A.D. 1308
unipg
DIPARTIMENTO
DI MATEMATICA E INFORMATICA

Computer Security: Principles and Practice
Lezione di Crittografia

Introduzione alla Sicurezza Informatica

Chiara Luchini

Anno Accademico 2024-2025

Symmetric Encryption Principles

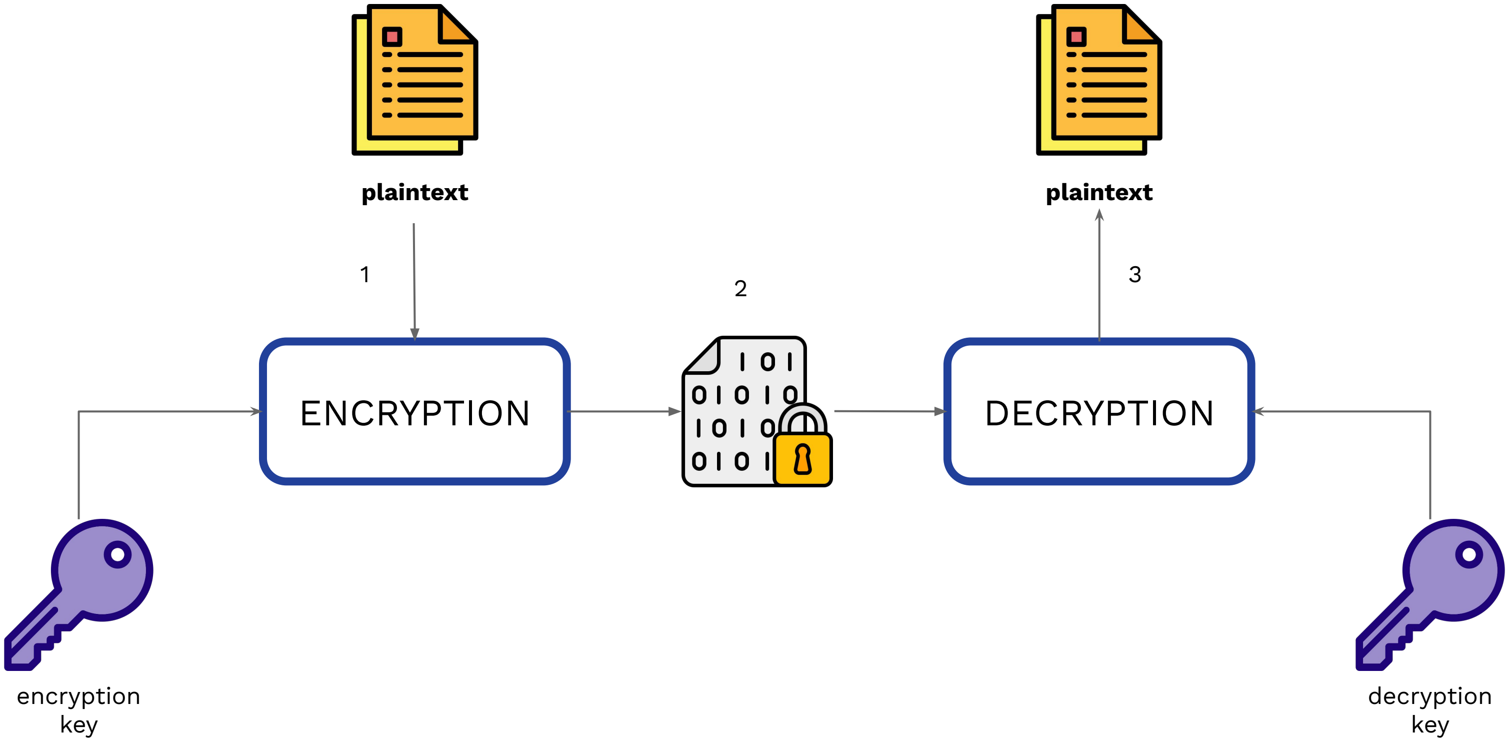
Cryptography

- Ancient science: encryption and decryption information
- Traces dating back to the time of Sparta
- World War II: ENIGMA
- Ancient: symmetric cryptography
- Modern: Asymmetric Cryptography (1977)

Cryptography

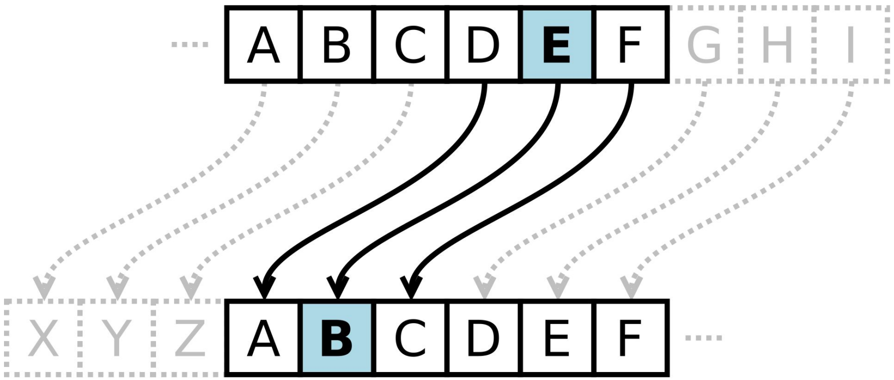
- **Encryption:** plaintext → ciphertext
- **Decryption**: ciphertext → plaintext
- Both based on: **algorithm** e **key**
    - Ex: “Shifting” of k position a string
- Public algorithm!
- Security comes from:
    - secrecy of the key
    - robustness of the algorithm

