Information Security Notes

CyberSecurity is the protection of information that is stored, transmitted, and processed in a networked system of computers, other digital devices, and network devices and transmission lines, including the Internet.

Protection encompasses confidentiality, integrity, availability, authenticity, and accountability.

As subsets of cybersecurity, we can define the following:

■ **Information security:** This term refers to preservation of confidentiality, integrity, and availability of information. In addition, other properties, such as authenticity, accountability, nonrepudiation, and reliability can also be involved.

■ **Network security:** This term refers to protection of networks and their service from unauthorized modification, destruction, or disclosure.

CIA and IAA

■ **Confidentiality:** Ensures that the information in a computer system and transmitted information are accessible only for reading by authorized parties.
E.g. Printing, displaying and other forms of disclosure.
Confidentiality is the protection of transmitted data from passive attacks.
This term covers two related concepts:

— **Data confidentiality:** Assures that private or confidential information is not made available or disclosed to unauthorized individuals.

— **Data integrity:** Assures that data (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner.

Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Integrity: Ensures that only authorized parties are able to modify computer system assets and transmitted information. Modification includes writing, changing status, deleting, creating and delaying or replaying of transmitted messages.

Availability: Assures that systems work promptly and service is not denied to authorized users.Requires that computer system assets be available to authorized parties when needed. A system is available if it provides services according to the system design whenever users request them

The CIA Triad—Confidentiality, Integrity, and Availability—is a guiding model in information security. The CIA triad guides the information security in a broad sense and is also useful for managing the products and data of research.

Confidentiality :Confidentiality refers to protecting information from unauthorized access.

Integrity: Integrity means data are trustworthy, complete, and have not been accidentally altered or modified by an unauthorized user.

Availability :Availability means means that the authorized users should be able to access data whenever required.

CIA Triad Examples

To have a better understanding of how the CIA triad works in practice, consider an ATM that allows users to access bank balances and other information. An ATM incorporates measures to cover the principles of the triad:

- The two-factor authentication (debit card with the PIN code) provides **confidentiality** before authorizing access to sensitive data.
- The ATM and bank software ensure data **integrity** by maintaining all transfer and withdrawal records made via the ATM in the user's bank accounting.
- The ATM provides **availability** as it is for public use and is accessible at all times.

To fight against confidentiality breaches, you can classify and label restricted data, enable access control policies, encrypt data, and use multi-factor authentication (MFA) systems.

- Categorize data and assets being handled based on their privacy requirements.
- Require data encryption and two-factor authentication to be basic security hygiene.
- Ensure that access control lists, file permissions and white lists are monitored and updated regularly.
- Ensure organization have the training and knowledge they need to recognize the dangers and avoid them

Ensuring data integrity involves protecting the data at all times, including when it is being used, transmitted, or stored. This includes implementing measures to prevent unauthorized access, data corruption, or tampering during these various stages. Countermeasures like encryption, **digital signatures**, hashing, and digital certificates can help maintain data integrity. Aside from these, intrusion detection systems, strong authentication mechanisms, version control, auditing, and access controls can ensure integrity.

Availability can be ensured through network, server, application, and service redundancy. Hardware fault tolerance in servers and storage is another good countermeasure to avoid violation of availability.

Authentication: Ensures that the origin of a message or electronic document is correctly identified, with an assurance that the identity is not false.
The **authentication** service is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function of the authentication service is to assure the recipient that the message is from the source that it claims to be from.

In information security, authentication is the set of methods used to establish whether a claim of identity is true.
Note that authentication does not decide what the party being authenticated is permitted to do; this is a separate task, known as authorization
Challenge/response: Party A, expecting a fresh message from B, first sendsB a nonce (challenge) and requires that the subsequent message (response)received from B contain the correct nonce value.

Challenge-response authentication refers to a set of protocols that helps validate actions to protect digital assets and services from unauthorized access. This

protocol usually has two components – a question and a response – where a verifier presents a challenge to a user, who must provide a correct answer for authentication. Challenge-response protocols can be as simple as a password or a dynamically generated request.

A challenge-response authentication mechanism, or CRAM, provides businesses with an easy-to-use tool that they can use to control access to sensitive information and identify bad actors.

To authenticate Alice, Bob will need to employ a challenge-response mechanism. That is, Bob will send a challenge to Alice, and the response from Alice must be something that only Alice can provide and that Bob can verify. To prevent a replay attack, Bob can incorporate a "number used once," or nonce, in the challenge. That is, Bob will send a unique challenge each time, and the challenge will be used to compute the appropriate response. Bob can

thereby distinguish the current response from a replay of a previous response.In other words, the nonce is used to ensure the freshness of the response.

One-Way Authentication

One-way authentication involves a single transfer of information from one user (A) intended for another (B). In its simplest form, one way authentication would establish the identity of A, the identity of B, and establish that some sort of authentication token actually was generated by A and actually was intended to be sent to B. An email message is an example of an application that lends itself to one-way authentication.

Multi-factor Authentication (MFA) is an authentication method that requires the user to provide two or more verification factors to gain access to a resource such as an application, online account, or a VPN.

Let’s return to the ATM example because it illustrates multifactor authentication well. In this case, you use something you know (your PIN) and something you have (your ATM card). Your ATM card serves as both a factor for authentication

and a form of identification. Another example of multifactor authentication is writing checks. In this case, you're using something you have (the checks themselves) and something you do (signing them).

Mutual authentication is an authentication mechanism in which both parties in a transaction authenticate each other. These parties are typically software-based. In the standard, one-way authentication process, the client authenticates to the server. In mutual authentication, not only does the client authenticate to the server, but the server authenticates to the client. Mutual authentication often relies on digital certificates.

Authorization: Authorization follows authentication. During authorization, a user can be granted privileges to access certain areas of a network or system. Authorization is different from authentication in that authentication only checks a user's identity, whereas authorization dictates what the user is allowed to do.

Accounting: Accounting keeps track of user activity while users are logged in to a network by tracking information such as how long they were logged in, the data they sent or received, their Internet Protocol (IP) address, the Uniform Resource Identifier (URI) they used, and the different services they accessed.

Other Defns

Non repudiation: Requires that neither the sender nor the receiver of a message be able to deny the transmission.
Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message

Access control: Requires that access to information resources may be controlled by or the target system.
In the context of network security, access control is the ability to limit and control the access to host systems and applications via communications links. To achieve this, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

Security attack – Any action that compromises the security of information owned by an organization.
Security mechanism – A mechanism that is designed to detect, prevent or recover from a security attack.
Security service – A service that enhances the security of the data processing systems and the information transfers of an organization.

Threats and Attacks

■ **Threat:** Any circumstance or event with the potential to adversely impact organizational operations,organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.

■ **Vulnerability**: A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

■ **Attack:** Any kind of malicious activity that attempts to collect, disrupt, deny, degrade, or destroy information system resources or the information itself.

A passive attack attempts to learn or make use of information from the system but does not affect system resources.
Two types of passive attacks are the release of message contents and traffic analysis. A second type of passive attack, traffic analysis, is subtler. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.Passive attacks are very difficult to detect because they do not involve any alteration of the data. the emphasis in dealing with passive attacks is on prevention rather than detection.

An active attack attempts to alter system resources or affect their operation.
Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into **four categories**: replay, masquerade, modification of messages, and denial of service.

A **masquerade** takes place when one entity pretends to be a different entity.
A masquerade can take the form of a man-in-the-middle attack (Figure 1.3c). In this type of attack, the attacker intercepts masquerades as the client to the server and as the server to the client. More generally, it can be used to impersonate the two ends of a legitimate communication. Another form of masquerade is illustrated in Figure 1.3d. Here, an attacker is able to access server resources by masquerading as an authorized user.

Replay involves the passive capture of a data unit and its subsequent re-transmission to produce an unauthorized effect.
Figure 1.3e illustrates the replay attack. As in a passive attack, the attacker does not disturb the information flow between client and server, but does capture client message. The attacker can then subsequently replay any client message to the server.

Data modification simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect.

The **denial of service** prevents or inhibits the normal use or management of communication facilities. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.
Figure 1.3d also illustrates denial of service in the context of a client/server environment. The denial of service can take two forms: (1) flooding the server with an overwhelming amount of data; and (2) triggering some action on the server that consumes substantial computing resources.

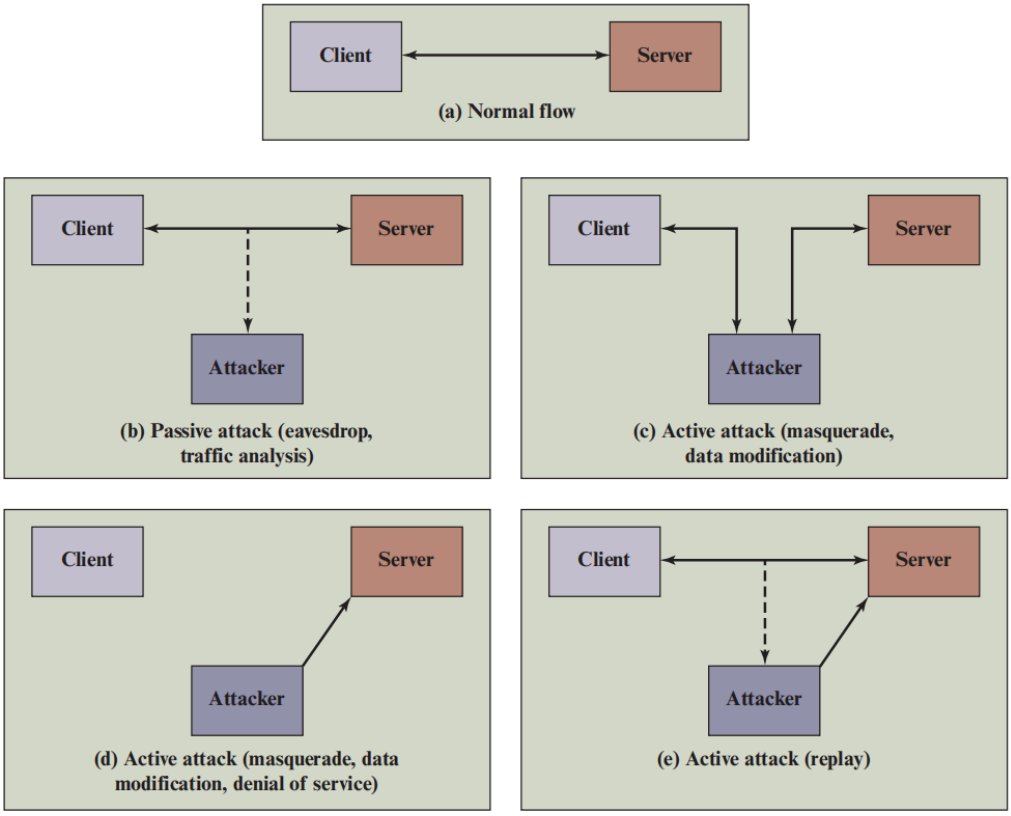


Figure 1.3 Security Attacks

Because the integrity service relates to active attacks, we are concerned with detection rather than prevention.

Cryptography is a branch of mathematics that deals with the transformation of data.

Cryptographic algorithms: We can distinguish between reversible cryptographic mechanisms and irreversible cryptographic mechanisms. A reversible cryptographic mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible cryptographic mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.

Digital signature: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery

Notarization: The use of a trusted third party to assure certain properties of a data exchange.

