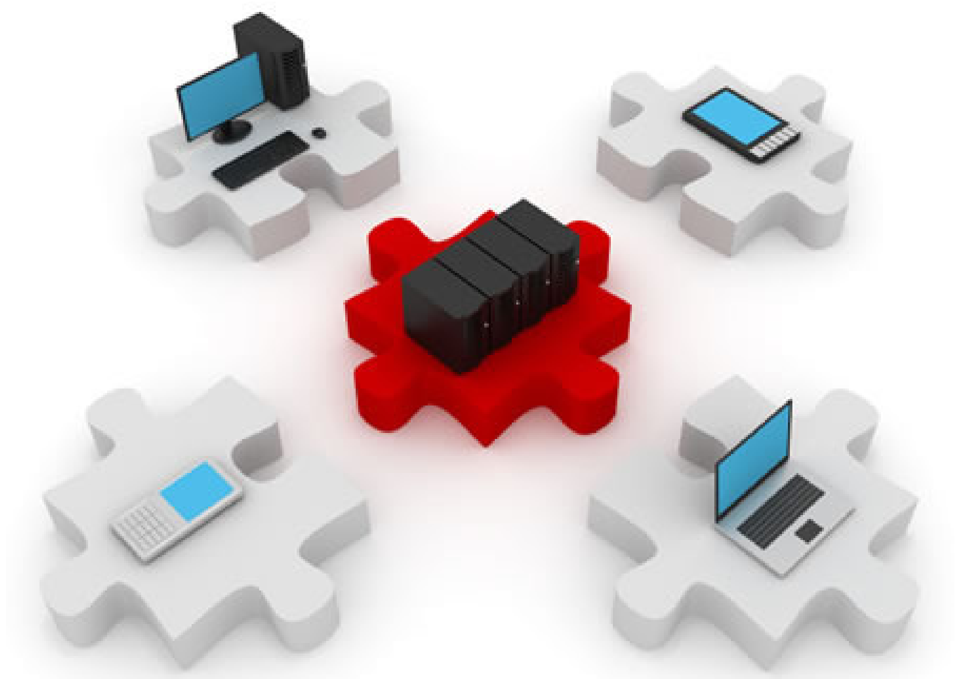


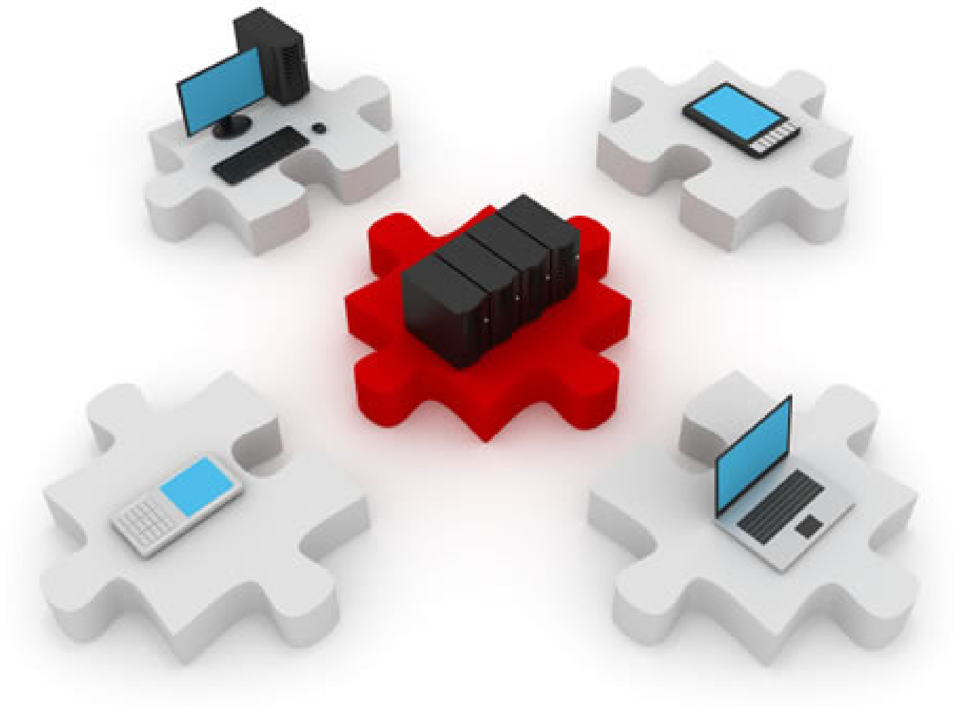
Cryptography

3-dec-2012

What this lecture is about:

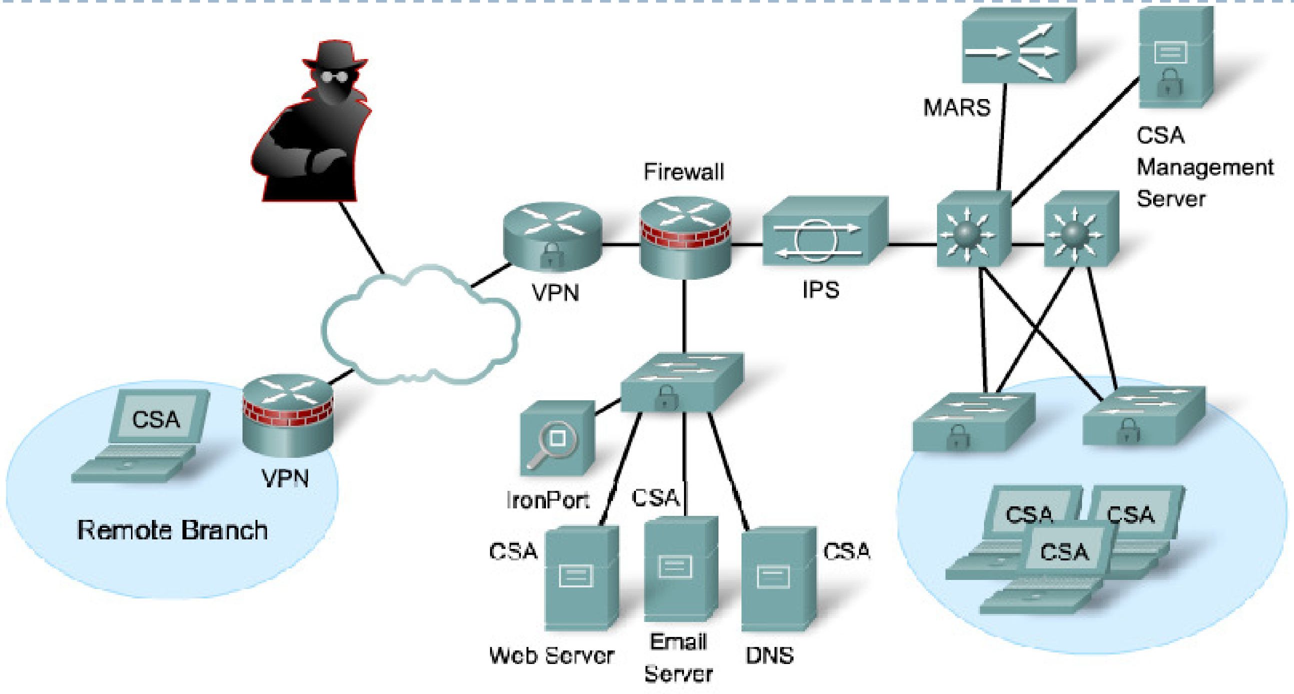
- Cryptology, including:
  - Cryptography
  - Cryptanalysis

