
Cryptography

Domain 5

Overview

Cryptography is both an art and a science – the use of deception and mathematics, to hide data, as in steganography, to render data unintelligible through the transformation data into an unreadable state, and to ensure that a message is not been altered in transit as well as to provide assurance of who sent the message, authentication of source, and proof of delivery

Key Areas of Knowledge

- ➡ **Application and use of cryptography**
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Agenda

- ➡ **Application and use of cryptography**
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Cryptography Goals

➡ Confidentiality

- Unauthorized parties cannot access information

➡ Authenticity

- Validating the source of the message to ensure the sender is properly identified

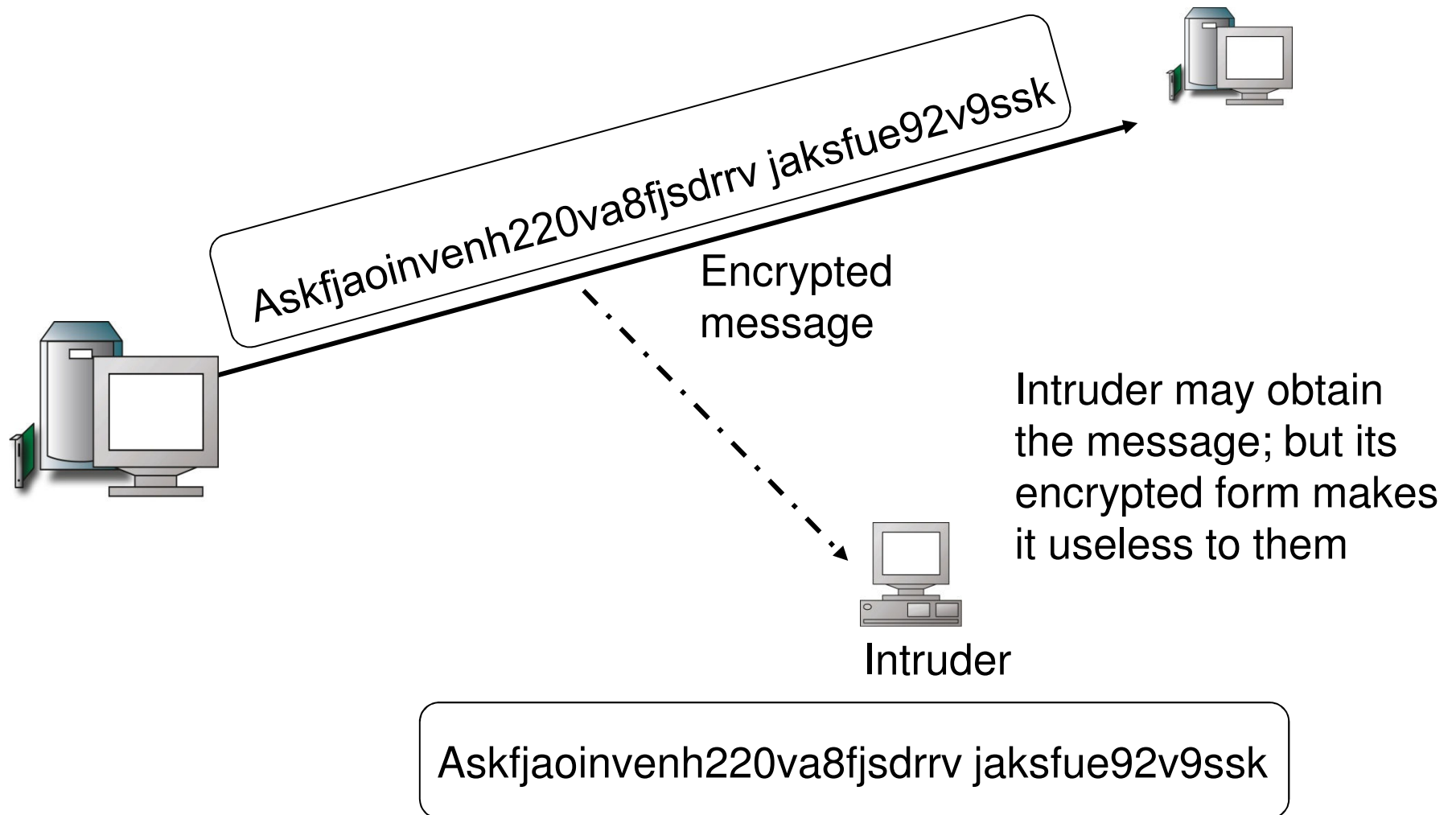
➡ Integrity

- Assurance that the message was not modified during transmission, accidentally or intentionally

➡ Non-repudiation

- A sender cannot deny sending the message at a later date

Protection by Encryption



Traditional vs. Modern Cryptography

➡ **Cryptography originally used for secrecy**

➡ **Cryptography now used for many things:**

- Prevent unauthorized disclosure of information
- Detect tampering or injection of false data
- Detect deletion of data
- Prevent repudiation
- Ensure protection of “data at rest” and “data in transit”

Modern Cryptography Components

➡ **Binary Operations**

➡ **Large Key Spaces**

➡ **Mathematical Algorithms**

➡ **Symmetric Ciphers**

- Block
- Stream

➡ **Public Key Algorithms**

Binary Operations and Key Components

➡ Key is just a string of bits

- May be a single large number or group of numbers
- Number of possible keys = 2^n

➡ Plaintext is digital representation of data

- ASCII, MS Word, Excel, e-mail, etc.

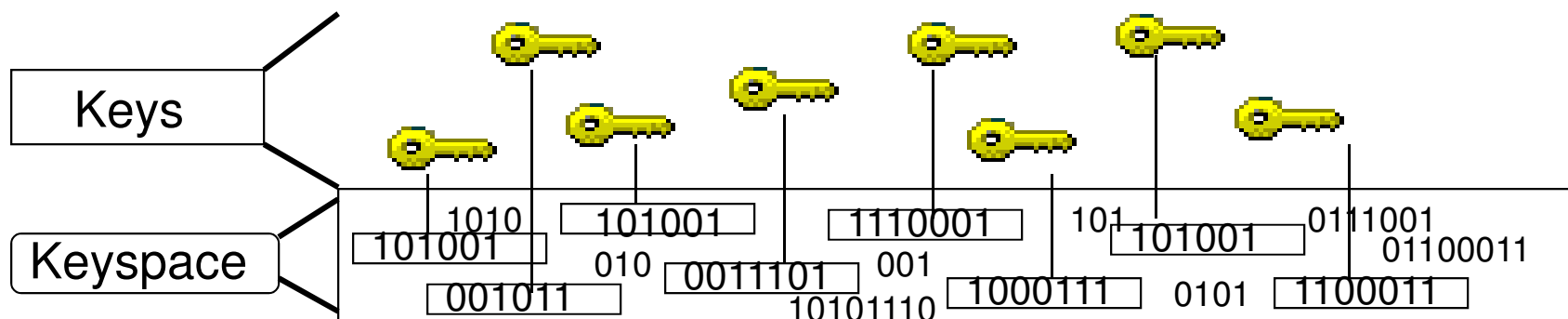
➡ Encryption and decryption operations (the algorithm)

- bit-wise operations
 - XOR, shift left/right, substitutions/permutations
- mod n arithmetic using numerical values
 - add, subtract, multiply, divide, raise to power

Possible Key Values

➔ Keyspace

- Range of possible values that can be used to construct a key
- The larger the keyspace, the more possible key values, and the more random the whole process, which increases the cryptosystem's strength



Symmetric Keyspace

➡ For a small 16-bit key, the keyspace is 2^{16} , or 65,536 keys

➡ For DES (56 bits): 2^{56} is over 72,000,000,000,000,000 (72 quadrillion) possible keys



=

0000 ... 00000

.

.

.

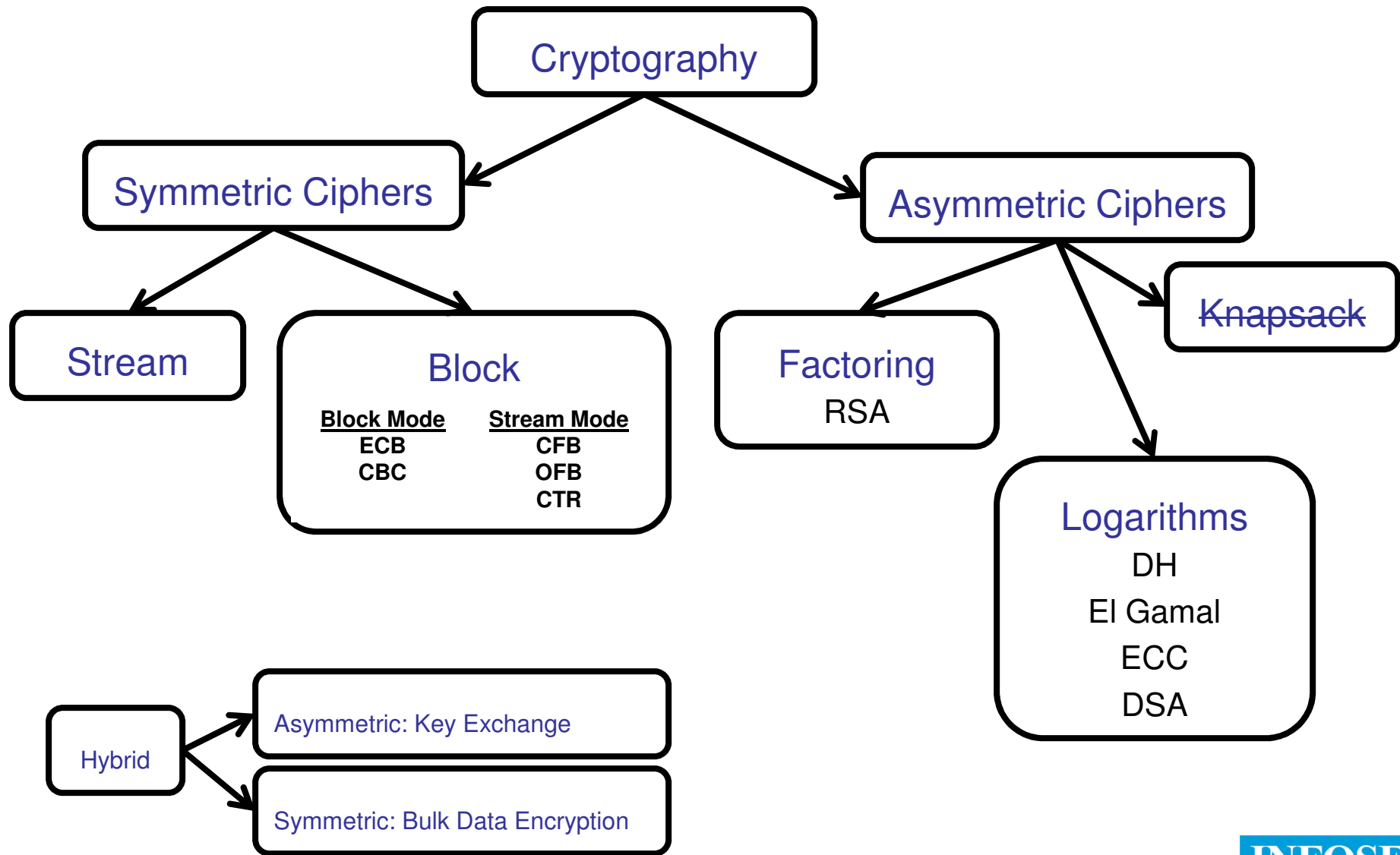
1111 ... 11111

Strength of a Cryptosystem

➡ Effectiveness of a Cryptosystem

- Strength of the encryption method
 - Algorithm
 - Secrecy of the key
 - Length of the key
- Strength of the protection mechanism should be used in correlation to the sensitivity of the data being encrypted
- Even if the algorithm is very complex and thorough, there are other issues within encryption that can weaken the strength of encryption methods

Components of Cryptography



Components of Cryptography

➡ Symmetric Ciphers (Algorithms)

- Stream Ciphers
 - Exclusive OR (XOR)
- Block Ciphers
 - Substitution Ciphers - replacing one value for another
 - Transposition/Permutation Ciphers - change in relative position

➡ Asymmetric Ciphers

- Public Key Cryptography – public/private key pairs

➡ Cyptosystem Solutions

- Confusion – hiding patterns in the plaintext by substitution
- Diffusion – transposing the plaintext throughout the ciphertext
- Avalanche – a change in one bit of the plaintext causes a change in half the resultant ciphertext

Different Security Goals

➡ Authenticity

- Open message format (public message)
 - Encrypt with sender's private key
 - No confidentiality because anyone with public key can decrypt

➡ Confidentiality

- Secure message format (private message)
 - Encrypt with receiver's public key
 - Only person with private key can decrypt

➡ Authentication and Confidentiality

- Secure and signed format
 - Encrypt with sender's private key and encrypt again with receiver's public key
 - Provides mutual authentication

Agenda

- ➡ Application and use of cryptography
- ➡ **Cryptographic lifecycle and encryption concepts**
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

History of Cryptography

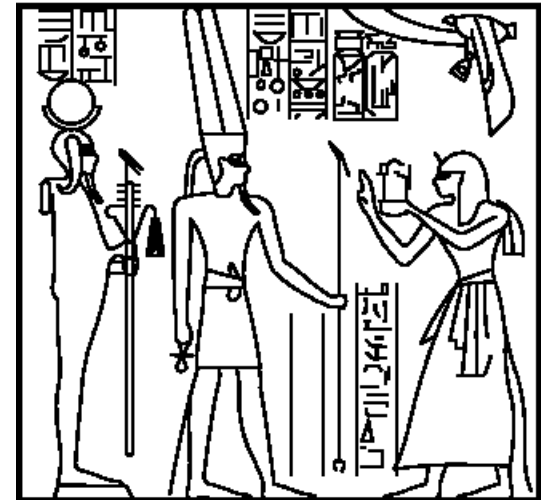
➡ Symmetric Ciphers

- Hieroglyphics
- Scytale Cipher
- Caesar Cipher
- Vigenere Cipher
- Vernam Cipher

History of Cryptography

➔ Hieroglyphics

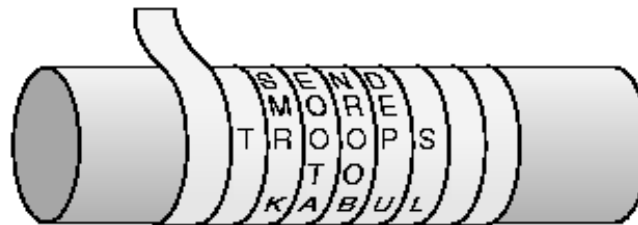
- 2000 BC
- First known cryptographic method
- Not really for secrecy
- Use of “non-standard” hieroglyphics



Scytale Cipher

➔ Spartans wrapped papyrus around a rod to encrypt and decrypt a message

- 400 B.C.
- Used to convey military directives



Substitution Cipher

➡ Caesar Cipher

- One character is replaced with another character
- When only one set of characters is used for substitution, it is a mono-alphabetic substitution algorithm
- Julius Caesar also used a similar algorithm that shifted characters three places

Caesar Cipher

➡ Caesar Cipher Example:

➡ Shift each letter in the alphabet forward by K = 3 positions:

Standard Alphabet:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cryptographic Alphabet:

DEFGHIJKLMNOPQRSTUVWXYZABC

➡ Example:

- Plaintext: CISSP DOMAINS
- Ciphertext: FLVVS GRPDLQV

➡ Attack:

- Letter Frequency Analysis: e, t, o, a, n, r, i, s, h
- Other common “letter” patterns

Polyalphabetic Substitution

➔ Vigenere Cipher

- Proposed by Blaise de Vigenere from the court of Henry III of France in the sixteenth century

➔ Polyalphabetic Substitution

- A polyalphabetic substitution using two or more cipher alphabets

Vigenere Cipher

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

Key Word:

"Intense"

Plain Text:

"CISSP DOMAINS"

Cipher Text:

"kvlw cvru nbre"

Plain Text: **ciss pdom ains**

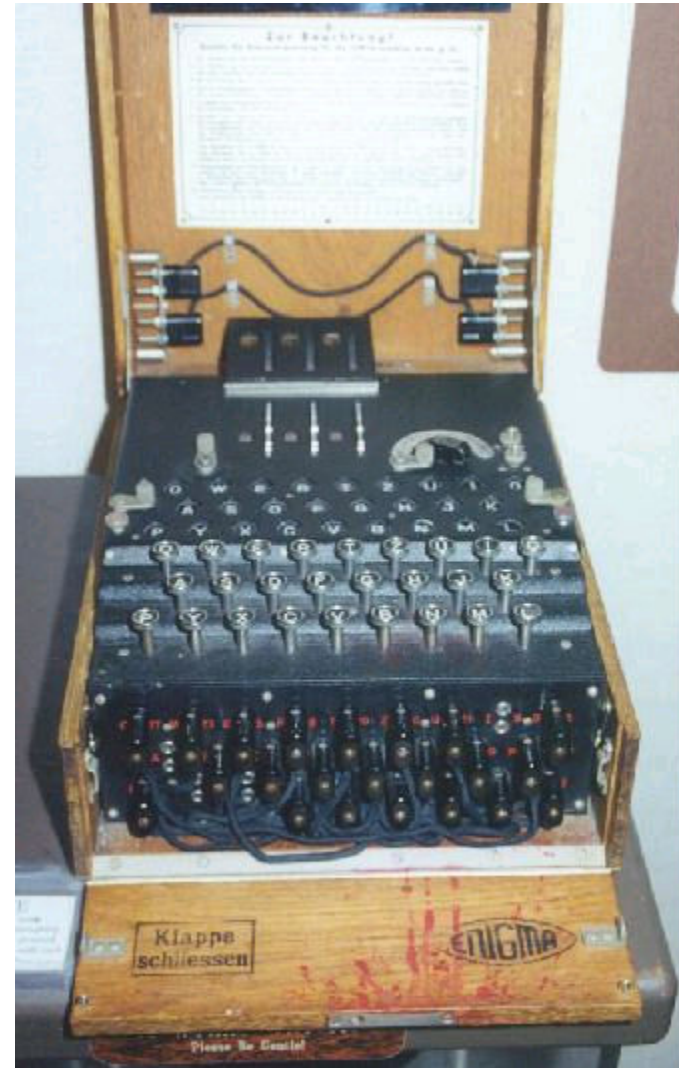
Key Word: **inte nsei nten**

Cipher Text: **kvlw cvru nbre**

Cryptography in War

German Enigma

- Used in World War II to encrypt telegraphic communication
- Rotor cipher machine that used polyalphabetic substitution
- Key was the original setting of the rotors and the sequence of advancement for each rotor
- Individual rotors are connected in a bank
- Character entered and substituted by each rotor for encryption



Cryptography Definitions

➡ Cryptography

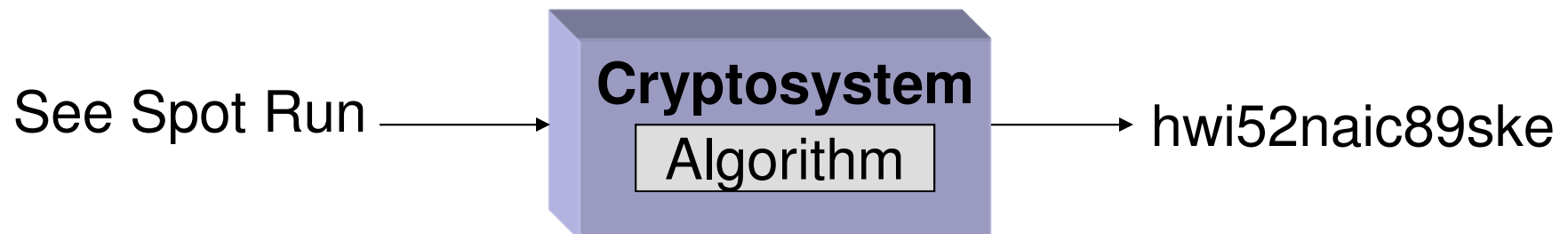
- Science of hiding meaning of communication

➡ Cryptanalysis

- Science of studying and breaking the secrecy of encryption algorithms and their necessary pieces

➡ Cryptosystem

- Mechanism that carries out the encryption process



Cryptography Definitions

➡ Work Factor

- The amount of time and resources needed to overcome protective measures of a cryptosystem; “breaking” is decreasing the work factor to a reasonable level

➡ Cryptographic Algorithm (Cipher)

- Procedure to encrypt plaintext into ciphertext and vice versa

➡ Cryptovvariable (Key)

- A variable used in conjunction with an algorithm to encrypt and decrypt data

➡ Key Space

- The range of available key values to be used by an algorithm

Cryptography Decoded

➡ Encryption

- The process of turning plaintext into ciphertext

➡ Decryption

- The process of turning ciphertext into plaintext

➡ Encryption/Decryption requires:

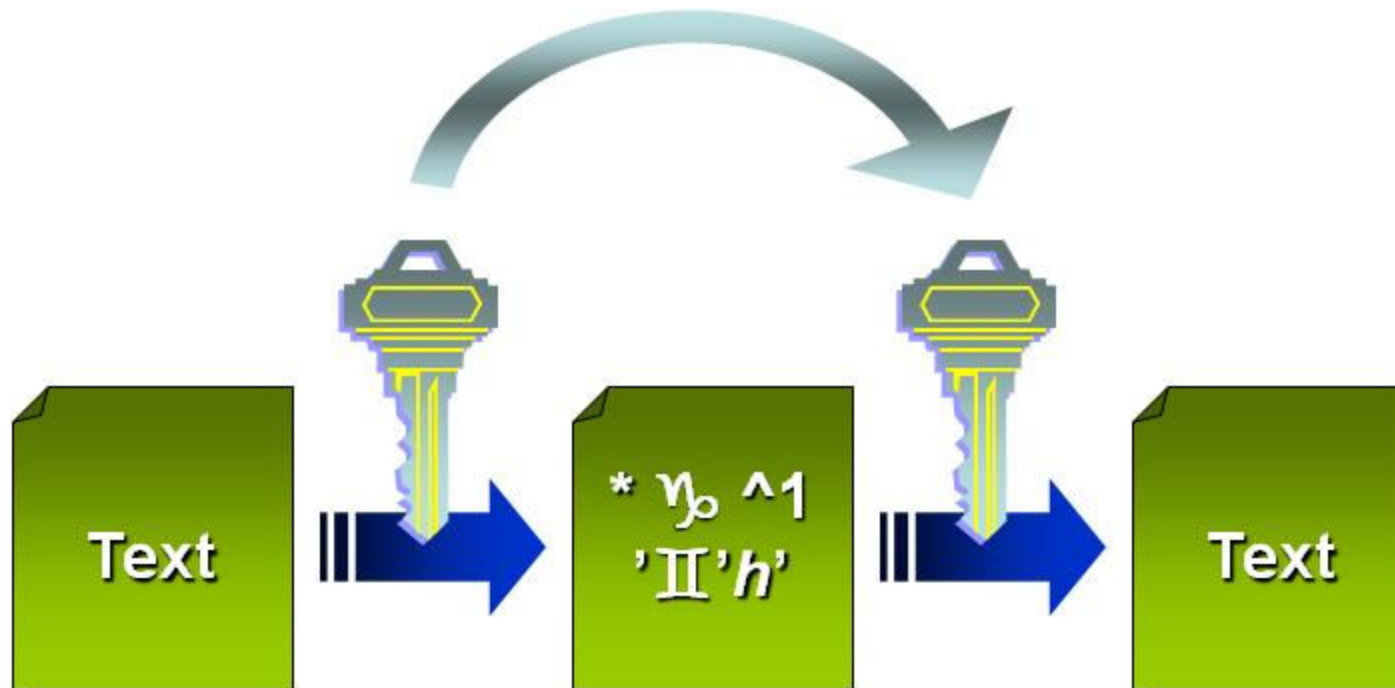
- An algorithm
- A key

➡ Two types of encryption operations:

- Symmetric and asymmetric

Symmetric Cryptography: The “Secret” Key

➡ One secret key shared for:
Enciphering and Deciphering



Asymmetric Cryptography or Public Key

➡ A Key Pair

- Public / Private key
- Bound mathematically
- Theoretically impossible to find a key based on the other key

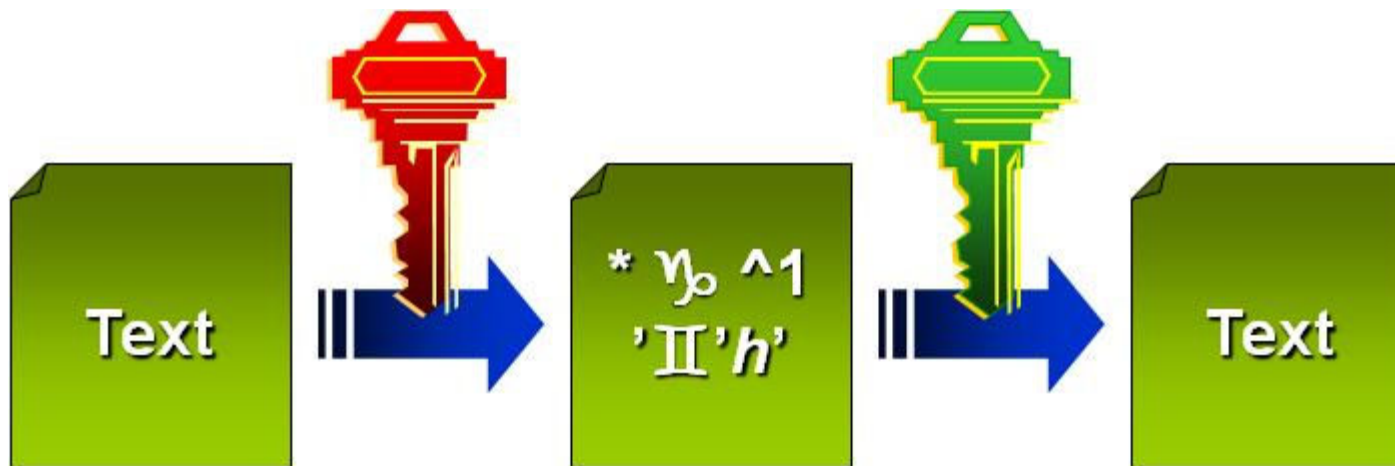
➡ No shared key between the two parties



Asymmetric Cryptography

➔ Two keys acting in conjunction with each other for:

- Enciphering
- Deciphering



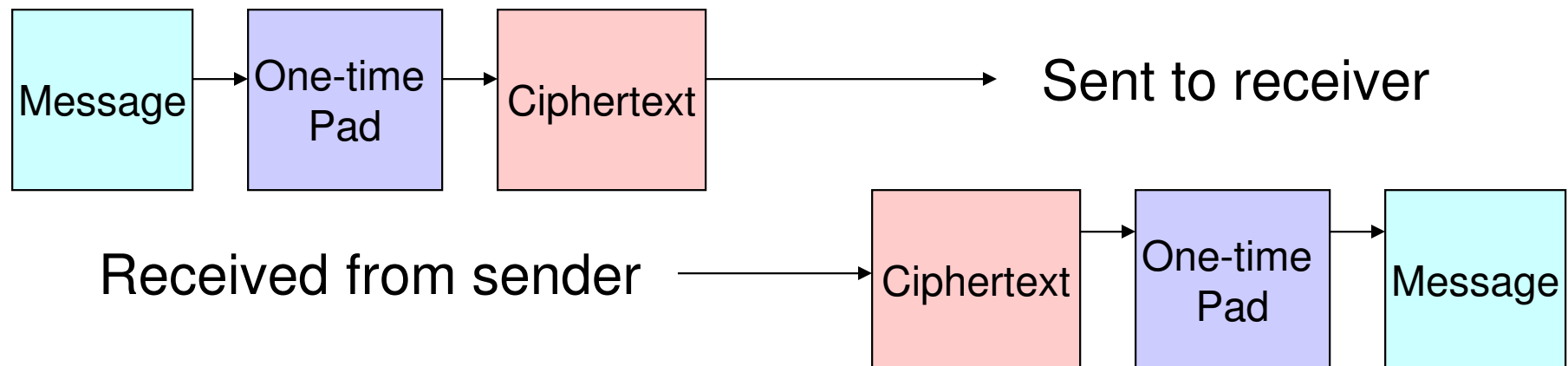
Vernam Cipher (aka One-time Pad)

➔ Devised by Gilbert Vernam in 1917

- Uses a one-time random “pad” that is at least as long as the message that is to be encrypted