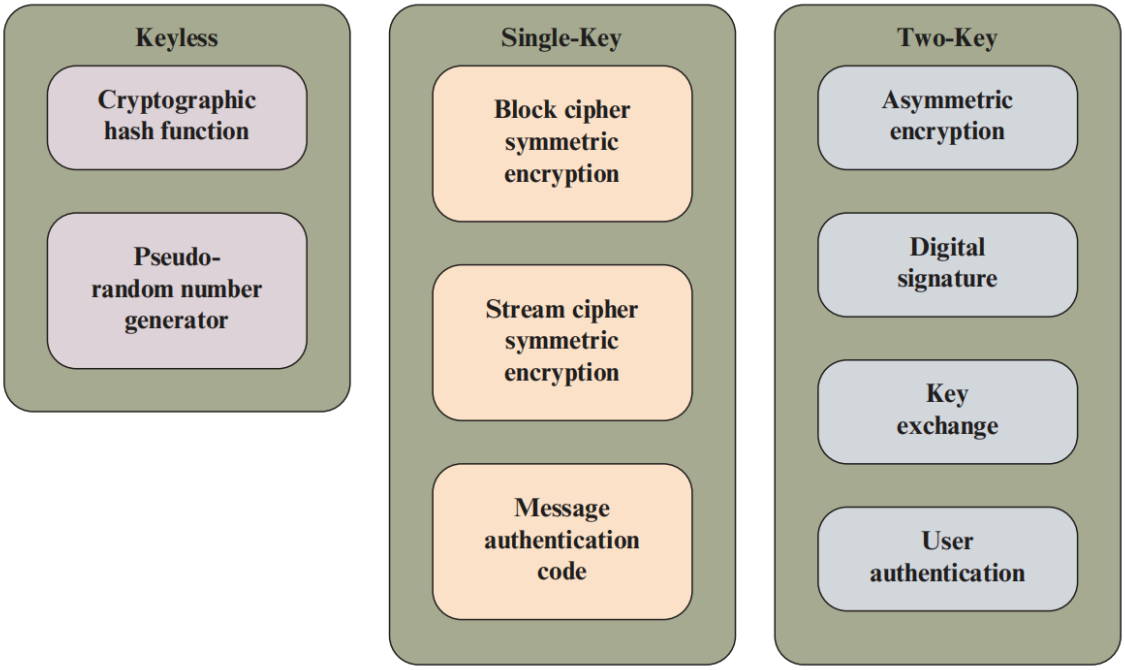


Figure 1.4 Cryptographic Algorithms

Cryptographic algorithms can be divided into three categories (Figure 1.4):

- **Keyless:** Do not use any keys during cryptographic transformations.
- **Single-key:** The result of a transformation is a function of the input data and a single key, known as a secret key.
- **Two-key:** At various stages of the calculation, two different but related keys are used, referred to as a private key and a public key.

A hash function turns a variable amount of text into a small, fixed-length value called a *hash value, hash code*, or *digest*. A **cryptographic hash function** is one that has additional properties that make it useful as part of another cryptographic algorithm, such as a message authentication code or a digital signature.
Encryption algorithms that use a single key are referred to as **symmetric encryption algorithms**

Block cipher: A block cipher operates on data as a sequence of blocks. A typical block size is 128 bits. In most versions of the block cipher, known as modes of operation, the transformation depends not only on the current data block and the secret key but also on the content of preceding blocks.

- **Stream cipher:** A stream cipher operates on data as a sequence of bits. Typically, an exclusive-OR operation is used to produce a bit-by-bit transformation. As with the block cipher, the transformation depends on a secret key.

MAC: Another form of single-key cryptographic algorithm is the **message authentication code** (MAC). A MAC is a data element associated with a data block or message. The MAC is generated by a cryptographic transformation involving a secret key and, typically, a cryptographic hash function of the message. The MAC is designed so that someone in possession of the secret key can verify the integrity of the message

Encryption algorithms that use two keys are referred to as **asymmetric encryption algorithms.** Asymmetric encryption has a variety of applications. One of the most important is the **digital signature algorithm**.

A digital signature is a value computed with a cryptographic algorithm and associated with a data object in such a way that any recipient of the data can use the signature to verify the data's origin and integrity. Typically, the signer of a data object uses the signer's private key to generate the signature, and anyone in possession of the corresponding public key can verify that validity of the signature.

Key exchange is the process of securely distributing a symmetric key to two or more parties.

User authentication is the process of authenticating that a user attempting to access an application or service is genuine and, similarly, that the application or service is genuine.

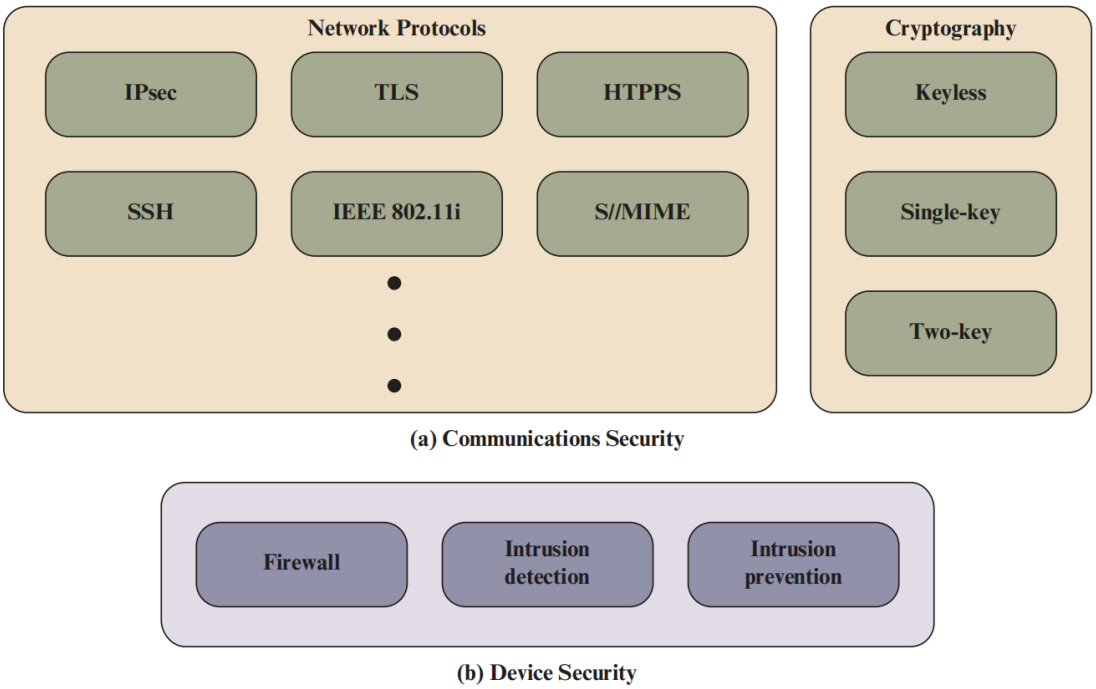


Figure 1.5 Key Elements of Network Security

A Trust Model

One of the most widely accepted and most cited definitions of trust in the organizational science literature is from [MAYE95], which defines **trust** as follows: the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the truster, irrespective of the ability to monitor or control that other party.

Three related concepts are relevant to a trust model:

- **Trustworthiness:** A characteristic of an entity that reflects the degree to which that entity is deserving of trust.
- **Propensity to trust:** A tendency to be willing to trust others across a broad spectrum of situations and trust targets. This suggests that every individual has some baseline level of trust that will influence the person's willingness to rely on the words and actions of others.
- **Risk:** A measure of the extent to which an entity is threatened by a potential circumstance or event, and typically a function of 1) the adverse impacts that would arise if the circumstance or event occurs; and 2) the likelihood of occurrence.

An original message is known as the **plaintext**
while the coded message is called the **ciphertext**.
The process of converting from plaintext to ciphertext is known as **enciphering** or **encryption**;

restoring the plaintext from the ciphertext is **deciphering** or **decryption**.
The many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** or a **cipher**.

Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis**. Cryptanalysis is what the layperson calls "breaking the code."
The areas of cryptography and cryptanalysis together are called **cryptology**.

A symmetric encryption scheme has five ingredients: Plaintext, Ciphertext, encryption algorithm, decryption algorithm, secret key.

We assume that it is impractical to decrypt a message on the basis of the ciphertext *plus* knowledge of the encryption/decryption algorithm. In other words, we do not need to keep the algorithm secret; we need to keep only the key secret. This feature of symmetric encryption is what makes it feasible for widespread use.

Cryptanalysis and Brute-Force Attack
Typically, the objective of attacking an encryption system is to recover the key in use rather than simply to recover the plaintext of a single ciphertext. There are two general approaches to attacking a conventional encryption scheme:
■ **Cryptanalysis:** Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.
■ **Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an understandable translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success.
That is, if there are X different keys, on average an attacker would discover the actual key after $X/2$ tries.

Table 3.1 Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext Only	■ Encryption algorithm ■ Ciphertext
Known Plaintext	■ Encryption algorithm ■ Ciphertext ■ One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	■ Encryption algorithm ■ Ciphertext ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	■ Encryption algorithm ■ Ciphertext ■ Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key ■ Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Caesar Cypher a **monoalphabetic substitution cipher**, because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

Then the algorithm can be expressed as follows. For each plaintext letter p, substitute the ciphertext letter C:[2]

$$C = \mathrm{E}(3, p) = (p + 3) \bmod 26$$

A shift may be of any amount, so that the general Caesar algorithm is

$$C = \mathrm{E}(k, p) = (p + k) \bmod 26 \qquad \textbf{(3.1)}$$

Plaintext is always in lowercase; ciphertext is in uppercase;

The best-known multiple-letter encryption cipher is the Playfair, which treats diagrams in the plaintext as single units and translates these units into ciphertext diagrams. The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword.

A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher .
A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64 or 128 bits is used

The majority of the encryption algorithms currently in use are block ciphers. Although block ciphers are often slower than stream ciphers, Block cipher tend to be more versatile.
Since block ciphers operate on larger blocks of the message at a time, they're usually more resource intensive and more complex to implement.
They're also more prone to errors in the encryption process. For example, an error in block cipher encryption would render a large segment of data unusable, whereas in a stream cipher, an error would corrupt only a single bit.

DES is a block cipher that uses a 56-bit key

Identification, as you just learned, is simply an assertion of who we are.
This may include who we claim to be as people, who a system claims to be over the network, or who the originating party of an email claims to be.

Passwords

Although they're only a single factor of authentication, passwords can represent a relatively high level of security when constructed and implemented properly

Drawbacks of password-based authentication

If you construct a password that uses lowercase letters only and is eight characters long, you can use a password-cracking utility to crack it quickly.

Don't write your password down and post it under your keyboard or on your monitor; doing so completely defeats the purpose of having a password in the first place.

Another common problem is the manual synchronization of passwords—in short, using the same password everywhere. If you use the same password for your email, for your login at work, and for your online knitting discussion forum, you're putting the security of all the accounts in the hands of those system owners. If any one of them is compromised, all of your accounts become vulnerable; all an attacker needs to do to access the others is look up your account name on the internet to find your other accounts and log in using your default password. By the time the attacker gets into your email account, the game is over because an attacker can generally use it reset account credentials for any other accounts you have.

Biometrics Authentication

You can use biometric systems in two ways. You can use them to verify the identity claim someone has put forth, as discussed earlier, or you can reverse the process and use biometrics as a method of identification. Processing the characteristic may also include noting elements that appear at certain parts of the image, known as minutiae. You can use the minutiae later to match the characteristic to the user.

Biometric factors are defined by seven characteristics: universality, uniqueness, permanence, collectability, performance, acceptability, and circumvention.

- Universality means you should be able to find your chosen biometric characteristic in the majority of people you expect to enroll in the system.

- Uniqueness is a measure of how unique a characteristic is among Individuals.

- Permanence tests how well characteristic resists change over time and with advancing age.

- Collectability measures how easy it is to acquire a characteristic.
- Circumvention describes how easy it is to trick a system by using a falsified biometric identifier.

The false acceptance rate (FAR) and false rejection rate (FRR) are two of these. FAR measures how often you accept a user who should be rejected. This is also called a false positive.
FRR measures how often we reject a legitimate user and is sometimes called a false negative.
You want to avoid both of these situations in excess. You should aim for a balance between the two error types, referred to as an equal error rate (EER). If you plot both the FAR and the FRR on a graph, the EER marks the point where the two lines intersect. We sometimes use EER as a measure of the accuracy of biometric systems.

AES

(Advanced Encryption Standard)

AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications.

AES STRUCTURE

Plain-text

1. The cipher takes a plain-text block size of 128 bits, or 16 bytes.
2. The key length can be 16, 24,or 32 bytes (128, 192, or 256 bits). The algorithm is referred to as AES-128, AES-192, or AES-256, depending on the key length.
3. The input is a single 128-bit block. This block is depicted as a 4 * 4 square matrix of bytes.
4. This block is copied into the **State** array, which is modified at each stage of encryption or decryption.
5. After the final stage, **State** is copied to an output matrix.