- Securing communications by ensuring:
  - Authentication
  - Data integrity
  - Confidentiality
  - Non-repudiation



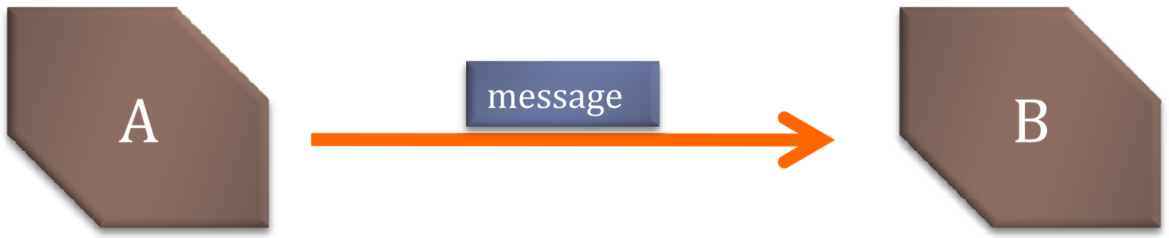
Securing communication

Securing communication in a network



- So far, you've learned that security must be applied to:
  - The network infrastructure (remember how?)
- The next goal is to secure data
  - Especially when it's travelling over the network

The basic problem



- A wants to send a message to B.
  - A does not want anyone else to read the message.
  - B wants to make sure that the message came from A.
  - B wants to make sure that the message it received is the same as the message sent by A.

- Ok, not A and B, but Alice and Bob ☺

How can you secure that message?

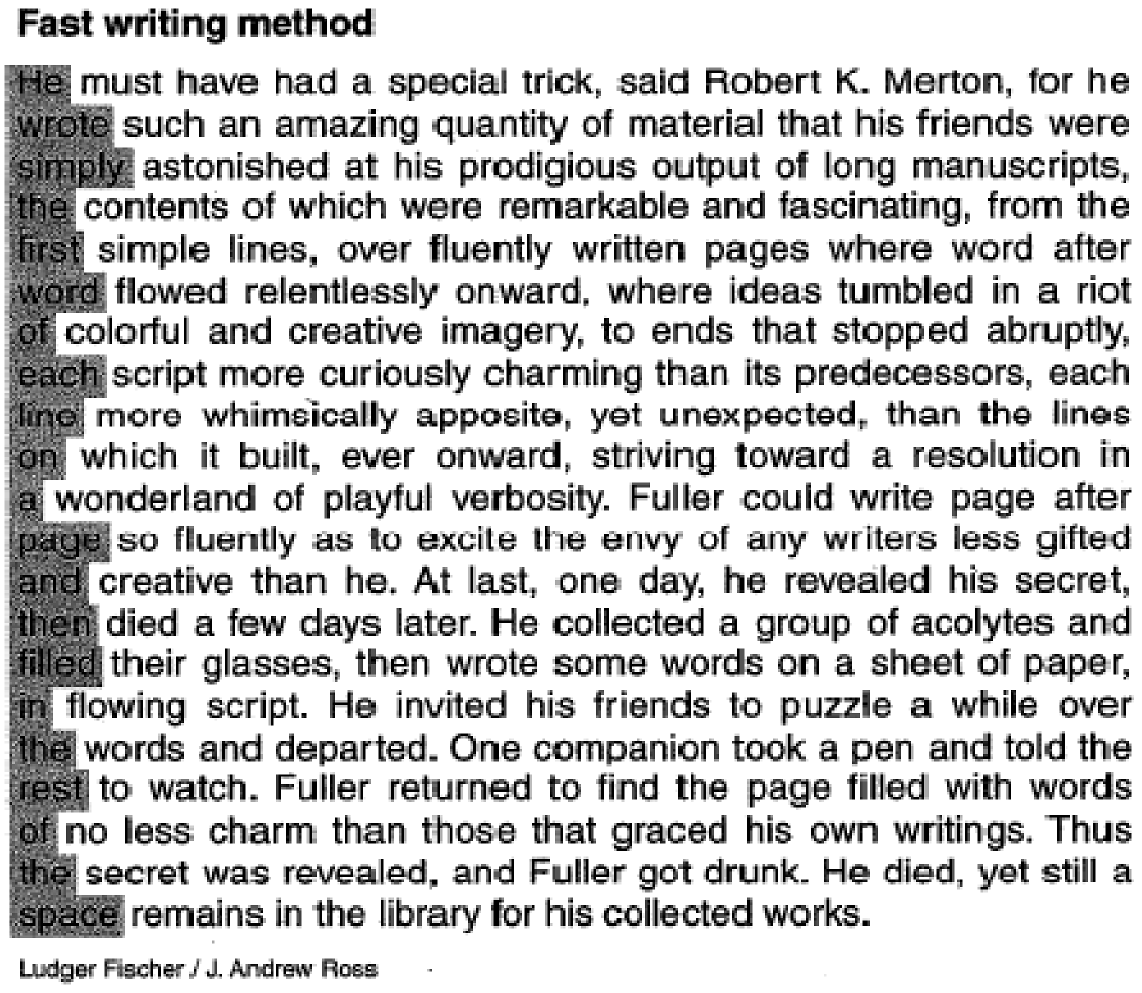
- One option: make sure that nobody else can receive that information.
  - Very difficult.
  - Very expensive.
  - Pretty-much impossible on the Internet

- A better option: hide your important data
  - Write the message in such a way so that only the recipient can understand it.
  - Hide the message inside a dummy message (steganography).
  - Encrypt/scramble the message to make it unreadable (cryptography).

Steganography

- An alternative to encryption.
- Hides the existence of the message.
  - An example of “security by obscurity”
- How?
  - Using a limited subset of marked symbols hidden in a longer message.
  - Using invisible ink.
  - Hiding information in pictures (in the binary file of an image).
- Disadvantage: requires large amounts of data to hide little information.
  - Once discovered, extremely easy to obtain the information.
  - Still, data can be encrypted before it is hidden.

Steganography example



Fast writing method

He must have had a special trick, said Robert K. Merton, for he
wrote such an amazing quantity of material that his friends were
simply astonished at his prodigious output of long manuscripts,
the contents of which were remarkable and fascinating, from the
first simple lines, over fluently written pages where word after
word flowed relentlessly onward, where ideas tumbled in a riot
of colorful and creative imagery, to ends that stopped abruptly,
each script more curiously charming than its predecessors, each
line more whimsically apposite, yet unexpected, than the lines
on which it built, ever onward, striving toward a resolution in
a wonderland of playful verbosity. Fuller could write page after
page so fluently as to excite the envy of any writers less gifted
and creative than he. At last, one day, he revealed his secret,
then died a few days later. He collected a group of acolytes and
filled their glasses, then wrote some words on a sheet of paper,
in flowing script. He invited his friends to puzzle a while over
the words and departed. One companion took a pen and told the
rest to watch. Fuller returned to find the page filled with words
of no less charm than those that graced his own writings. Thus
the secret was revealed, and Fuller got drunk. He died, yet still a
space remains in the library for his collected works.

Ludger Fischer / J. Andrew Ross

Fig. 14. Self-describing acrostic

Security Requirements

- The three main security requirements for network traffic are:
  - Authentication
    - Make sure the sender is who he/she claims to be.
    - Make sure the receiver is the one intended and not a “spy”.
  - Integrity
    - Guarantees that nobody has tampered with your data while it was in transit.
    - Similar to a checksum, but its purpose is not to detect errors, but undesired alterations.
  - Confidentiality
    - You cannot make a message by itself “un-capturable”
    - Any kind of traffic can and will be sniffed at some point.
    - But you can make sure that a message cannot be deciphered if captured.