➔ One-time pads are used in pairs

- One copy is used by the sender, and the other copy is used by the recipient
- Should only be used once



One-time Pad Encryption Scheme

- ➡ **Perfect encryption scheme**
- ➡ **Computationally “impossible” to break, provided pad is truly random**
 - Unbreakable by exhaustive search or brute force
- ➡ **Pad (key) of non-repeating set of random bits that are XORed against the message to create ciphertext**
- ➡ **Pad (key) is the same size as the message and only used once**
- ➡ **A software number generator creates the pseudo-random values to be used in the pad**

Binary Mathematics – Exclusive OR

➡ Exclusive OR (XOR) functionality:

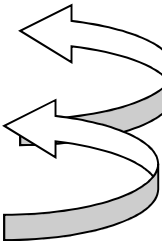
- Binary mathematical operation that is applied to two bits
- Two Rules:
 - If both bits are the same, the result is zero
 - If they are different than each other, the result is one
- Logical “either/or”
 - Output is true if either, but not both, of the inputs are true
 - Output is false if both inputs are false or both inputs are true

➡ Popular function in modern cryptography

Binary Mathematics – XOR

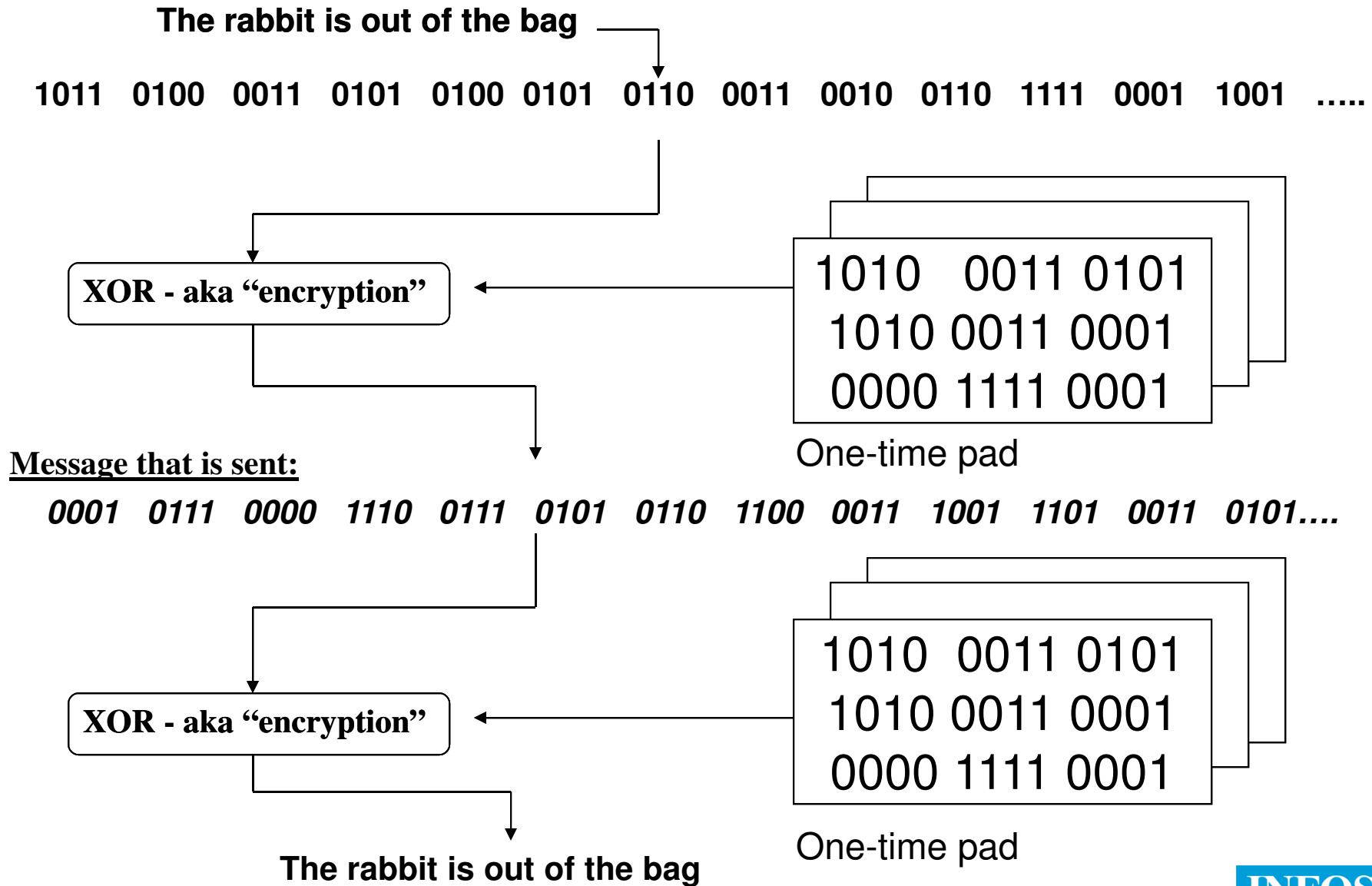
➔ The XOR Binary Function

- Rule #1: If both inputs are the same, the result = 0
- Rule #2: If both inputs are different, the result = 1

	1101001	1100001	Message	
<u>XOR</u>	0010101	1101011	Key	
	1111100	0001010	Cipher Text	

XOR Benefit → It is *reversible*

One-time Pad “Pages”



One-time Pad

➡ True One-Time Pad Use

- Impractical for most uses because of the overhead of creating these large keys and distributing them
 - Used on the hotline between U.S. and Russia
 - Used by some spy communications
- Pseudo one-time pad
 - SSL / TLS
 - VPN technology like IPsec

➡ Weakness

- Key exchange is cumbersome

Running Key Cipher

- ➡ Uses a key that does not require an electronic algorithm and bit alterations, but clever steps in the physical world
- ➡ Book number, page number, line number, word number
- ➡ Example – 3rd book, page 112, line 4, 6th word is “informative”:
 - InfoSec Institute is the most informative and useful education institution available

Symmetric Stream Cipher

- ➔ **Can be much faster than a block cipher**
- ➔ **Operates on smaller units of plaintext (bits), while block ciphers work on larger units (64 bits)**
 - Does not divide message into blocks, treats the message as a stream of bits
- ➔ **More suited for hardware implementation than a block cipher**
 - Performs mathematical functions on individual bits
 - Algorithm uses a keystream generator to create a keystream
 - Keystream is same length as the plaintext message
 - Keystream is XORed with plaintext for encryption

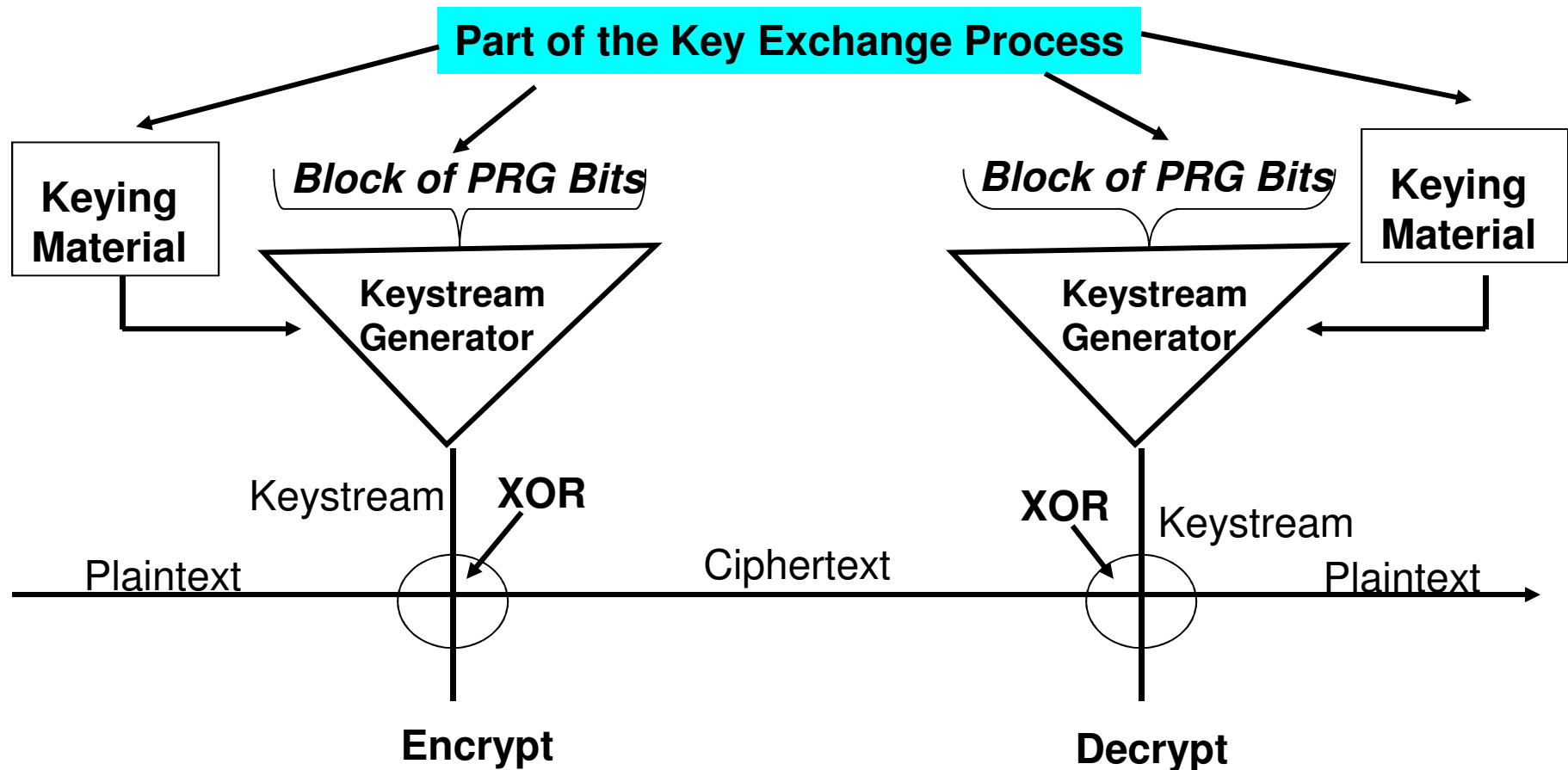
Strength of a Stream Cipher

➔ Strength and effectiveness of stream cipher depends upon:

- Long periods of no repeating patterns within keystream values
- Statistically unpredictable
- The keystream is not linearly related to the key
- Statistically unbiased keystream (as many 0's as 1's)
- Used for secure wireless communications
 - RC4 cipher - Wired Equivalent Privacy (WEP); Wi-Fi Protected Access (WPA)
 - Bluetooth “E0” cipher

Symmetric Stream Cipher

➡ Key into Keystream



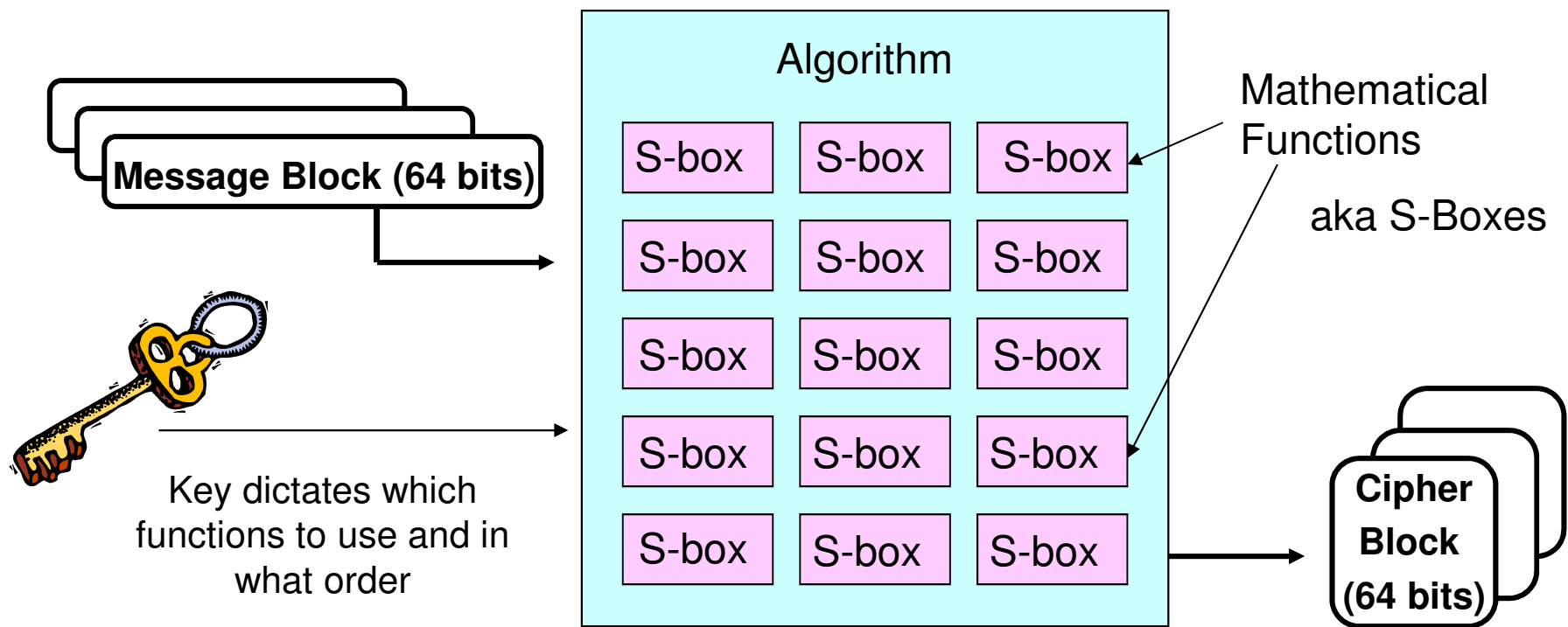
Symmetric Block Cipher

➡ Block Cipher

- Message is divided into blocks and put through mathematical functions called Substitution Boxes (S-Boxes)
- Algorithm dictates all possible functions, and the key determines which of these possibilities will be used and in what order
- Each function performs a different mathematical operation
- Cipher should contain confusion and diffusion
 - Strong algorithms make reengineering basically impossible

Symmetric Block Cipher

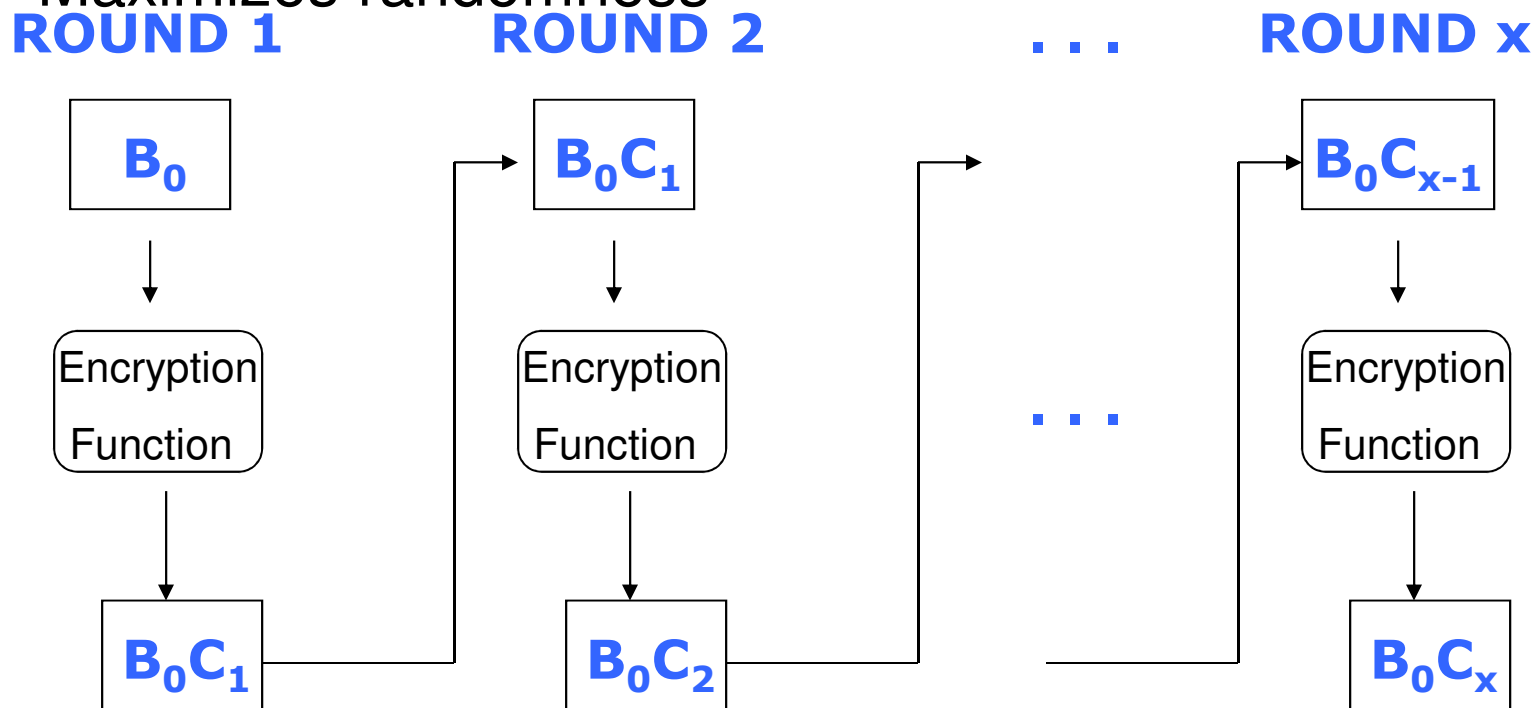
➡ Operation of a Block Cipher



Symmetric Block Cipher

➡ Rounds of Calculations

- The mathematical operation is performed several times on the same message block
- Maximizes randomness



Symmetric Block Cipher

➔ **Each round in a block cipher uses a different subkey**

➔ **Key Scheduling:**

- A large key (K) is created and broken into sub keys K_n
- The same Key schedule is used for both encryption and decryption

Block Cipher Example: 56-bit DES

➡ **DES has a key length of 56 bits + 8 parity bits**

➡ **Can be viewed as 8 keys of 7 bits + 1 parity bit:**

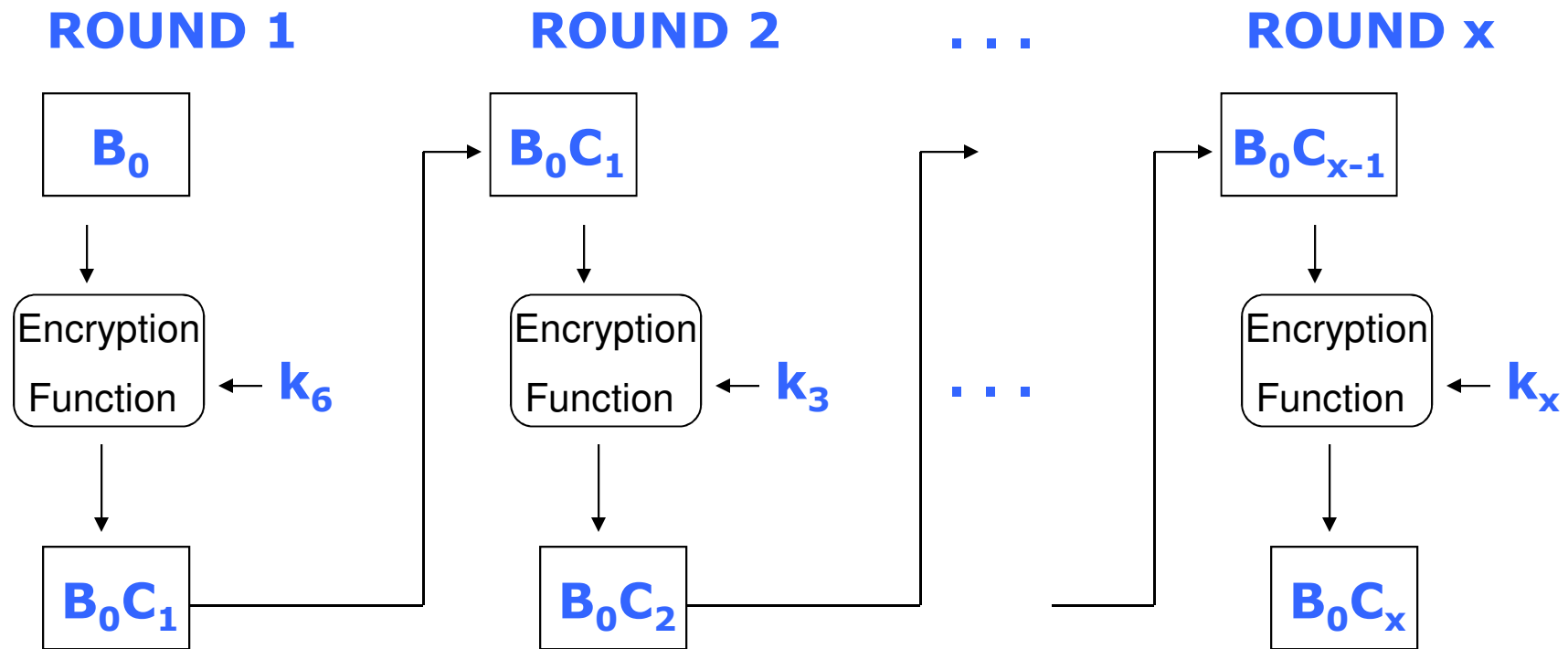
- K1 is Key 1 = 7 bits + 1 parity bit
- K2 is Key 2 = 7 bits + 1 parity bit
- K3 is Key 3 = 7 bits + 1 parity bit
- K4 is Key 4 = 7 bits + 1 parity bit
- K5 is Key 5 = 7 bits + 1 parity bit
- K6 is Key 6 = 7 bits + 1 parity bit
- K7 is Key 7 = 7 bits + 1 parity bit
- K8 is Key 8 = 7 bits + 1 parity bit
- Total: 56 8

➡ **Sample Key Schedule:**

- K6, K3, K5, K8, K7, K1, K2, K4

Symmetric Block Cipher

➡ Rounds and Key Scheduling



The Clipper Chip (US Government)

➡ Clipper Chip

- Protects private communications
- Agents can obtain the "keys" upon "legal authorization."
- Keys are held by two government "escrow agents"
- Developed by NSA
- Skipjack algorithm was classified as Secret
- Precluded any public scrutiny
- 80 bit key
- 16 bit checksum
- Several deficiencies
- Mainly a backdoor into your private data
- Declared really dead in 1996

Key Escrow (US Government)

- ➡ Behind the clipper chip was the key escrow
- ➡ At the factory any device would get a Clipper chip
- ➡ The crypto key had to be provided to the government
- ➡ The key was kept in Escrow by the government
- ➡ If agencies had "established their authority" to listen to a communication, then the password would be given to those government agencies
- ➡ They would then decrypt all data transmitted on the specific telephone device
- ➡ The EFF called this the "key surrender"
- ➡ Also called Fair Cryptosystem

Data Encryption Standard (DES)

➡ Data Encryption Standard (DES)

- IBM submitted their Lucifer Algorithm for the NSA DES standard in 1974
- Original Lucifer Algorithm used 48 to 128-bit keys, and NSA implemented 64-bit key
 - 56 bits make up the true key and 8 bits are used for parity
- Lucifer was altered and called Data Encryption Algorithm (DEA)

Data Encryption Standard (DES)

➡ Block Symmetric Algorithm

➡ Blocks of 64 bits are put through 16 rounds of transposition and substitution functions

- Order and type of functions is dictated by the key value

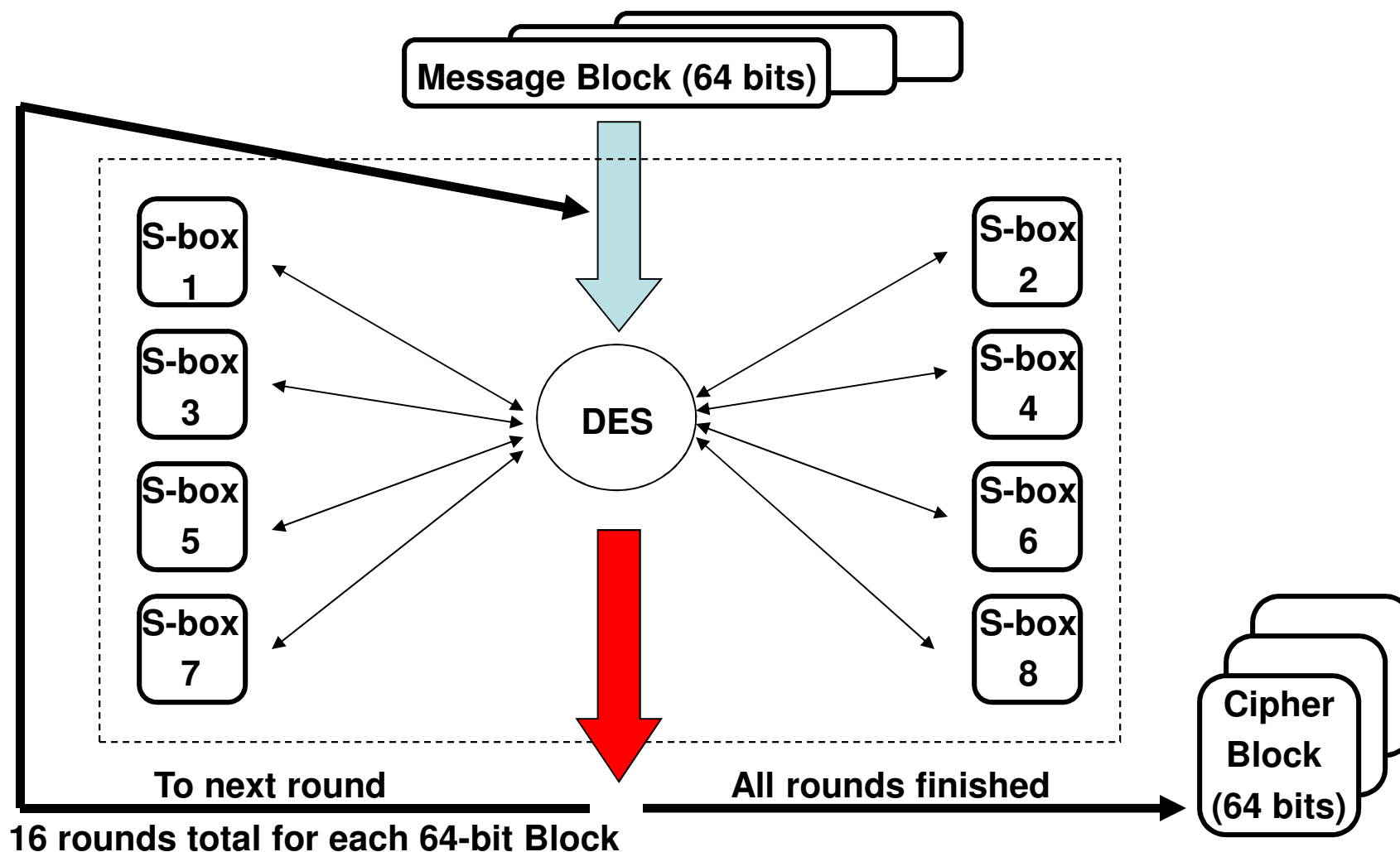
➡ For government agencies

- Used to protect sensitive but unclassified data

➡ Double DES

➡ Triple DES

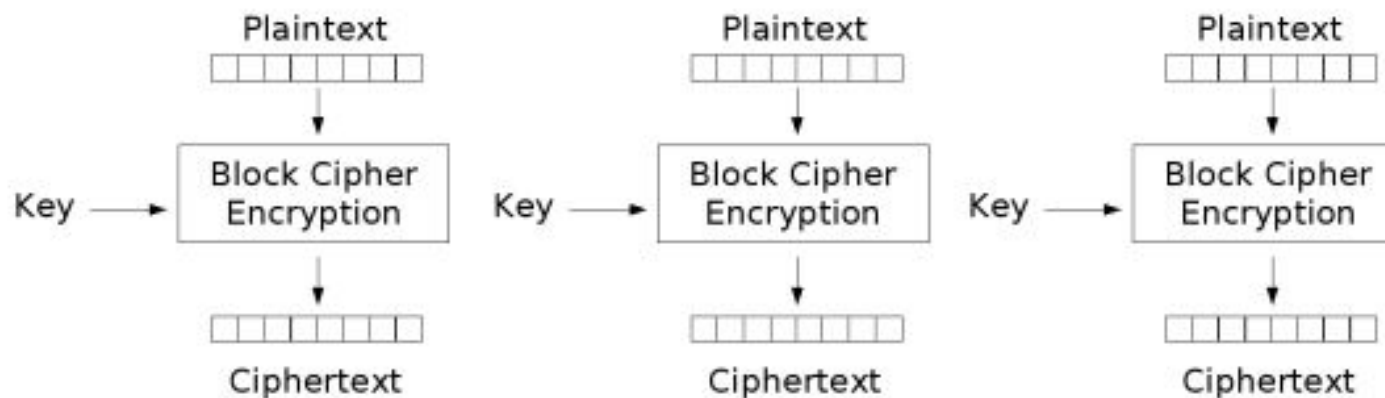
DES Conceptually - Review



Modes of Block Ciphers - ECB

➡ Electronic Code Book (ECB) Mode

- Same ciphertext is always produced for the same plaintext
- Easier to identify patterns
- It is best used on small amounts of data
- Each key indicates a different code book
- Uses MAC for integrity and authentication