Key

1. The key is represented as a square matrix of bytes.
2. This key is then expanded into an array of key schedule words.
3. Each word is four bytes, and the total key schedule is 44 words for the 128-bit key.

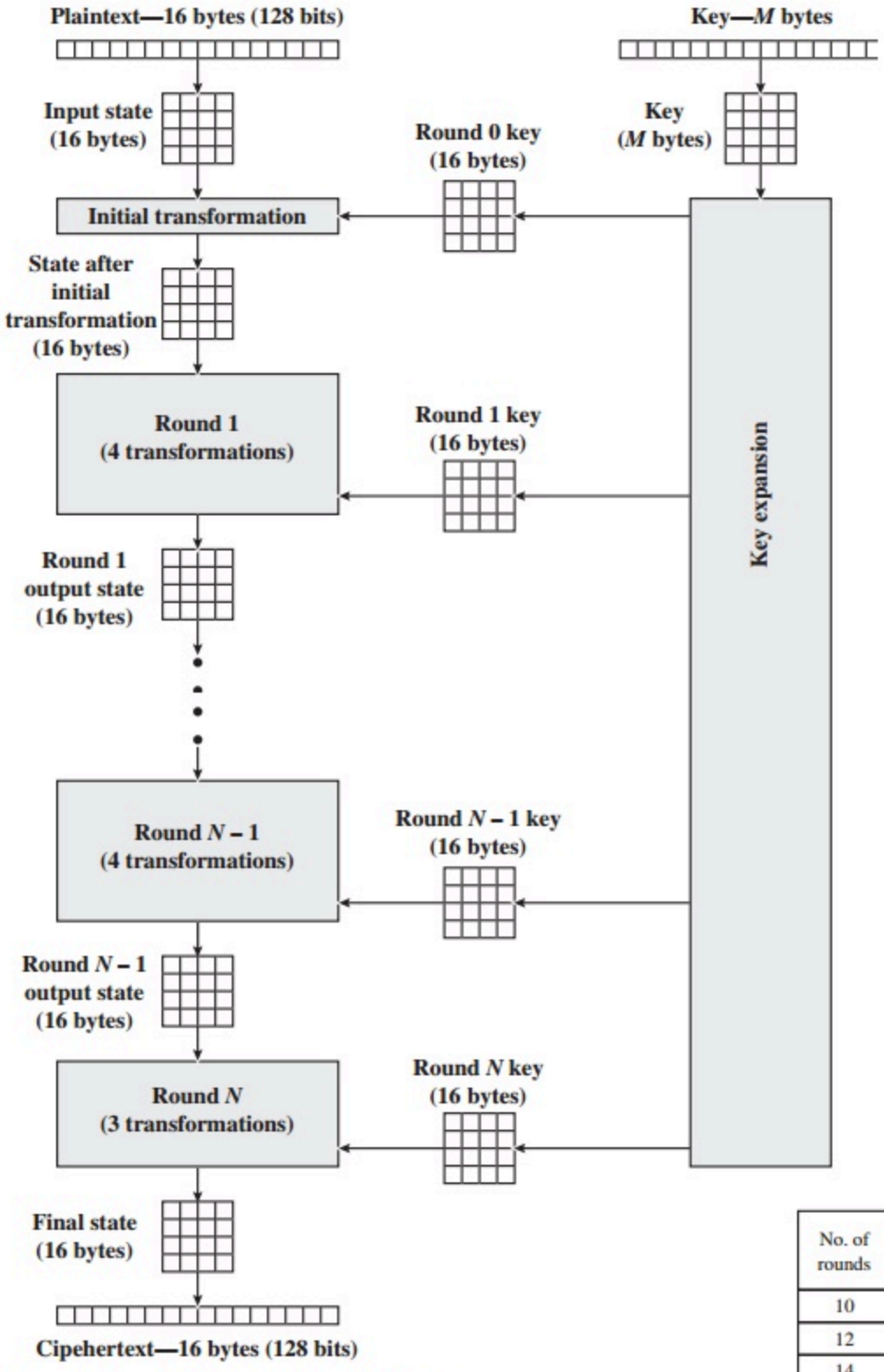
Note that the ordering of bytes within a matrix is by column.

So, for example, the first four bytes of a 128-bit plaintext input to the encryption cipher occupy the first column of the **in** matrix, the second four bytes occupy the second column, and so on. Similarly, the first four bytes of the expanded key, which form a word, occupy the first column of the **w** matrix.

The cipher consists of *N* rounds, where the number of rounds depends on the

key length: 10 rounds for a 16-byte key, 12 rounds for a 24-byte key, and 14 rounds for a 32-byte key .

The first *N* - 1 rounds consist of four distinct transformation functions: SubBytes, ShiftRows, MixColumns, and AddRoundKey.The final round contains only three transformations.



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 5.1 AES Encryption Process

Table 6.1 AES Parameters

Key Size (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Plaintext Block Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14
Round Key Size (words/bytes/bits)	4/16/128	4/16/128	4/16/128
Expanded Key Size (words/bytes)	44/176	52/208	60/240

Detailed Structure

- It is not a Feistel structure AES instead processes the entire data block as a single matrix during each round using substitutions and permutation.

- The key that is provided as input is expanded into an array of forty-four 32-bit words, **w**[*i*]. Four distinct words (128 bits) serve as a round key for each round;

- Four different stages are used, one of permutation and three of substitution:

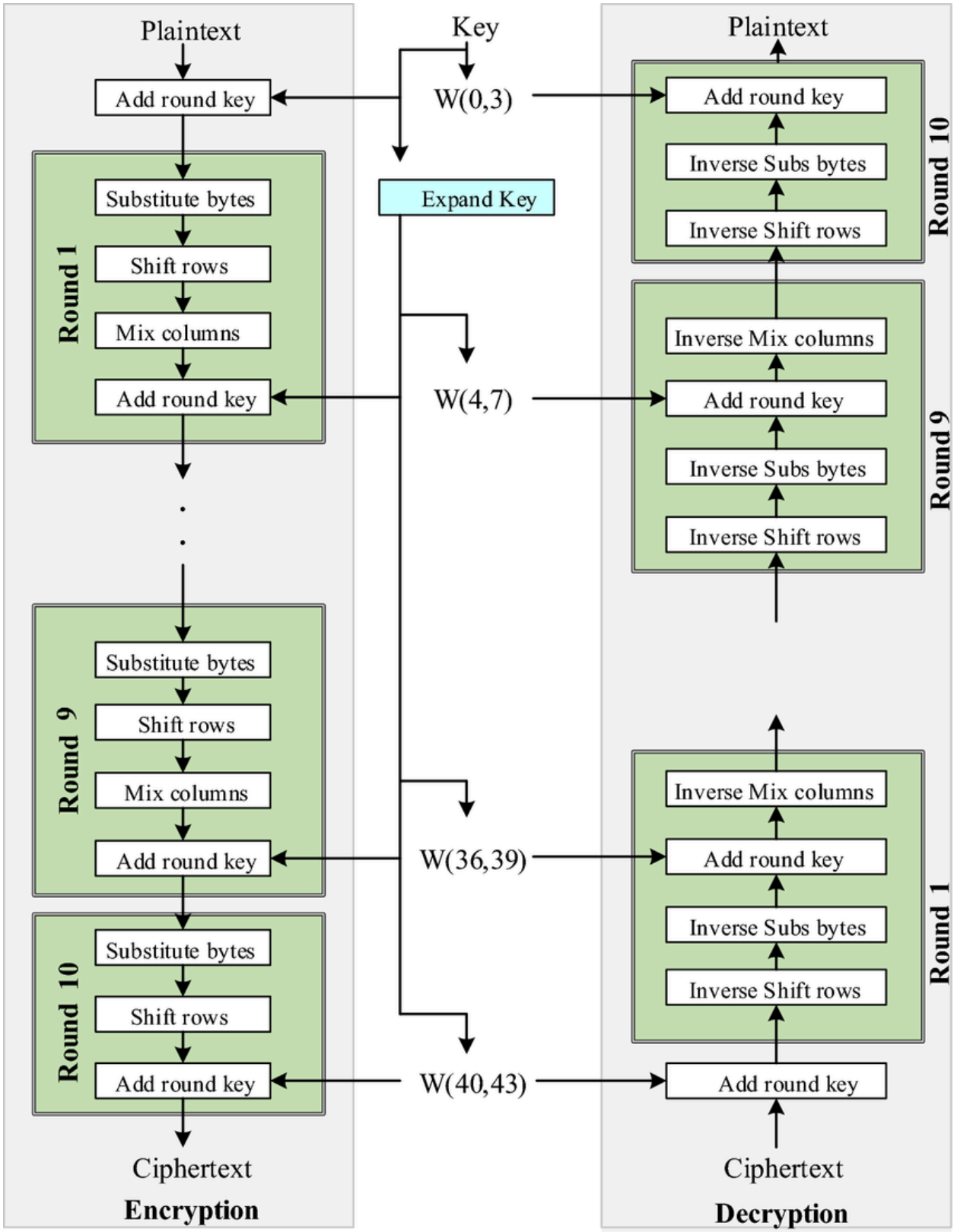
■ **Substitute bytes:** Uses an S-box to perform a byte-by-byte substitution of the block.

■ **ShiftRows:** A simple permutation.

■ **MixColumns:** A substitution that makes use of arithmetic over GF(28).

■ **AddRoundKey:** A simple bitwise XOR of the current block with a portion of the expanded key.

- Only the AddRoundKey stage(a form of Vernam cipher) makes use of the key.
- Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security


Plaintext
Add round key
Round 1
Substitute bytes
Shift rows
Mix columns
Add round key
Round 9
Substitute bytes
Shift rows
Mix columns
Add round key
Round 10
Substitute bytes
Shift rows
Add round key
Ciphertext
Encryption
Key
W(0,3)
Expand Key
W(4,7)
W(36,39)
W(40,43)
Plaintext
Add round key
Round 10
Inverse Subs bytes
Inverse Shift rows
Inverse Mix columns
Add round key
Round 9
Inverse Subs bytes
Inverse Shift rows
Inverse Mix columns
Add round key
Round 1
Inverse Subs bytes
Inverse Shift rows
Add round key
Ciphertext
Decryption

Substitute Bytes Transformation

Each individual byte of **State** is mapped into a new byte in the following way: The leftmost 4 bits of the byte are used as a row value and the rightmost 4 bits are used as a column value. These row and column values serve as indexes into the S-box to select a unique 8-bit output value.

For example, the hexadecimal value {95} references row 9, column 5 of the S-box, which contains the value {2A}. Accordingly, the value {95} is mapped into the value {2A}

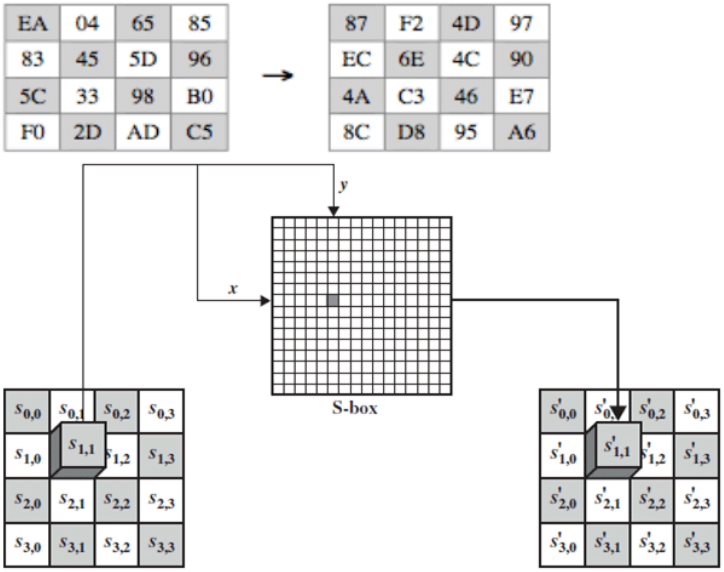


Table 5.2 AES S-Boxes

	y															
	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	2C	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	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	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

x

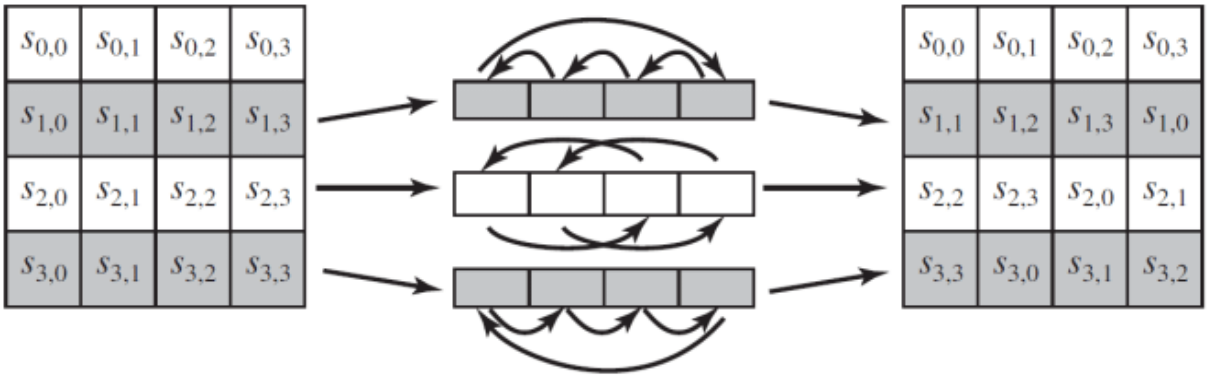
(a) S-box

ShiftRows Transformation

The first row of **State** is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed.

