 FEE0

  - Semi-weak keys generate only two forms of the round key, each repeated eight times.
  - Two semi-weak keys paired as one produce the same round key, just in a different order.

Break DES

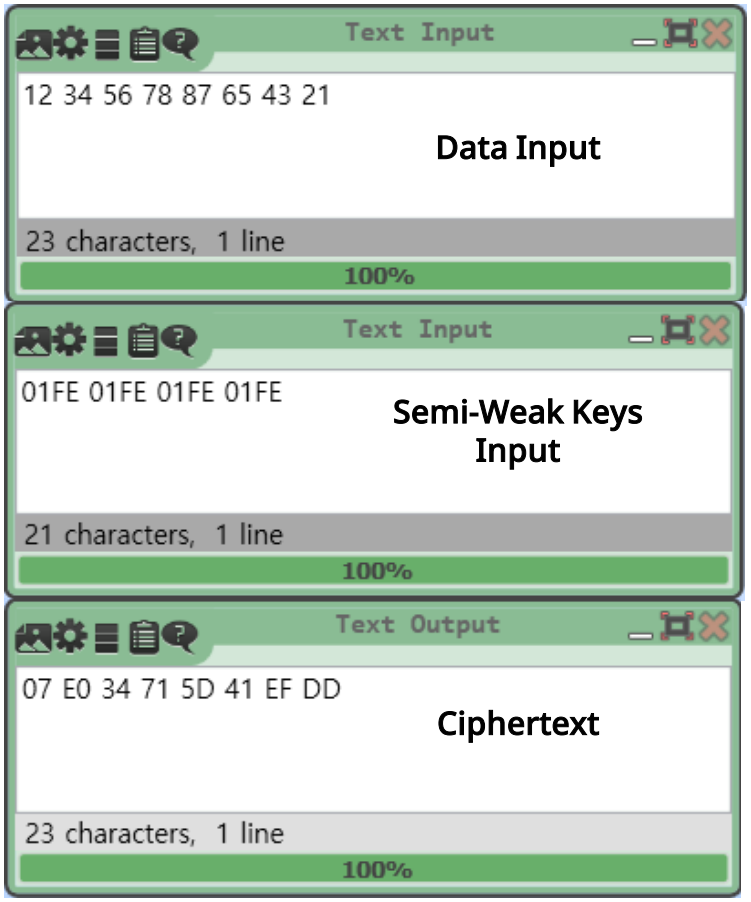
DES semi-weak-key encryption

- Create a ciphertext using the DES created in the previous section.
  - Perform an encryption test with semi-weak keys
    - Data : 12 34 56 87 65 43 21
    - Key : 01 FE 01 FE 01 FE 01 FE
    - Ciphertext : 07 E0 34 71 5D 41 EF DD
  - Put the result of this encryption back as input and encrypt it again with the semi-weak key pair.
    - Data : 07 E0 34 71 5D 41 EF DD
    - Key : FE 01 FE 01 FE 01 FE 01
    - Ciphertext : 12 34 56 78 87 65 43 21
  - See the source message printed when performing encryption using different keys instead of the same key.