Authentication

- A PIN is also a form of authentication.
- The PIN is a “shared secret” between the bank and the client.
- Cryptographic methods can also provide authentication.
- Many protocols and applications do not provide built-in authentication mechanisms.
  - They are vulnerable to spoofing attacks

Non-repudiation

- Another aspect of authentication: non-repudiation.
- Is a service that allows the sender of a message to be uniquely identified.
  - More exactly, the sender cannot deny being the source of a message.

- Non-repudiation specifies that only the sender has the unique characteristics or signature for how the message was treated.
  - Not even the receiver can pretend to be the source.
  - Proper security must be able to uniquely identify the origin of a message.

Integrity

- Security must also ensure that data was not altered in transit.
- Communication integrity confirms that the message that has been received is the message originally sent.
- Wax seals were used to make sure that nobody has read or altered the message.
- Nowadays, message hashes are used as a "signature" only to prevent modifications.

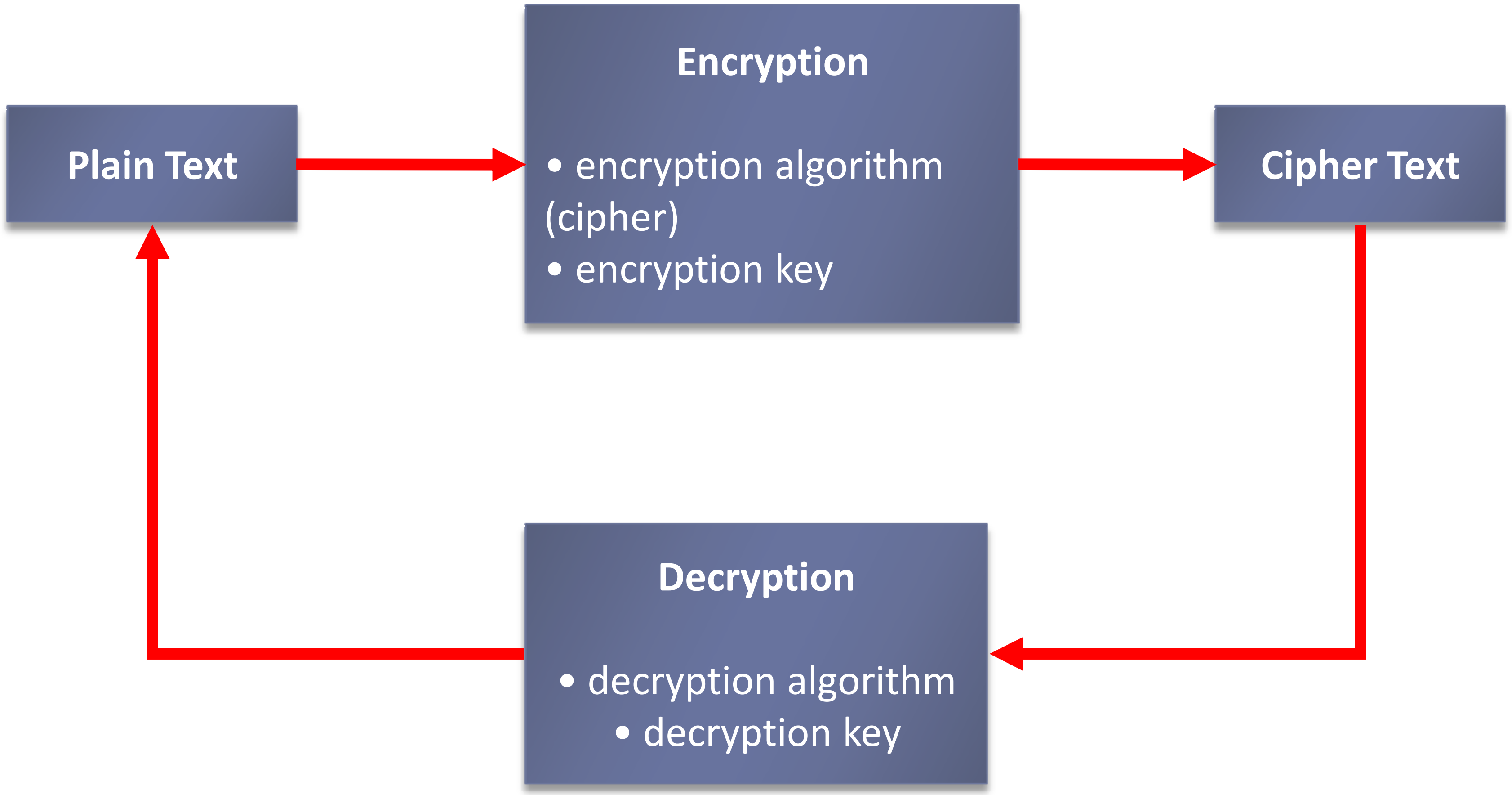
Confidentiality

- A confidential message can only be understood by the destination.
- **Encryption** converts clear-text data into encrypted data, called a cipher-text.
  - The reverse process is called **decryption**.
  - The encryption algorithm is public in most cases
  - The link between the clear-text and the cipher-text is called a key.
  - The key can be a string of letters/numbers or a method (algorithm) for interpreting the cipher-text.

Cryptographic algorithms

- Technologies that are used as a ground for developing cryptographic systems.
- Usually have their basis in mathematics.
- Combinations of underlying algorithms can lead to sophisticated and highly secure systems.

The cryptographic process



Symmetric encryption

- The only encryption method until 1970.
  - That's when public key cryptography was invented.

- Also called private-key / secret key encryption.
- The sender and the recipient share the same secret key.
- In order to be used, both peers must know the key.
  - It can be statically configured on each one.
  - Or it can be sent in a secure manner from one to the other.

- Still widely used today
  - Because it is the least CPU-intensive method.
  - Suitable for resource-constrained devices.

Requirements for symmetric encryption

- Two requirements:
  - A strong algorithm for encryption (E)
    - How do you define “strong”?
  - A secret key (K) known only to the sender and the receiver.
- The decryption algorithm (D) is the inverse of E and can be easily deducted:
  - Cipher = E_K(Msg)
  - Msg = D_K(Cipher)
- Assumptions:
  - D and E are relatively fast algorithms.
  - The algorithm is public, the key is secret.
  - The same key is used for encryption and decryption as well.

Symmetric encryption types

- By the operation executed:
  - Substitution ciphers
    - Change the message’s characters.
  - Transposition ciphers
    - Permute the message’s characters.
  - Product ciphers
    - A combination of both.

- By the way that plaintext is processed:
  - Stream ciphers
    - Independently encodes every character of the plaintext.
  - Block ciphers
    - Splits the plaintext into blocks of characters and applies the encryption algorithm on each block.

Symmetric encryption examples

- Substitution ciphers:
  - Monoalphabetic (Caesar)
  - Polyalphabetic (Alberti, Vigenere)
  - Monophonic
- Transposition ciphers:
  - Columnar (Rail fence)
  - Block reversal
- Product ciphers:
  - Enigma machine
  - DES, 3DES (Data Encryption Standard)
  - AES (Advanced Encryption Standard)

Substitution ciphers

- Units of plaintext are substituted with cipher-text according to a regular system.