ShiftRow -Left/right circular shift.

InverseShiftRow -Opposite direction circular shift.



The following is an example of ShiftRows:

87	**F2**	**4D**	**97**
EC	**6E**	**4C**	**90**
4A	**C3**	**46**	**E7**
8C	**D8**	**95**	**A6**

→

87	**F2**	**4D**	**97**
6E	**4C**	**90**	**EC**
46	**E7**	**4A**	**C3**
A6	**8C**	**D8**	**95**

MixColumns Transformation

Each byte of a column is mapped into a new value that is a function of all four bytes in that column.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \quad \textbf{(5.3)}$$

The MixColumns transformation on a single column of **State** can be expressed as

$$\begin{aligned} s'_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\ s'_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j}) \end{aligned}$$

AddRoundKey Transformation

In the **forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key. The operation is viewed as a columnwise operation between the 4 bytes of a **State** column and one word of the round key; it can also be viewed as a byte-level operation.

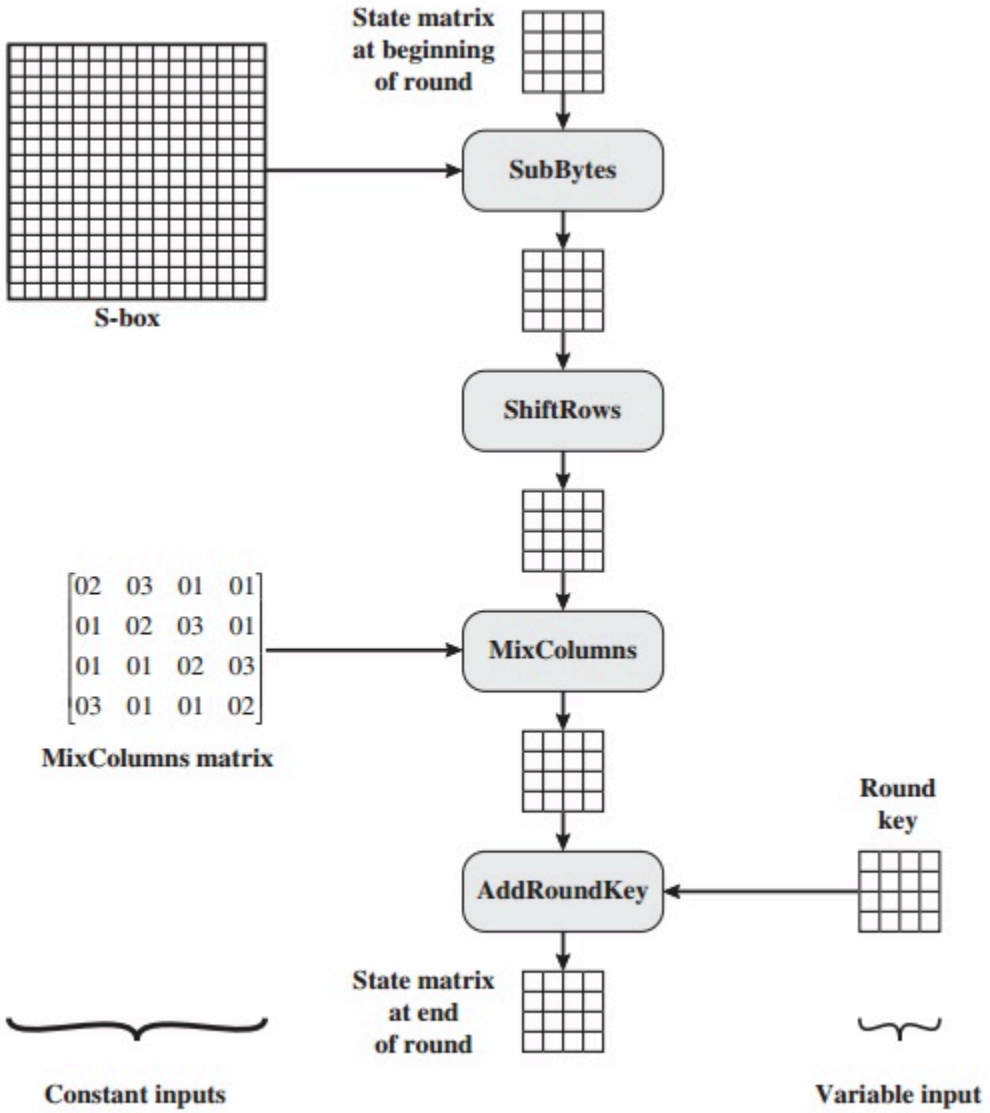


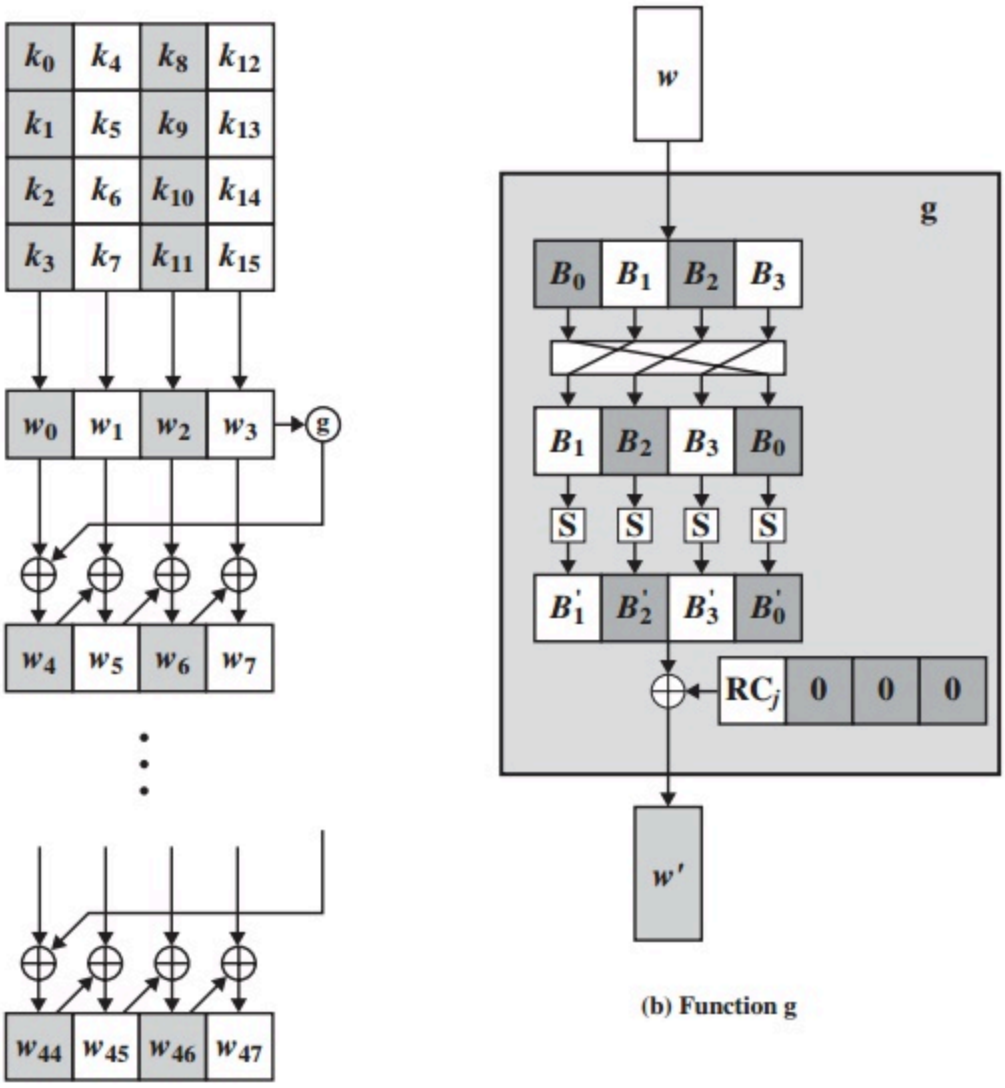
Figure 5.8 Inputs for Single AES Round

Key Expansion Algorithm

The AES **key expansion** algorithm takes as input a four-word (16-byte) key and produces a linear array of 44 words (176 bytes). This is sufficient to provide a four

word round key for the initial AddRoundKey stage and each of the 10 rounds of the cipher.

```
KeyExpansion (byte key[16], word w[44])
{
    word temp
    for (i = 0; i < 4; i++)    w[i] = (key[4*i], key[4*i+1],
                                       key[4*i+2],
                                       key[4*i+3]);
    for (i = 4; i < 44; i++)
    {
     temp = w[i - 1];
     if (i mod 4 = 0)    temp = SubWord (RotWord (temp))
                                ⊕ Rcon[i/4];
     w[i] = w[i-4] ⊕ temp
    }
}
```



(a) Overall algorithm

(b) Function g

Figure 5.9 AES Key Expansion

1. RotWord(Row shift) performs a one-byte circular left shift on a word. This means that an input word [B0, B1, B2, B3] is transformed into [B1, B2, B3, B0].

2. SubWord(S box substitute byte) performs a byte substitution on each byte of its input word, using the S-box (Table 6.2a).

3. The result of steps 1 and 2 is XORed with a round constant, Rcon[j].

The round constant is a word in which the three rightmost bytes are always 0.

Thus, the effect of an XOR of a word with Rcon is to only perform an XOR on the leftmost byte of the word.

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

For example, suppose that the round key for round 8 is

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Then the first 4 bytes (first column) of the round key for round 9 are calculated as shown in Table 6.3

Table 6.3 Example Round Key Calculation

Description	Value
i (decimal)	36
temp = w[i − 1]	7F8D292F
RotWord (temp)	8D292F7F
SubWord (RotWord (temp))	5DA515D2
Rcon (9)	1B000000
SubWord (RotWord (temp)) ⊕ Rcon (9)	46A515D2
w[i − 4]	EAD27321
w[i] = w[i − 4] ⊕ SubWord (RotWord (temp)) ⊕ Rcon (9)	AC7766F3

Cyber Security Controls: Physical, Technical, Procedural, Legal.

Cybersecurity controls are mechanisms used to prevent, detect and mitigate cyber threats and attacks. Mechanisms range from physical controls, such as security guards and surveillance cameras, to technical controls, including firewalls and multifactor authentication.

There are 4 major categories of security controls that can be used when constructing cybersecurity protection:
1. Physical
2. Technical
3. Procedural
4. Legal (also referred to as regulatory or compliance controls)

physical security *– measures designed to deter, prevent, detect or alert unauthorized real-world access to a site or material item.*

These controls include restrictions on physical access, (such as security guards at building entrances), locks, closed-circuit security cameras, and perimeter fences.

technical control *– the use of an electronic or digital method to influence or command how something like a* ***digital device*** *can or cannot be used.*

For example, removing the ability to cut or paste information on a smartphone is an example of a technical control that can be used to minimize security risks.

procedural control *– an instruction during a sequence of required steps to limit how something is or is not permitted to be used.*

An example of a procedural control is to require a minimum of 2 authorized people to approve any access request. Procedural controls use any process whose purpose is to help strengthen a security position.

legal control *– the use of legislation to help promote and invest in positive security methods and also to deter, punish and correct infringements.*

Whenever you hear about a large financial penalty being imposed on an organization, this is an example of the consequences of not meeting a legal control requirement.
Many companies seek to pass some of their legal financial responsibilities onto their employees or suppliers as an incentive to promote good practices. It is also normal for any breach in legal controls to result in disciplinary action.