Encryption and Decryption



Caesar cipher

It is a type of **substitution** cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
Example: key = 23 right shift



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

↓

```
Plaintext:  THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
```

Vigenere Cipher

The Vigenère cipher is a method of **encrypting alphabetic** text where each letter of the plaintext is encoded with a different Caesar cipher, whose **increment** is determined by the **corresponding letter** of another text, the **key**.

```
Testo in chiaro   - RAPPORTOIMMEDIATO
Verme             - VERMEVERMEVERMEVE
Testo cifrato     - MEGBSMXFUQHIUUEOS
```

Attacking Symmetric Encryption

Cryptanalytic Attacks

- Rely on:
  - Nature of the algorithm
  - Some knowledge of the general characteristics of the plaintext
  - Some sample plaintext-ciphertext pairs
- Exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
- If successful all future and past messages encrypted with that key are compromised

Brute-Force Attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
  - On average half of all possible keys must be tried to achieve success

Cryptanalysis

The process of attempting to **discover the plaintext** or **key** is known as cryptanalysis.

- The strategy used by the cryptanalyst depends on the **nature** of the encryption scheme and the **information available** to the cryptanalyst.

Table 2.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext only	• Encryption algorithm • Ciphertext to be decoded
Known plaintext	• Encryption algorithm • Ciphertext to be decoded • One or more plaintext–ciphertext pairs formed with the secret key
Chosen plaintext	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	• Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Cryptanalysis

An encryption scheme is **computationally secure** if the ciphertext generated by the scheme meets one or both of the following criteria:

- The **cost** of breaking the cipher exceeds the **value** of the encrypted information.
- The **time** required to break the cipher exceeds the **useful lifetime** of the information.

-> Difficult to estimate.

Brute force attack (trying all possible keys) -> $\frac{keys}{2}$ tentatives

Feistel Cipher Structure

- Base for many symmetric block encryption algorithms, including DES
- **Symmetric block ciphers:** consists of a **sequence of rounds**, with each round performing substitutions and permutations conditioned by a secret key value.
- It depends on different parameters and design features

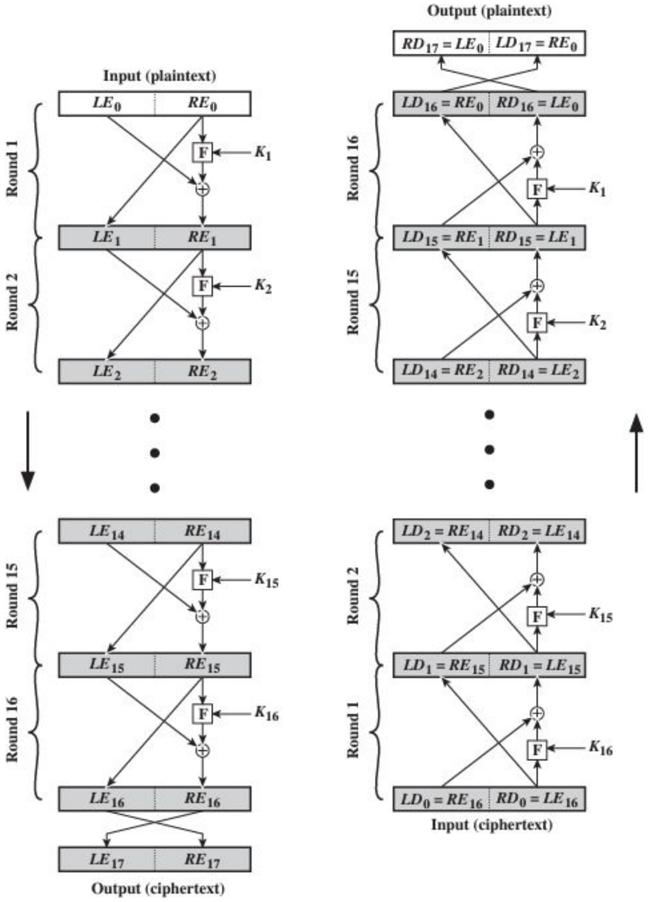


Figure 2.2 Feistel Encryption and Decryption (16 rounds)

Symmetric Block Encryption Algorithms

- Same key for encryption and decryption
- Secrecy, authentication, integrity from the secrecy of the key
- Data Encryption Standard (**DES**), triple DES (**3DES**),and the Advanced Encryption Standard (**AES**).

A **block cipher** processes the plaintext input in **fixed-sized blocks** and produces a block of ciphertext of equal size for each plaintext block.