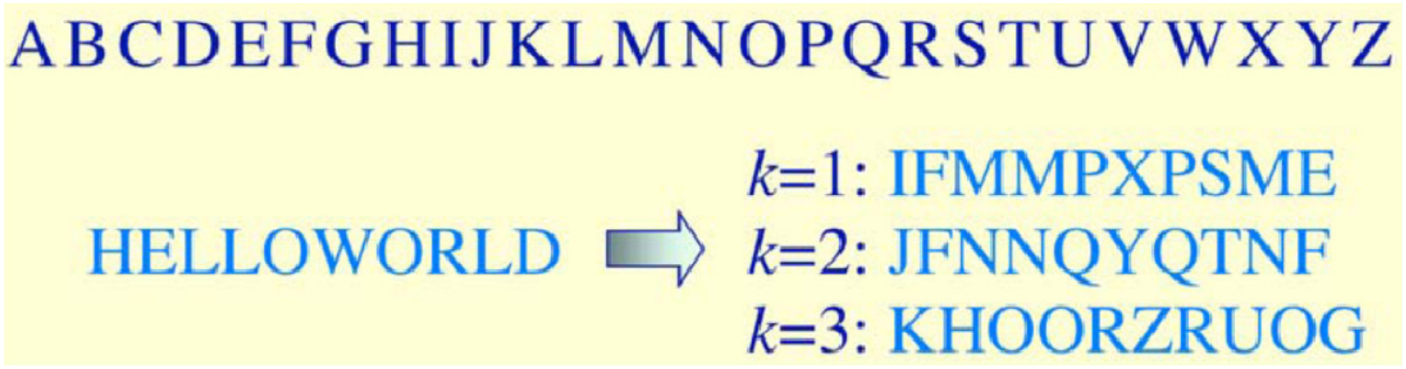
- Monoalphabetic substitution
  - Each letter in the plaintext is encoded by only one letter from the cipher alphabet (and vice-versa).
  - One-to-one relationship.

- Polyalphabetic substitution
  - Each letter in the plaintext can be encoded by several characters from the cipher alphabet (and vice versa).
  - Many-to-many relationship.

Monoalphabetic – the Caesar cipher

- Shift the plaintext characters with k characters to the right
  - Apply modulus when overflow occurs.

- The result for shifting “HELLOWORLD” with 1, 2 and 3 characters to the right:



- What is the key of this cipher?
- Answer: k

The Caesar cypher – another version

- The Caesar cipher is weak, having only 26 possible keys.
- Besides shifting every letter with the same key, a different key can be used for every letter.
- Each plaintext letter maps to a different “random” cipher-text letter.
  - Now we have 26-letter long keys.
  - Which give us a whopping number of 26! keys (! Is factorial)