Modes of Block Ciphers – CBC

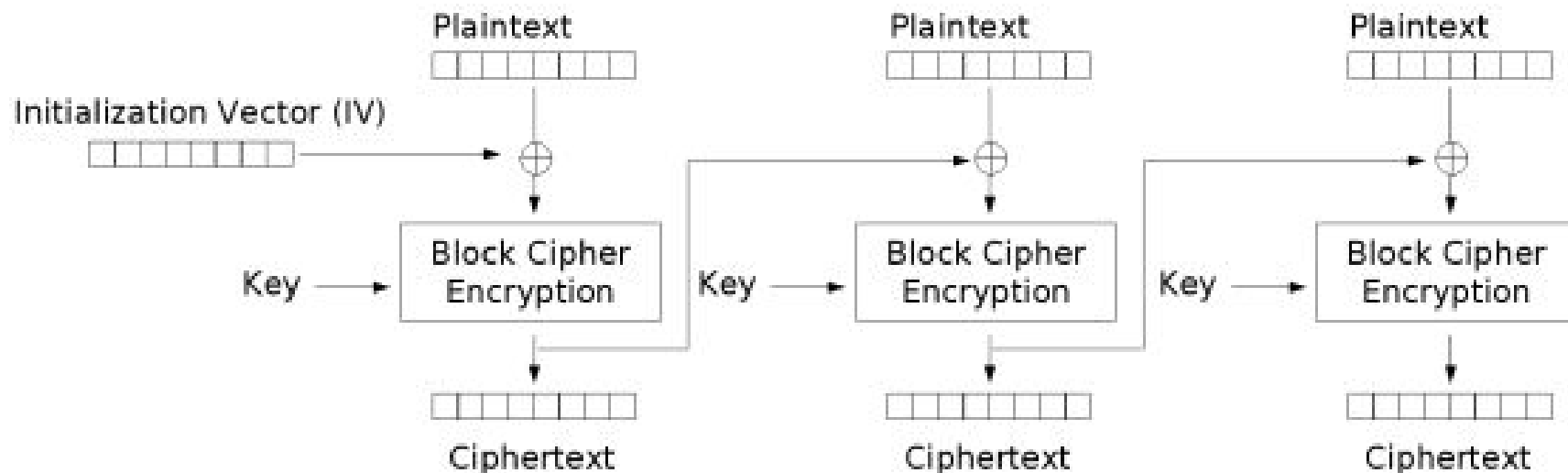
➡ Cipher Block Chaining (CBC)

- Encryption is dependent on values from the previously encrypted block
- First block will be XORed against the IV as no previous ciphertext exist for that first block
- Each block of encrypted ciphertext is XORed with the next plaintext block to be encrypted

Modes of Block Ciphers – CBC

➡ Cipher Block Chaining (CBC)

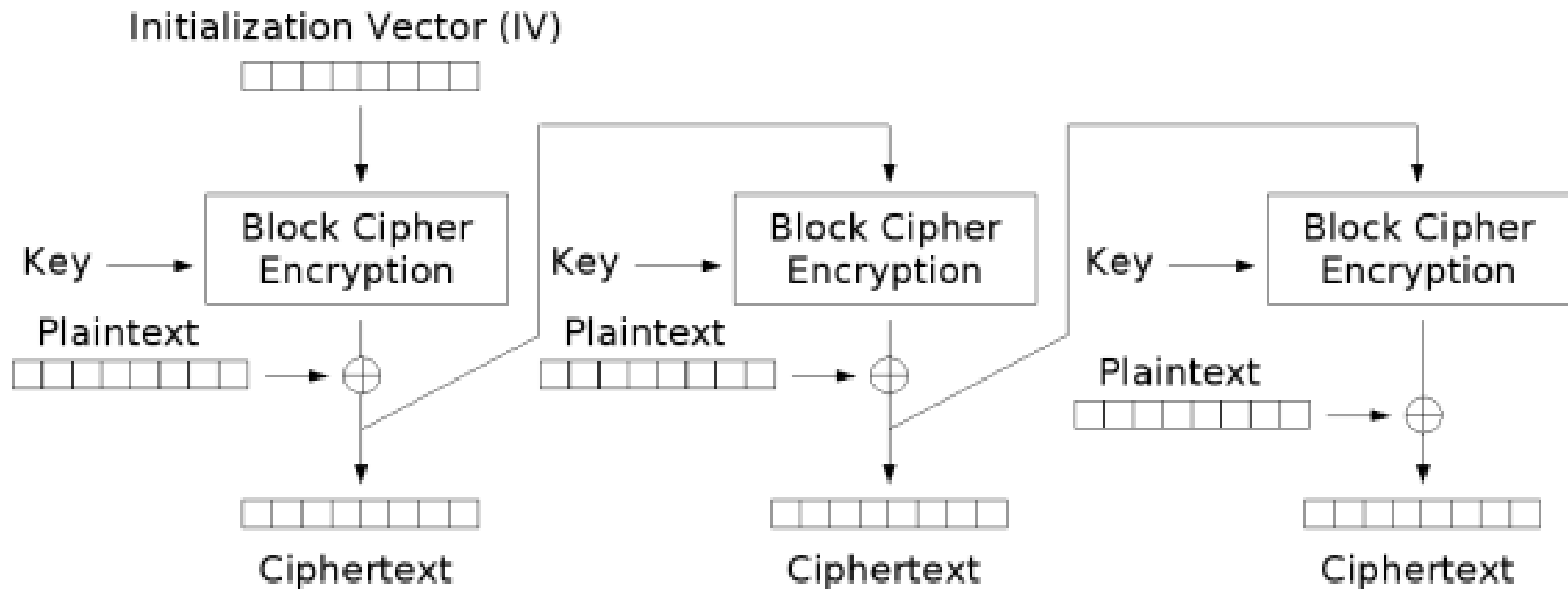
- Encrypts 64-bit blocks, but chains them together by XORing each ciphertext block with the next plaintext block before encryption
 - Encrypted data stream is more randomized
 - Does not show patterns as does ECB



Modes of Block Ciphers – CFB

➡ Cipher Feedback Mode (CFB)

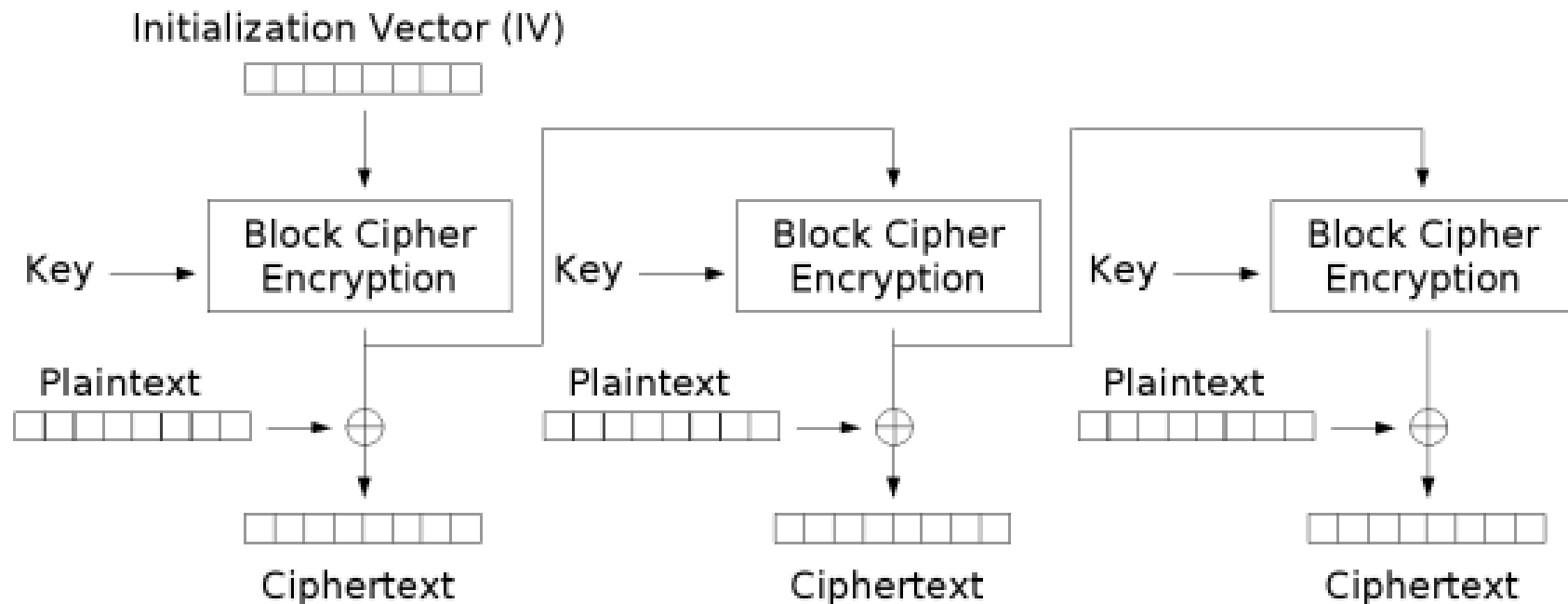
- Previous ciphertext is used to encrypt the next block of data
- Often used to encrypt individual characters (terminals)



Modes of Block Ciphers – OFB

➔ Output Feedback (OFB) Mode

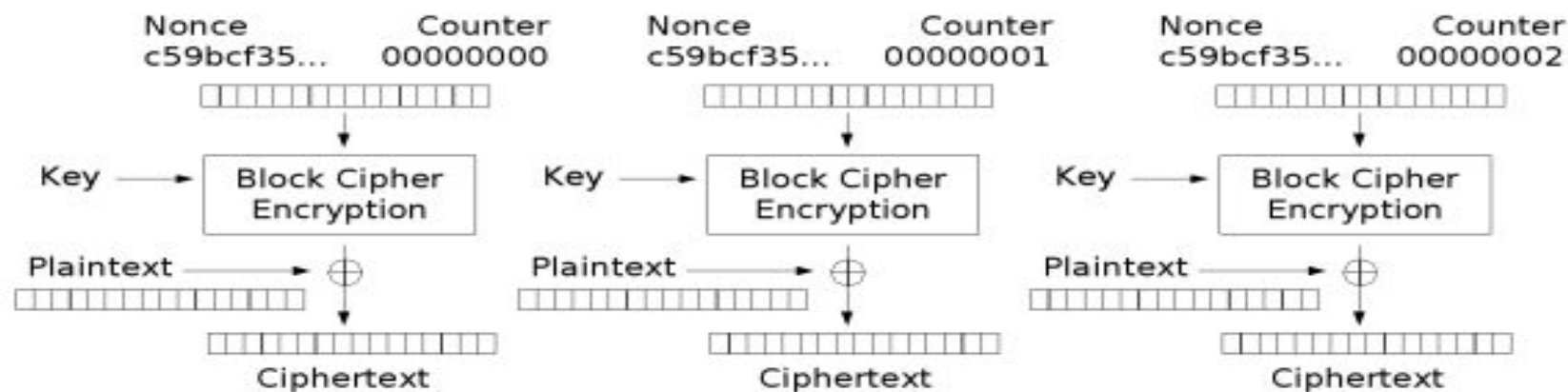
- The entire output of the previous block's calculation is used as input for the next block's encryption
- Often used to encrypt satellite communications



Modes of Block Ciphers – CTR

➡ Counter (CTR) Mode

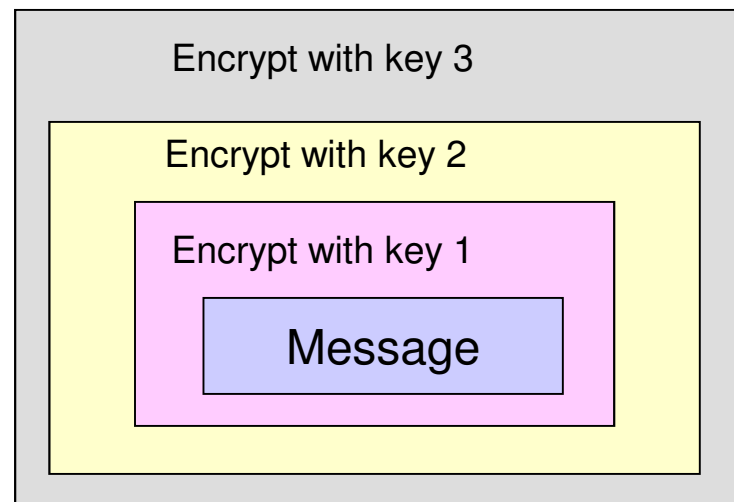
- Similar to OFB, but the IVs are successive values of a "counter"
- CTR mode is well suited to operation on a multi-processor machine because the encryption of each block can be performed in parallel
- Note that the nonce in this graph is the same thing as the IV in the other graphs and is concatenated, added, or XORed with the counter value



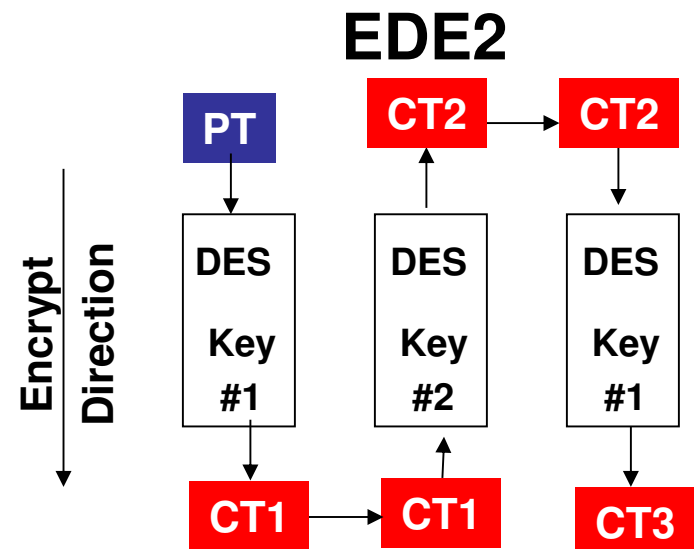
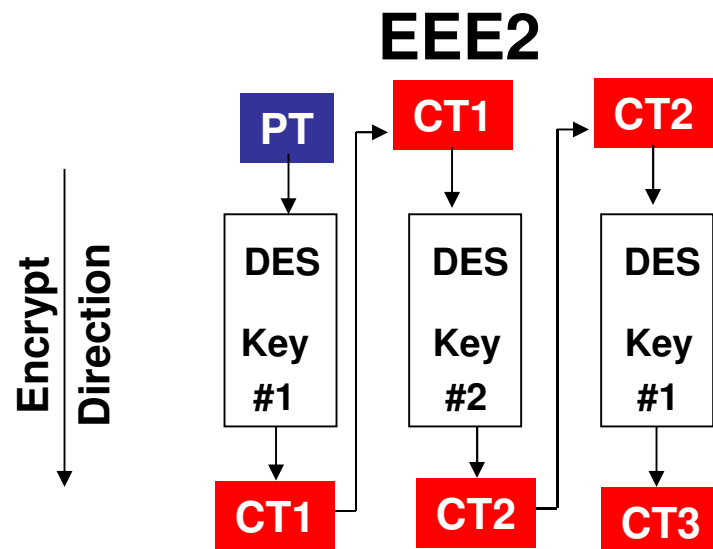
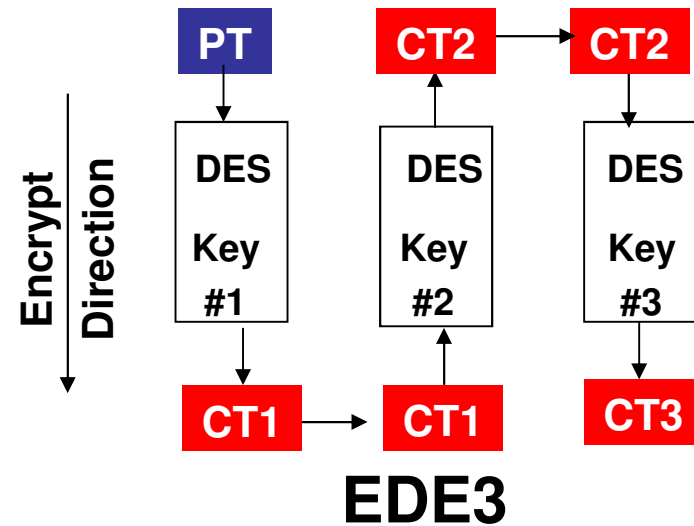
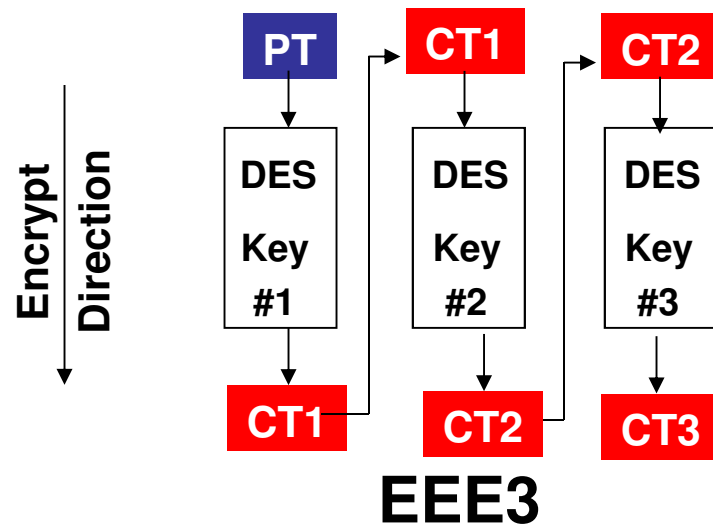
Counter (CTR) mode encryption

Triple DES

- ➔ As processing power increased, DES was 'broken'
- ➔ Encrypts message three times with multiple keys
 - DES-EEE3 uses three keys for encryption
 - DES-EDE3 uses 3 keys, encrypts, decrypts, and encrypts data
 - DES-EEE2, DES-EDE2 are the same as the previous mode, but the first and third operation uses same key
- ➔ Performance hit



Triple DES



Advanced Encryption Standard

➡ Rijndael Algorithm

➡ U.S. Official Standard for sensitive but unclassified data encryption

- Effective as of May 26, 2002

➡ Block Symmetric Encryption Algorithm

- Block sizes of 128, 192, 256

➡ Key sizes of 128, 192, 256

- Key size is variable

Advanced Encryption Standard

➡ In the 1990's a “DES Cracker” machine was built that could recover a DES key in a few days

- Today, there are systems that can crack DES in 3½ minutes

➡ If a machine was built to recover a DES key in one second, it would take that system 149 trillion years to crack a 128-bit AES Key

- According to NIST

Other Symmetric Algorithms

➡ International Data Encryption Algorithm (IDEA)

- Block Cipher
- Operates on 64-bit blocks of data
- Key length is 128-bits
- Uses 8 rounds of 16-bit sub-blocks

Other Symmetric Algorithms

➡ RC5

- Block Cipher that uses variable block lengths, key lengths, and rounds
- Block sizes are 32, 64, and 128
- Rounds up to 255
- Key sizes up to 2048 bits

➡ Twofish

- Block Cipher
- 128-bit blocks in 16 rounds
- Key sizes up to 256 bits

Symmetric Key Cryptography Review

➡ Number of keys and maintenance

- Requires secure mechanism for key exchange
- Number of keys grows as number of users increases
- N users would need:
 - 3 users require 3 keys
 - 10 users require 45 keys
 - 1000 users require 499,500 keys
- Number of necessary keys:
 - $\# = N(N - 1) / 2$

➡ Types

- Block
- Stream

Symmetric Key Cryptography Review

➡ Symmetric Key Cryptography

- Secret keys
- Same key to encrypt and decrypt a message
- Secrecy dependent on users keeping keys secret and protected
- No automated key distribution and management
 - Out-of-band key exchange is performed by sneaker net or courier
- Can provide confidentiality but NOT non-repudiation
- Very fast, and if key is kept secret and key is large, hard to break

Symmetric Key Cryptography Review

➡ Issues:

- A distinct key for each couple communicating
 - Many users requires many keys to manage
 - Not scalable
 - Costly to change secret keys regularly
- The more the key is used to encrypt large blocks of data, the more the key is exposed
 - Finding the secret key means access to the data encrypted in the past and the present
 - Consider the need to change keys often
- Cannot be used for digital signatures

Symmetric Key Cryptography Review

➡ Weaknesses

- Key distribution – It requires a secure mechanism to deliver keys properly
- Scalability – Each pair of users needs a unique pair of keys, so the number of keys grow exponentially
- Limited security – It can provide confidentiality, but limited authenticity and cannot provide non-repudiation

Symmetric Key Cryptography Review

➡ Symmetric Algorithms

- DES – Data Encryption Standard (56 bits)
- 3DES – 3 DES keys
- AES – 128, 192 and 256 bits
- IDEA – 128 bits
 - International Data Encryption Algorithm
- Blowfish – up to 448 bits
- Twofish – up to 256 bits
- RC4 (variable) – stream cipher
- RC5 – up to 2048 bits
- RC6 – up to 2048 bits
- CAST – 40, 64, 128, and 256 bits
- SAFER – block cipher developed by the co-creator of IDEA
- Serpent – Runner-up cipher in the AES competition

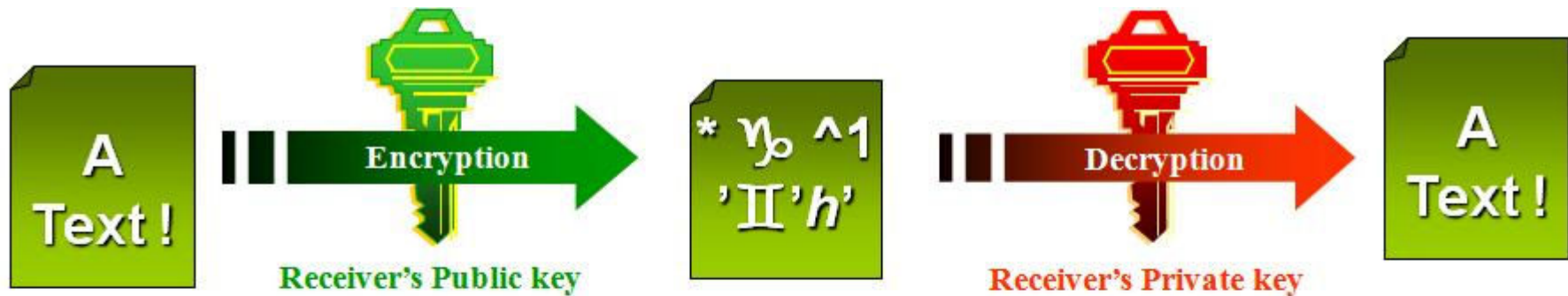
Asymmetric Cryptography

➡ Asymmetric Key Systems

- Public Key Cryptography: two keys - one public and one private key
 - Public key can be given to anyone
 - Private key should only be in possession of the owner
- The public and private keys are mathematically related, but cannot be derived from each other
- No prior relationship required for encrypted communication, as necessary with symmetric keys

Asymmetric Cryptography

One key encrypts the data, and the other decrypts it.