Comparison of Three Popular Symmetric Encryption Algorithms

	DES	3DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192 or 256

DES= Data Encryption Standard
AES= Advanced Encryption Standard

Symmetric Encryption

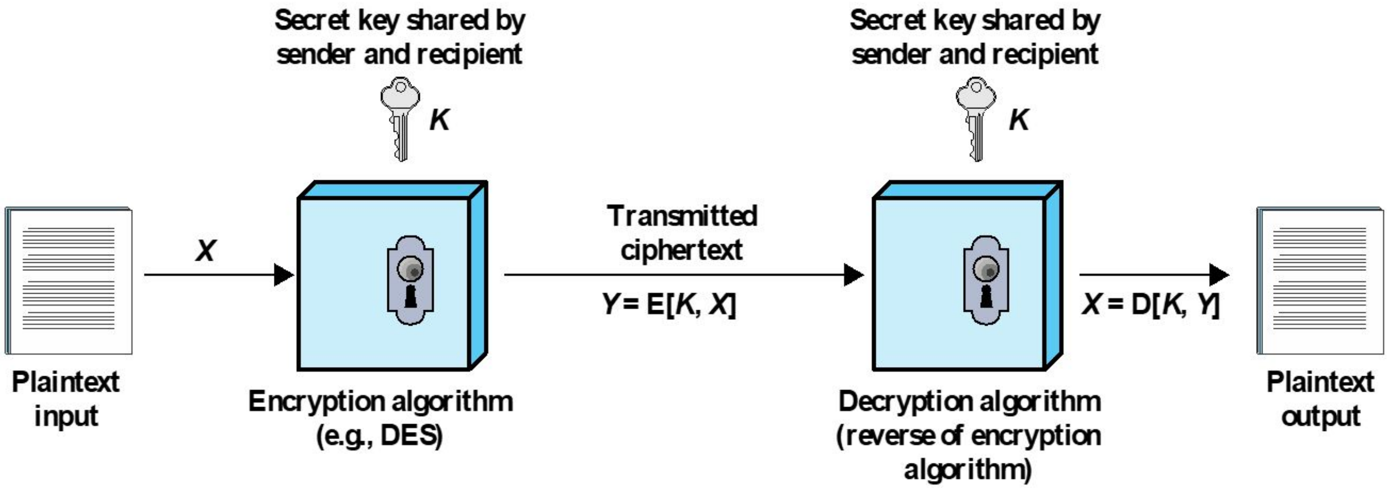


Figure 2.1 Simplified Model of Symmetric Encryption

Data Encryption Standard (DES)

- Until recently was the most widely used encryption scheme
  - FIPS PUB 46
  - Referred to as the Data Encryption Algorithm (DEA)
  - Uses **64 bit plaintext** block and **56 bit key** to produce a **64 bit ciphertext block**
- Strength concerns:
  - Concerns about the **algorithm itself**
    - DES is the most studied encryption algorithm in existence
  - Concerns about the use of a **56-bit key**
    - The speed of commercial off-the-shelf processors makes this key length woefully inadequate

Data Encryption Standard (DES)

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

Average Time Required for Exhaustive Key Search

Triple DES (3DES)

- **Repeats** basic DES algorithm **three times** using either two or three unique keys
- First standardized for use in financial applications in ANSI standard X9.17 in 1985
- Attractions:
  - **168-bit key length** overcomes the vulnerability to brute-force attack of DES
  - Underlying encryption algorithm is the same as in DES
- Drawbacks:
  - Algorithm is **sluggish** in software
  - Uses a **64-bit block size**

3DES

- **encrypt-decrypt-encrypt (EDE)** sequence
- Encryption:
  - C=E(K3,D(K2,E(K1,P)))
- Decryption:
  - P=D(K1,E(K2,D(K3,C)))

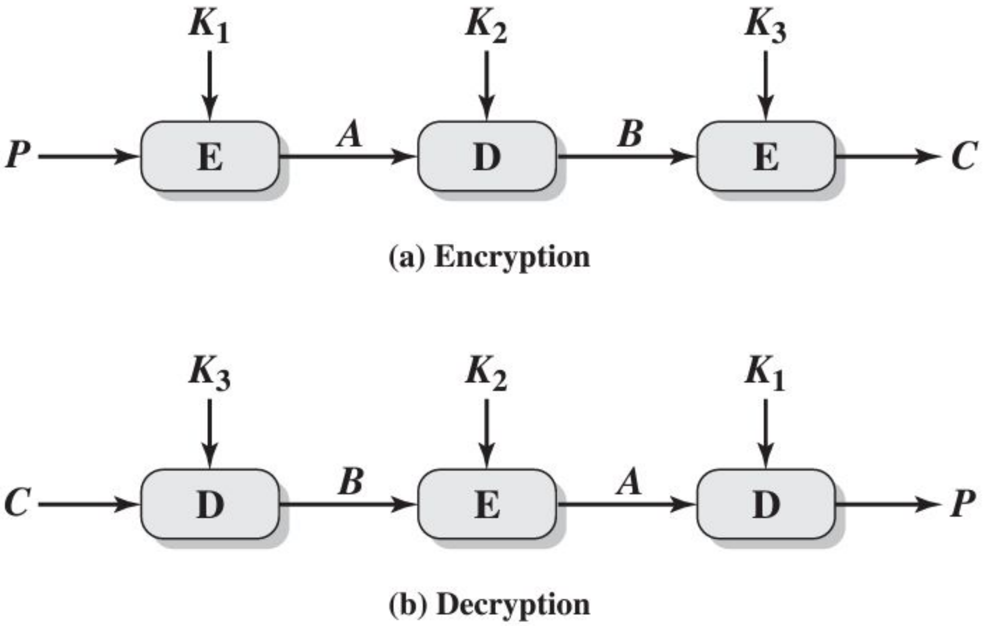


Figure 2.4 Triple DES

Advanced Encryption Standard (AES)