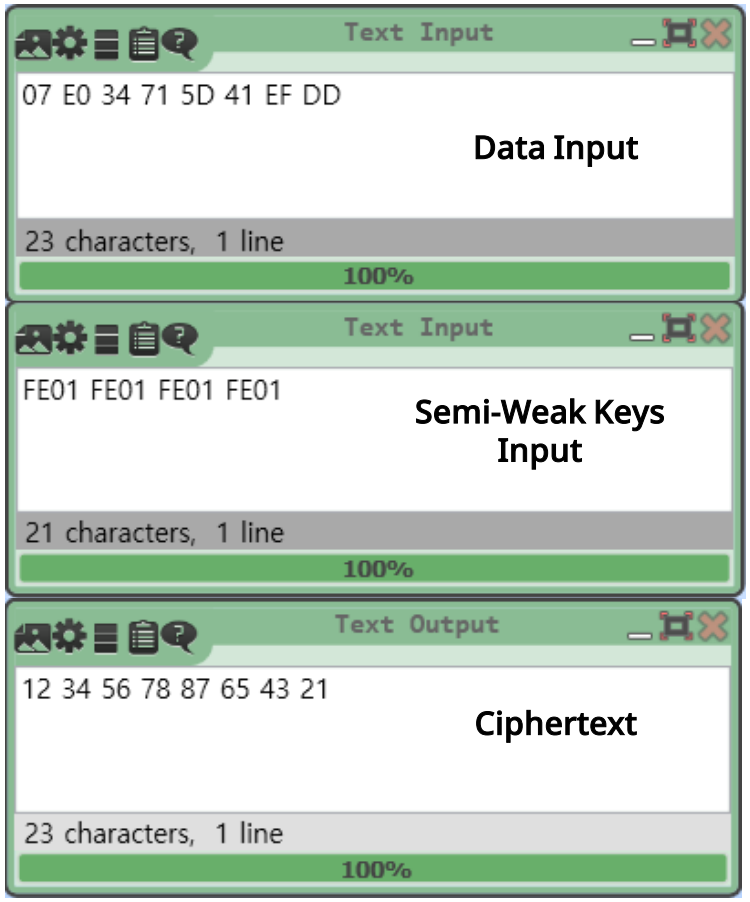
Break DES

DES semi-weak-key encryption

First encryption



Second encryption



Break hash with Hashcat

Hashcat is a password recovery tool and open source software. Algorithms that can be cracked with Hashcat include LM hash, MD4, MD5, and the SHA family.

- Hashcat is a CPU-based password recovery tool
  - GPU-enabled variants of oclHashcat/cudaHashcat also exist
  - Based on flaws in other software discovered by Hashcat's creators
  - Many algorithms supported by legacy Hashcat can be cracked in a fraction of the time with GPU-based Hashcat.
  - Not all algorithms are GPU accelerated.
    - Bcrypt : not available due to factors such as data-dependent branching, serialization, and memory.

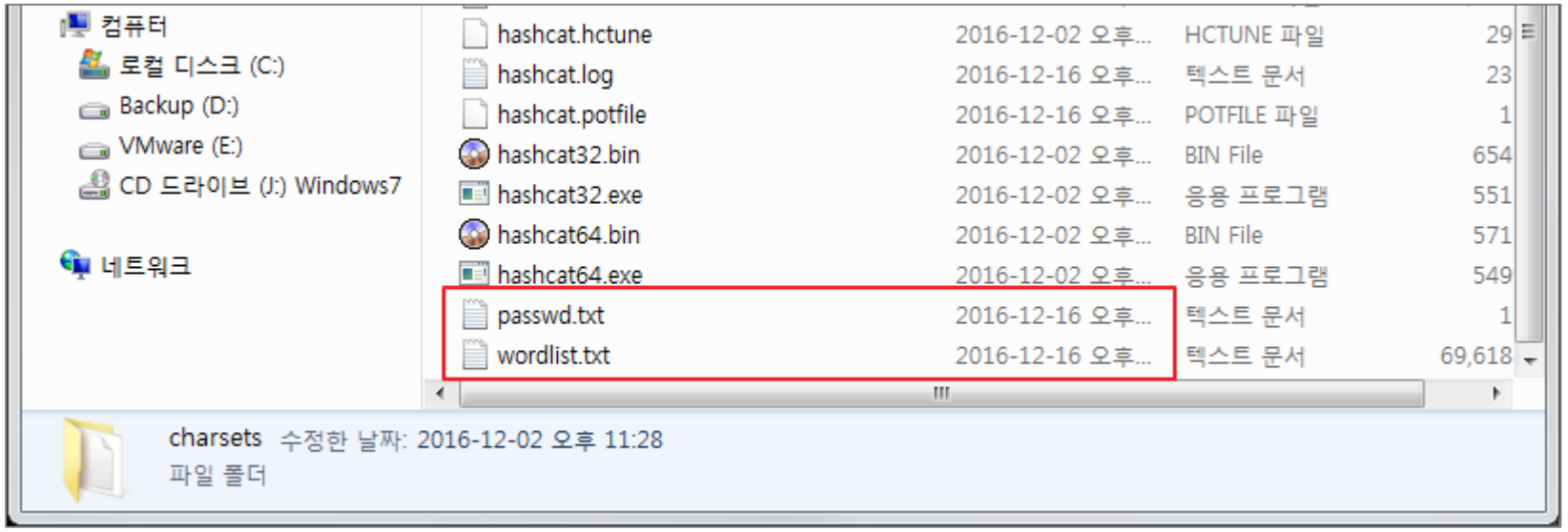
Break hash with Hashcat

Hashcat is a password recovery tool and open source software. Algorithms that can be cracked with Hashcat include LM hash, MD4, MD5, and the SHA family.