8 Best Practices for Security Controls

1. **Keep strong passwords.** Implement stringent password policies, including unique passwords for each user, a mix of characters, and regular password expiration to prevent unauthorized access and data breaches.
2. **Enact user access restrictions.** Limit user access through access controls and permissions, following the principle of least privilege to minimize the potential for unauthorized access, malware spread, and improve compliance and audit processes.
3. **Embrace patch management.** Perform regular updates and patches to address vulnerabilities in operating systems and third-party software, reducing the risk of cyberattacks and improving system security and performance.
4. **Use firewall protection.** Employ firewalls to regulate incoming and outgoing network traffic, establishing a strong barrier against unauthorized access and potential threats to your business.
5. **Use VPN encryption.** Use a virtual private network (VPN) to encrypt data and protect privacy when connecting to public networks, assuring anonymity and safeguarding against potential hackers.
6. **Use antivirus software.** Install enterprise-grade antivirus and anti-malware software on all devices and systems to detect and remove malicious threats automatically.
7. **Encourage multi-factor authentication (MFA).** Implement MFA to add an extra layer of security beyond passwords, requiring additional verification factors like PINs, authenticator apps, or biometric data to assure secure user authentication.
8. **Schedule regular data backup.** Back up important files, including sensitive information and cloud applications, to protect against data loss due to

compromise or unforeseen events, ensuring minimal disruption to operations.

Cyber Security Terminology, reporting software flaws, computer crime, and difference between law and ethics

1 Computer Security Terminology

Adversary (threat agent)

An entity that attacks, or is a threat to, a system.

Attack

An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

Countermeasure

An action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective
action can be taken.

Risk

An expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular harmful result.

Security Policy

A set of rules and practices that specify or regulate how a system or organization provides security services to protect sensitive and critical system resources.

System Resource (Asset)

Data contained in an information system; or a service provided by a system; or a system capability, such as processing power or communication bandwidth; or an item of system equipment (i.e., a system component—
hardware, firmware, software, or documentation); or a facility that houses system operations and equipment.

Threat

A potential for violation of security, which exists when there is a circumstance, capability, action, or event, that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

Vulnerability

A flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.

What are the most effective ways to report software bugs in software design?

- Identify the bug.
- Use a bug tracking system.
- Write a clear and concise title.
- Describe the impact and priority.
- Provide a solution or suggestion.
- Follow up and update.

Computer crime, or **cybercrime**, is a term used broadly to describe criminal activity in which computers or computer networks are a tool, a target, or a place of criminal activity.

Difference between Law and Ethics

Law and Ethics share the goal of regulating behavior and promoting societal order. Law is a formal, codified system enforced by the state, **whereas** ethics is a set of principles based on personal and societal values that guide individual conduct. Laws provide a legal framework, while ethics provide a moral framework for behavior.

What is Law?
Law refers to a system of rules, regulations, and principles established by a governing authority to regulate behavior within a society. Laws are created to maintain order, protect individual rights, and provide a framework for resolving disputes. They play a crucial role in shaping social, political, and economic structures

Ethics refers to a system of moral principles that relates to the benefits and harms of particular actions, and to the rightness and wrongness of motives and ends of those actions.

Law and Ethics

- **Laws** are rules that mandate or prohibit certain behavior in society; they are drawn from ethics, which define socially acceptable behaviors.
- The key difference between laws and ethics is that laws carry the sanctions of a governing authority and ethics do not.
- **Ethics** in turn are based on cultural mores: the fixed moral attitudes or customs of a particular group. Some ethics are recognized as universal.
  - For example, murder, theft, assault, and arson are commonly accepted as actions that deviate from ethical and legal codes in the civilized world.

BASIS FOR COMPARISON	LAW	ETHICS
Meaning	The law refers to a systematic body of rules that governs the whole society and the actions of its individual members.	Ethics is a branch of moral philosophy that guides people about the basic human conduct.
What is it?	Set of rules and regulations	Set of guidelines
Governed By	Government	Individual, Legal and Professional norms
Expression	Expressed and published in writing.	They are abstract.
Violation	Violation of law is not permissible which may result in punishment like imprisonment or fine or both.	There is no punishment for violation of ethics.
Objective	Law is created with an intent to maintain social order and peace in the society and provide protection to all the citizens.	Ethics are made to help people to decide what is right or wrong and how to act.

BASIS FOR COMPARISON	LAW	ETHICS
Binding	Law has a legal binding.	Ethics do not have a binding nature.

Human Factors in Information Security:; Social Engineering Attacks, techniques and prevention mechanisms.

Why human are the weakest point in Information security?

Humans are susceptible to cognitive biases, such as the tendency to prioritize convenience over security or to underestimate risks when they perceive a task as familiar. By recognizing these biases, cybersecurity professionals can tailor their approach to mitigate human error effectively. This might involve simplifying security protocols, implementing user-friendly authentication methods, or leveraging behavioral psychology principles to promote adherence to security guidelines.

Insider threats

Another critical aspect of the human factor in cybersecurity is the insider threat. While external threats often dominate headlines, insider threats—whether intentional or unintentional—pose a significant risk to organizations. Employees with access to sensitive information can inadvertently leak data through negligent actions or intentionally exploit their privileges for personal gain or malicious purposes. Addressing insider threats requires a multifaceted approach that combines technical controls with policies and procedures designed to detect, deter, and respond to suspicious behavior.

Social engineering

Social engineering remains a prevalent tactic used by cybercriminals to exploit the human element in cybersecurity. Whether through phishing emails, pretexting, or baiting, attackers leverage psychological manipulation to deceive individuals into divulging confidential information or performing actions that compromise security. Mitigating the risks associated with social engineering requires a combination of

technical controls, user awareness training, and robust incident response procedures.

Burnout, fatigue, and cognitive overload can impair decision-making and undermine the effectiveness of security measures. Organizations must prioritize employee well-being and provide adequate resources and support to prevent fatigue and maintain optimal performance. Additionally, fostering a culture of collaboration and knowledge sharing can enhance the collective resilience of cybersecurity teams in responding to evolving threats.

The human factor in cybersecurity extends beyond individual users and organizations to encompass broader societal issues. Cybersecurity policies and regulations must strike a delicate balance between protecting privacy and promoting security, ensuring that measures designed to enhance security do not infringe upon individual rights and freedoms. Moreover, addressing the root causes of cybercrime, such as economic inequality and geopolitical tensions, requires a multifaceted approach that goes beyond technical solutions.

Final thoughts

By understanding the complexities of human behavior, organizations can develop more effective strategies for mitigating risks and safeguarding digital assets. From raising awareness and fostering a culture of security consciousness to addressing insider threats and combating social engineering tactics, integrating the human element into cybersecurity initiatives is essential for protecting against evolving threats in an increasingly digital world.

Manipulating people to do certain things is not the only human factor that can create security gaps. The most significant human factors are:
Inadequate cybersecurity subject knowledge leading to the presence of large amounts of open vulnerabilities.
Poor capture and communication of risks leading to repeated, unanticipated cybersecurity failures.

1) The risk does not directly impact the person's immediate location, department or budget. This is an example of *silo thinking*.
2) There are sometimes negative personal or career consequences for reporting risks. Some enterprises believe that formal procedures for reporting risks conflict with the organization's risk appetite.

Employees then reason that if there are no easy mechanisms or rewards for reporting suspected risks, why do it?**

3) If the process for filtering and escalating risks is not very well developed, the recipient of any reported risk information may be more inclined to bury it than to communicate and manage it.

Culture and relationship issues, both in the enterprise itself and/or in key suppliers, creating disinterested and disaffected personnel with insider knowledge.

Many cybersecurity threats are created from within. A corporate culture that creates disaffected or disinterested staff is much more likely to lead to this type of threat than one that fosters employee satisfaction.
In your organization, do people generally like each other, get along nicely and believe that the company invests in them and considers them to be more than assets with id numbers?
When people feel no connection to or support from their organization, they are more likely to seek opportunities to take personal advantage of their position. This is because these individuals often seek to retaliate for the lack of support they receive from the employer.
An organization's whistleblowing process is another factor that reflects the enterprise's culture and also impacts its cybersecurity posture.

Under-investment in security training resulting in a low level of awareness about the security risks we all manage (even if we are not cybersecurity experts).

When people have access to an organization's digital systems, their actions can affect employees, customers and others as well. This is why anyone with access needs practical and regular awareness training on what the potential security threats are, how to avoid them and how to report any suspected or confirmed security problems.
Security awareness and advice needs to include specific and practical content about security threats to any relevant electronic information or systems to which a person may have access. For example:
Do not leave your computer or mobile device unlocked when you are not with it and using it.
Never mix alcohol with using any digital device (phone, tablet or computer) that can access systems at your workplace.

Never discuss or speak about work when intoxicated.Be aware that malicious software can be loaded onto your
computer, phone or tablet simply by clicking on a link. For that reason, do not click on any link that you believe may not be safe

Using trust instead of procedures, especially for personnel with privileged access to information, systems or devices.

In many organizations, especially in those that are growing, a few select individuals enjoy unbridled privileges and are considered to be completely trustworthy. They have always been there, they have always done the right thing and to add in procedures that move away from the trust system can seem both expensive and unnecessary.
What I have written in the paragraph above is the usual explanation that is used just after an organization was badly burned because trusting an insider proved to be a huge mistake.

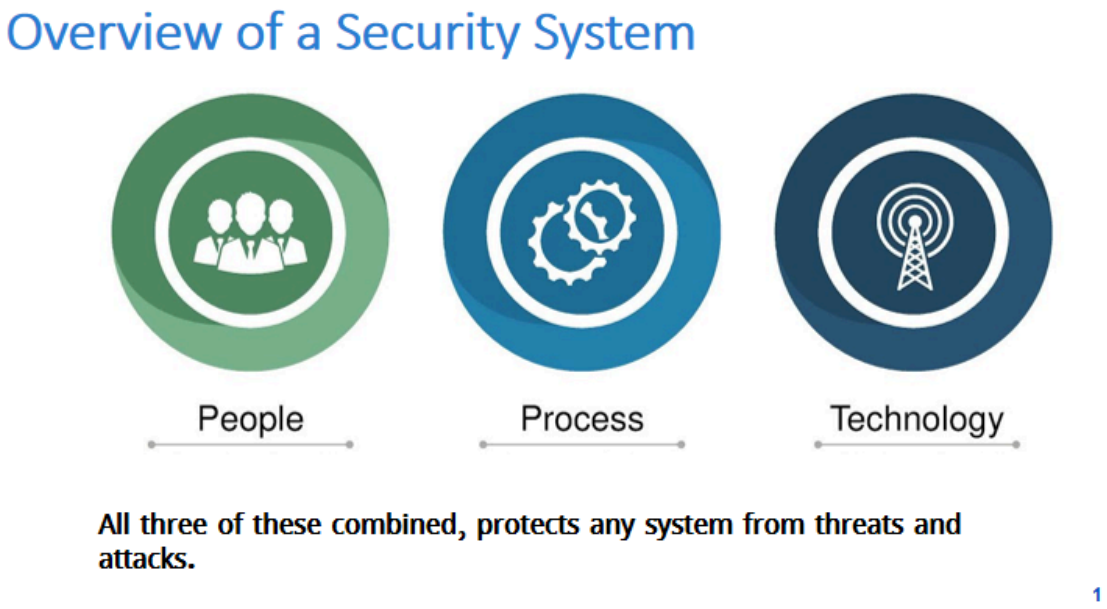
Absence of a single point of accountability. When more than one person is accountable, nobody is.

single point (of) accountability *(SPA or SPOA) – the principle that all critical* ***assets****, processes and actions must have clear ownership and traceability to a single person. The rationale is that the absence of a defined, single owner is a frequent cause of process or asset protection failure. Shared ownership is regarded as a significant security gap due to the consistent demonstration that security flaws have an increased probability of persisting when more than one person is accountable.*
Using a single point of accountability has been demonstrated to work incredibly well; it is proved to help control highly regulated systems successfully.