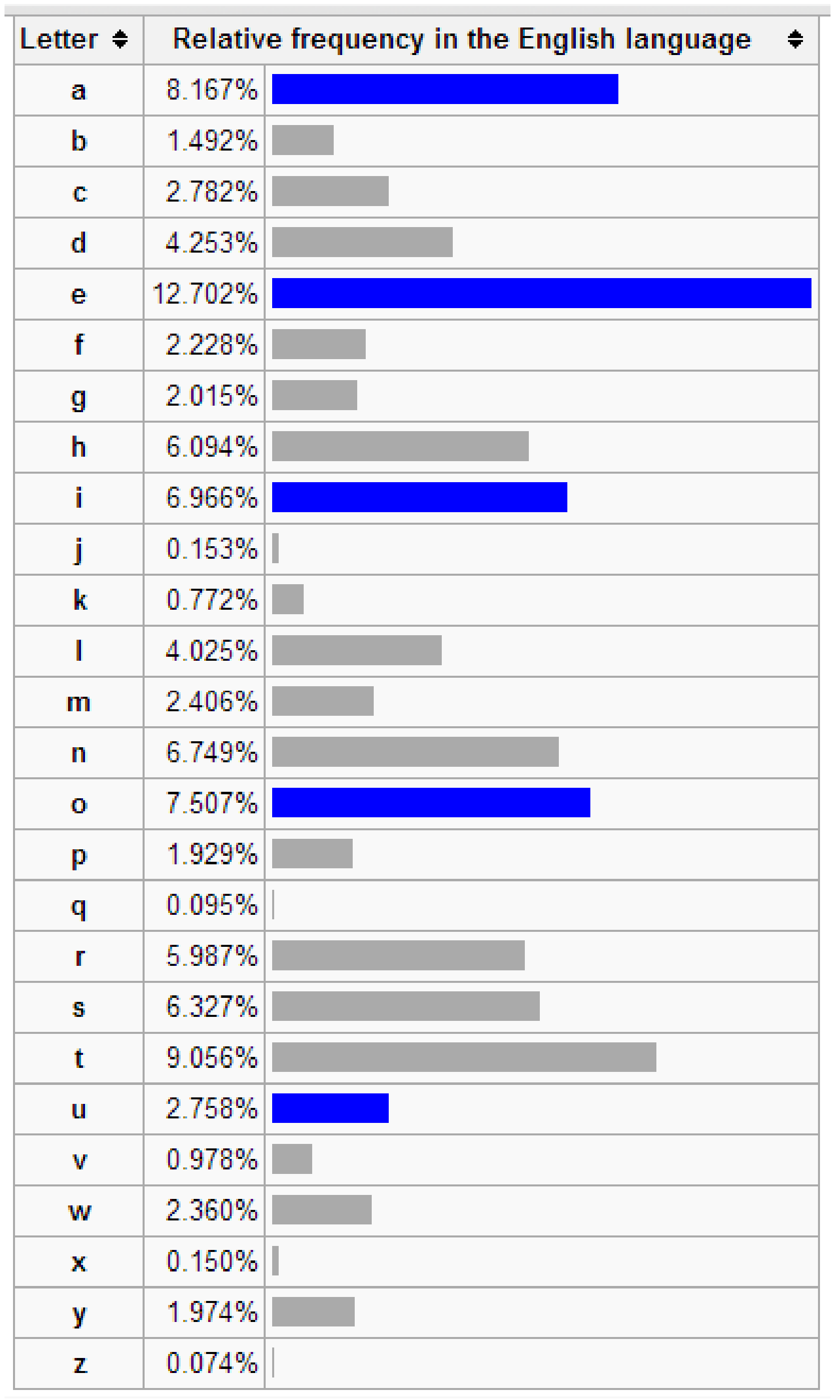
```
Plain:  abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext:  ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

Cracking monoalphabetic ciphers

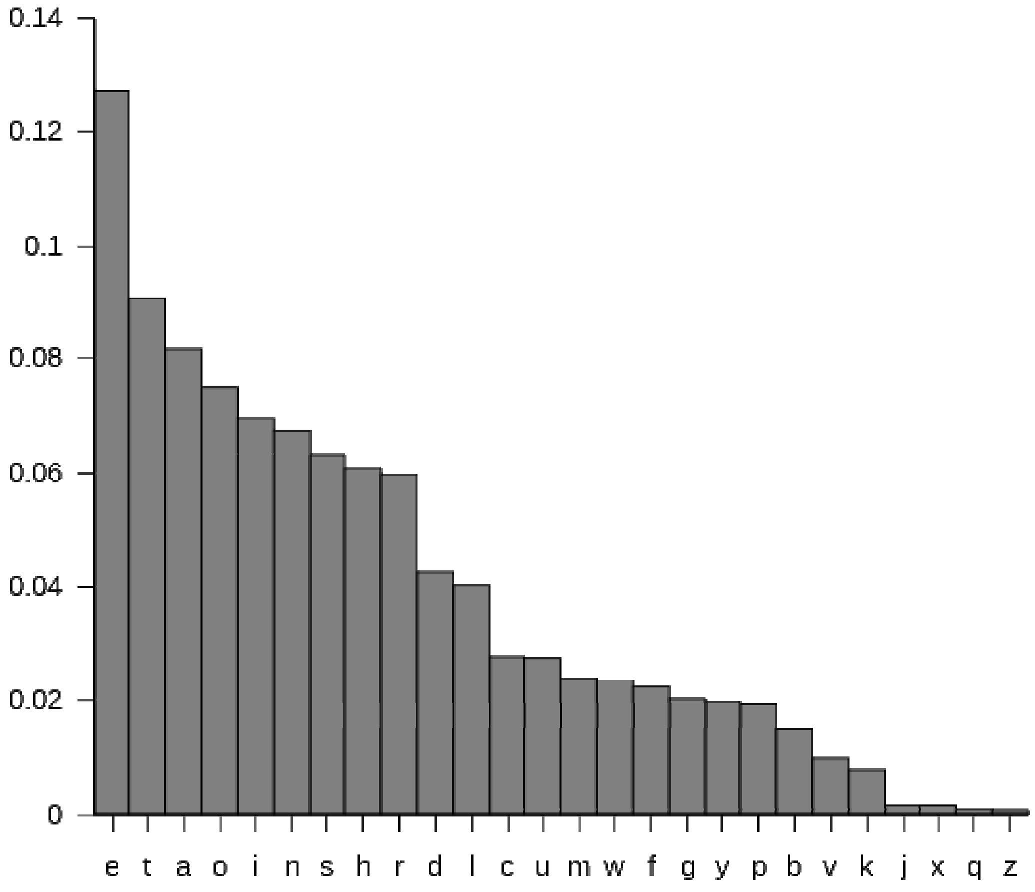
- For use in cryptanalysis, the following concept is useful:
  - Mono-alphabetic substitution ciphers do not change relative letter frequency.

- Calculate letter frequencies for a giver cipher-text.
- Compare counts against known values.
- Look for common highest and lowest frequencies.

- For example, the English language letter frequency is:
  - 12.702% for the “e” letter
  - 0.074% for the “z” letter

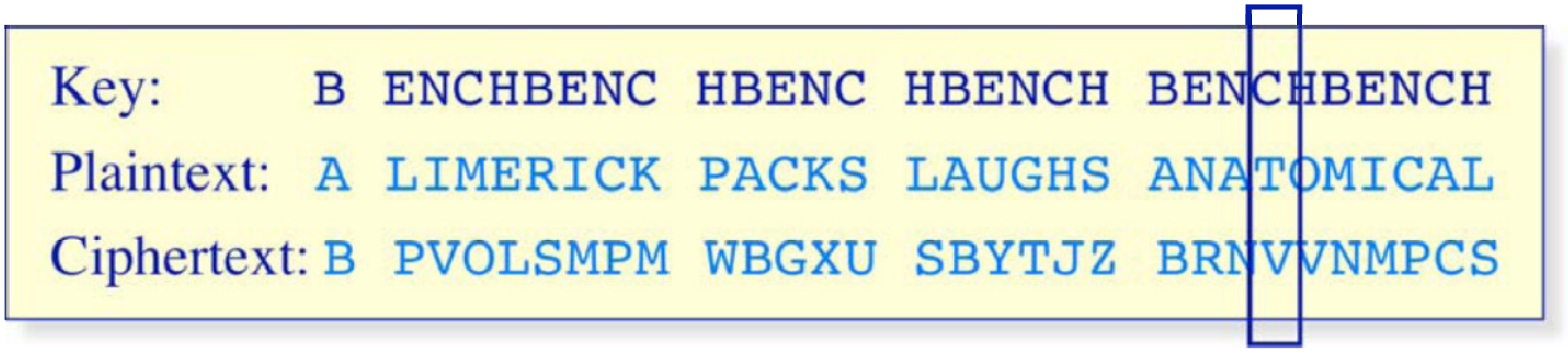


Letter ◆	Relative frequency in the English language ◆
a	8.167%
b	1.492%
c	2.782%
d	4.253%
e	12.702%
f	2.228%
g	2.015%
h	6.094%
i	6.966%
j	0.153%
k	0.772%
l	4.025%
m	2.406%
n	6.749%
o	7.507%
p	1.929%
q	0.095%
r	5.987%
s	6.327%
t	9.056%
u	2.758%
v	0.978%
w	2.360%
x	0.150%
y	1.974%
z	0.074%



Poly-alphabetic substitution ciphers - Vigenere

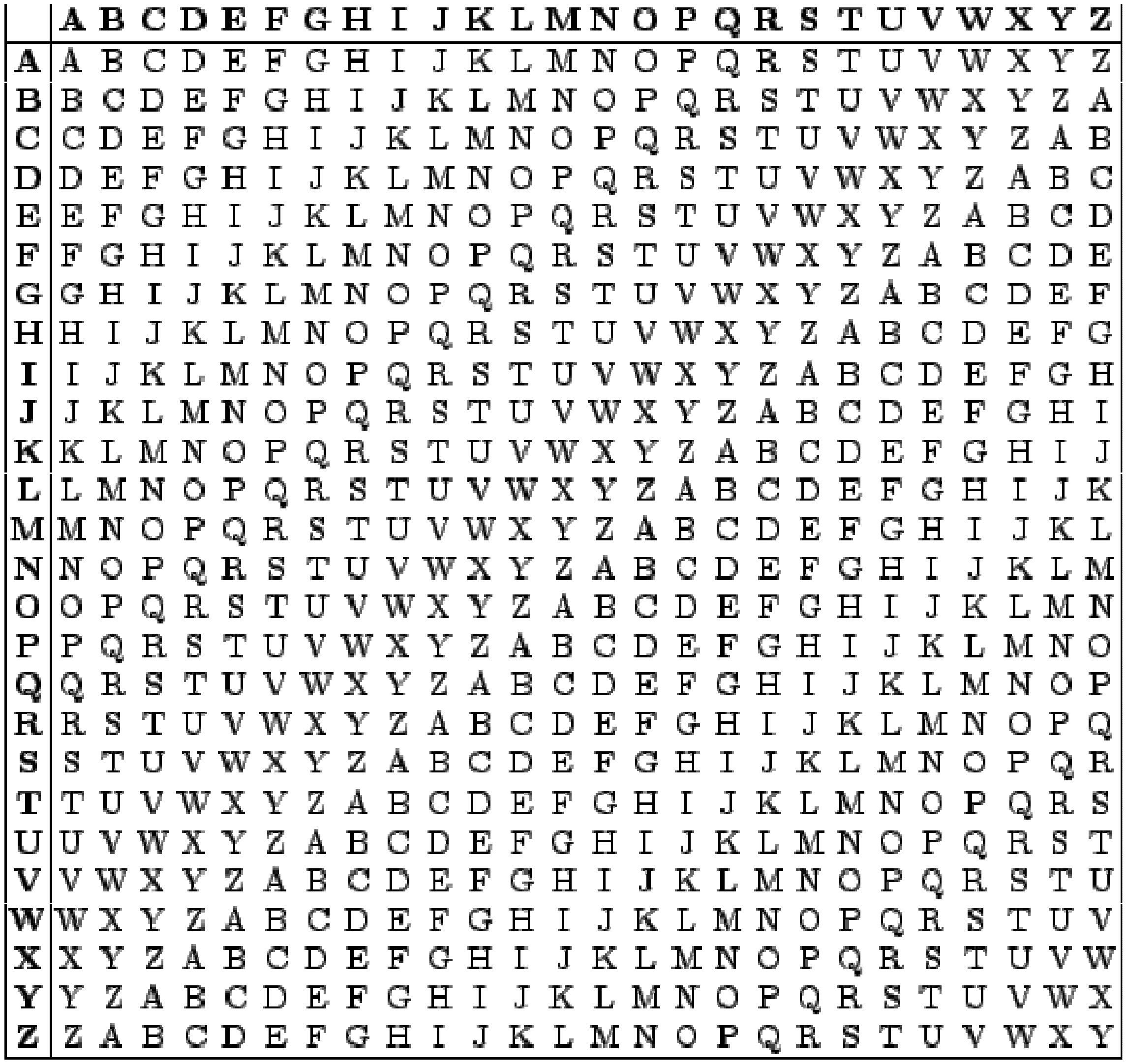
- Consists of multiple Caesar ciphers based on a codeword.
- Consider the word “BENCH” as the key (codeword).
- Apply each character in the key in sequence to the plaintext as in Caesar cipher:



Key:	B	ENCHBENC	HBENC	HBENCH	BENCHBENCH
Plaintext:	A	LIMERICK	PACKS	LAUGHS	ANATOMICAL
Ciphertext:	B	PVOLSMPM	WBGXU	SBYTJZ	BRNVVNMPCS

- Easier to do with a table:

The Vigenere table