- Types of attacks supported by Hashcat
  - Brute force/dictionary attack
  - Combinator attack
  - Fingerprint attack
  - Hybrid attack
  - Mask attack
  - Permutation attack
  - Rule-based attack
  - Table-lookup attack (CPU only)
  - Toggle-case attack
  - PRINCE attack (in CPU version 0.48 or later only)

Break hash with Hashcat

- How to crack with Hashcat
  - Download a cracking tool, Hashcat.
    - Virtual machine (X)
      - Not used because GPUs are hard to use in virtual machines
  - Copy the created dictionary file you created (wordlist.txt) and the extracted WordPress password (passwd.txt).



Break hash with Hashcat

- How to crack with Hashcat
  - Start cracking as follows :
  - Crack command can be found at https://hashcat.net/wiki/doku.php?id=hashcat

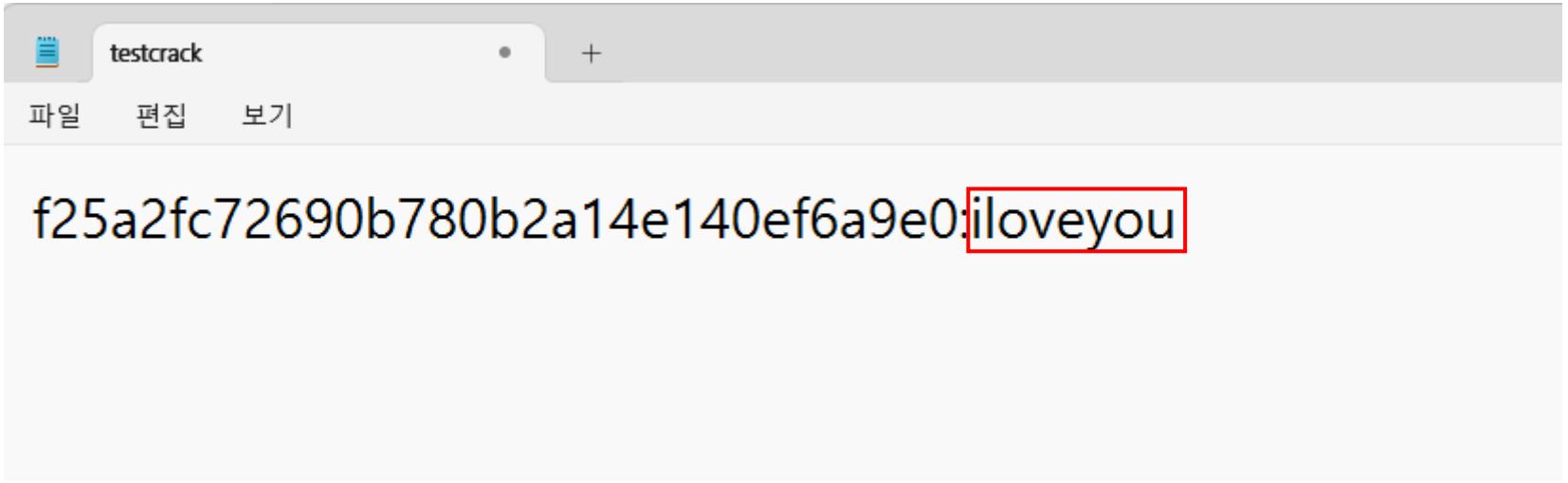
```
hashcat.exe -a 3 -m 0 -d 2 test.txt -o testcrack.txt
```

```
Session..........: hashcat
Status...........: Running
Hash.Type........: phpass, MD5(Wordpress), MD5(phpBB3), MD5(Joomla)
Hash.Target......: $P$BwBGCV1t7piyvdrer7hCLAITAi6w0y.
Time.Started.....: Wed Mar 08 16:50:03 2017 (1 min, 12 secs)
Time.Estimated...: Wed Mar 08 16:53:39 2017 (2 mins, 24 secs)
Input.Base.......: File (wordlist.txt)
Input.Queue......: 1/1 (100.00%)
Speed.Dev.#3.....:    55042 H/s (8.99ms)
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 3919872/11881376 (32.99%)
Rejected.........: 0/3919872 (0.00%)
Restore.Point....: 3919872/11881376 (32.99%)
Candidates.#3....: ipaqi -> irphf
HWMon.Dev.#3.....: Temp: 81c

[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>
```

Break hash with Hashcat

- How to crack with Hashcat
  - When the crack is complete, a crack.txt is generated that stores the hash and password.



  - Alternatively, you can use the following command to check.

```
hashcat.exe -a 3 -m 0 -d 2 test.txt --show
```

```
PS C:\Users\      \Downloads\hashcat-6.2.5> .\hashcat.exe -a 3 -m 0 -d 2 test.txt --show
f25a2fc72690b780b2a14e140ef6a9e0:iloveyou
PS C:\Users\      \Downloads\hashcat-6.2.5>
```