Social engineering, which can involve various methods of leveraging insiders' access or knowledge to create opportunities that bypass other security controls. These methods may include picking up information from personnel through traditional espionage techniques or manipulating them to do specific

things.

social engineering* – *is the art of manipulating people through personal interaction to gain* ***unauthorized access*** *to something.*



The People

- **People** are the weakest link to security programs (most neglected factor).
- Technology and processes are mostly reliable.
- Human being are unpredictable, susceptible to attacks and can be stressed into doing something out of protocol.
- **Human error is still the greatest cause of data breaches and security failures** (Source: Helpnet Security).
- A single human error is enough to bring down the whole system

1

Social Engineering Attacks

- **Social engineering attack** uses psychological manipulation to trick users into making security mistakes or giving away sensitive information.
- Attacker needs valid data to trick users.
- One need to know how attacker collect data for **social engineers** attacks.

1

Gathering Information

- A staggering wealth of information exists in online databases, public records, and social media sites, and in many cases, this data is free for the taking.
- Human intelligence is data gathered by **talking to people**.
- Google hacking

Pretexting Attack

- Attackers use information to pretend to be a trusted entity to a specific user.
- Create scenarios where the user is convinced to trust them
- Finally, give up sensitive information and perform activities that render the user vulnerable.

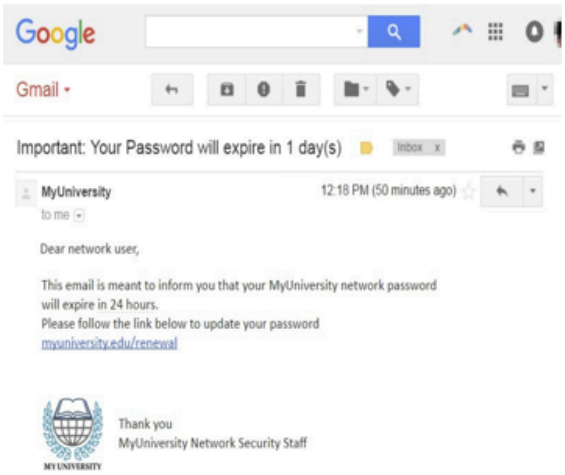
Example

Assume someone calls an employee and pretends to be someone in power, such as the CEO or on the information technology team.

The attacker convinces the victim that the scenario is true and collects information that is sought.

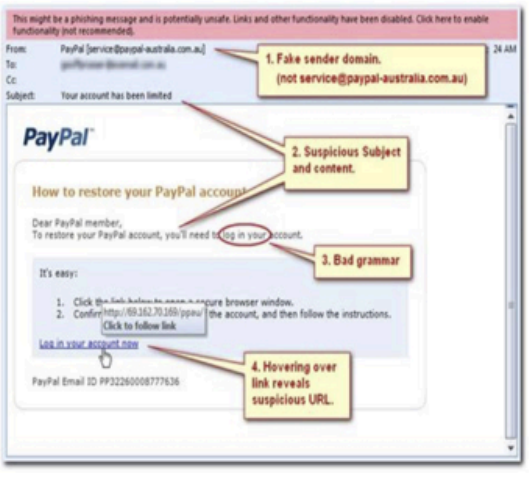
Phishing Attack

- Attacker uses electronic communications such as email, texting, or phone calls to **convince** the target to click a malicious link.
- Goal is to collect the target's **personal information** or **install malware** on their system.



How to Tackle Phishing Attack

- Don't click any links on an email unless you can guarantee who its from
- Look details of such an email carefully
- Use a trusted method of contacting via a phone number, or website
- Mark the email as spam
- Using updated browser



Vishing and Smishing Attack

- Vishing and Smishing are similar to phishing.
- Vishing is convincing a target to give access to computer over telephone.
- Smishing is sending fraudulent links over SMS to bait a victim.

Scareware

- Attackers use to scare people into downloading malicious software.
- For example, rogue scareware or fake software include Advanced Cleaner, System Defender, and Ulti

Prevention

- Use software from trusted companies.
- Avoid popups and use adblocker.

Baiting

- Baiting means offering something enticing or curious in front of the victim to lure them into a social engineering trap.
- For example, encouraging a person to provide bKash PIN in exchange for free money.

Prevention

- Staying vigilant about suspicious offers.
- Conduct organized simulated attacks to check employee awareness.

Quid Pro Quo

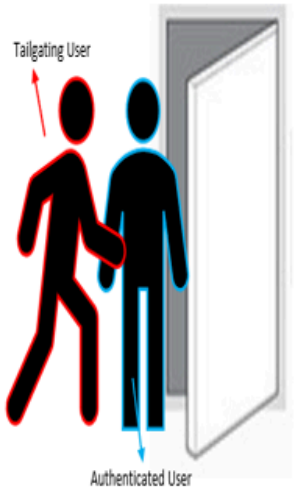
- Quid Pro Quo means something for something.
- Quid pro quo usually provides sensitive information in exchange for a service.
- For example, social media like Facebook offers free services in exchange for a ton of user data.

Prevention

- Reading privacy policies and terms and conditions before signing up for a free service.
- User Awareness.
- Remembering any free service is not always free

Tailgating

- Tailgating, or piggybacking, is the act of following someone through an access control point, **instead of using the credentials normally needed to enter**.
- For example, a user fails to properly log off their computer, allowing an unauthorized user to "piggyback" on the authorized user's session.



Prevention

- Always log off from public computers
- Avoid logging into sensitive sites using public networks
- Avoid physical tailgating through video surveillance

Social Engineering Training

Training users to recognize and respond to social engineering attacks can be an incredibly arduous task because such attacks **take advantage of our behavioral norms and tendencies.**

- Users should be taught to be **suspicious** of anything that seems **unusual**
- Ask people to **trust but verify** when faced with even the slightest doubt
- Users may flood security operations center with calls and emails, but at least they won't fall victim.
- Teach **password hygiene**
- **Create policies** regarding social engineering attacks

Passwords Hygiene

To Do

- Use 1 password per account/Servic
- Use **strong passwords**: Use long passwords and combine uppercase letters, lowercase letters, numbers, and symbols.
- **Change** your password frequently**. Never reuse** passwords
- Be careful where you enter your password (**protect shoulder surfing**)
- Enable **Two-Factor Authentication** where required
- Password managers can be helpful to store your passwords
- Create policies regarding password hygiene

1

DIFFIE–HELLMAN KEY EXCHANGE

The Algorithm

Figure 10.1 summarizes the Diffie–Hellman key exchange algorithm. For this scheme, there are two publicly known numbers:

a **prime number *q*** and an **integer a** that is a primitive root of q.

Suppose the users Alice and Bob wish to create a shared key.

Alice selects a random integer $XA < q$ and computes $YA = a^{Xa} \bmod q$.

Similarly, Bob independently selects a random integer $XB < q$ and computes

$YB = a^{XB} \bmod q$.

Each side keeps the X value private and makes(Exchanges) the Y value available publicly to the other side. Thus, XA is Alice's private key and YA is Alice's corresponding public key, and similarly for Bob.

Alice computes the key as $K = (YB)^{XA} \bmod q$ and Bob computes the key as $K = (YA)^{XB} \bmod q$. These two calculations produce identical results:

Man-in-the-Middle Attack

The protocol depicted in Figure 10.1 is insecure against a **man-in-the-middle attack**. Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows (Figure 10.2).

1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. Alice transmits Y_A to Bob.
3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates $K2 = (Y_A)^{X_{D2}} \bmod q$.
4. Bob receives Y_{D1} and calculates $K1 = (Y_{D1})^{X_B} \bmod q$.
5. Bob transmits Y_B to Alice.
6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K1 = (Y_B)^{X_{D1}} \bmod q$.
7. Alice receives Y_{D2} and calculates $K2 = (Y_{D2})^{X_A} \bmod q$.

At this point, Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key $K1$ and Alice and Darth share secret key $K2$. All future communication between Bob and Alice is compromised in the following way.

1. Alice sends an encrypted message M: $E(K2, M)$.
2. Darth intercepts the encrypted message and decrypts it to recover M.
3. Darth sends Bob $E(K1, M)$ or $E(K1, M')$, where M' is any message. In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob.