Confidentiality Mode

Authentication Mode



Different Public Key Ciphers

➡ Diffie-Hellman

➡ RSA

➡ ECC

➡ El Gamal

➡ DSA

Asymmetric Algorithms

➡ RSA

- Developed by Ron Rivest, Adi Shamir and Leonard Adleman
- Digital signatures, key distribution, encryption
- Difficulty of factoring large prime numbers
- Key sizes: 512, 1024, 2048, 4096, 8192

➡ El Gamal

- Digital signatures, encryption, and key exchange
- Based on calculating discrete logarithms in a finite field

➡ Elliptic Curve Cryptography (ECC)

- Digital signatures, key distribution, encryption
- More efficient than other algorithms
 - Used in devices with limited processing power
 - Does not require longer key to provide higher protection

Public Key Cryptography Advantages

➡ **Allows parties to communicate securely without previously sharing secret information**

- Solves the fundamental problem with symmetric cryptography

➡ **Scales very well**

- 1000 users = 2000 keys

➡ **Enables digital signatures**

Public Key Cryptography Disadvantages

➡ Very slow compared to symmetric cryptography

- 100 to 1000 times slower

➡ Size of encrypted data limited by performance considerations

- Not suitable for encrypting large amounts of data

➡ Asymmetric Algorithms

- Diffie-Hellman
- RSA
- Elliptic Curve Cryptography
- El Gamal
- DSA

Symmetric vs. Asymmetric Key Systems

Attributes	Symmetric	Asymmetric
Keys	One key is shared between two or more entities	One entity has a public key, and the other entity has a private key
Key Exchange	Out-of-band	Symmetric key is encrypted and sent with message; thus, the key is distributed by in-bound means
Speed	Algorithm is less complex and faster	Algorithm is more complex and slower
Number of Keys	Grows as users grow	Does not grow exponentially
Use	Bulk encryption, which means encrypting files and communication paths	Key encryption and distributing keys
Security Service Provided	Confidentiality	Confidentiality, authentication, and non-repudiation

Addressing Cryptographic Weaknesses

➡ Solution to symmetric and public key cryptography weaknesses:

- Combine both techniques using a hybrid approach

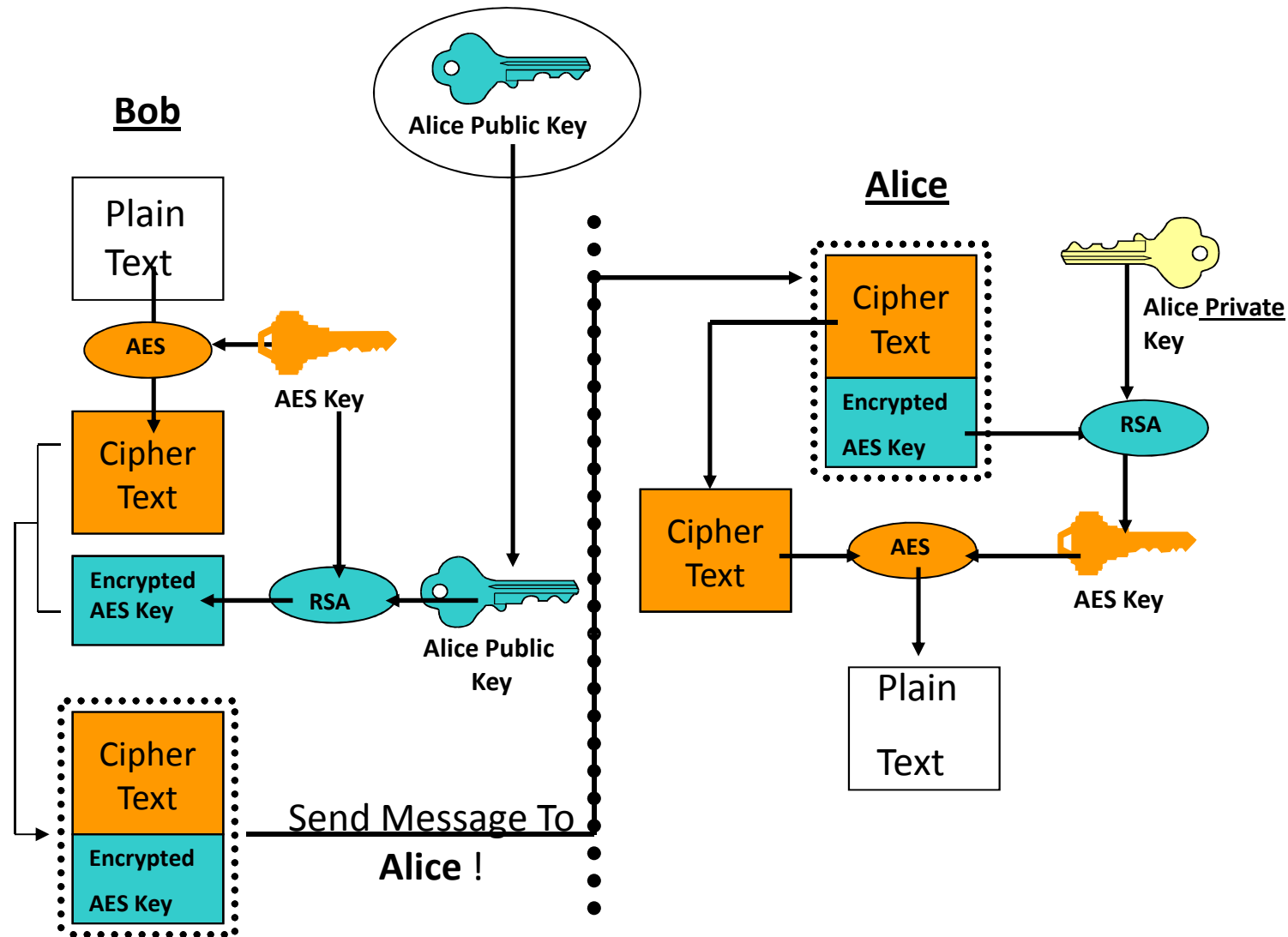
➡ Symmetric Algorithms

- Symmetric algorithm is used to encrypt the bulk of the data using a session key
- Session keys are randomly generated

➡ Asymmetric Algorithms

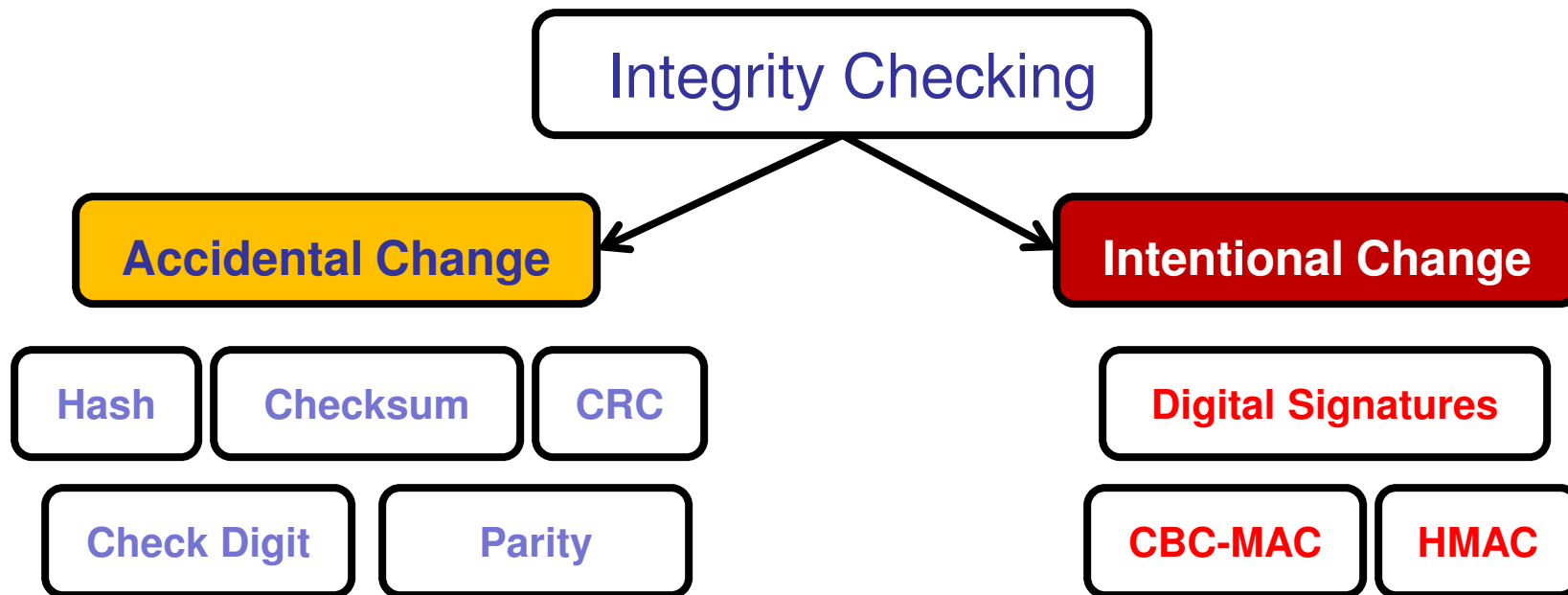
- Asymmetric algorithm provides the public and private keys to each party
- Recipient's public key is used to encrypt the session key to be transmitted with the encrypted message

Components of Hybrid Cryptography



Message Integrity Controls

➔ We can encrypt data so that it is private; how do we know it has not been altered accidentally or intentionally?



Data Integrity – Hashing Algorithms

➡ Use Hashing Algorithms:

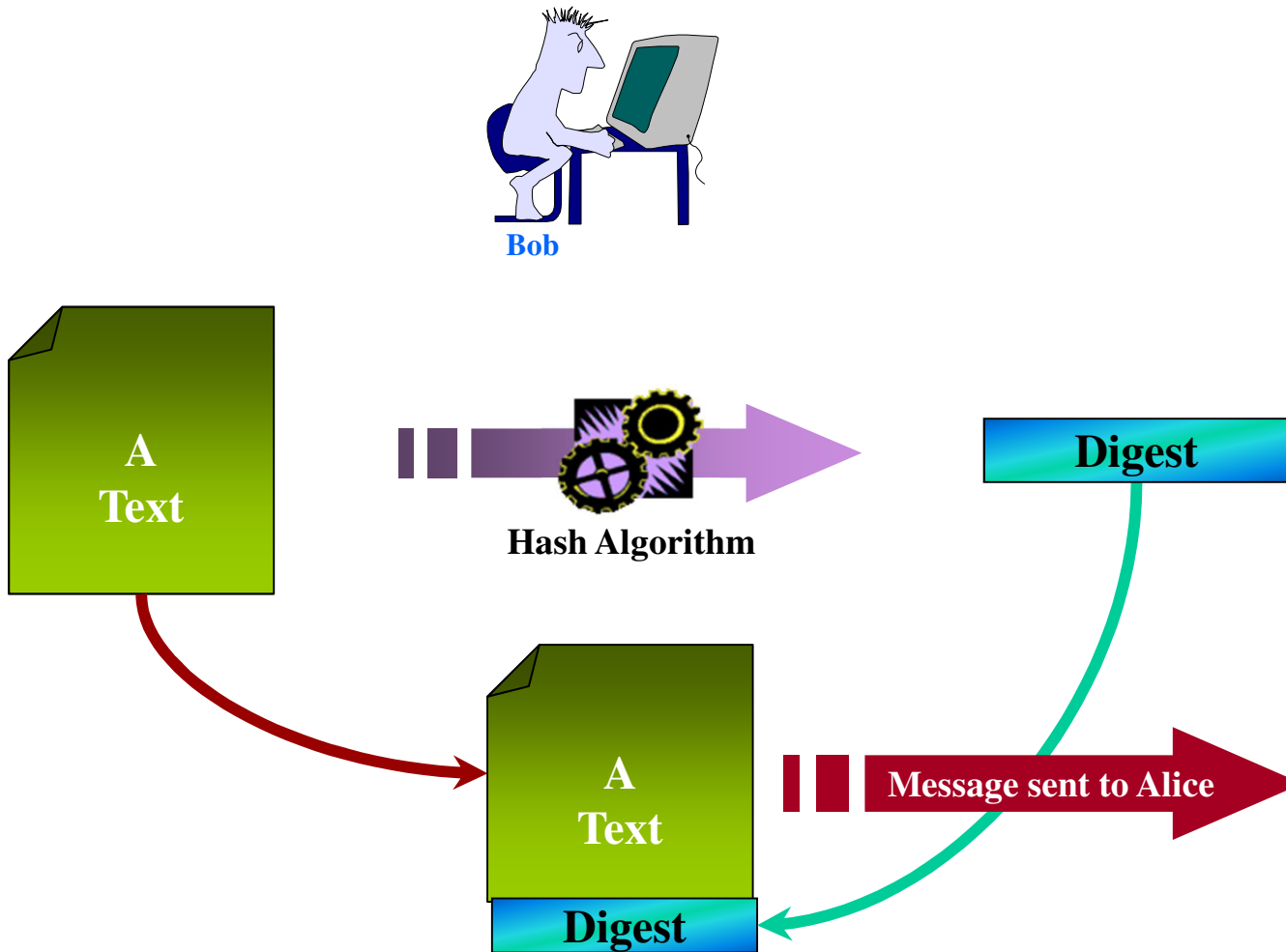
- A one-way function comparable to a CRC check
- Usually 128-bit or 160-bit “message digest”
- “One-way function” – message is not disclosed by its digest
- One bit modified on the message affects half of the bits of the digest
- Two different messages should not produce the same digest

Message Integrity

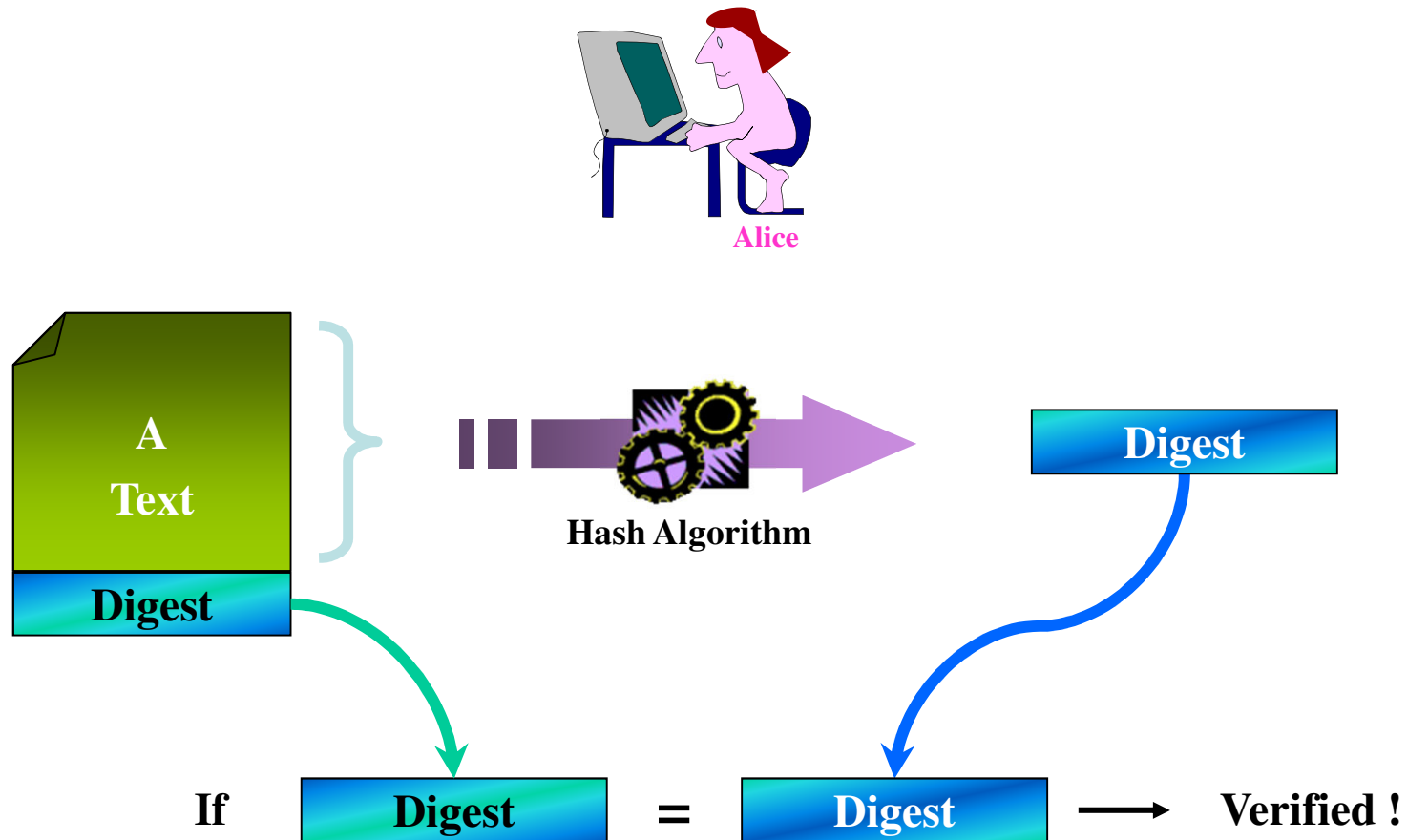
➡ One-way Hash

- Accepts variable-length string (message)
- Generates a fixed-length value (hash value)
- Hash value also called message digest
- A “fingerprint” of message
- Sender and receiver generate separate values
- Receiver compares to ensure message has not been altered

Data Integrity – Creating a Hash



Data Integrity – Verifying a Hash



One-way Hash

➡ Algorithms are publicly known

- The secrecy is in the “one-wayness”

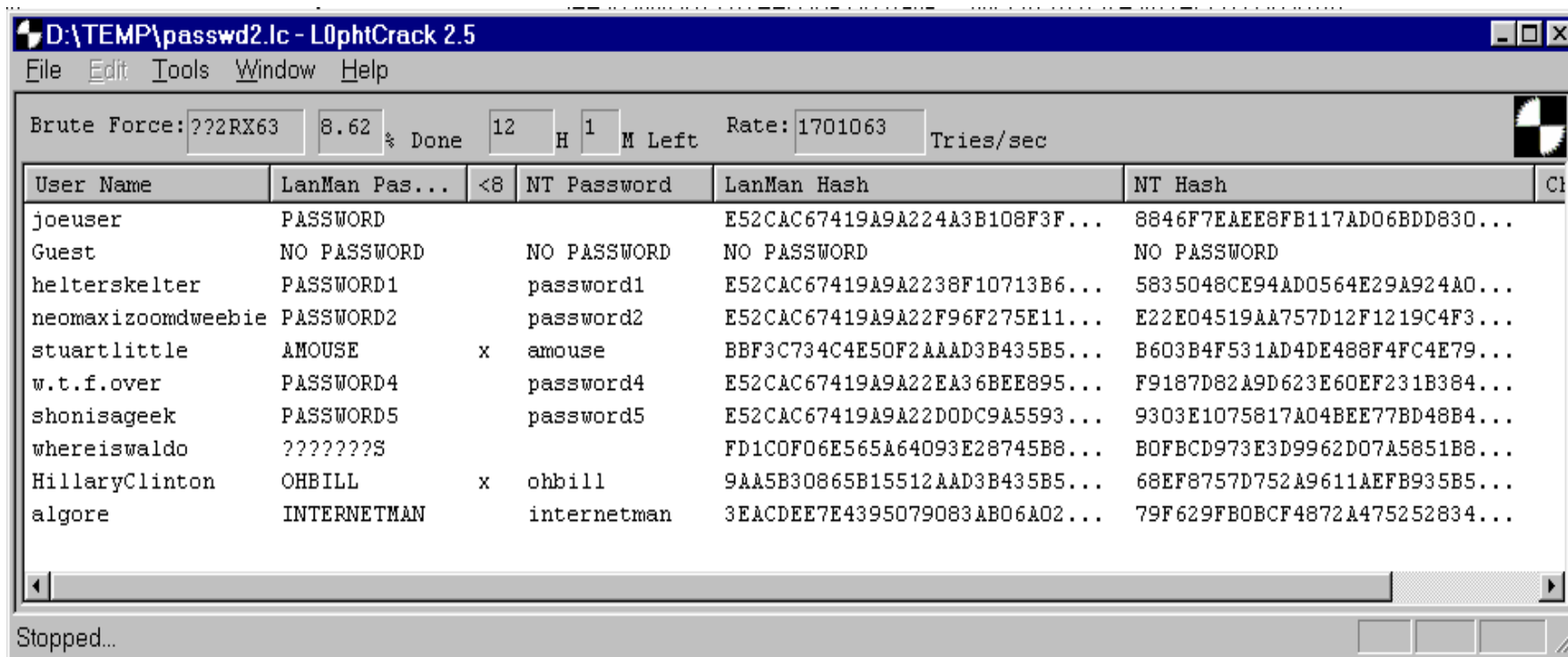
➡ Different than a one-way function in public key cryptography

- Security is from the difficulty of reversing the one-way function
 - Requires knowledge of key (trapdoor) to reverse process
- A one-way hash is never run in reverse

➡ One-way function requires keys; a one-way hash does not

- Anyone can perform a one-way on a hash; even an intruder

Hashing Example



Data Integrity - Hashing Algorithms

➡ Example of Hashing Algorithms:

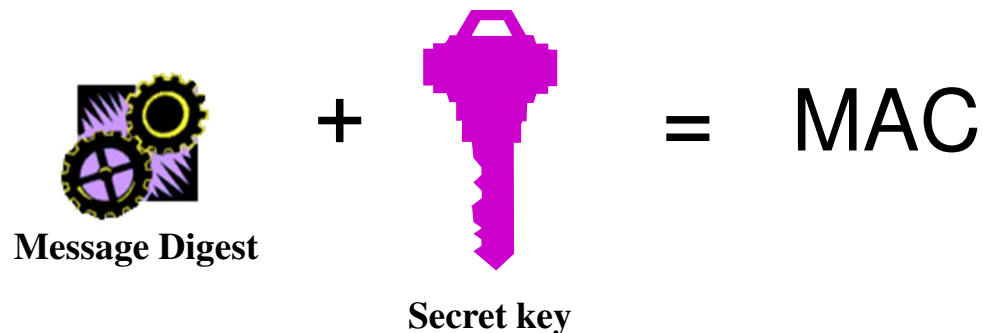
- MD2 (128-bit digest)
- MD4 (128-bit digest)
- MD5 (128-bit digest – Ron Rivest, RFC 1321)
- SHA-1 (160-bit digest – NIST); SHA-256; SHA-384; SHA-512
- HAVAL (Variation of MD5; variable-length message digests)
- Other Hashing Algorithms: RIPEMD, Tiger, WHIRLPOOL

Authentication

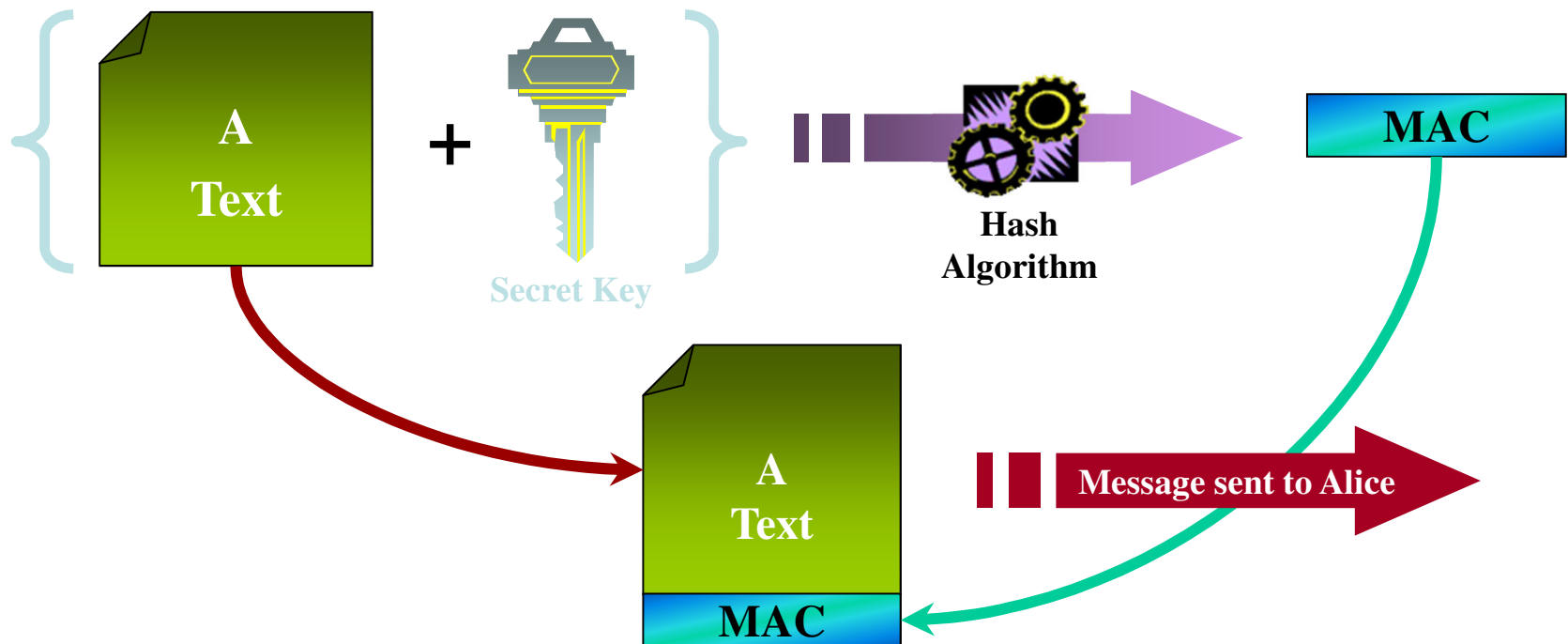
➡ How does Alice know the message is coming from Bob?

- By keying the hash
- Bob combines the hash function with a shared secret key

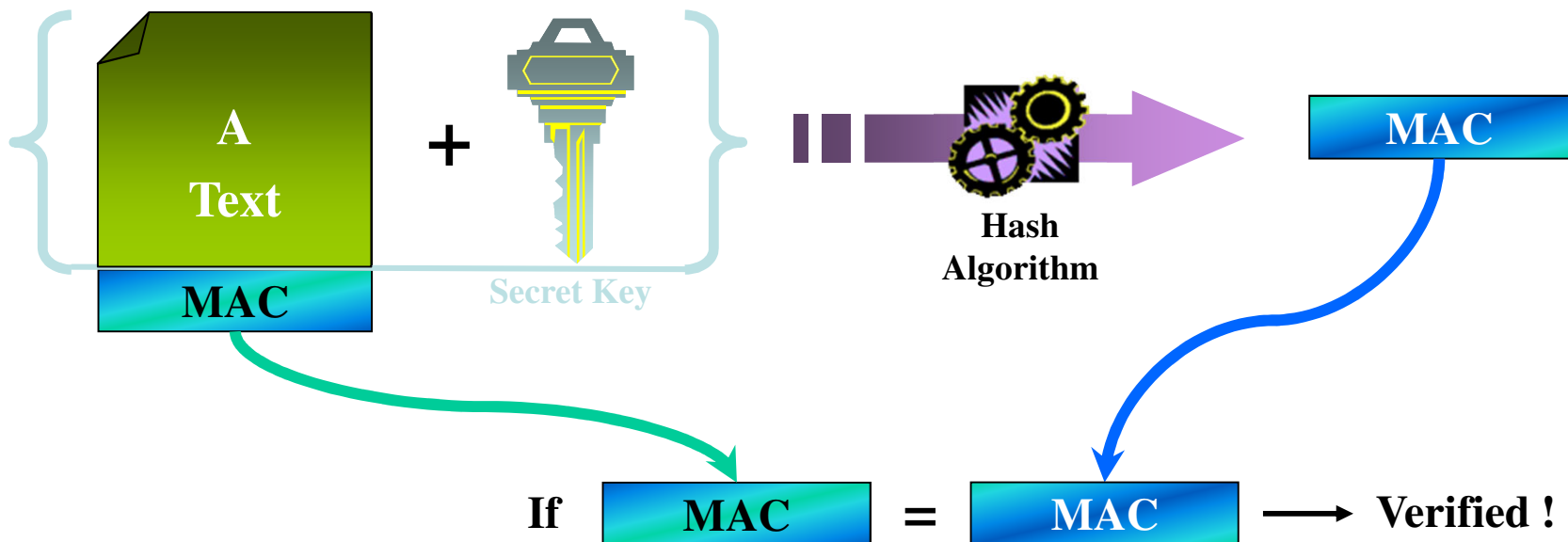
➡ Hash with a shared secret is a Message Authentication Code



Secret Key Authentication – Creating MAC

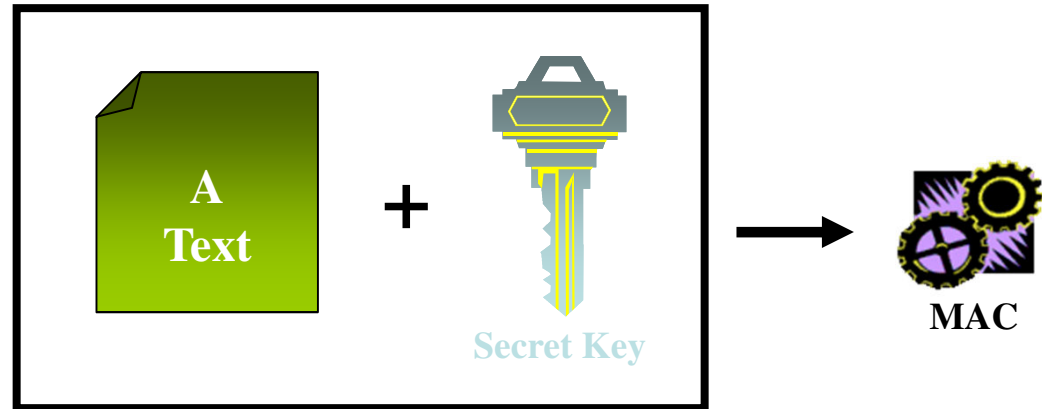


Secret Key Authentication – Verifying MAC

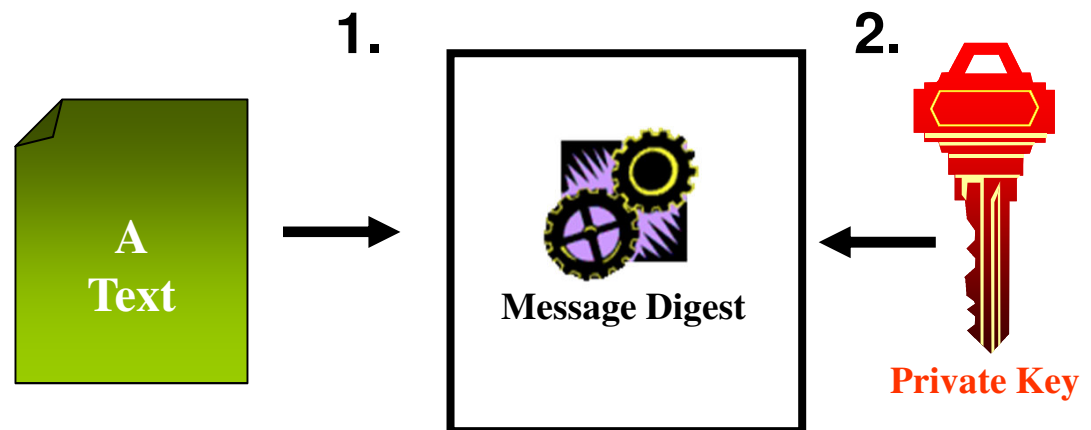


Authentication Methods Comparison

MAC:



**Digital
Signature:**



What is Quantum Cryptography

- ➡ **Set of protocols, systems, and procedures that can be use to create and distribute secret keys**
- ➡ **Secret keys can be used in traditional cryptosystems**
- ➡ **It is not used encryption, transferring encrypted data, or to store encrypted data**
- ➡ **Allow secure key exchange with complete security**
- ➡ **Solve the key exchange problem**
- ➡ **It can detect an attacker trying to gain knowledge of the keys being exchanged**
- ➡ **It is also called Quantum Key Cryptography**

Basis of Quantum Cryptography

- ➡ **Classical public-key cryptography relies on complex mathematical problems such as integer factorization**
- ➡ **Quantum cryptography relies on the laws of quantum mechanics**
- ➡ **Quantum cryptographic devices typically employ individual photons of light and relies either on the uncertainty principle and photon polarization**
- ➡ **Uncertainty is an integral part of quantum mechanics**
- ➡ **It is possible to encode information into quantum properties of a photon in such a way that any effort to monitor them disturbs them in some detectable way**