Needed a replacement for 3DES

3DES was not reasonable for long term use

NIST called for proposals for a new AES in 1997

- Should have a security strength equal to or better than 3DES
- Significantly improved efficiency
- Symmetric block cipher
- 128 bit data and 128/192/256 bit keys

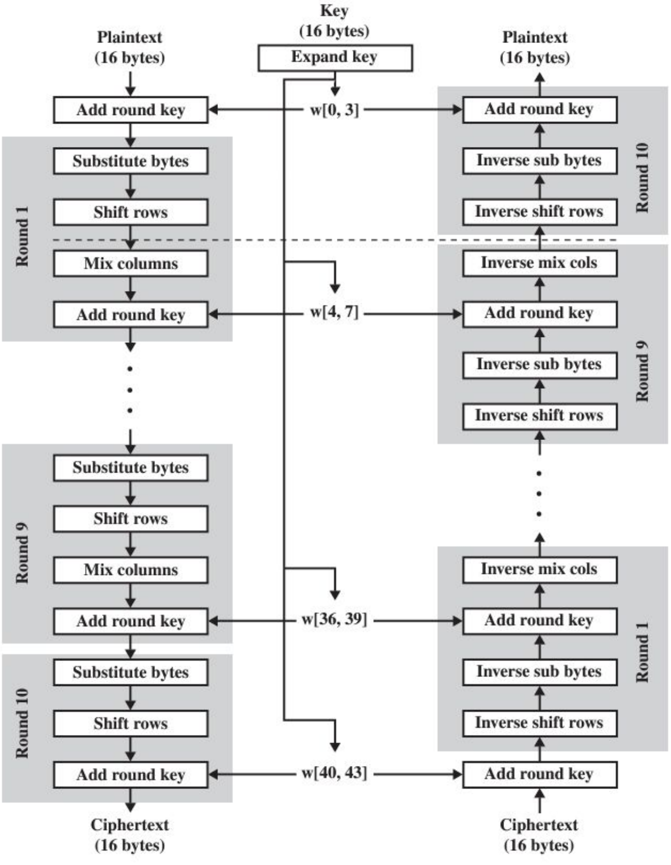
Selected Rijndael in November 2001

Published as FIPS 197

AES algorithm

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

- Input: **State** array
- **Substitute bytes:** Uses a table, referred to as an S-box, to perform a byte-by-byte substitution of the block
- **Shift rows:** A simple permutation that is performed row by row.
- **Mix columns:** A substitution that alters each byte in a column as a function of all of the bytes in the column.
- **Add round key:** A simple bitwise XOR of the current block with a portion of the expanded key.



(a) Encryption
(b) Decryption

Random and Pseudorandom numbers

The Use of Random Numbers

- Generation of keys for RSA and public-key algorithms.
- Generation of a stream key for symmetric stream cipher.
- Generation of a symmetric key for use as a temporary session key.
- Key distribution scenarios (Kerberos)

Randomness & Unpredictability

Randomness

The following criteria are used to validate that a sequence of numbers is random:

- **Uniform distribution:** The distribution of bits in the sequence should be uniform; that is, the frequency of occurrence of ones and zeros should be approximately the same.
- **Independence:** No one subsequence in the sequence can be inferred from the others.

Unpredictability

In reciprocal authentication and session key generation, the requirement is the successive members of the sequence are unpredictable.

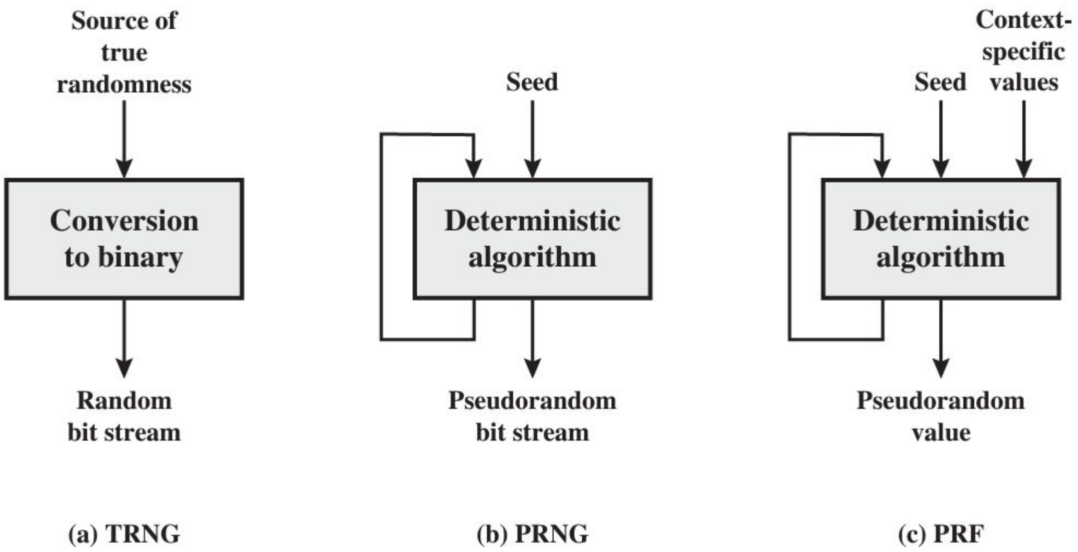
- true random numbers are not always used -> sequences of **“random” numbers** are generated by some algorithm.
- an opponent **not be able to predict** future elements of the sequence on the basis of earlier elements.