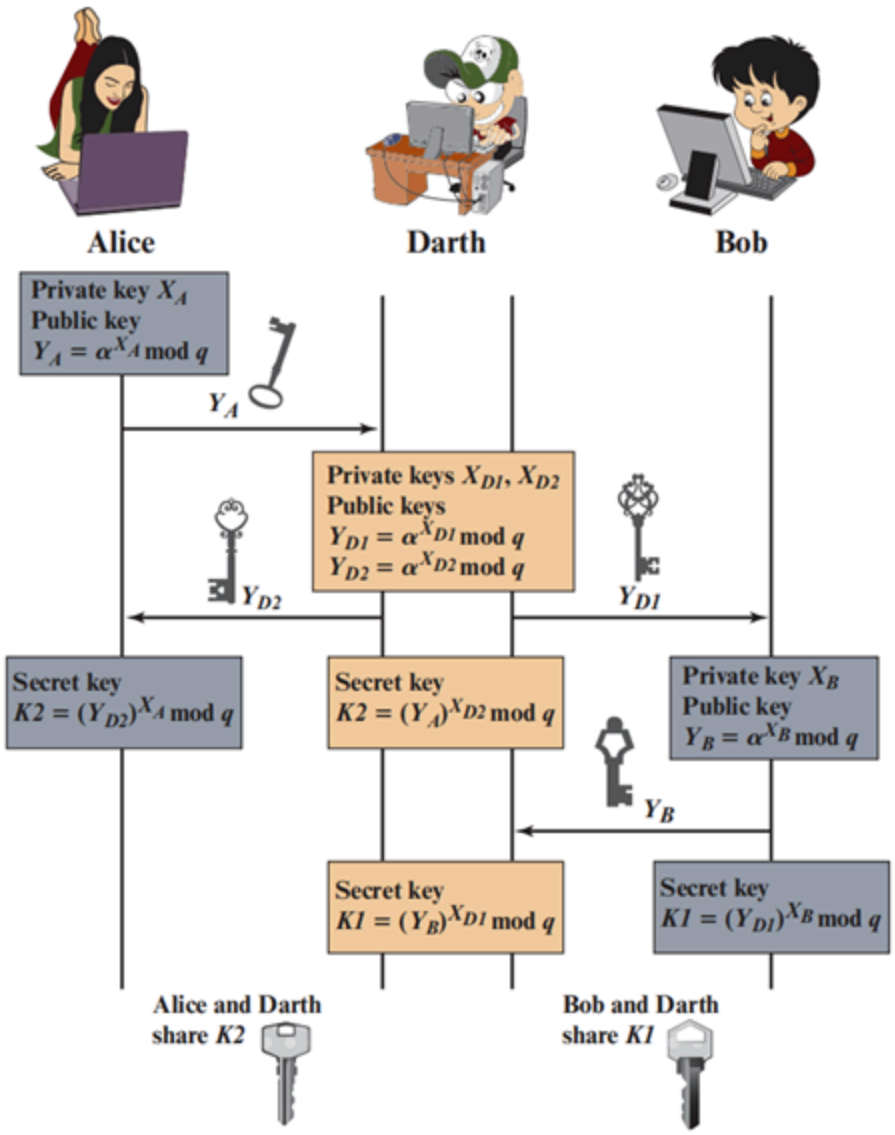


Figure 10.2 Man-in-the-Middle Attack

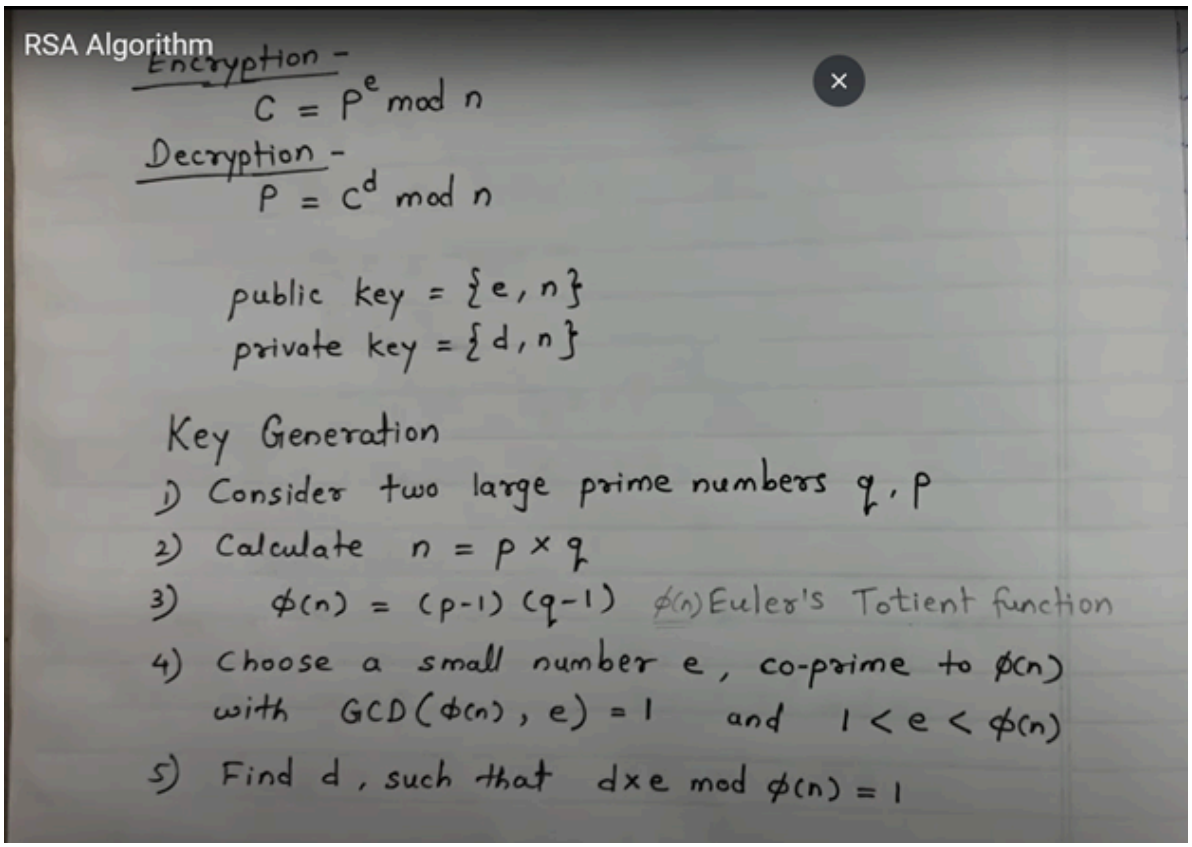
The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates; these topics are explored in

Limitations of Diffie-Hellman

- Does not authenticate either party involved in the exchange.it is vulnerable to man-in-the-middle attack.

- It cannot be used for asymmetric exchange.
- It cannot be used to encrypt messages.
- Digital signature cannot be signed using Diffie-Hellman algorithm
- As it is computationally intensive, it is expensive in terms of resources and CPU performance time.

Cryptographic algorithm: Introduction to modular arithmetic, detail operation of RSA



where $2^i < n \leq 2^{i+1}$. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C.

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

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$. For

1. Select two prime numbers, $p = 17$ and $q = 11$.
2. Calculate $n = pq = 17 \times 11 = 187$.
3. Calculate $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.
4. Select e such that e is relatively prime to $\phi(n) = 160$ and less than $\phi(n)$; we choose $e = 7$.
5. Determine d such that $de \equiv 1 \pmod{160}$ and $d < 160$. The correct value is $d = 23$, because $23 \times 7 = 161 = (1 \times 160) + 1$; d can be calculated using the extended Euclid's algorithm (Chapter 2).

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$. The example shows the use of these keys for a plaintext input of $M = 88$. For encryption, we need to calculate $C = 88^7 \bmod 187$. Exploiting the properties of modular arithmetic, we can do this as follows.

$$\begin{aligned}
88^7 \bmod 187 &= [(88^4 \bmod 187) \times (88^2 \bmod 187) \\
&\quad \times (88^1 \bmod 187)] \bmod 187 \\
88^1 \bmod 187 &= 88 \\
88^2 \bmod 187 &= 7744 \bmod 187 = 77 \\
88^4 \bmod 187 &= 59{,}969{,}536 \bmod 187 = 132 \\
88^7 \bmod 187 &= (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11
\end{aligned}$$

For decryption, we calculate $M = 11^{23} \bmod 187$:

$$\begin{aligned}
11^{23} \bmod 187 &= [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \\
&\quad \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187 \\
11^1 \bmod 187 &= 11 \\
11^2 \bmod 187 &= 121 \\
11^4 \bmod 187 &= 14{,}641 \bmod 187 = 55 \\
11^8 \bmod 187 &= 214{,}358{,}881 \bmod 187 = 33 \\
11^{23} \bmod 187 &= (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 \\
&= 79{,}720{,}245 \bmod 187 = 88
\end{aligned}$$

Modular Arithmetic as Remainders

The easiest way to understand modular arithmetic is to think of it as finding the remainder of a number upon division by another number. For example, since both 15 and -9 leave the same remainder 3 when divided by 12, we say that

$$15 \equiv -9 \pmod{12}.$$

This allows us to have a simple way of doing modular arithmetic: first perform the usual arithmetic, and then find the remainder. For example, to find $123 + 321 \pmod{11}$, we can take

$$123 + 321 = 444$$

and divide it by 11, which gives us

$$123 + 321 \equiv 4 \pmod{11}.$$

However, this could get messy when the numbers get larger. One approach that we could take is to first find the remainders of 123 and 321 when divided by 11 (the remainders are both 2), perform the usual arithmetic, and find the remainder again. In this example, since $123 \equiv 2 \pmod{11}$ and $321 \equiv 2 \pmod{11}$, we can conclude that

$$\begin{aligned} 123 + 321 &\equiv 2 + 2 \pmod{11} \\ &\equiv 4 \pmod{11}. \end{aligned}$$

Congruence

For a positive integer n, the integers a and b are congruent $\bmod n$ if their remainders when divided by n are the same.

EXAMPLE

$$52 \equiv 24 \pmod{7}$$

As we can see above, 52 and 24 are congruent (mod 7) because $52 \pmod{7} = 3$ and $24 \pmod{7} = 3$.

Note that $=$ is different from $\equiv$.

Another way of defining this is that integers a and b are congruent $\bmod n$ if their difference $(a - b)$ is an integer multiple of n, that is, if $\frac{a-b}{n}$ has a remainder of 0.

EXAMPLE

$$36 \equiv 10 \pmod{13}$$

36 and 10 are said to be congruent (mod 13) because their difference $36 - 10 = 26$ is an integer multiple of $n = 13$, that is, $26 = 2 \times 13$.

Digital Signature, RSA-based Digital Signature Generation and Verification

In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature must have the following properties:

■ It must verify the author and the date and time of the signature.

■ It must authenticate the contents at the time of the signature.

■ It must be verifiable by third parties, to resolve disputes.

Thus, the digital signature function includes the authentication function.

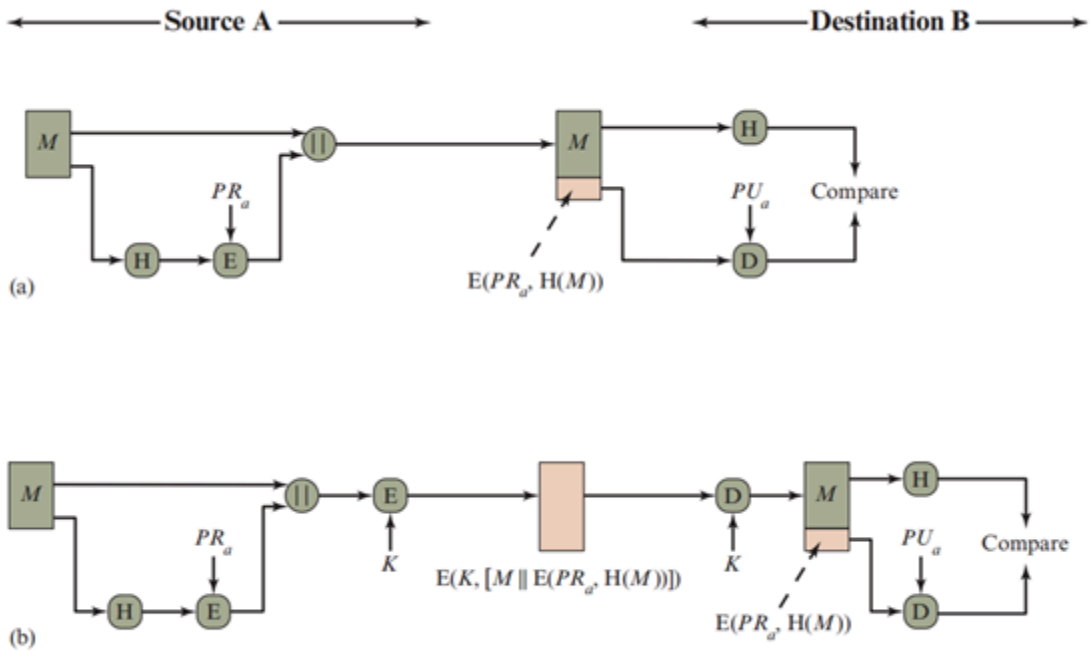


Figure 11.4 Simplified Examples of Digital Signatures

Digital Signature

Public-key encryption can be used for authentication, as suggested by Figure 2.6b.

Suppose that Bob wants to send a message to Alice. Although it is not important that the message be kept secret, he wants Alice to be certain that the message is indeed from him. For this purpose, Bob uses a secure hash function, such as SHA-512, to generate a hash value for the message and then encrypts the hash code with his private key, creating a **digital signature**. Bob sends the message with the signature attached. When Alice receives the message plus signature, she (1) calculates a hash value for the message; (2) decrypts the signature using Bob's public key; and (3) compares the calculated hash value to the decrypted hash value. If the two hash values match, Alice is assured that the message must have been

signed by Bob. No one else has Bob's private key and therefore no one else could have created a ciphertext that could be decrypted with Bob's public key. In addition, it is impossible to alter the message without access to Bob's private key, so the message is authenticated both in terms of source and in terms of data integrity. It is important to emphasize that the digital signature does not provide confidentiality. That is, the message being sent is safe from alteration but not safe from eavesdropping. This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear. Even in the case of complete encryption, there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key.

RSA (Rivest-Shamir-Adleman) is a widely used public-key cryptosystem that can be used for digital signatures. A digital signature is a mathematical scheme that is used to verify the authenticity of digital documents or messages.

Here's a step-by-step approach to implementing RSA for digital signatures:

1. Key generation: Generate a public and private key pair using the following steps:

a. Select two large prime numbers, p and q.

b. Calculate n = p*q.

c. Calculate phi(n) = (p-1)*(q-1).

d. Choose a public exponent e, such that 1<e<phi(n) and gcd(e, phi(n))=1. e. Calculate the private exponent d, such that d is the modular inverse of e modulo phi(n). That is, (d*e) % phi(n) = 1.

The public key consists of the modulus n and the public exponent e. The private key consists of the modulus n and the private exponent d.

2. Signing: To sign a message M, the sender computes the message hash, h = hash(M), using a cryptographic hash function. Then, the sender calculates the signature s as:

s = h^d mod n

Here, d is the private exponent of the sender's key pair.

3. Verification: To verify the signature s, the receiver computes the message hash h' = hash(M). Then, the receiver computes the value of the original message hash h using the signature s and the sender's public key:

h = s^e mod n

If h equals h', the signature is valid. Otherwise, the signature is invalid.

4. Security considerations: The security of the RSA digital signature scheme depends on the security of the underlying RSA key generation and the cryptographic hash function used to compute the message hash. The prime numbers p and q should be large enough to prevent factorization attacks. The public exponent e should be chosen randomly and should not be too small. The hash function should be collision-resistant, meaning that it is computationally

infeasible to find two messages with the same hash. Additionally, the signature scheme should be protected against replay attacks and other forms of attacks.