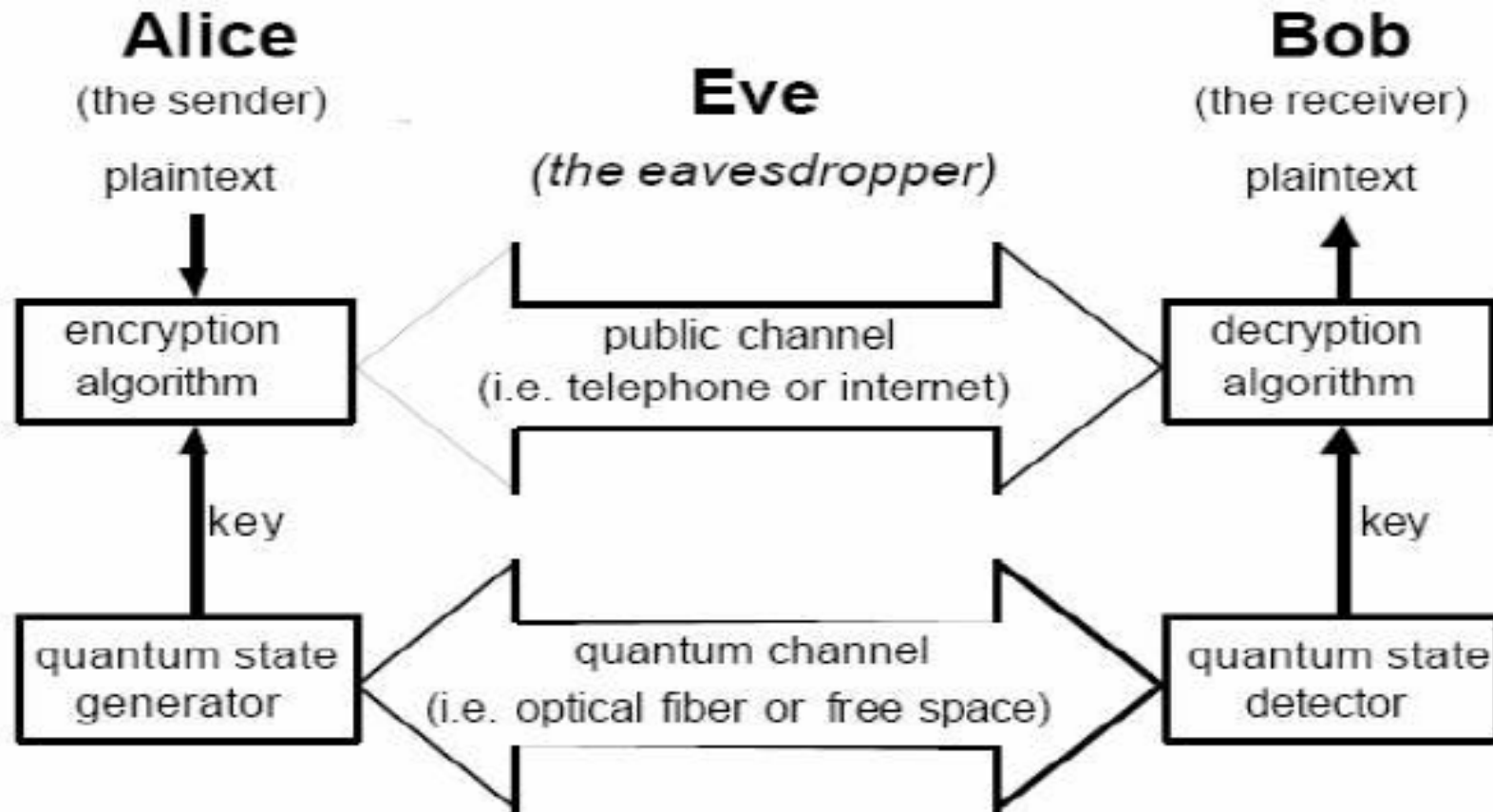
Steps of Quantum Cryptography

- ➡ **Two parties need highly secure key exchange**
- ➡ **They choose standard protocols and technologies to encrypt and send the data in encrypted form**
- ➡ **They use the Quantum Cryptography Channel to generate and exchange the secret keys needed**
- ➡ **They use those keys with classical cryptosystems**
- ➡ **Once they have the keys they can communicate securely using classical cryptosystems**
- ➡ **Two channels are used:**
 - One for the transmission of the quantum material
 - One for the encrypted user traffic

State of Quantum Cryptography

- ➡ **Experimental implementations since 1990.**
- ➡ **In 2004 QC was performed over distances of 30 to 40 kilometers using optical fiber**
- ➡ **Repeaters are needed for longer distance**
 - Practical Repeaters are a long way in the future
 - Repeater is seen as an eavesdropper on link
- ➡ **NIST did testing with Infrared Laser**
 - Great speed at short distance
 - Low speed over long distance
- ➡ **The speed is the price to pay for complete secrecy and detection of eavesdropping**

Quantum Key Distribution



Graphic from: http://stanford.edu/~adityaj/quantum_cryptography.pdf

Agenda

- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ **Key management processes**
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Protocols Exchanging Keys

➡ Agreeing upon a secret key

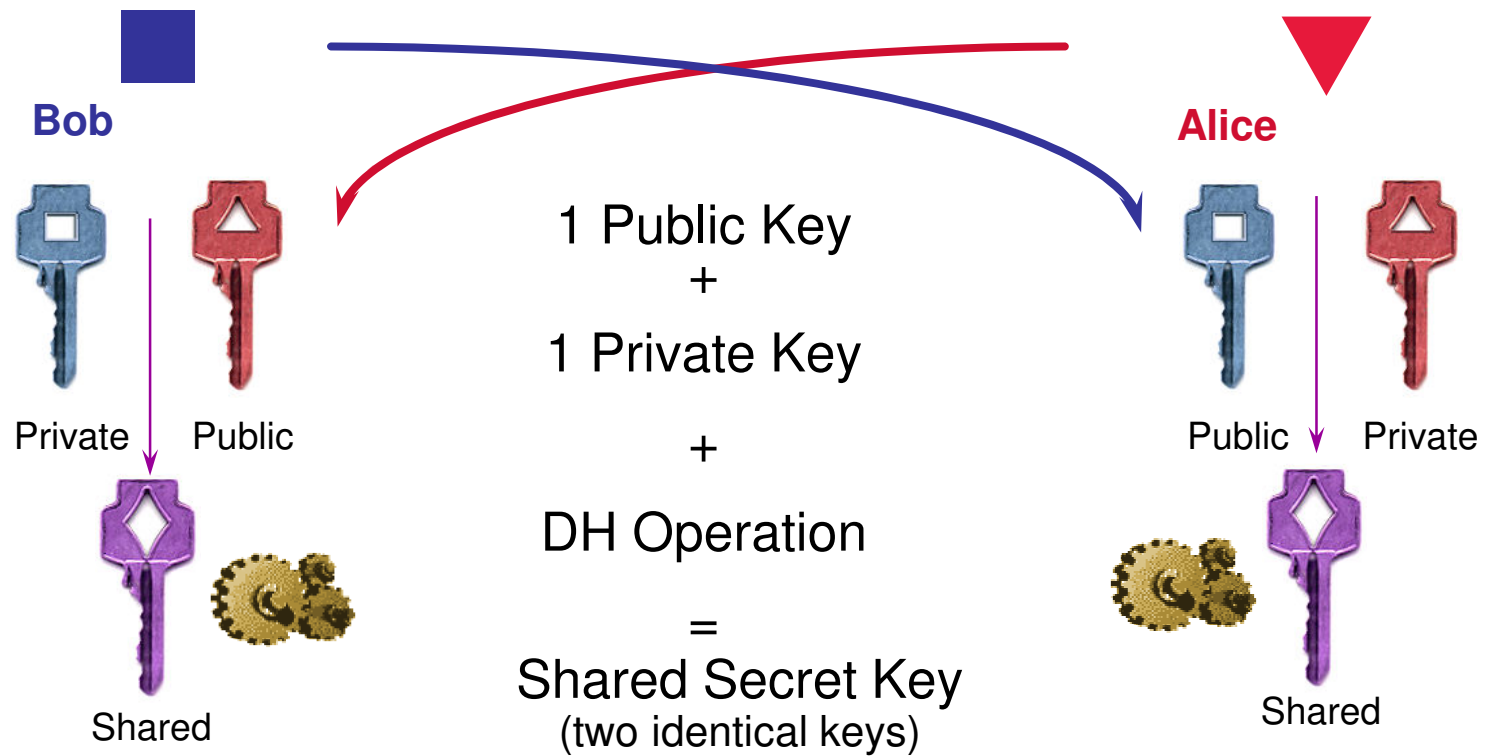
- A key exchange protocol uses a series of steps to agree upon a shared secret key
- This does not require a previous relationship between the two parties
- Key exchange can be done in a secure manner
- Diffie-Hellman is one example

Asymmetric Algorithm

➡ Diffie-Hellman

- First asymmetric algorithm
- Allows users to agree on a symmetric key over a non-secure medium
- Does not provide data encryption or digital signatures
- Security based on calculating discrete logarithms in a finite field
- Vulnerable to man-in-the-middle attacks
 - lack of authentication
 - Can be countermeasured with digital signatures

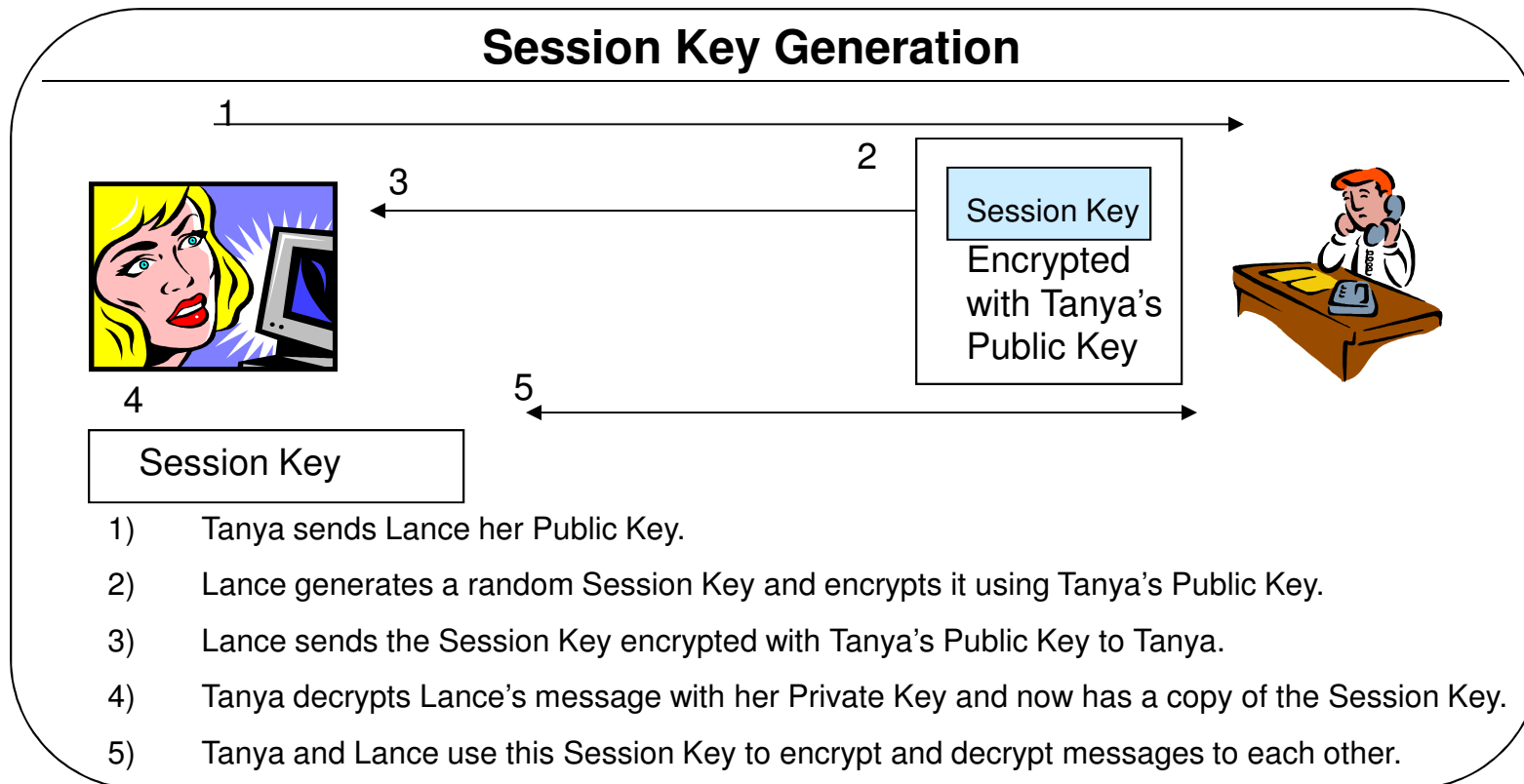
Diffie-Hellman Key Agreement



- 1) Generate a value from a random string
- 2) Exchange value with other party
- 3) Complete calculation using the local value and the received value
- 4) A unique, mathematically identical key is created

Session Keys

- ➡ Secret symmetric key used to encrypt messages
- ➡ Only good for one session and then destroyed
- ➡ If same key is used again, it could be compromised



Key Management

➔ Functions of Key Management:

- Need an automated way of creating and distributing keys
- Key recovery
- Key storage and destruction
- Key strength
 - Complexity of the key
 - Truly random
 - Suitable length
 - Secrecy of the key

Key Recovery

- ➔ **Plan for users leaving company, losing key, or somehow not available to decrypt crucial information**

- ➔ **Different methods for key recovery**
 - A copy is encrypted with the public key of a private/public pair just for key recovery

 - Copy kept some where secure and decrypted when needed

 - Private key can be broken down into several pieces and given to different people in the company
 - Dual control
 - Split Knowledge

Key Management Issues

- ➡ **Dual control should be used to protect centrally stored keys**
- ➡ **Multiple copies of keys increases chance of disclosure and compromise**
- ➡ **Use one key per application; using the same key for multiple uses increases potential for compromise**
- ➡ **Multiparty control for emergency key recovery**
 - Improves accountability
 - Reduces potential for abuse

What makes key management secure?

- ➡ Keys should be stored and transmitted by secure means
- ➡ Keys should be extremely random and use the full spectrum of the keyspace
- ➡ The key's lifetime should correspond with the sensitivity of the data it is protecting (less secure data may allow for a longer key lifetime, whereas more sensitive data might require a shorter key lifetime)
- ➡ Keys should be properly destroyed when their lifetimes end
- ➡ Keys should not be presented in cleartext anywhere

Protecting Keys

➡ Separate Keys

- Most implementations of public key cryptography have separate keys for digital signatures and encryption
- This separation can add layers of necessary protection
- Each key type can have its own expiration date, backup procedures, storage (hard drive, database, or smart card), and strength (1024-bit encryption and 2048-bit for signature)
- One person may have different digital signatures with different strengths; for example:
 - 1024 bit key used for e-mail signatures
 - 2048 bit key for Purchase Order signing
- Keys must be stored separately

IPsec Key Management

➡ Manual

- Administrator configures each system with keying material and security association information

➡ Internet Key Exchange (IKE) is the de facto standard

- Negotiation of services is automatic
- Hybrid of Internet Security Association and Key Management Protocol (ISAKMP) and Oakley Key Exchange
 - Phase 1: IKE peers establish a secure, authenticated channel so that the IPsec negotiation can take place
 - Phase 2: SA's are negotiated for keying material and parameter negotiation

IKE Services within IPsec

➡ OAKLEY Protocol

- Oakley Key Determination Protocol
- Negotiates key information using the Diffie-Hellman Algorithm

➡ Internet Security Association and Key Management Protocol (ISAKMP)

- Creates framework for key negotiation
 - Agrees on protocols
- Does not do the actual negotiation
 - Oakley does

Trusted Platform Module

- ➡ The Trusted Platform Module offers facilities for the secure generation of cryptographic keys, and limitation of their use, in addition to a hardware pseudo-random number generator. It also includes capabilities such as remote attestation and sealed storage.
- ➡ Software can use a Trusted Platform Module to authenticate hardware devices. Since each TPM chip has a unique and secret RSA key burned in as it is produced, it is capable of performing platform authentication. For example, it can be used to verify that a system seeking access is the expected system.

Agenda

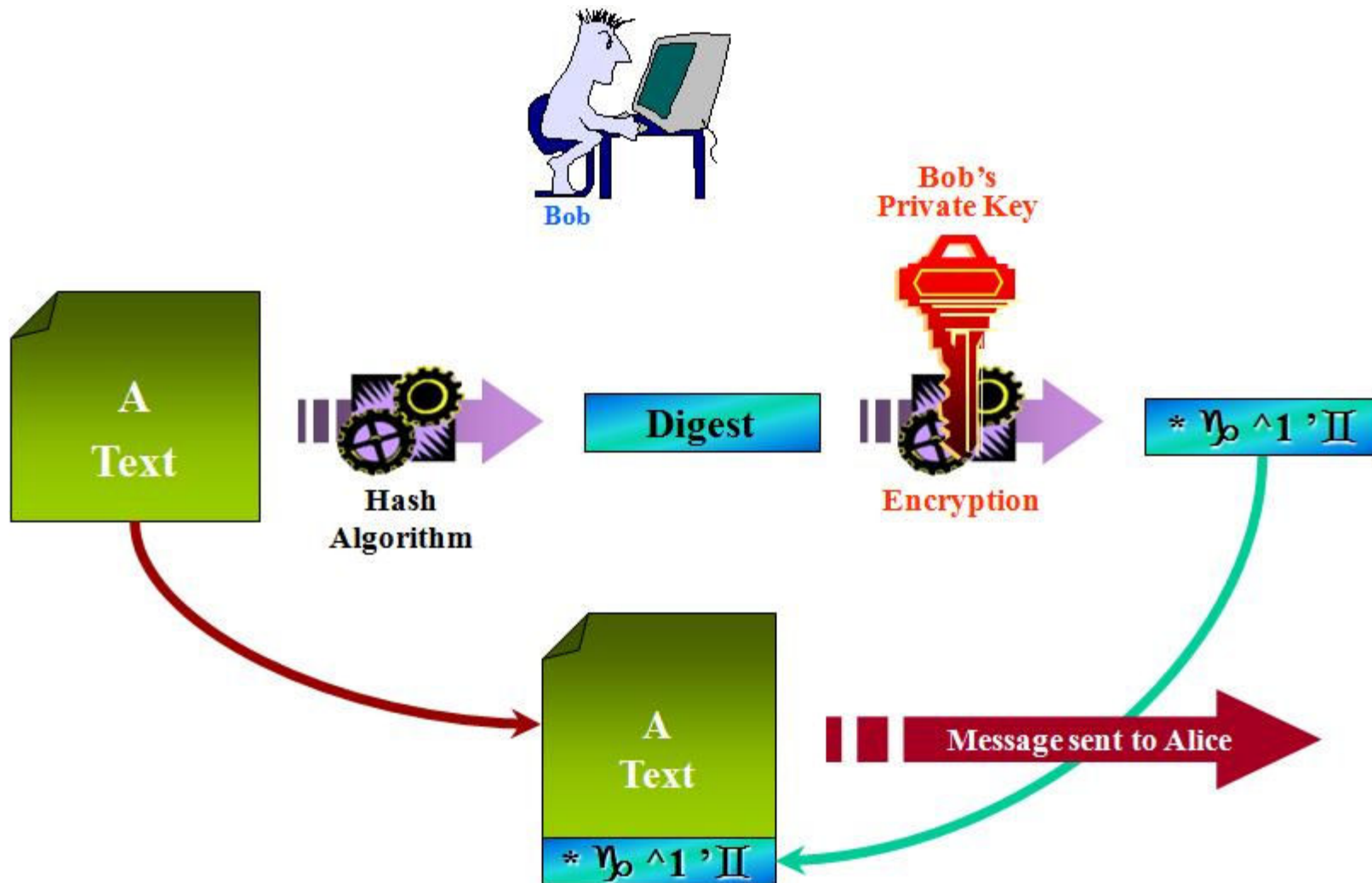
- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ **Digital signatures and non-repudiation**
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Electronic Signing

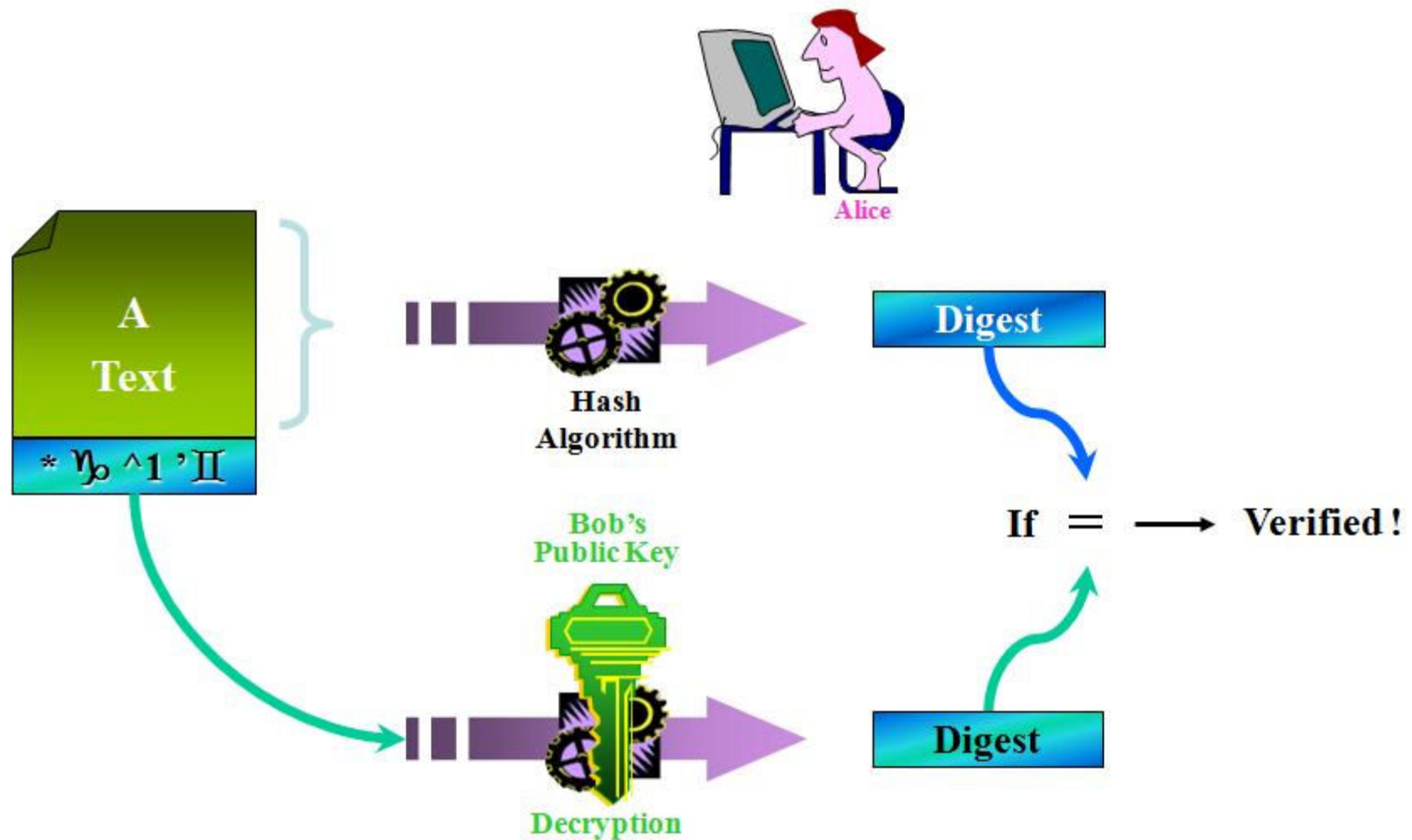
➔ Digital Signatures

- Authentication tool used to verify sender of message
- Message digest is created – input into a digital signature algorithm
 - Algorithm uses private key to encrypt digest
- Encrypting a hashing value with a private key performs “signing”
- Public key verifies signature
 - If can decrypt digest, sender’s identity is verified
 - Receiver computes digest and compares with sent digest

Applying a Digital Signature



Verifying a Digital Signature



Digital Signature Services

- ➡ Provides integrity, authentication, and non-repudiation
- ➡ A message can be encrypted, which provides confidentiality
- ➡ A message can be hashed, which provides integrity
- ➡ A message can be digitally signed, which provides authentication, integrity, and non-repudiation
- ➡ A message can be encrypted and digitally signed, which provides confidentiality, authentication, non-repudiation, and integrity

DSS

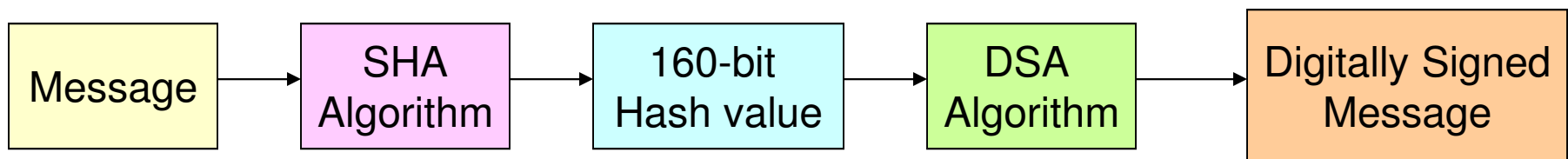
➔ Digital Signature Standard (DSS)

- Developed by NIST for digital signatures to be used within government facilities
 - SHA used for message digest – integrity
 - Digital Signature Algorithm (DSA) to sign message digest – authentication and non-repudiation
 - Private key is used for signing, and public key is used for signature verification
 - DSS can now use DSA, RSA, and ECDSA for digital signatures
 - ECDSA: elliptic curve digital signature algorithm, used with SHA-1

Hashing Algorithm for Digital Signatures

➔ SHA

- Some government documentation requires more secure digital signatures
- NIST and NSA developed SHA to be used in their Digital Signature Standard (DSS)
- SHA computes a 160-bit hash value, which is then input into the Digital Signature Algorithm (DSA)