TRNGs,PRNGs,and PRFs

Algorithm for generating a random number is **deterministic**
Random numbers -> Pseudorandom numbers

- **True random number generator (TRNG):** takes as input a source that is effectively random (**entropy source**) drawn from the physical environment. Conversion of an analog source to a binary output.
- **Pseudorandom number generator (PRNG)**: An algorithm that is used to produce an open-ended sequence of bits (i.e. input of symmetric stream cipher).
- **Pseudorandom function (PRF):** A PRF is used to produce a pseudorandom string of bits of some fixed length (i.e. symmetric encryption keys and nonces)

TRNGs,PRNGs,and PRFs



TRNG = true random number generator
PRNG = pseudorandom number generator
PRF = pseudorandom function

Figure 2.7 Random and Pseudorandom Number Generators

Public-key cryptography and Message Authentication

Message Authentication

- Protects against active attacks
- Verifies received message is authentic
  - Contents have not been altered
  - From authentic source
  - Timely and in correct sequence
- Can use conventional encryption
  - Only sender and receiver share a key

Message Authentication Without Confidentiality

- Message encryption by itself does not provide a **secure form of authentication**
- It is possible to combine authentication and confidentiality in a single algorithm by **encrypting a message** plus its **authentication tag**
- Typically message authentication is provided as a separate function from message encryption
- Situations in which message authentication without confidentiality may be preferable include:
  - There are a number of applications in which the same message is broadcast to a number of destinations
  - An exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages
  - Authentication of a computer program in plaintext is an attractive service
- Thus, there is a place for both authentication and encryption in meeting security requirements

Message Authentication Code (MAC)

Usage of a secret key to generate small block of data (**MAC**)
Two parties A and B share the same secret key $K_{A,B}$

- A wants to send a msg to B
- A calculates MAC
  - $MAC_M = F(K_{A,B}, M)$
- Msg + MAC_M sent to B
- B calculates MAC_M
- B compares MAC received and calculated

1. The receiver is assured that the msg has not been **altered.**
2. The receiver is assured that the msg is from the **alleged sender**.
3. If the msg includes a sequence number, then the receiver can be assured of the **proper sequence**.

Message Authentication Code (MAC)

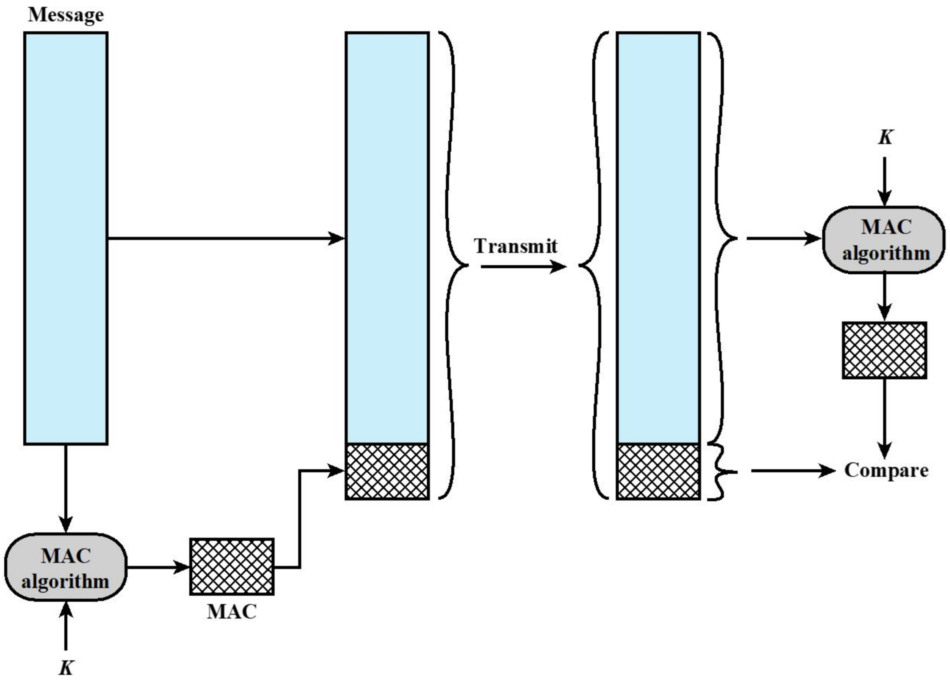


Figure 2.3 Message Authentication Using a Message Authentication Code (MAC).