	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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

Transposition ciphers

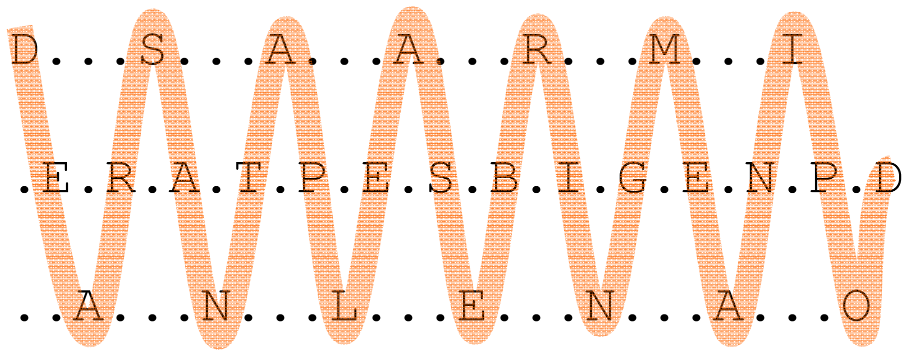
- Transposition ciphers do not change any of the symbols used to represent the clear-text.
  - Symbols are only rearranged

- If intercepted, the text appears readable, but scrambled.

- Some examples:
  - **`reeb sdder ekil i`**
    - Read backwards: `i like redds beer`
  - **`epgniusn rae omtsyl aft`**
    - Switch every two adjacent letters: `penguins are mostly fat`

Transposition ciphers – “rail fence” ciphers

- Try decoding the following message:
- Now try reading it like this:


D . . . S . . . A . . . A . . . R . . . M . . . I
. E . R . A . T . P . E . S . B . I . G . E . N . P . D
. . A . . . N . . . L . . . E . . . N . . . A . . . O

- This is called a “rail fence” cipher.
  - The key of such a cipher is the number of lines needed to represent and decode it.
  - Here, the key is 3.
- Transposition methods are still used by modern algorithms, like DES and 3DES.
  - …so it isn’t as childish as it looks ☺

Product ciphers

- Ciphers based on just transpositions or just substitutions are not secure enough.

- Consider using several ciphers in sequence:
  - Two substitutions make a more complex substitution
  - Two transpositions make a more complex transposition
  - A substitution followed by a transposition makes a much more complex cipher.

- The last one is the bridge from classical to modern ciphers.

Cryptanalysis methods