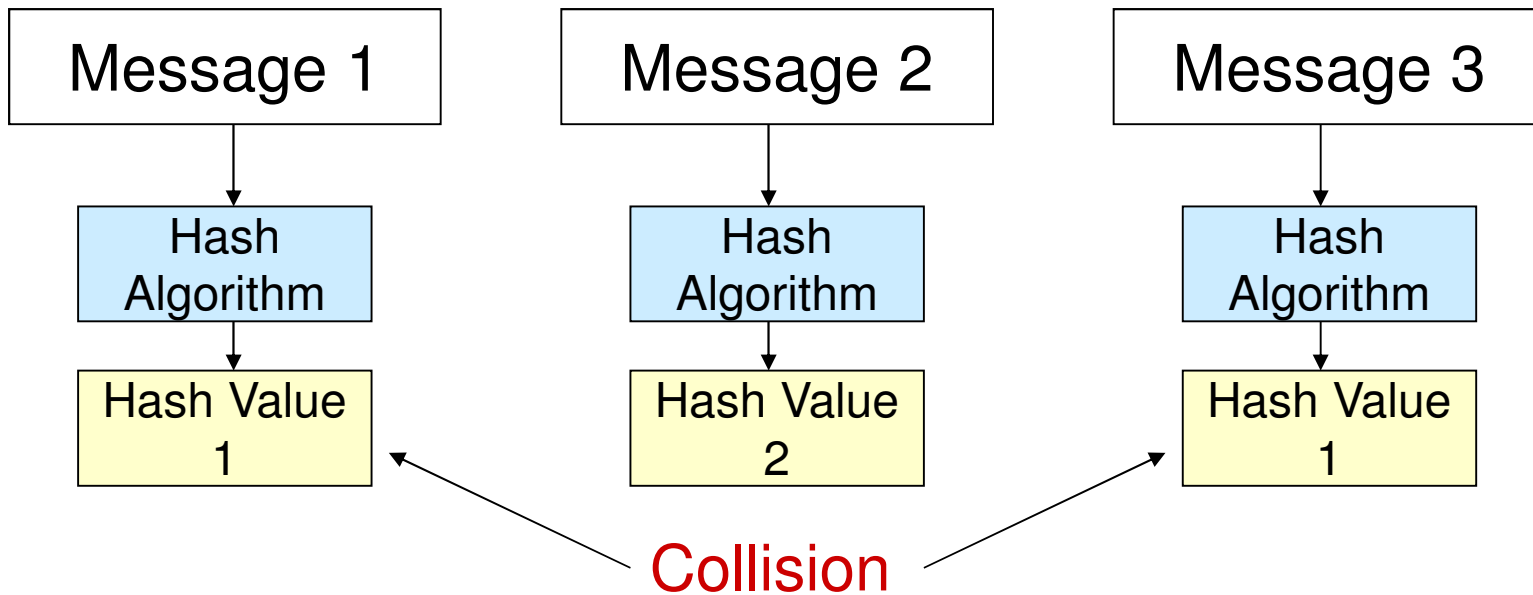


Security in Hashing

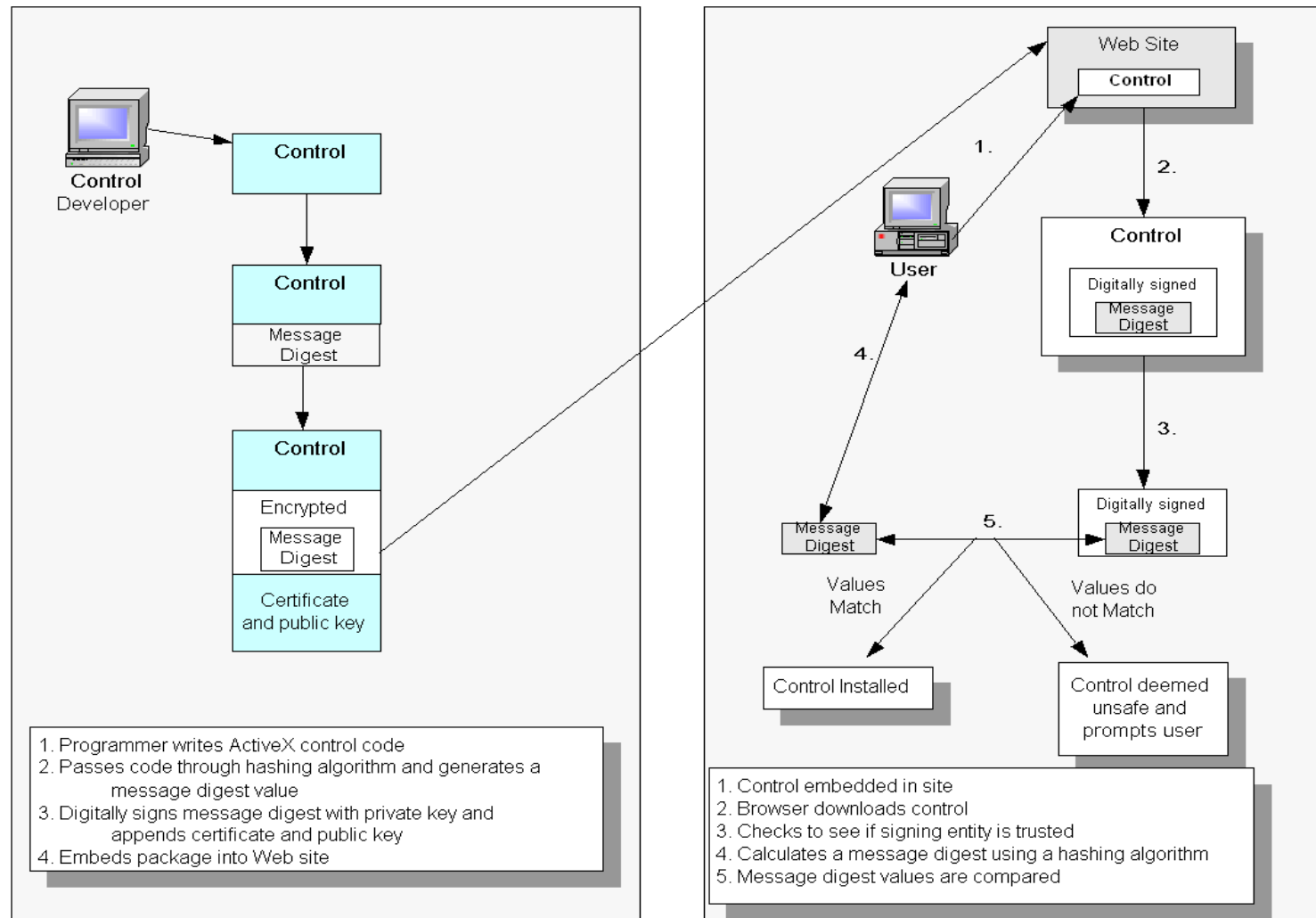
➡ What makes a strong Hashing Algorithm?

- The hash should be computed on the entire message
- The hash should be a one-way function so that messages are not disclosed by their signatures
- It should be impossible, given a message and its hash value, to compute another message with the same hash value
- It should be resistant to birthday attacks:
 - An attacker should not be able to find two messages with the same hash value

Security in Hashing



Digital Signing of ActiveX Controls



Agenda

- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ **Methods of cryptanalytic attacks**
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Attacking Encryption Systems

➡ Brute Force Attack

- If the cryptographic algorithm is strong, the only way of defeating it is through brute force
- The attacker attempts all possible combinations of a given key space to derive the key
- Most systems are created so that even identifying half of the possibilities is impossible with today's processing power

Attacking Encryption Systems

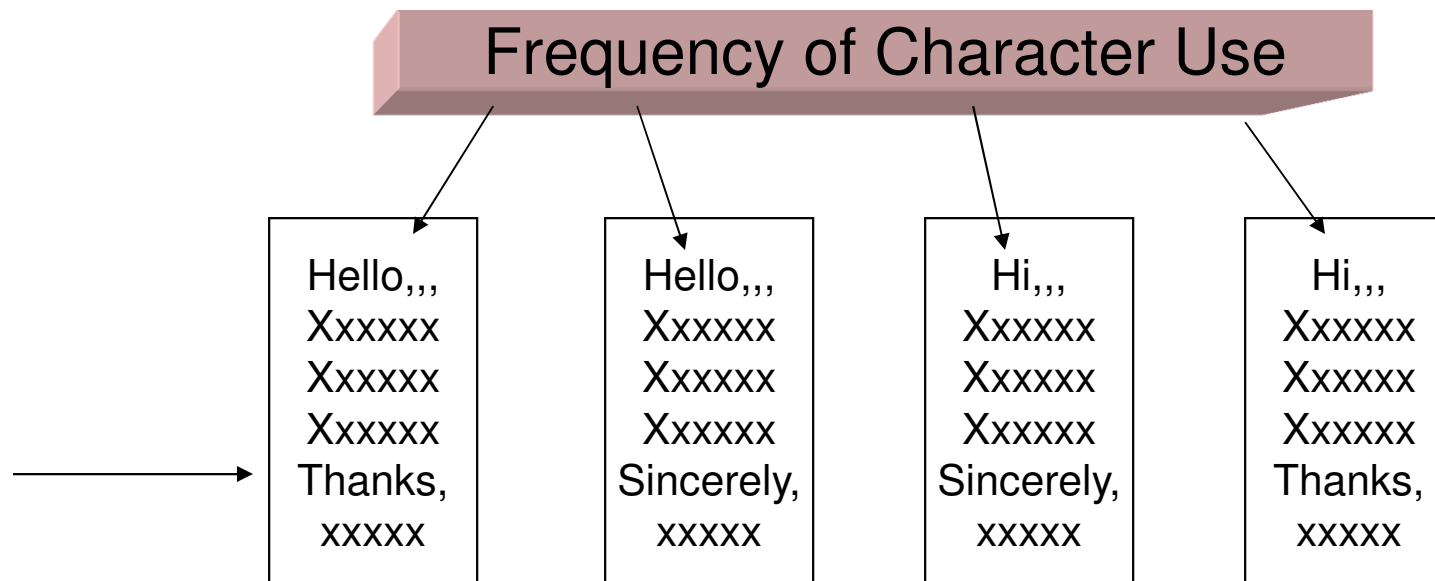
➡ Frequency Analysis Attack

- Calculating the frequency characteristics in a particular language helps to break substitution and transposition cryptosystems
- Common pattern of messages, sentences, words, and characters helps attackers calculate plaintext from ciphertext

Attacking Encryption Systems

➔ Frequency Analysis Attack

- Patterns are revealed
- More complex and strong cryptosystems should not reveal any patterns



Trapdoor of RSA

- ➡ **Mathematical function that is easier to compute in one direction than in the opposite direction**
 - Analogy is dropping a glass on the floor
- ➡ **Easy direction of computation is multiplying two large prime numbers**
 - Hard direction is factoring a large number into two prime numbers
 - RSA is based on the difficulty of finding prime factors of large (100 – 200 decimal digits) numbers
- ➡ **Trapdoor one-way functions**
 - Trapdoor is a secret mechanism that enables the decrypter to reverse the function in a one-way function
 - A trapdoor one-way function is a function for which the inverse direction is easy when a piece of information is provided, which is the trapdoor
 - $Y=f(x)$ – easier to get y if you know x
 - Telephone book – easier to find number by looking up name than the other way around

Attack on Hashing Functions

➡ Birthday Attack

- How many people must be in the same room for the chance to be over 50% that someone has the same birthday as you?
 - 253 people
- How many people must be in the same room for the chance to be over 50% that two people have the same birthday?
 - 23 people

Attack on Hashing Functions

➡ Birthday Attack

- It would be easier for an attacker to find two messages with the same digest value than match a specific value
- Hashing value = n
Brute force to find one specific hash value = 2^n
Brute force to find any two matching hashes = $2(n/2)$
- A hashing algorithm with a hash output of 64 bits is more vulnerable to the birthday attack than a hash with a 128-bit value

Attacks on Cryptosystems

➡ **Ciphertext-Only Attack**

- Captured ciphertext only
- Most common attack

➡ **Known-Plaintext Attack**

- Captured ciphertext and plaintext

➡ **Chosen-Plaintext Attack**

- Captured ciphertext and plaintext and can choose what plaintext gets encrypted
- Attacker sends a message they think the victim will encrypt and send out to others

➡ **Chosen-Ciphertext Attack**

- Attacker can choose the ciphertext to be decrypted and has access to the resulting decrypting plaintext

More Attacks

➡ Replay Attack

- Attacker obtains a set of credentials and sends them to an authentication service
 - Capture username and password, token, ticket
- Timestamps and sequence numbers are used to protect against this attack

➡ Man-in-the-middle Attack

- Attacker injects itself between two users and reads messages going back and forth or manipulates messages
- Sequence numbers and digital signatures are used as countermeasures to this type of attack

➡ Meet-in-the-Middle Attack

- An attack designed to compromise algorithms that use multiple keys, such as 3DES

Agenda

- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ **Using cryptography to maintain network security**
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Using Cryptography to Maintain Network Security

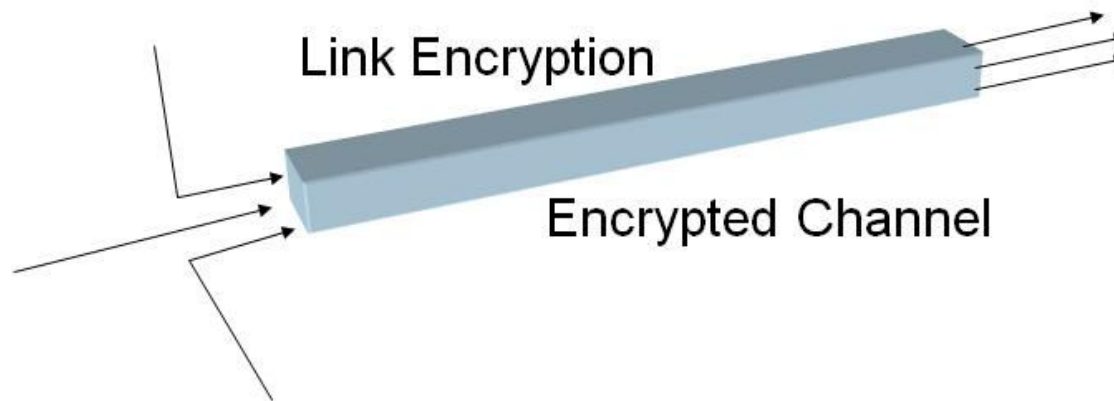
➔ Implementations of Cryptography

- Link encryption
- End-to-end encryption
- E-mail encryption
- Protocol encryption

Encryption at Different Layers

➡ Link Encryption

- Payload, headers, and trailers are encrypted – all data along a communication path
 - Telephone circuit, T1, satellite link
- Usually provided by service providers
- Each hop has to decrypt headers – if a node is compromised, all traffic going through that node can be compromised



Link Encryption

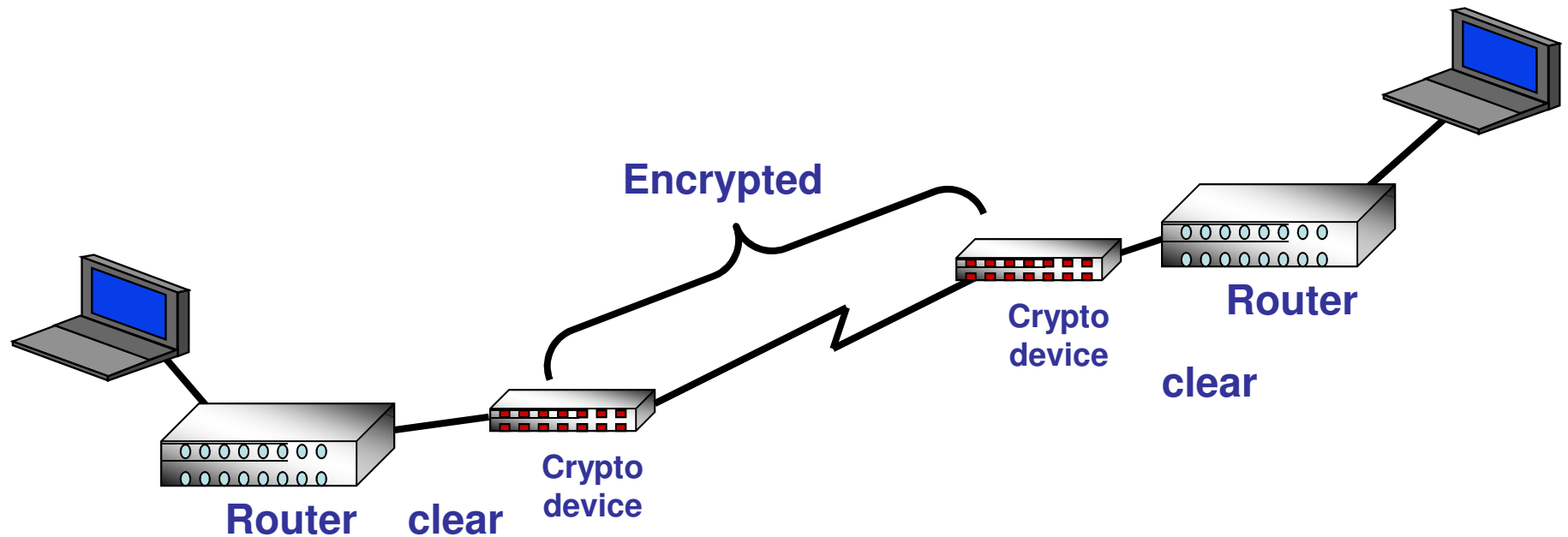
➔ Advantages of link encryption include the following

- All data is encrypted: including headers, addresses, and routing information
- Users do not need to do anything to initiate it; it works at a lower layer in the OSI Model

➔ Disadvantages of link encryption include the following

- Key distribution and management is more complex because each hop computer must receive a key, and when the keys change, each must be updated
- Messages are decrypted at each hop; thus, there are points of vulnerability

Link-level Encryption



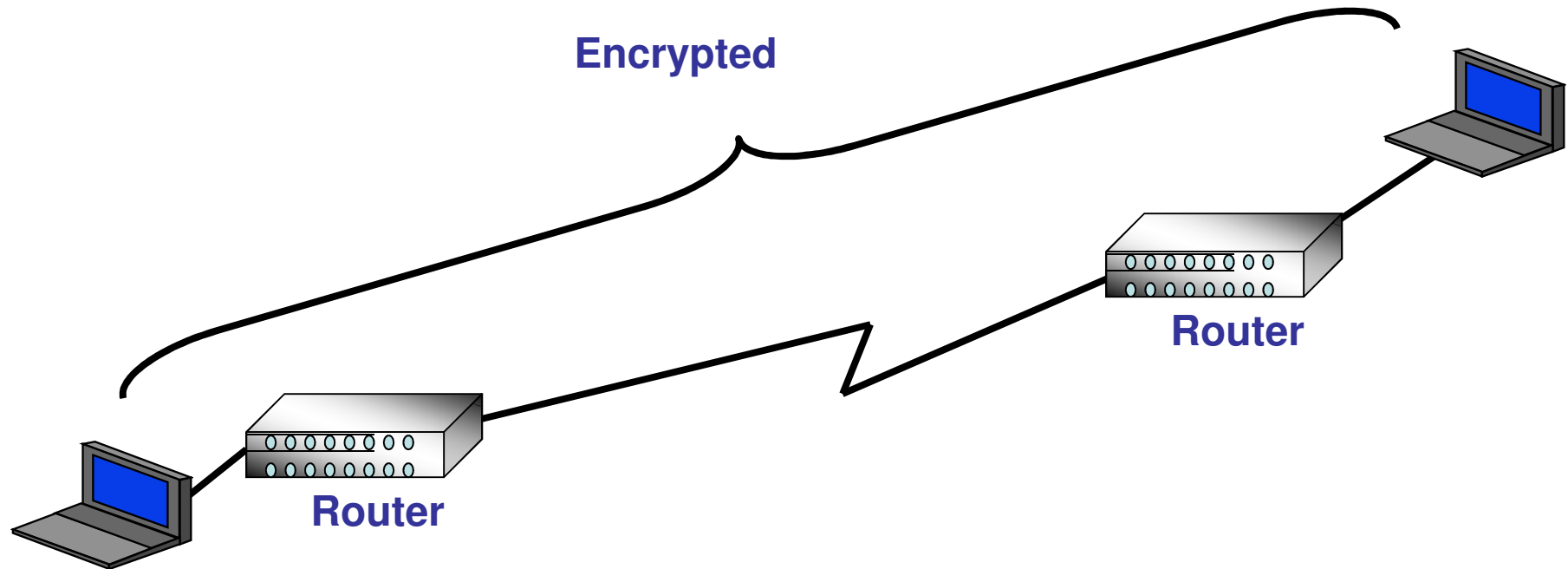
The message is protected as it travels across the wire, but is in the clear in each intermediate node

Encryption at Different Layers

➡ End-to-end Encryption

- Only payload is encrypted
- Headers and trailers are not encrypted
- Hops do not need to decrypt headers

End to End Encryption



The message is protected across the wire, and in each intermediate node

SSL 101

➔ Steps of setting up an SSL Connection:

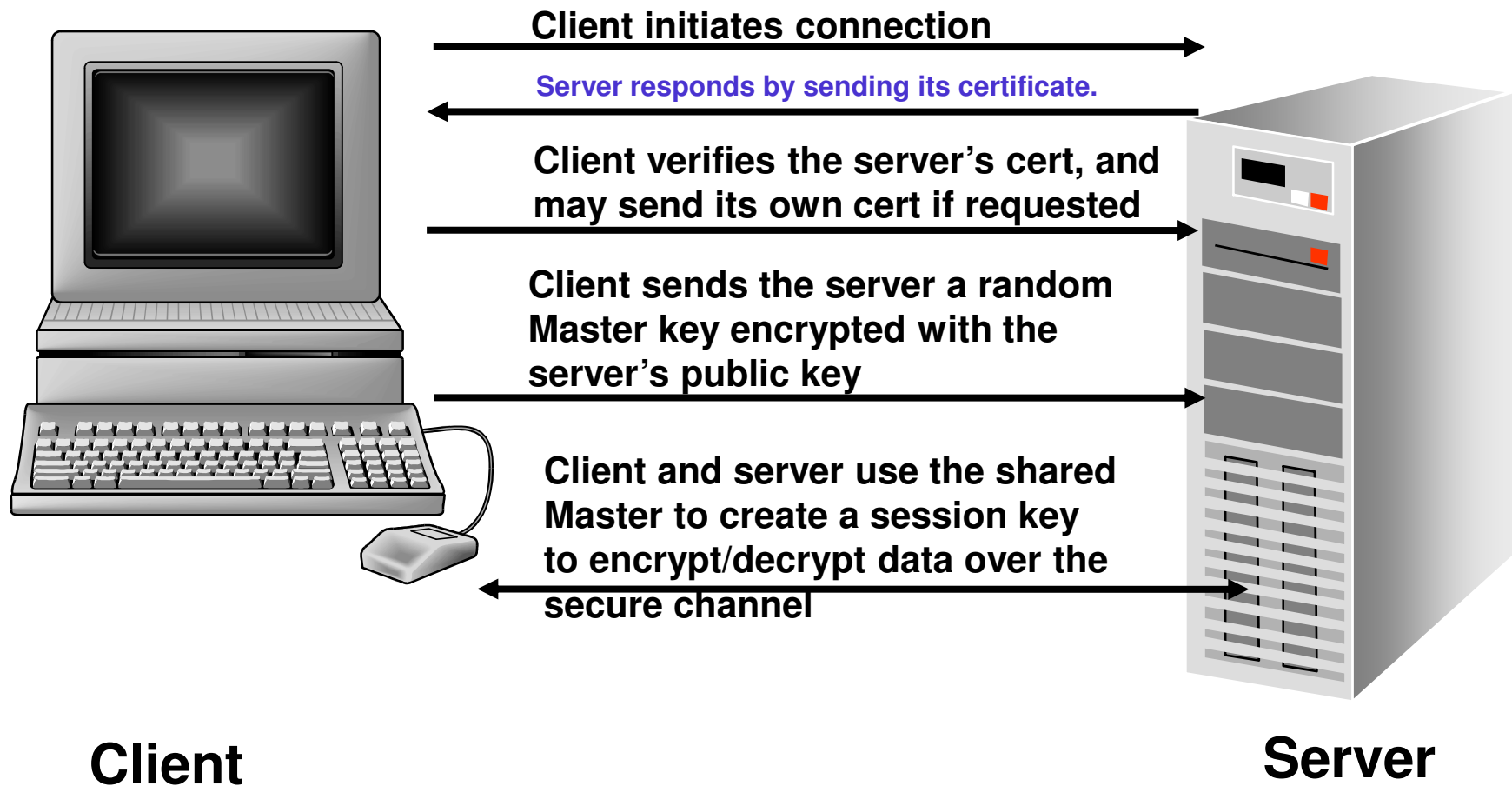
- Client initiates connection to server
- Server sends client the server's certificate
- Client checks to see if signing CA is in trusted list in browser
- Client computes hash of certificate and compares message digest of certificate by decrypting using the CA's public key (CA signed the certificate)
- Client checks validity dates in certificate

SSL 101

➔ Steps of setting up an SSL Connection:

- Client will check URL in certificate compared to URL it is communicating with
- Client extracts server's public key from certificate
- Client creates a session key (symmetric)
- Client encrypts session key with server's public key and sends it over
- Server decrypts with private key

SSL and TLS



Other Network Security Technologies

➡ Domain Name Service Security (DNSSEC)

- DNS server distributes public keys
- Secure Distributed Name Services

➡ Generic Security Services API (GSSAPI)

- Key exchange, generic authentication, provides encryption interface for different authentication methods and systems

➡ S-RPC

- Secure Remote Procedure Call for computer to computer communication
- Security protocol that uses DES to encrypt messages
- Uses Diffie-Hellman to create key pair

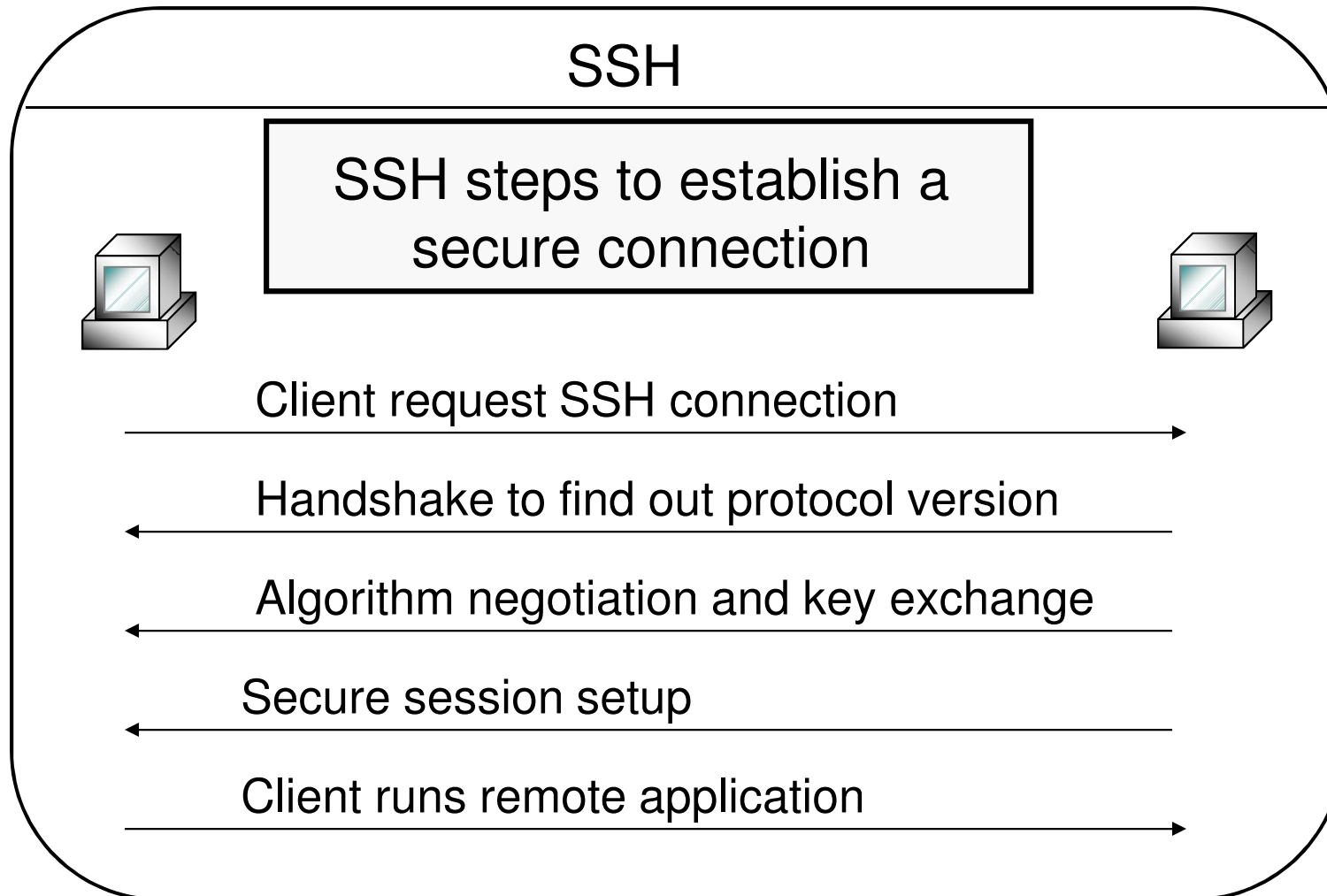
SSH Security Protocol

➔ Secure Shell (SSH)

- Secure terminal sessions
- Functions like a tunneling protocol
- Provides terminal-like access to remote system
- Should be used instead of telnet, UNIX r-utilities



SSH Steps



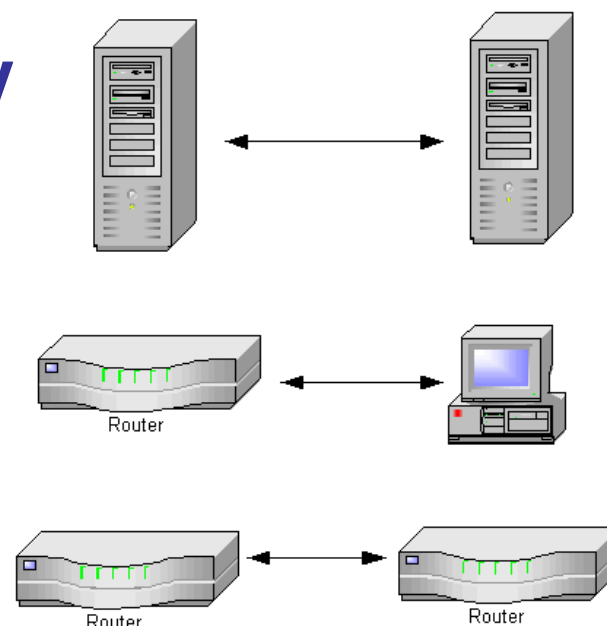
Security Related Protocols

➡ Secure Electronic Transaction (SET)