Hash: Properties of Hash functions, hash algorithm variants

Hash Functions:

Hash functions represent a third type of modern cryptography, which we call keyless cryptography. Instead of using a key, hash functions, or message digests, convert the plaintext into a largely unique and fixed length value, commonly referred to as a hash. You can think of these hash values as fingerprints because they're unique identifiers of a message. Moreover, hashes of similar messages look completely different.

Hash Function Requirements

The purpose of a hash function is to produce a"fingerprint" of a file, message, or other block of data. To be useful for message **authentication**, a hash function H must have the following properties:

1. H can be applied to a block of data of any size.

2. H produces a fixed-length output.

3. H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical.

4. For any given code h, it is computationally infeasible to find x such that H(x) = h. A hash function with this property is referred to as one-way or pre-image resistant.The fourth property is the one-way property: It is easy to generate a code given a message, but virtually impossible to generate a message given a code.

5. For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$. A hash function with this property is referred to as second preimage resistant. This is sometimes referred to as weak collision resistant.

The fifth property guarantees that it is impossible to find an alternative message with the same hash value as a given message.

6. It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

A hash function with this property is referred to as collision resistant. This is sometimes referred to as strong collision resistant. The first three properties are requirements for the practical application of a hash function to message authentication.

Secure Hash Functions are:

Preimage resistant	2^n
Second preimage resistant	2^n
Collision resistant	$2^{n/2}$

Hashing algorithms are just as abundant as encryption algorithms, but there are a few that are used more often than others. Some common hashing algorithms include MD5, SHA-1, SHA-2, NTLM, and LANMAN.

MD5: This is the fifth version of the Message Digest algorithm. MD5 creates 128-bit outputs. MD5 was a very commonly used hashing algorithm. That was until weaknesses in the algorithm started to surface. Most of these weaknesses manifested themselves as collisions. Because of this, MD5 began to be phased out.

SHA-1: This is the second version of the Secure Hash Algorithm standard, SHA-0 being the first. SHA-1 creates 160-bit outputs. SHA-1 is one of the main algorithms that began to replace MD5, after vulnerabilities were found. SHA-1 gained widespread use and acceptance.

SHA-2: This is actually a suite of hashing algorithms. The suite contains SHA-224, SHA-256, SHA-384, and SHA-512. Each algorithm is represented by the length of its output. SHA-2 algorithms are more secure than SHA-1 algorithms, but SHA-2 has not gained widespread use.

LANMAN: Microsoft LANMAN is the Microsoft LAN Manager hashing algorithm. LANMAN was used by legacy Windows systems to store passwords. LANMAN used DES algorithms to create the hash.

Triple DES: The life of DES was extended by the use of triple DES (3DES), which involves repeating the basic DES algorithm three times, using either two or three unique keys, for a key size of 112 or 168 bits.

3DES has two attractions that assure its widespread use over the next few years.First, with **its 168-bit key length**, it overcomes the vulnerability to brute-force attack of DES.

Second, the underlying encryption algorithm in 3DES is the same as in DES. This algorithm has been subjected to more scrutiny than any other encryption algorithm over a longer period of time, and **no effective cryptanalytic attack based on the algorithm rather than brute force has been found.** Accordingly, there is a high level of confidence that 3DES is very resistant to cryptanalysis. If security were the only consideration, then 3DES would be an appropriate choice for a standardized encryption algorithm for decades to come.

The principal drawback of 3DES is that the algorithm is relatively **sluggish** in software. The original DES was designed for mid-1970s hardware implementation and does not produce efficient software code. 3DES, which requires three times as many calculations as DES, is correspondingly slower. A secondary drawback is that both DES and 3DES use a **64-bit block size**. For reasons of both efficiency and security, a larger block size is desirable.

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

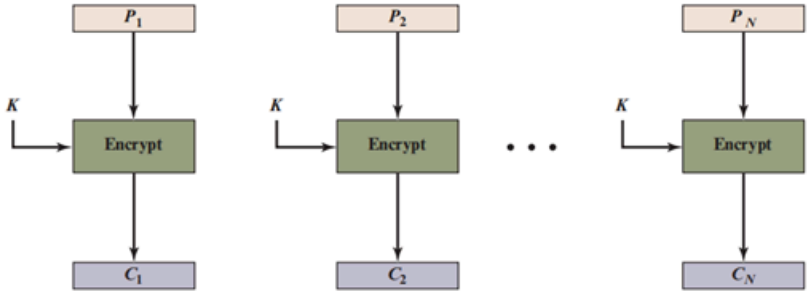
• The cost of breaking the cipher exceeds the value of the encrypted information.

• The time required to break the cipher exceeds the useful lifetime of the information.

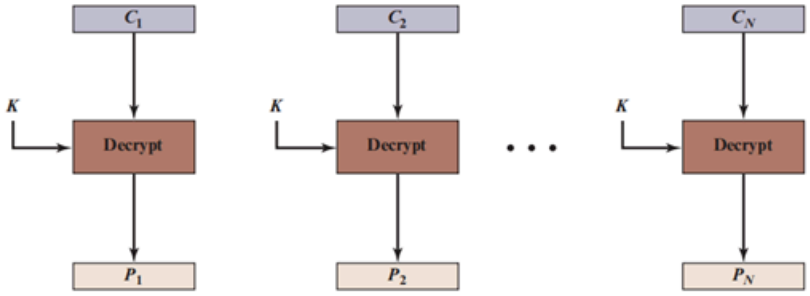
An encryption scheme is **unconditionally secure** if the ciphertext generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext, no matter how much ciphertext is available. That is, no matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there. With the exception of a scheme known as the **onetime pad**, there is no encryption algorithm that is unconditionally secure.

Table 7.1 Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of plaintext bits is encoded independently using the same key.	• Secure transmission of single values (e.g., an encryption key)
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next block of plaintext and the preceding block of ciphertext.	• General-purpose block-oriented transmission • Authentication
Cipher Feedback (CFB)	Input is processed s bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding encryption output, and full blocks are used.	• Stream-oriented transmission over noisy channel (e.g., satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is incremented for each subsequent block.	• General-purpose block-oriented transmission • Useful for high-speed requirements

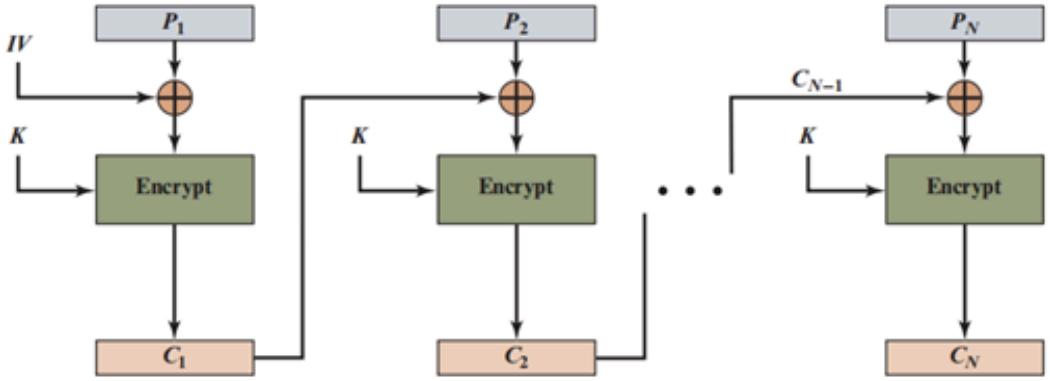


(a) Encryption

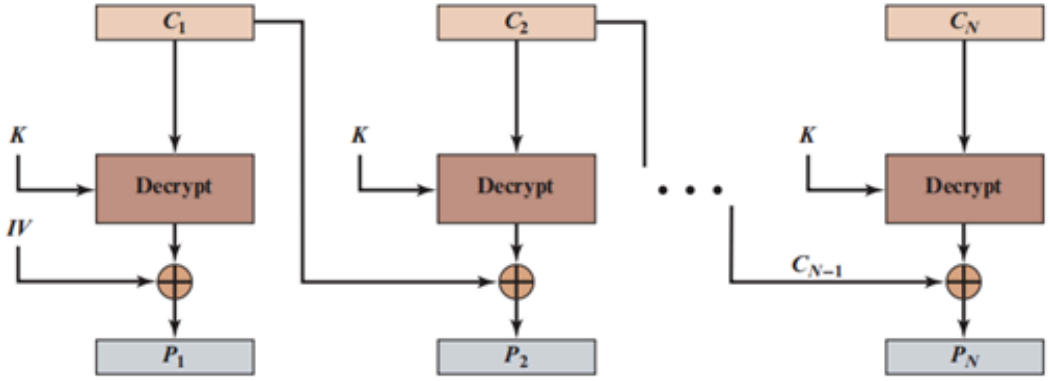


(b) Decryption

Figure 7.3 Electronic Codebook (ECB) Mode

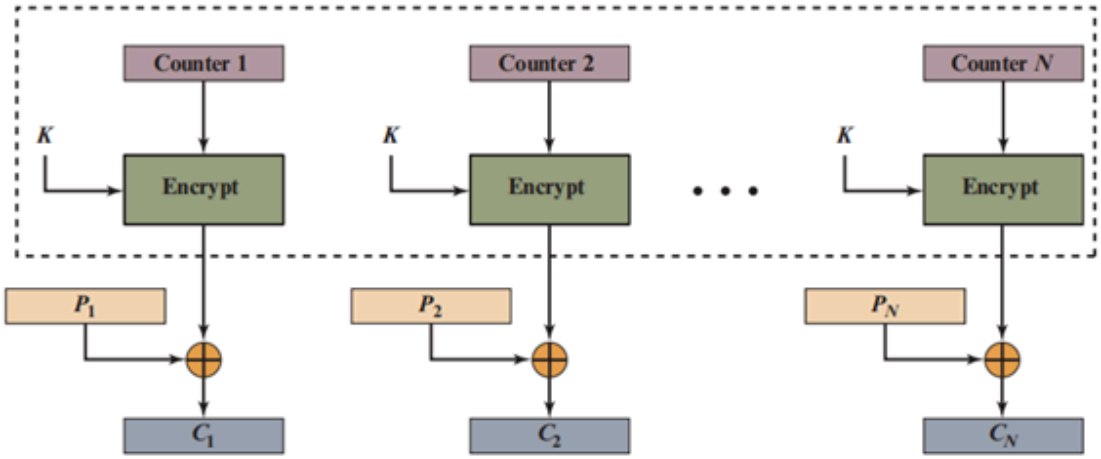


(a) Encryption

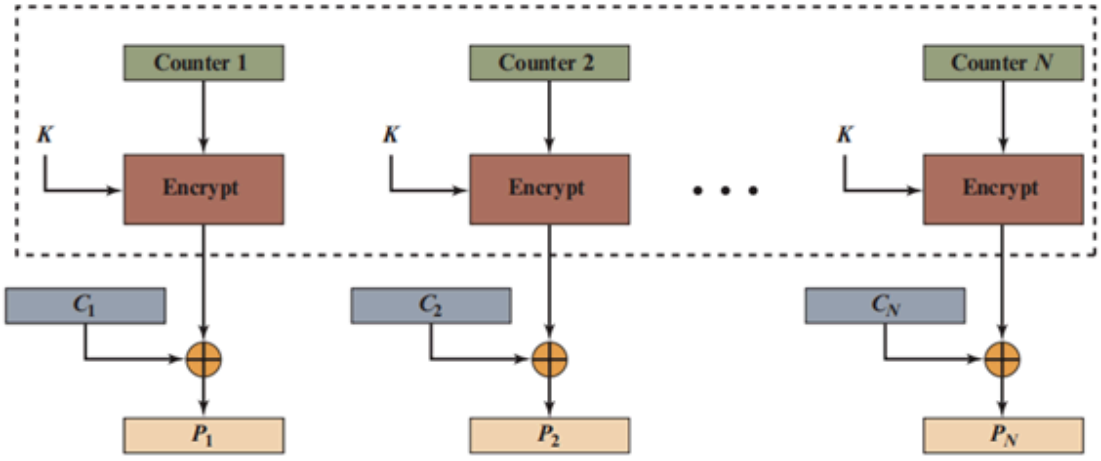


(b) Decryption

Figure 7.4 Cipher Block Chaining (CBC) Mode



(a) Encryption



(b) Decryption

Figure 7.7 Counter (CTR) Mode