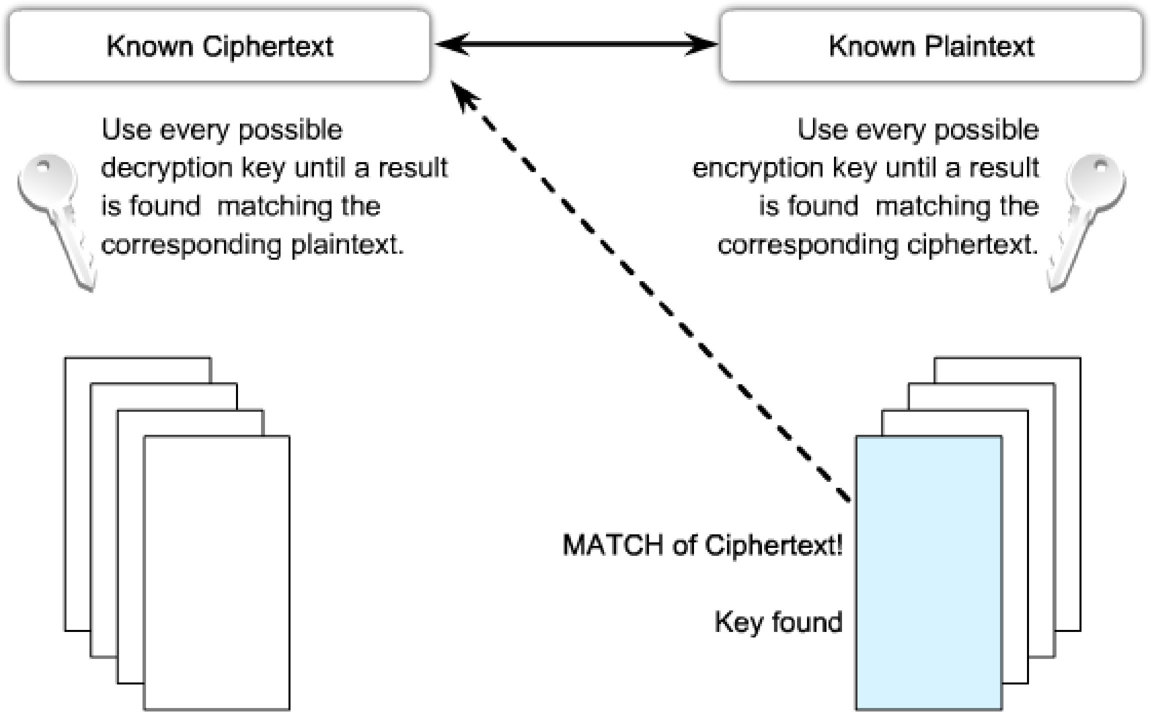
- Brute-force attacks
  - Attacker tries every possible key with the known decryption algorithm and, eventually, one of them will work.
  - **All encryption algorithms are vulnerable to brute-force.**
  - Modern-day cryptography's objective is to have a possible number of keys large enough that is takes too much money and time to run a brute-force attack.
- Ciphertext-only attacks
  - The attacker has the ciphertext of several messages, encrypted with the same key and algorithm.
  - The attacker knows nothing about the plaintext.
  - Statistical analysis can be used to deduce the encryption key.
  - Modern algorithms produce pseudorandom outputs that are resistant to statistical analysis.

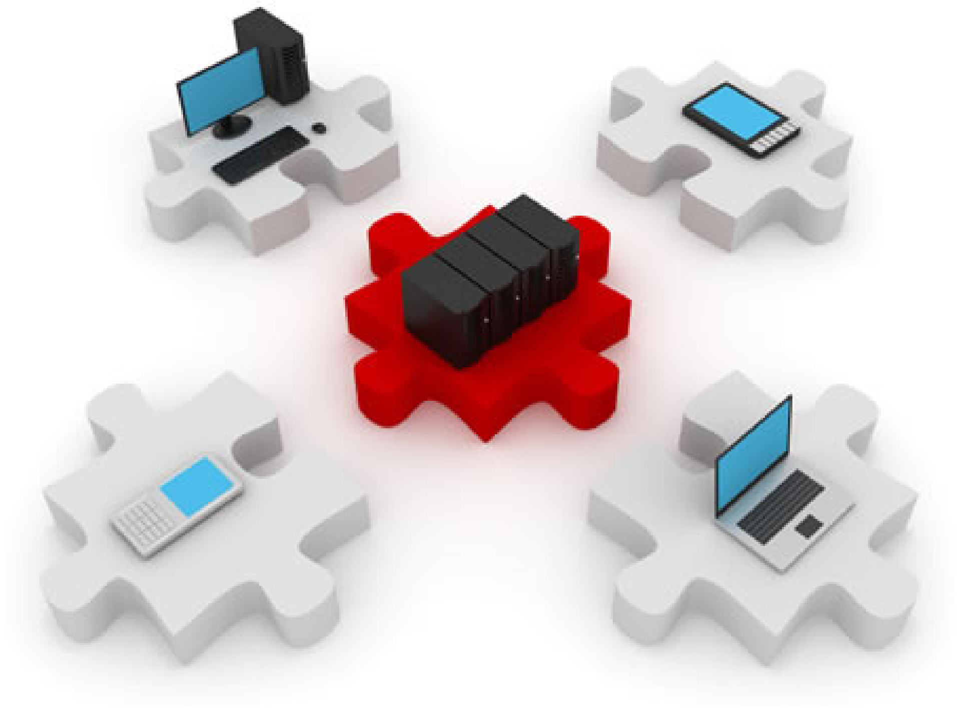
Cryptanalysis methods continued

- Chosen-plaintext attack
  - The attacker can test encryption on any given plaintext.
  - The attacker can find information about the key much faster.
  - Unlikely to have access to the ciphertext AND the corresponding desired plaintext as well.
- Chosen-ciphertext attack
  - The attacker can choose different ciphertexts to be decrypted and has access to their plaintexts.
  - The attacker can also find information about the key much faster.
  - Just as unlikely as above.

Cryptanalysis methods continued

- “Meet-in-the-middle” method.
- The attacker knows a piece of both the ciphertext and the plaintext.

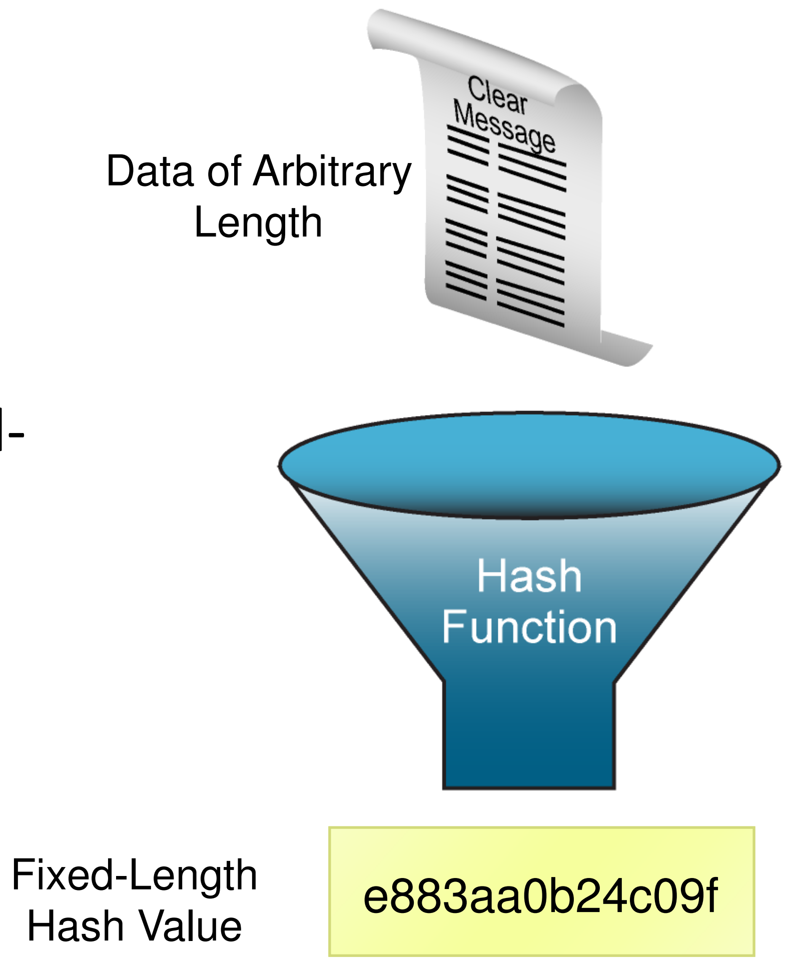

Known Ciphertext
Known Plaintext