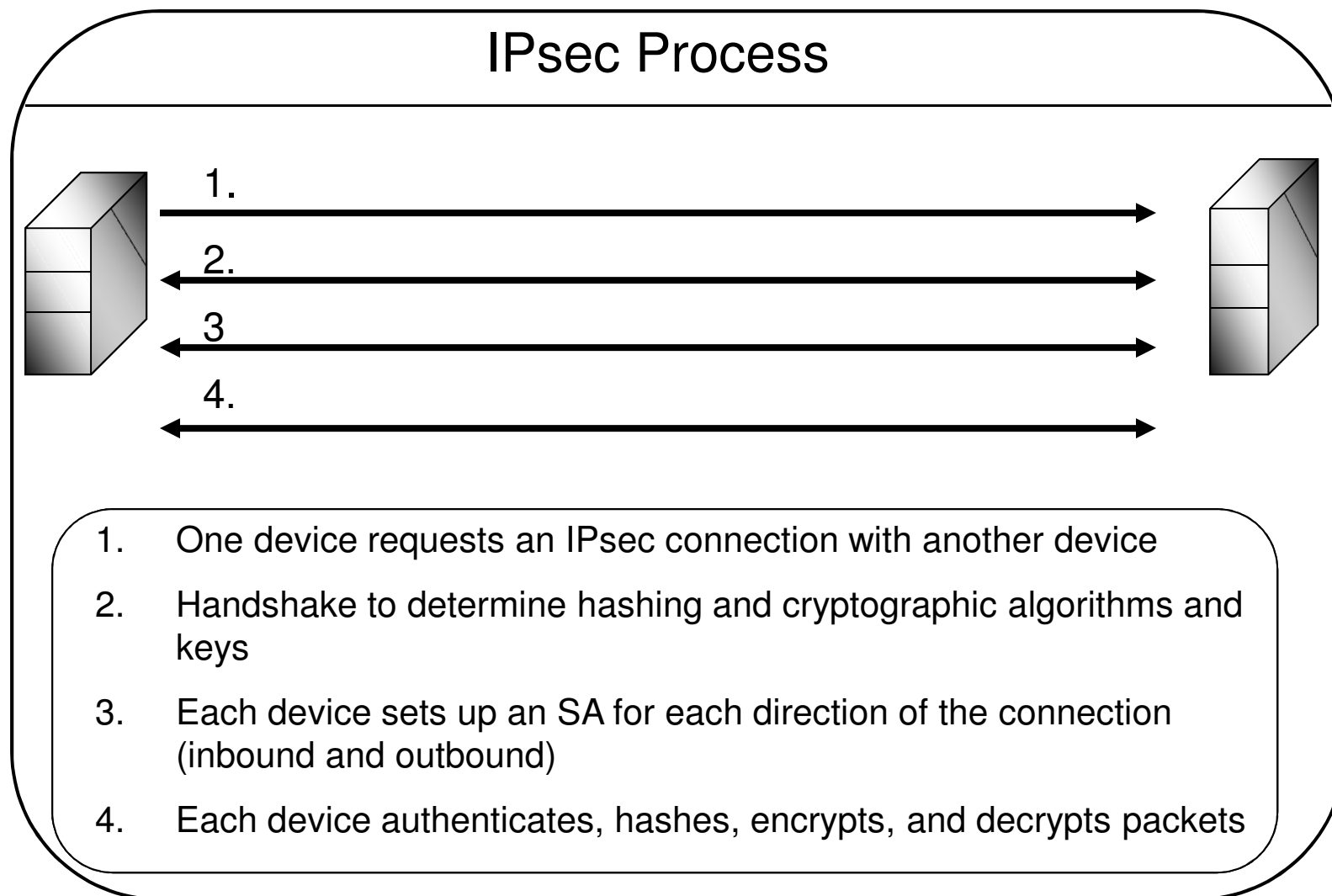
- Developed by Visa and MasterCard to allow for more secure monetary transactions over the Internet
- Cryptographic protocol that encrypts and sends credit card numbers over the Internet
- Uses public key cryptography
- Protects payment cards and cardholders' data
- Was developed to replace SSL, but it is slow in acceptance
- Characteristics: Confidentiality through DES; Digital signatures using RSA

IPsec

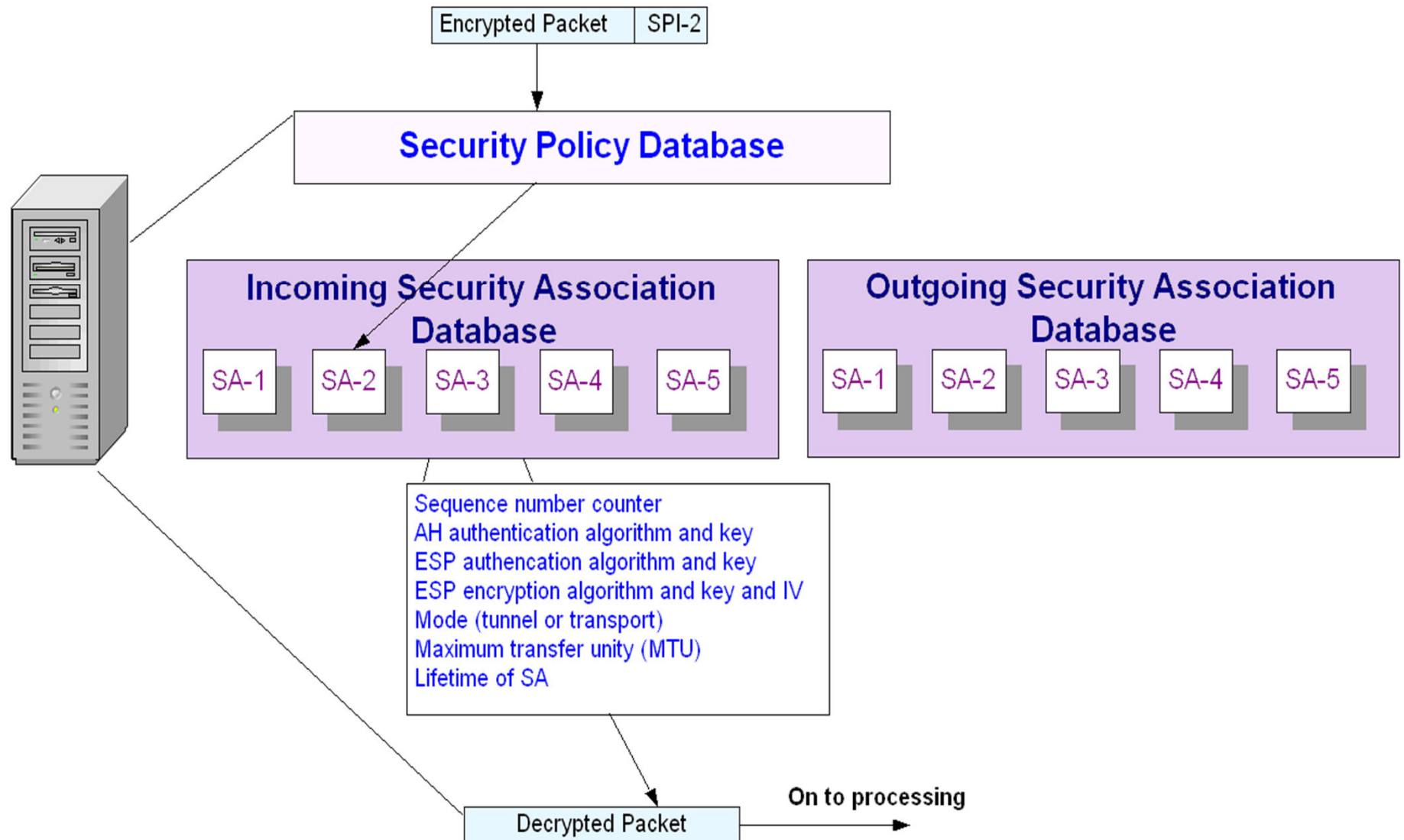
- ➔ Framework of open standards for ensuring secure communications over IP networks
- ➔ Network layer security
- ➔ Security between two nodes instead of two applications, as seen in SSL
- ➔ Uses public key cryptography
- ➔ Transport mode
 - Payload is protected
- ➔ Tunnel mode
 - Entire packet is protected
- ➔ Can provide host-to-host, host-to-subnet, and subnet-to-subnet links



IPsec Process



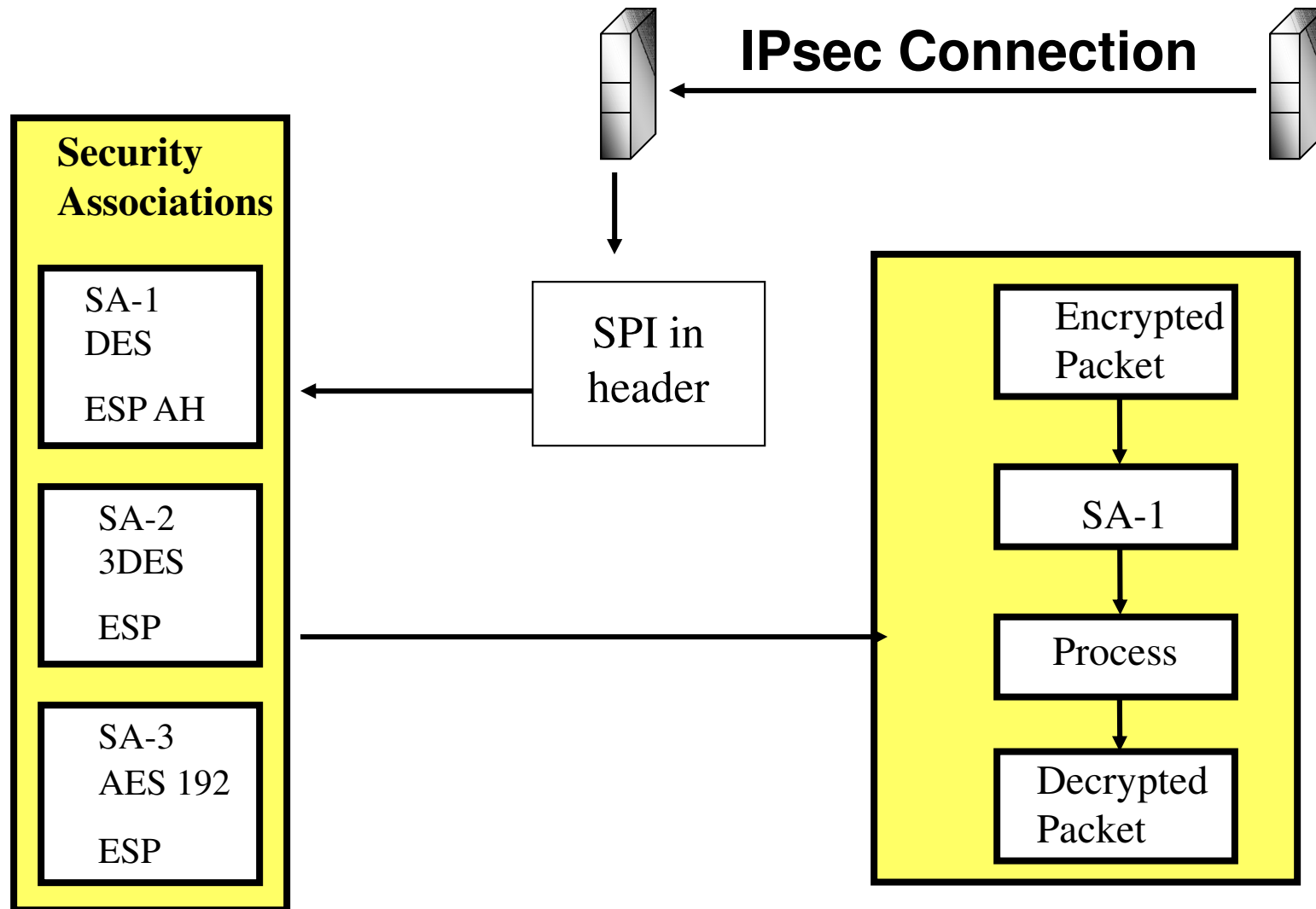
SA's in Use



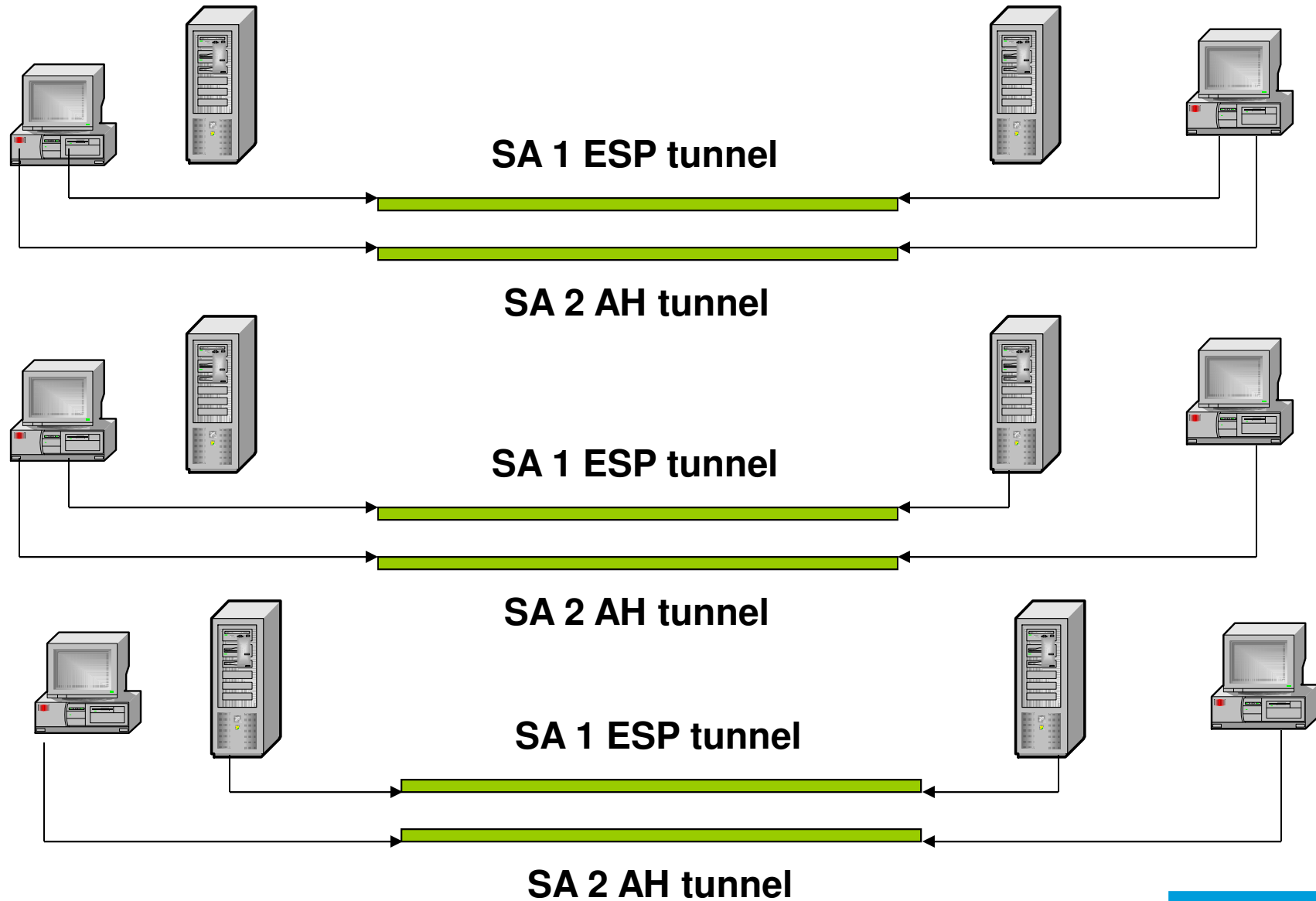
IPsec Components

- ➔ **Security Association (SA) defines Parameters for one specific active connection**
- ➔ **Security Parameter Index (SPI) Points to the correct SA**
- ➔ **SPI holds SA information and is put into header so both sides know what parameters to use to communicate**

IPsec Components



Combining Security Associations



IPsec

➔ Authentication Header (AH) Protocol

- Computes an Integrity Check Value (ICV) over entire IP packet except header field value that might change
 - Integrity through ICV
 - Data origin through MAC
 - Replay protection through sequence numbers

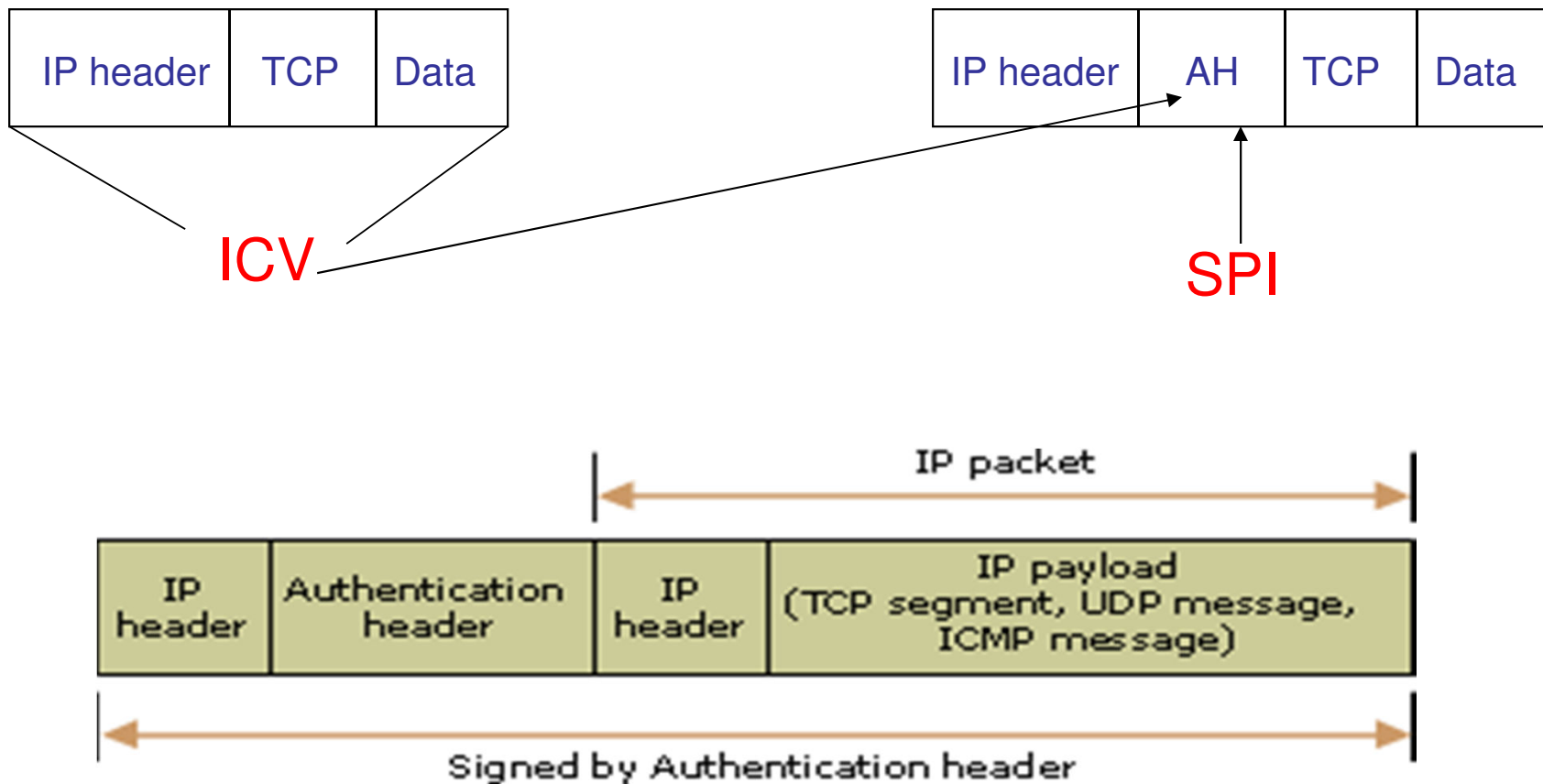
➔ Encapsulating Security Payload (ESP)

- Similar functionality as AH, but adds encryption
 - Adds confidentiality service

Authentication Header

- ➔ **Uses keyed-hash function instead of digital signatures because signatures are too slow**
 - MD5, SHA-1 used with symmetric keys (MAC)
- ➔ **ICV is calculated over data and IP header fields that do not change in transit**
- ➔ **Security Parameters Index**
 - Value uniquely identifies a security association for this packet
 - indicates a set of security parameters for use in this connection
- ➔ **Receiving end will calculate message digest and compare results for integrity**

AH Protocol – Transport & Tunnel Modes



Encapsulating Security Payload Protocol

➔ **Actual format depends on type of encryption and mode being used**

➔ **Encryption Algorithms**

- 3DES, RC5, IDEA, CAST, Blowfish, AES

➔ **Transport mode:**

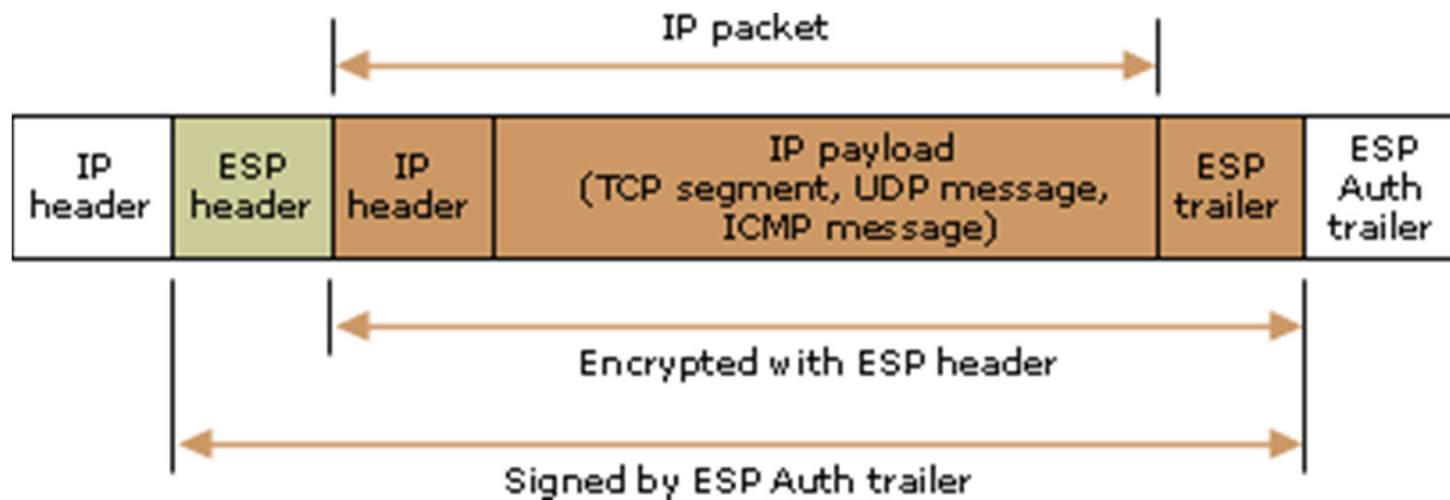
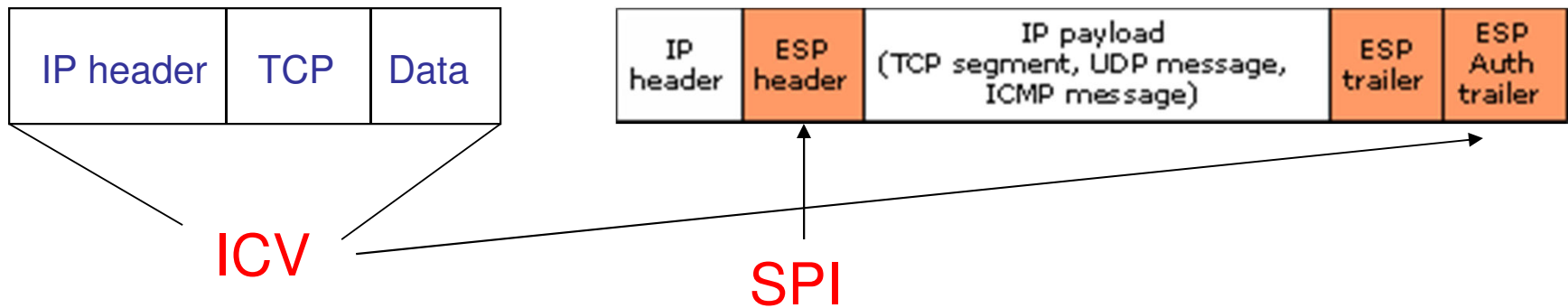
- Protects upper protocols, but not IP header

➔ **Tunnel mode:**

- All of packet including IP header protected – new IP header made

New IP header	ESP header	Original IP header	TCP	Data	ESP trailer
---------------	------------	--------------------	-----	------	-------------

ESP Protocol – Transport & Tunnel Modes



Agenda

- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ **Using cryptography to maintain application security**
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ Information hiding alternatives

E-mail Standards

➡ Privacy-Enhanced Mail (PEM)

- Internet E-mail Standard
- Framework that will allow different algorithms to be plugged in

➡ Message Security Protocol (MSP)

- Military's PEM – another framework
- Developed by NSA to provide secure e-mail exchanges

➡ Pretty Good Privacy (PGP)

- Free e-mail security program
- Developed by Phil Zimmerman
- Uses pass phrases instead of passwords
- Web of trust instead of hierarchy of CA's
- Keys are kept in a key ring file

Secure E-mail

➔ Secure MIME (S/MIME)

- Secure Multipurpose Internet Mail Extensions
 - Extends MIME Standard
- Application-layer Protocol
- Standard for encrypting and digitally signing e-mail containing attachments
- Developed to countermeasure message interception and forgery
- Provides data integrity, confidentiality, and authentication

E-mail Security

➡ Sender functions:

- Calculate hash value on message
- Encrypt message with session key
- Encrypt hash value with private key
- Encrypt session key with receiver's public key

➡ Receiver functions:

- Decrypt session key with private key
- Decrypt hash value with sender's public key
- Decrypt message
- Calculate hash value and compare with one sent

Secure Protocols

➡ Secure Hypertext Transport Protocol (S-HTTP)

- Protects each message – not communication channel

➡ HTTPS

- HTTP plus SSL – protects entire communication channel

➡ Secure Sockets Layer (SSL)

- Originally developed by Netscape
- Uses public key cryptography to protect communication channel
- Server authenticates to client, optionally client can authenticate to server
- Used for WWW connections
- Works at transport layer

Agenda

- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ **Public Key Infrastructure (PKI)**
- ➡ Certificate related issues
- ➡ Information hiding alternatives

Why Do We Need an Infrastructure?