One-way Hash Function

To be useful for message authentication, a hash function H must have the following properties:

- Can be applied to a block of data of any size
- Produces a fixed-length output
- H(x) is relatively easy to compute for any given x
- **One-way or preimage resistant**
  - Computationally infeasible to find x such that H(x) = h
- **Second preimage resistant** or **weak collision resistance**:
  - Computationally infeasible to find y ≠ x such that H(y) = H(x)
- **Collision resistant** or **strong collision resistance**
  - Computationally infeasible to find any pair (x,y) such that H(x) = H(y)

One-way Hash Function

1. **Symmetric encryption**
   a. The msg digest is encrypted with the key shared only by the sender and receiver (authenticity)
2. **Public-key encryption**
   a. The msg digest is encrypted with the private key of sender
   b. The receiver decipher the encrypted msg with public key of sender
3. **Secret value**
   a. A e B share a secret value S_{AB}
   b. A computes $M_{DM} = H(S_{AB}||M)$ and sends $[M||M_{DM}]$
   c. B computes $H(S_{AB}||M)$ and verifies M_{DM}

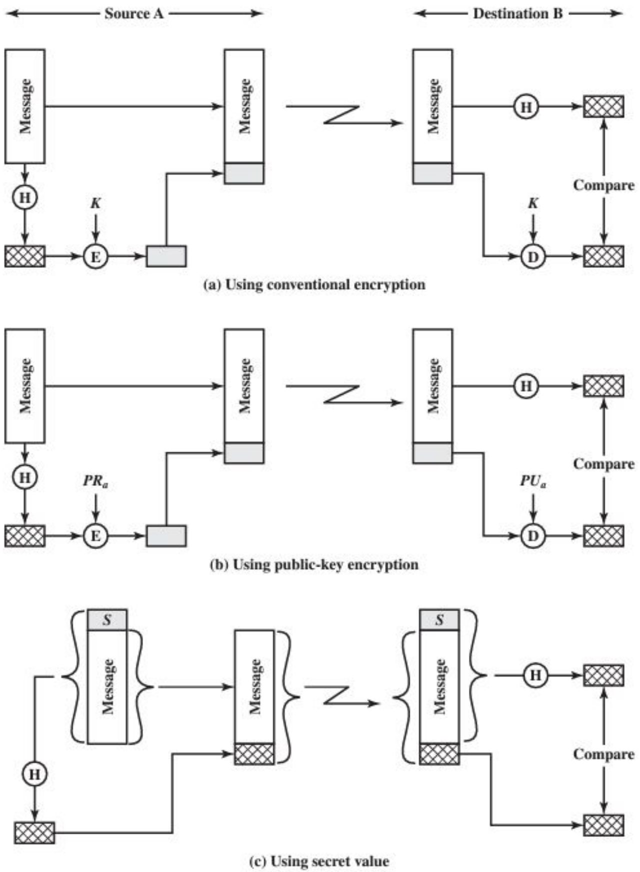


Figure 3.2 Message Authentication Using a One-Way Hash Function

Security of Hash Functions

SHA most widely used hash algorithm

There are two approaches to attacking a secure hash function:

Cryptanalysis
- Exploit logical weaknesses in the algorithm

Brute-force attack
- Strength of hash function depends solely on the length of the hash code produced by the algorithm

Additional secure hash function applications:

Passwords
- Hash of a password is stored by an operating system

Intrusion detection
- Store H(F) for each file on a system and secure the hash values

Public-Key Encryption Structure

Publicly proposed by **Diffie** and **Hellman** in **1976**

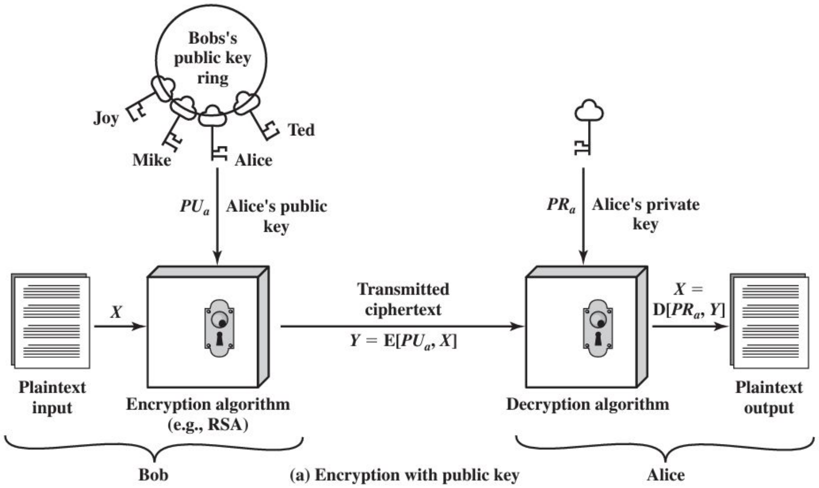
Based on **mathematical functions**

Asymmetric

- Uses two separate keys
- Public key and private key
- Public key is made public for others to use