➡ Public Key Cryptography Issues

- Systems can generate their own public/private key pairs and exchange them – but what trust level is there?
- If a public key stored in a database were changed out with the attacker's key, then he can encrypt and sign items with his private key
 - Masquerading as someone else
- Need a trusted third party to vouch for the identity of the owner of a public key

What Makes Up a PKI

- ➡ CA
- ➡ RA
- ➡ Certificate Repository
- ➡ Certificate Revocation System
- ➡ Key Backup and Recovery System
- ➡ Automatic Key Update
- ➡ Management of Key Histories
- ➡ Time Stamping
- ➡ Client-side Software

What Makes Up a PKI

➡ Security Services Provided by PKI

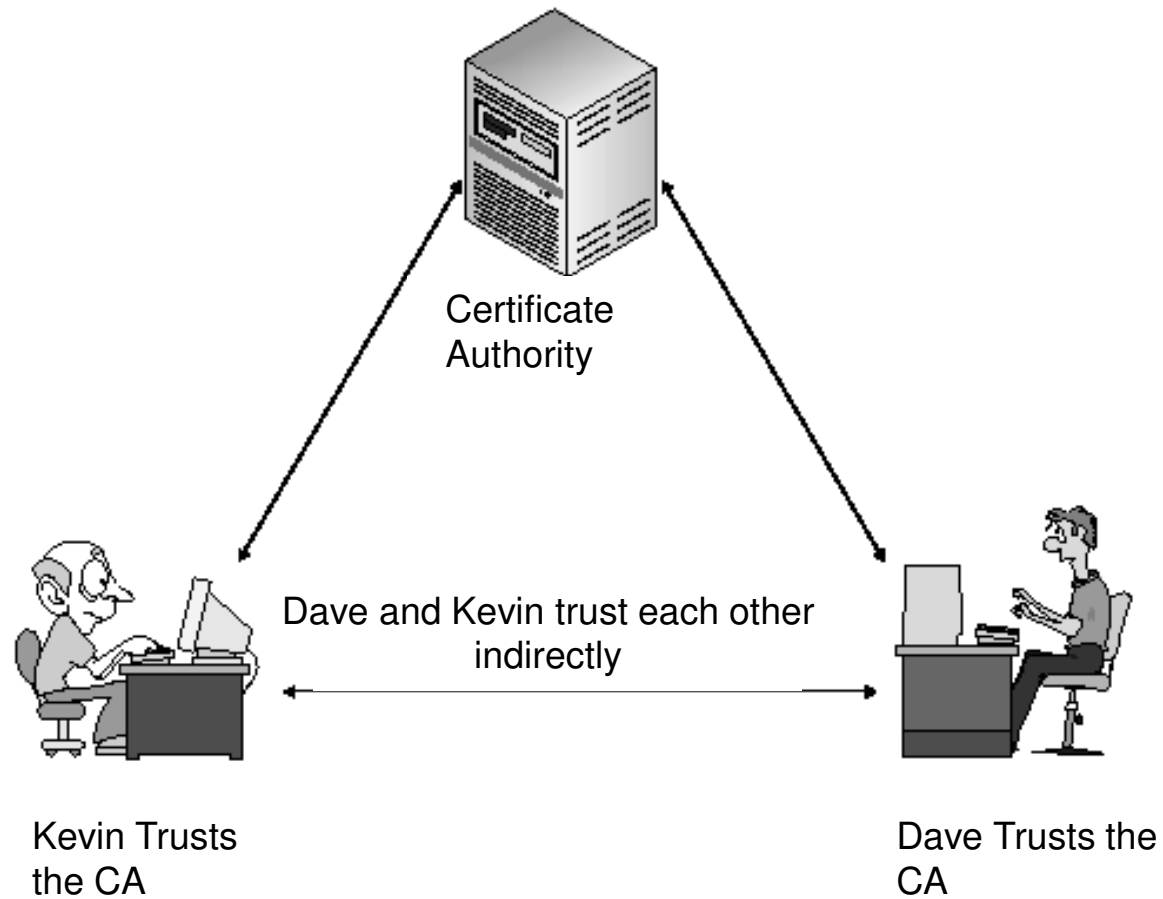
- Confidentiality
- Access Control
- Integrity
- Authentication
- Non-repudiation

➡ Certificates and digital signatures are based on the X.509 Version 3 Standard

Components of PKI – CA

- ➡ **Entity that issues digital certificates**
 - Public directory maintains public key certificates
- ➡ **Signs certificate with its private key**
- ➡ **Users trust CA and then each other indirectly**
 - Users are referred to as subscribers
- ➡ **Public CA's operate via the Internet**
- ➡ **Private CA's operate in organizations**

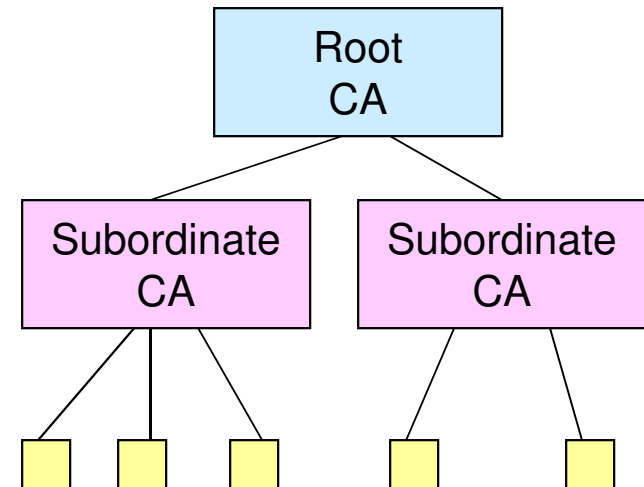
Components of PKI – CA



CA Hierarchy

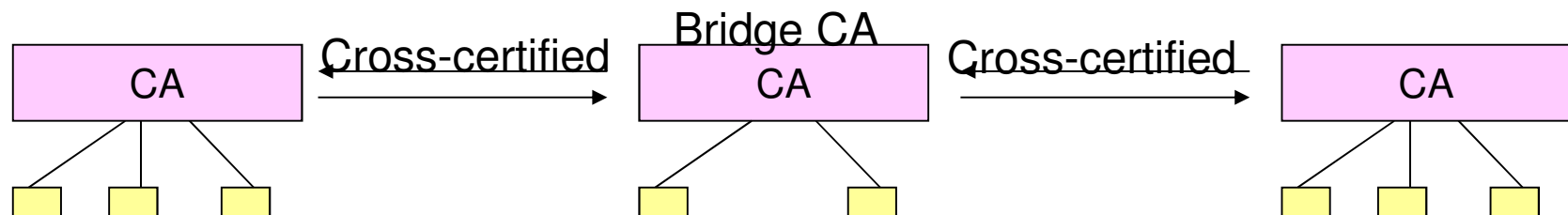
➔ How different CA's can communicate

- Certificate Hierarchy:
 - CA delegates authority to subsidiary CA's
- Root CA
 - Initiates all trust paths
 - All certificate holders and relying parties are given self-signed root CA certificates
 - Verisign, Entrust
- Subordinate CA
 - Does not begin trust path
 - May have subordinate CA's of its own which it issues certificates to



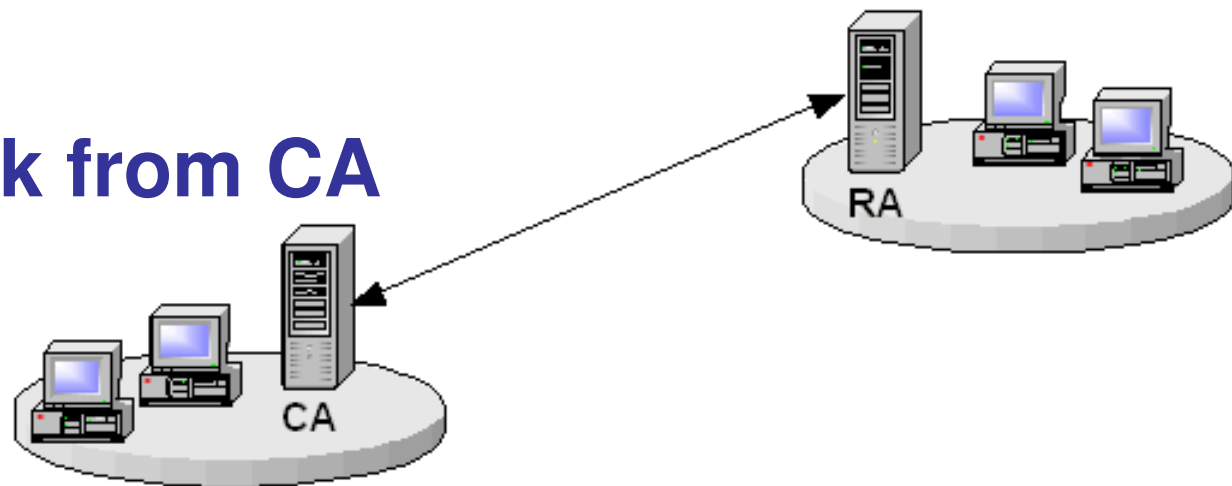
Cross-certification structure

- ➔ When two PKI's are established between organizations or businesses
- ➔ Nonhierarchical trust path – mutual trust
 - Requires two certificates – one for each direction
- ➔ *Develop a cross-certification agreement*



Registration Authority Responsibilities

- ➔ Accept and verify registration information
- ➔ Accept and authorize requests for certification revocation
- ➔ Cannot issue a certificate
- ➔ Offload work from CA



Components of PKI - Directory

➡ Certificate Directory

- Storage of certificates
- No required directory standard
 - Lotus Notes and Microsoft Exchange
 - Other directories based on X.500 are used
 - Certificates can be stored with other network data – user information, network device data, etc.
- Clients can locate and access directories via Lightweight Directory Access Protocol (LDAP)

Components of PKI – CRL

- ➔ **CRL is a signed data structure containing a time-stamped list of revoked certificates**
 - List can be distributed or stored in a directory
- ➔ **If a certificate has been revoked for some reason, it will be listed on the CRL**
 - Compromise of a private key, employee left company
- ➔ **When users verify another's identity through a CA, they will check the CRL to make sure it is still valid**

PKI and Trust

➔ Level of trust required for reliable use of digital certificates

- A certain level of trust is required to complete transactions reliably
- Trust is derived from many techniques:
 - The CA
 - The steps the CA goes through to identify the individual
 - Use of the digital certificate
 - Protection and security of private keys

Agenda

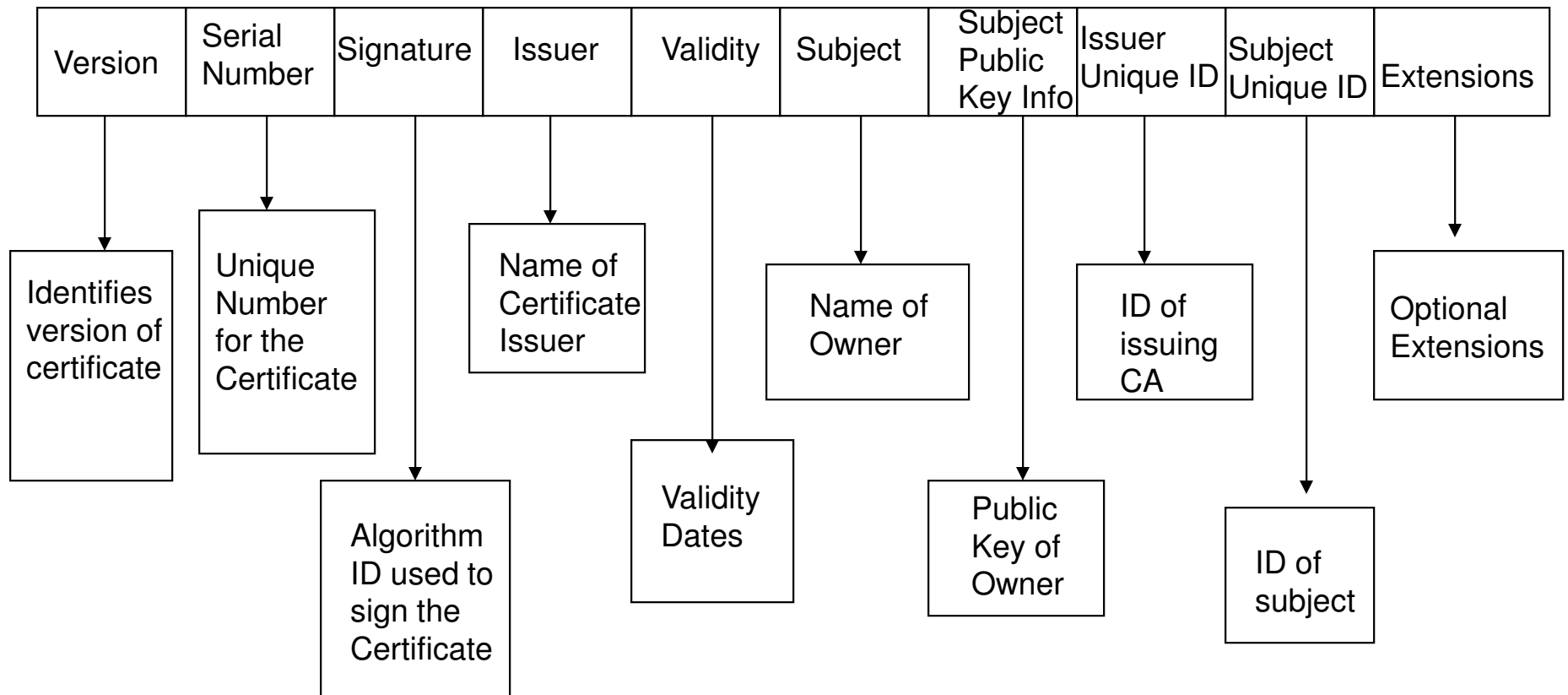
- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ **Certificate related issues**
- ➡ Information hiding alternatives

Components of PKI – Certificates

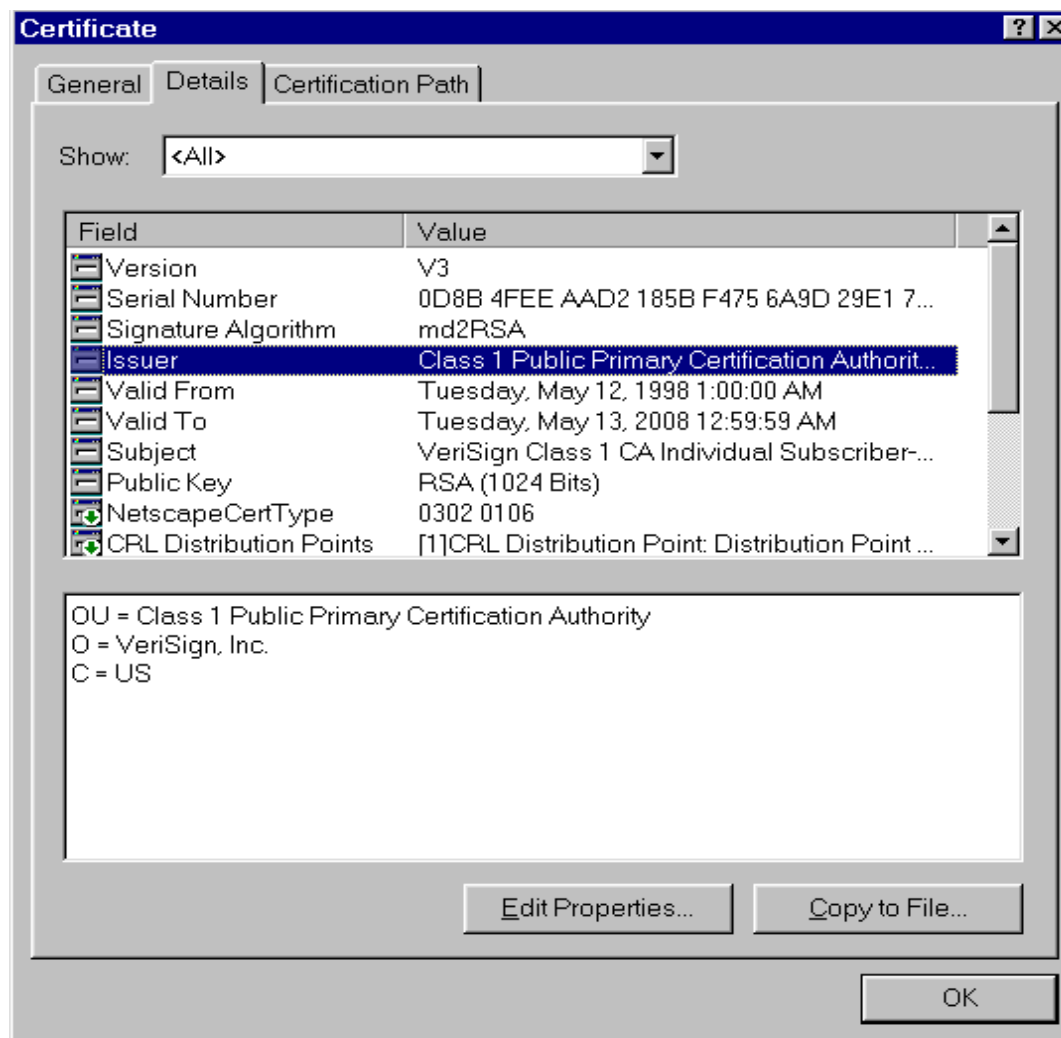
➡ X.509 Version 3 Certificates

- Secure means of distributing public keys
- Associates public key with owner
- All of this data is digitally signed by trusted anchor
 - Thus protected by digital signature
 - RSA and MD5 or DSA and SHA-1

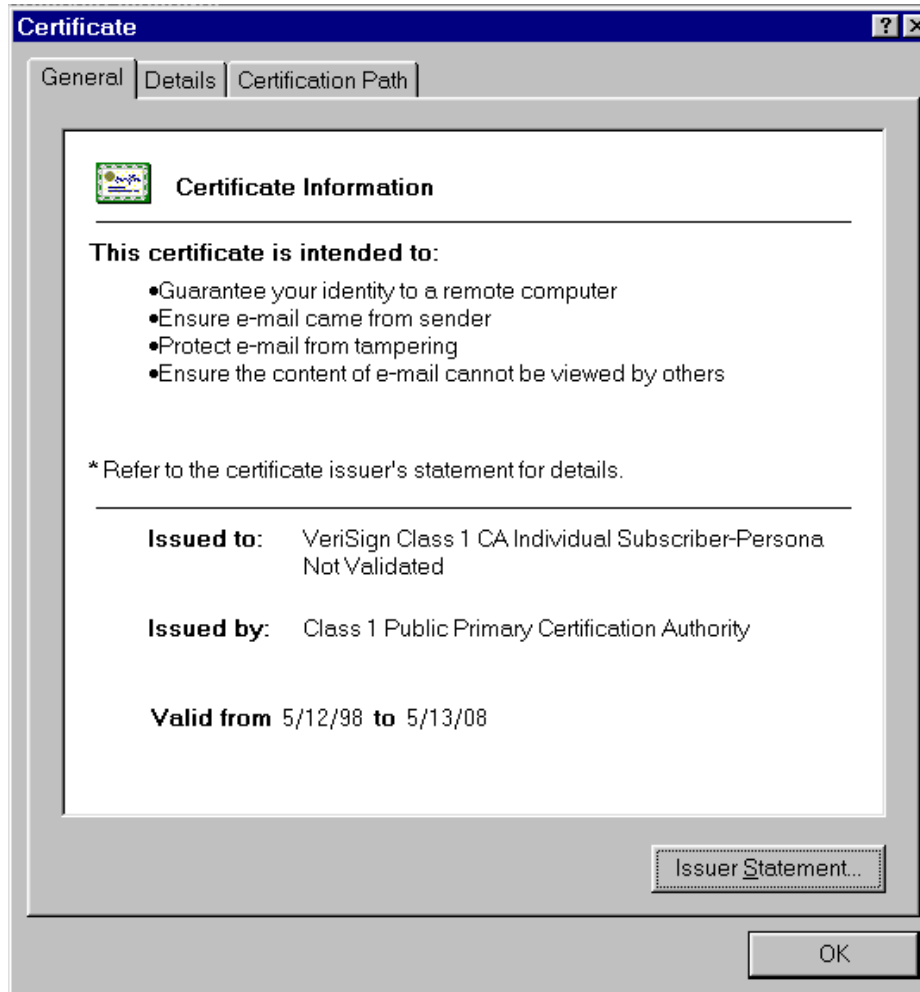
Components of PKI – Certificates



Certificate Details



Uses for Certificates



Receiving a Certificate

- ➡ **Sender signs a message digest and sends a certificate containing the necessary public key**
- ➡ **Steps to verify a digital signature – part 1:**
 - Receiver runs certificate through hashing algorithm
 - Receiver decrypts hash in certificate to ensure the trusted CA signed the certificate
 - Receiver compares two results to ensure that the certificate has not been modified
 - Receiver extracts public key from certificate

Receiving a Certificate

➡ Part 2

- Receiver runs message through hashing algorithm to calculate a new hash
- Receiver decrypts original hash (in digital signature format) with public key from certificate
- Receiver compares results of message digests for message to ensure its integrity has not been compromised

Agenda

- ➡ Application and use of cryptography
- ➡ Cryptographic lifecycle and encryption concepts
- ➡ Key management processes
- ➡ Digital signatures and non-repudiation
- ➡ Methods of cryptanalytic attacks
- ➡ Using cryptography to maintain network security
- ➡ Using cryptography to maintain application security
- ➡ Public Key Infrastructure (PKI)
- ➡ Certificate related issues
- ➡ **Information hiding alternatives**

Concealment (or Null) Cipher

➡ **True letters are hidden or disguised by a device or algorithm**

- The true message is hidden

➡ **For example: every third word in a sentence:**

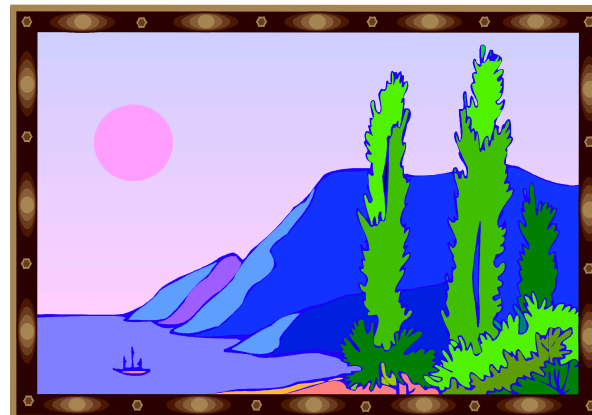
- “The old red rooster hit head first bypassing rules.”
- The secret message is “red head rules”

Hiding Messages in Media

➔ Steganography

- Hiding the very existence of data within another message or media
- Digital watermark to detect illegal copies of digital images
- No algorithm or key; just placing data in a place that people would not usually look

The money is hidden under the dock by the bench.



Concealment (or Null) Cipher

➡ **True letters are hidden or disguised by a device or algorithm**

- The true message is hidden

➡ **For example: every third word in a sentence:**

- “The old red rooster hit head first bypassing rules.”
- The secret message is “red head rules”

Steganography

- ➡ From the Greek word *steganos* which means hidden or covered, plus *graphein* which means to write
- ➡ Hiding in plain sight
- ➡ Hiding a message within another medium
- ➡ You cannot even claim there is a message
- ➡ Can hide information in a variety of formats:
 - High Quality graphic files
 - MP3 and other audio files
 - Video clips
 - Even text files
- ➡ Invisible and hard to detect without tools

Steganography Jargon

➡ Carrier File

- Simple a file that has information hidden in it

➡ Steganalysis

- Detection of steganography usage in a medium

➡ Stego Medium

- The medium in which information is hidden

➡ Redundant Bits

- Bits that can be overwritten or altered within the medium

➡ Payload

- Information concealed within another file or medium

Steganography Usage

➡ Illegal usage

- Claimed to be used by terrorist group
 - Posting items for sale on Ebay
 - Posting other high quality pictures
- Leaking files and information in and out of your company

➡ Legit usage

- Used by graphic artists to watermark their work
- Used by Photographers to watermark their photos
- Watermarking of documents and other work

➡ Researchers could not prove it is used at large

➡ Learn more by reading Hiding in Plain Sight

- From Eric Cole, ISBN-13: 978-0471444497

Steganography Explained

Dark Green Background

Byte 1 - Red	Byte 2 - Green	Byte 3 - Blue
0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0

0 1 0 0 0 0 0 1	0 1 1 0 1 0 0 0
0 1 1 0 0 0 0 1	0 0 1 0 0 0 0 1

Pixel	Byte 1 - Red	Byte 2 - Green	Byte 3 - Blue
1	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0
2	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
3	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0
4	0 1 0 1 0 1 1 1	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0
5	0 1 0 1 0 1 1 1	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
6	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 1
7	0 1 0 1 0 1 1 1	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
8	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 1
9	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 1
10	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 0	0 0 0 1 1 1 1 0
11	0 1 0 1 0 1 1 0	0 1 1 1 1 0 1 1	0 0 0 1 1 1 1 0

1 pixel = 24 bits or 3 bytes

256 values for each color

$256 * 256 * 256 = 16777216$ or 16.7 Million colors

← the four characters of Aha! - in ASCII binary

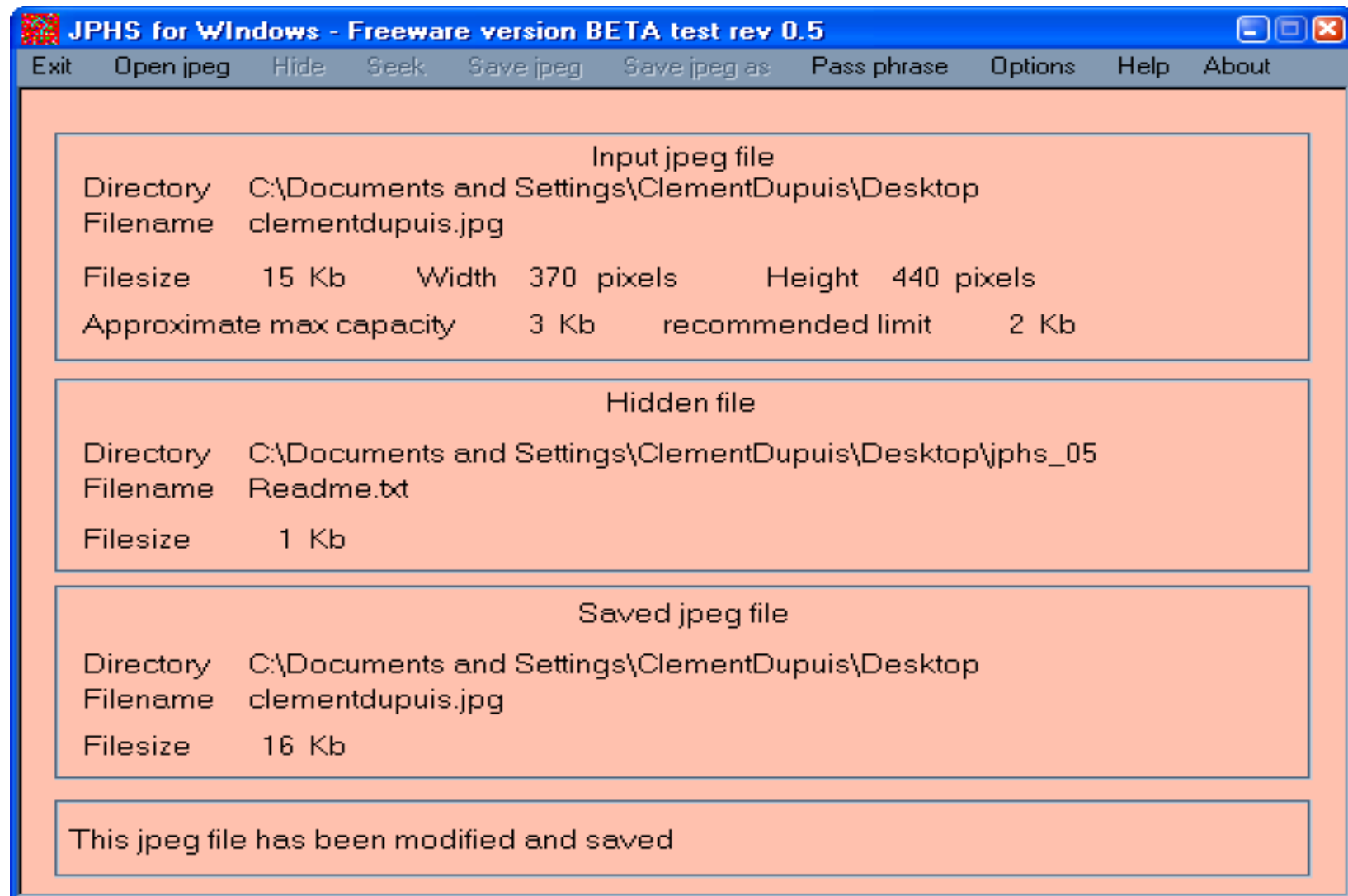
← We now hide one bit at the end of each of the color code bytes of 11 different pixels.

← In the diagram, each maroon or gold box represents a bit that had to be changed to include the hidden message.

← Notice that only 15 of 264 bits (less than 6%) had to be changed and only 8 of the 11 pixels were altered.

← Changes will not be visible to a human eye
Our depth of perception is only 6 bits

JPHide and JPSeek



The JPHide and JPSeek Tool



Without Data



With 1K of Data Hidden

StegDetect Tool