Some form of protocol is needed for distribution

Public-Key Cryptography



- **Plaintext**
  - Readable message or data that is fed into the algorithm as input
- **Encryption algorithm**
  - Performs transformations on the plaintext
- **Public and private key**
  - Pair of keys, one for encryption, one for decryption
- **Ciphertext**
  - Scrambled message produced as output
- **Decryption key**
  - Produces the original plaintext

Public-Key Cryptography

- User encrypts data using his or her own private key
- Anyone who knows the corresponding public key will be able to decrypt the message

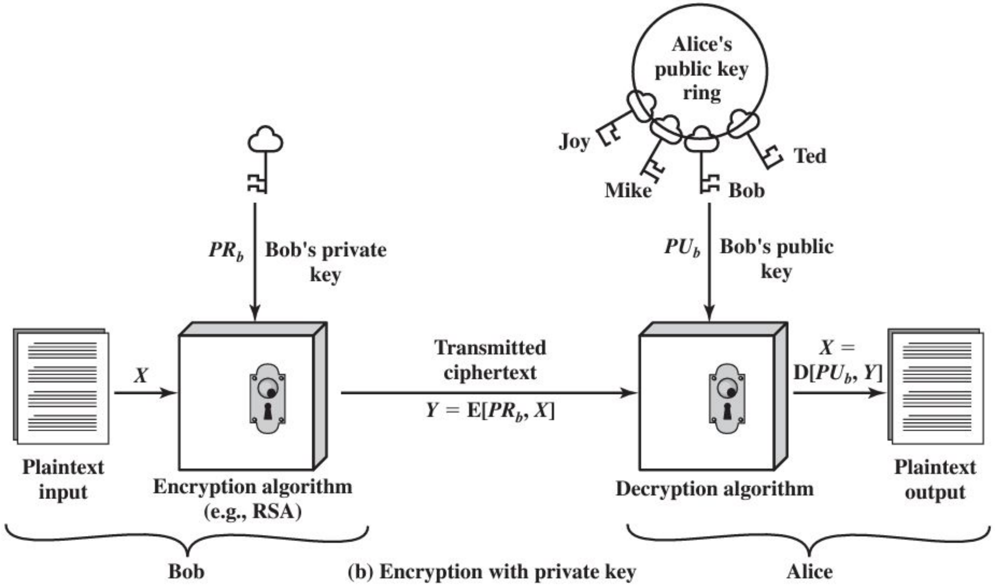


Figure 3.9 Public-Key Cryptography

Applications for Public-Key Cryptosystems

ALGORITHM	DIGITAL SIGNATURE	SYMMETRIC KEY DISTRIBUTION	ENCRYPTION OF SECRET KEY
RSA	Yes	Yes	Yes
Diffie Hellman	No	Yes	No
Digital Signature Standard (DSS)	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

Requirements for Public-Key Cryptosystems

1. Computationally easy to **create key pairs**
2. Computationally **easy** for sender knowing public key **to encrypt messages**
3. Computationally **easy** for receiver knowing private key to **decrypt ciphertext**
4. Computationally **infeasible** for opponent to **determine private key** from public key
5. Computationally **infeasible** for an opponent, knowing the public key, and a ciphertext to **recover the original message**
6. Useful if either key can be used for each role

Asymmetric Encryption Algorithms

RSA (Rivest, Shamir, Adleman)	Developed in 1977	Most widely accepted and implemented approach to public-key encryption	Block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.
Diffie-Hellman key exchange algorithm	Enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages	Limited to the exchange of the keys	
Digital Signature Standard (DSS)	Provides only a digital signature function with SHA-1	Cannot be used for encryption or key exchange	
Elliptic curve cryptography (ECC)	Security like RSA, but with much smaller keys		

RSA (Rivest, Shamir, Adleman)

Encryption and decryption are of the following form period for some plaintext block M and ciphertext block C:

- $C = M^e \bmod n$
- $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

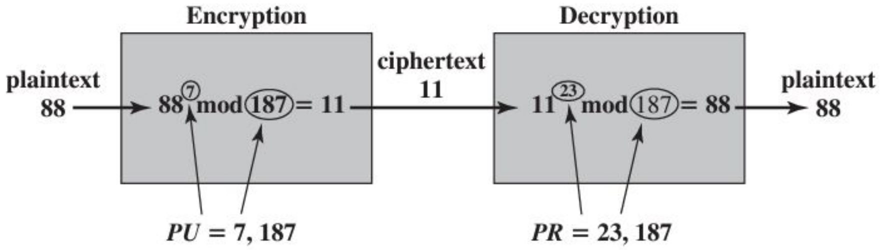


Figure 3.11 Example of RSA Algorithm

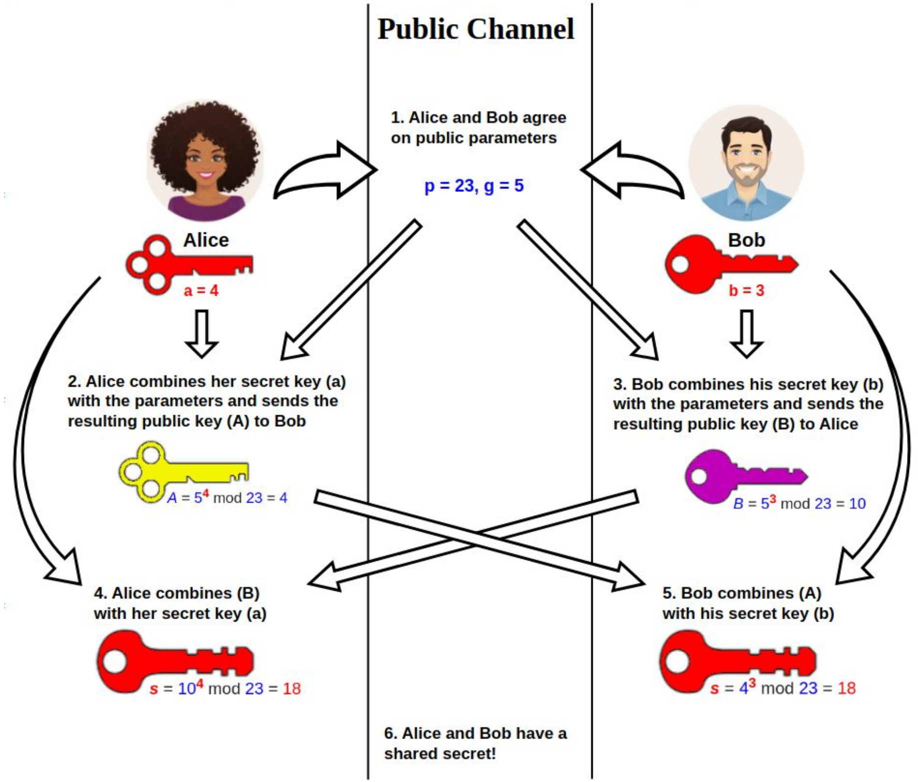
Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

Figure 3.10 The RSA Algorithm

Diffie Hellman Key Exchange



Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Generation of Secret Key by User A

$K = (Y_B)^{X_A} \bmod q$

Generation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

Figure 3.12 The Diffie-Hellman Key Exchange Algorithm

Digital Signatures

- NIST FIPS PUB 186-4 defines a digital signature as:

  "The result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity and signatory non-repudiation."
- Thus, a digital signature is a **data-dependent bit pattern**, generated by an agent as a function of a file, message, or other form of data block
- FIPS 186-4 specifies the use of one of three digital signature algorithms:
  - Digital Signature Algorithm (DSA)
  - RSA Digital Signature Algorithm
  - Elliptic Curve Digital Signature Algorithm (ECDSA)

Digital Signatures

- Based on asymmetric cryptography
- You only get authentication and integrity
- Signing is not exactly encrypting
- Verify that a signature is not exactly decrypting

- **Authenticity:** The message comes from the person who claims to be the sender
- **Integrity:** The message has not undergone modifications or tampering

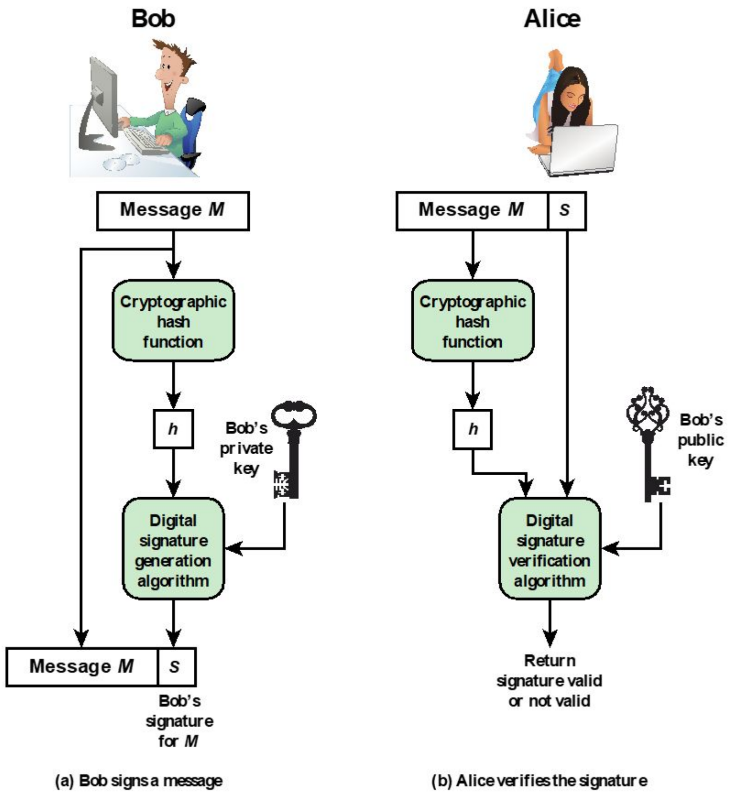


Figure 2.7 Simplified Depiction of Essential Elements of Digital Signature Process

References

- Stallings, William. Network security essentials: applications and standards. Pearson, 2016.
- Wikipedia contributors. (2024, October 12). Diffie–Hellman key exchange. In Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Diffie%E2%80%93Hellman_key_exchange&oldid=1250718908
-