Use every possible decryption key until a result is found matching the corresponding plaintext.
Use every possible encryption key until a result is found matching the corresponding ciphertext.
MATCH of Ciphertext!
Key found



Data integrity and authenticity

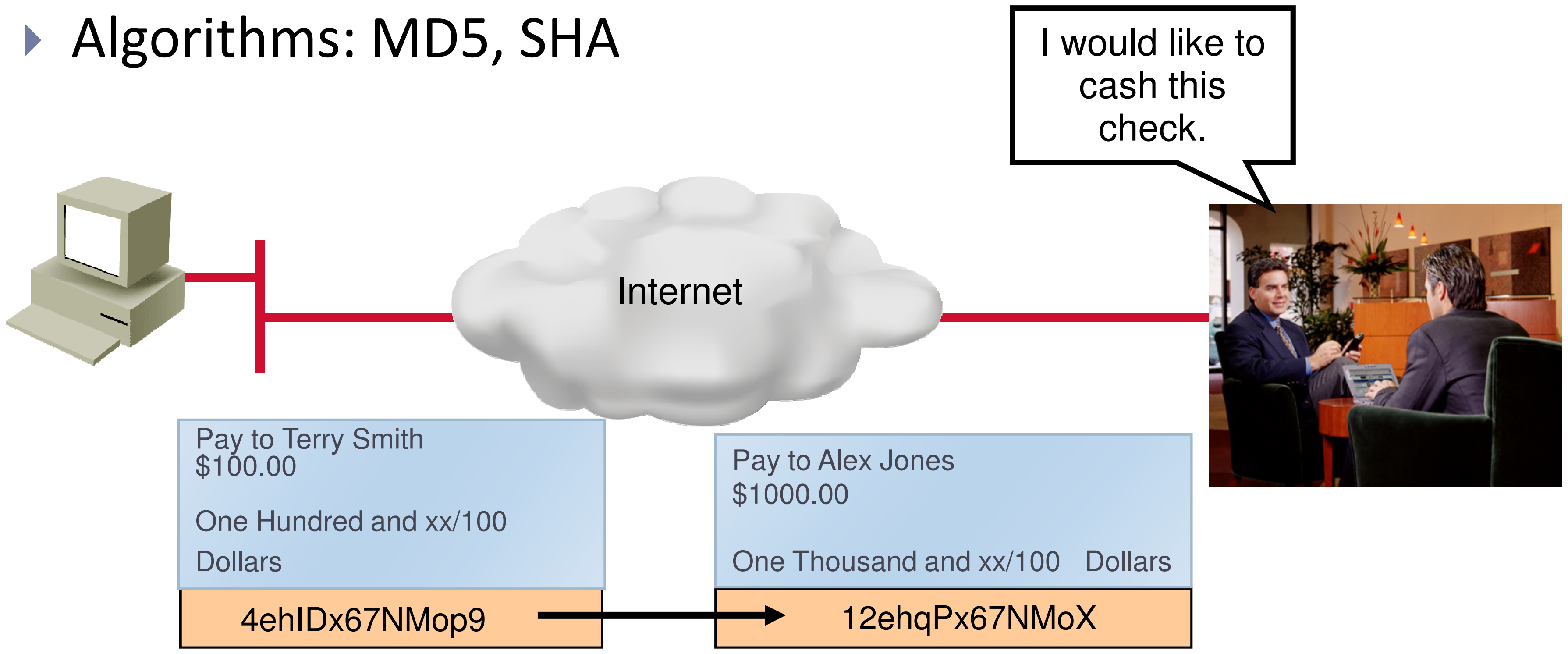
Cryptographic hashes

- Hashes are used to ensure integrity.
- Hashes are (must be) one-way functions.
- The hash function hashes an arbitrary-length data set into a fixed-length value.
- The hash value is also known as:
  - Digest value
  - Message digest
  - Fingerprint



Hashing in action

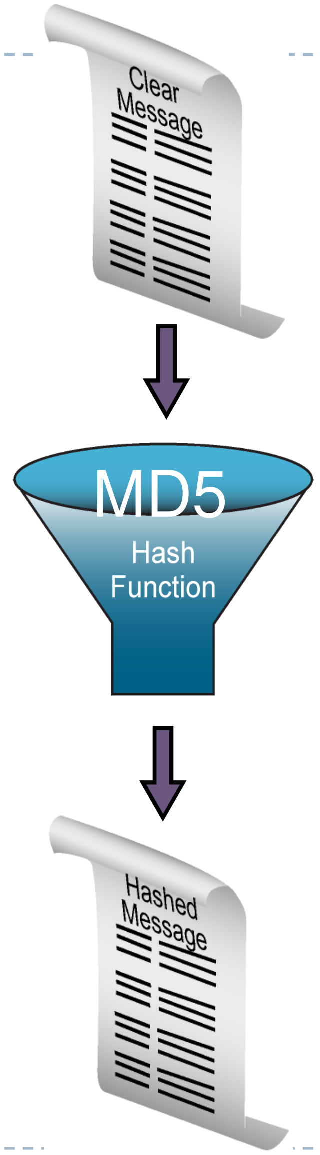
- Vulnerable to man-in-the-middle attacks.
- Hashes DO NOT provide security.
- Algorithms: MD5, SHA



Match = No changes
No match = Alterations

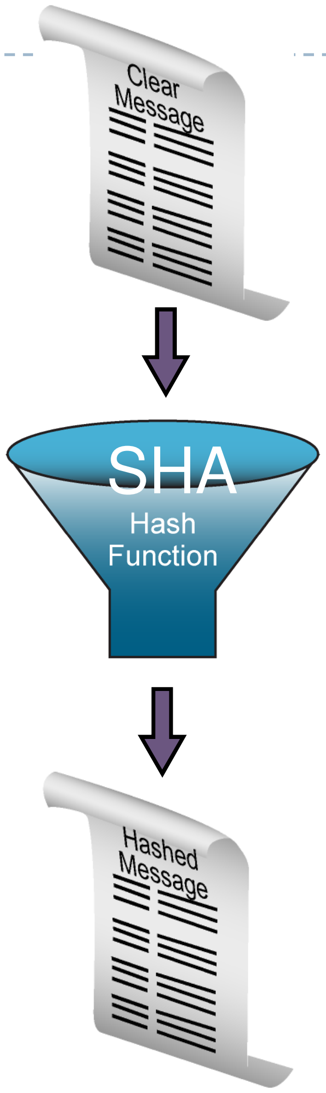
MD5

- Message-Digest 5
- MD5 is a widely-spread hashing algorithm.
- One-way function
  - Hash is easy to compute.
  - Unable to recover original data from hash.
- Complex sequence of simple binary operations (XOR, shifting, etc).
- Produces a sequence of 128 bits.

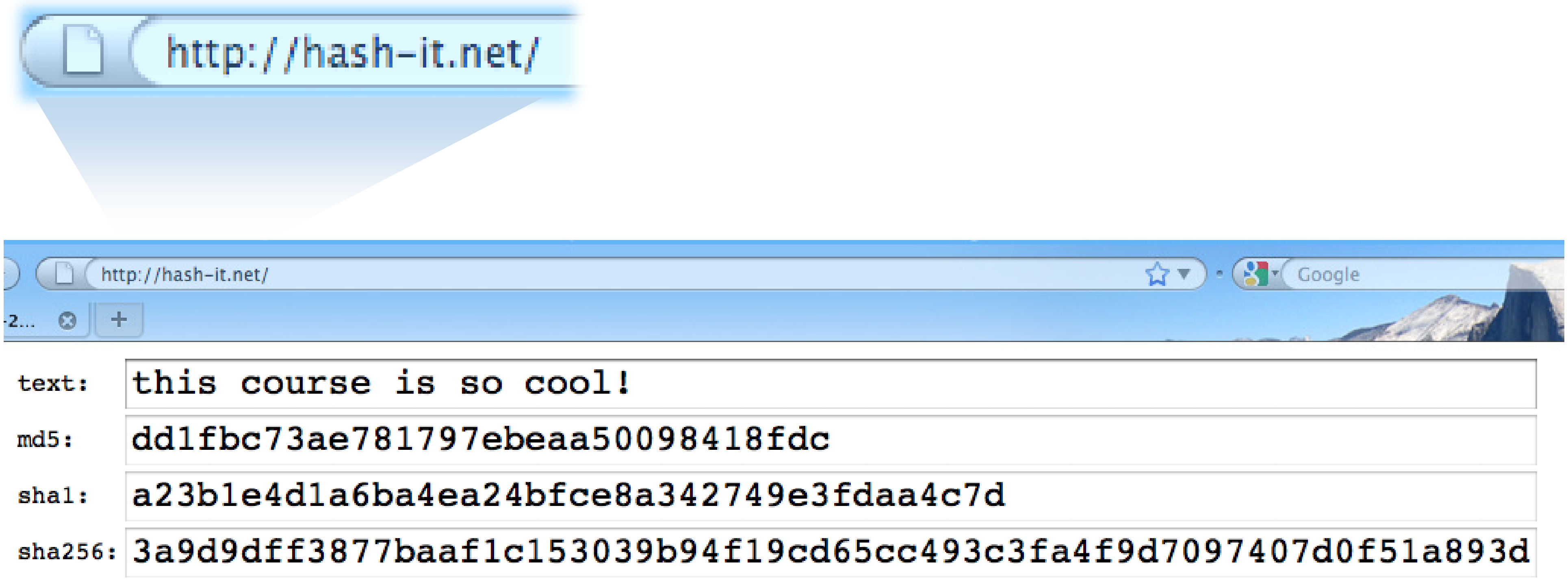


SHA

- Secure Hash Algorithm
- Similar to MD5
  - Accepts no more than 2^{64} bits of input
  - Returns an 160-bit message digest
- Slower than MD5
- SHA-1 is a revision that corrected an unpublished flaw in the original SHA algorithm.
- SHA-224, SHA-256, SHA-384 and SHA-512 are newer and more secure versions of SHA and are collectively known as SHA-2.



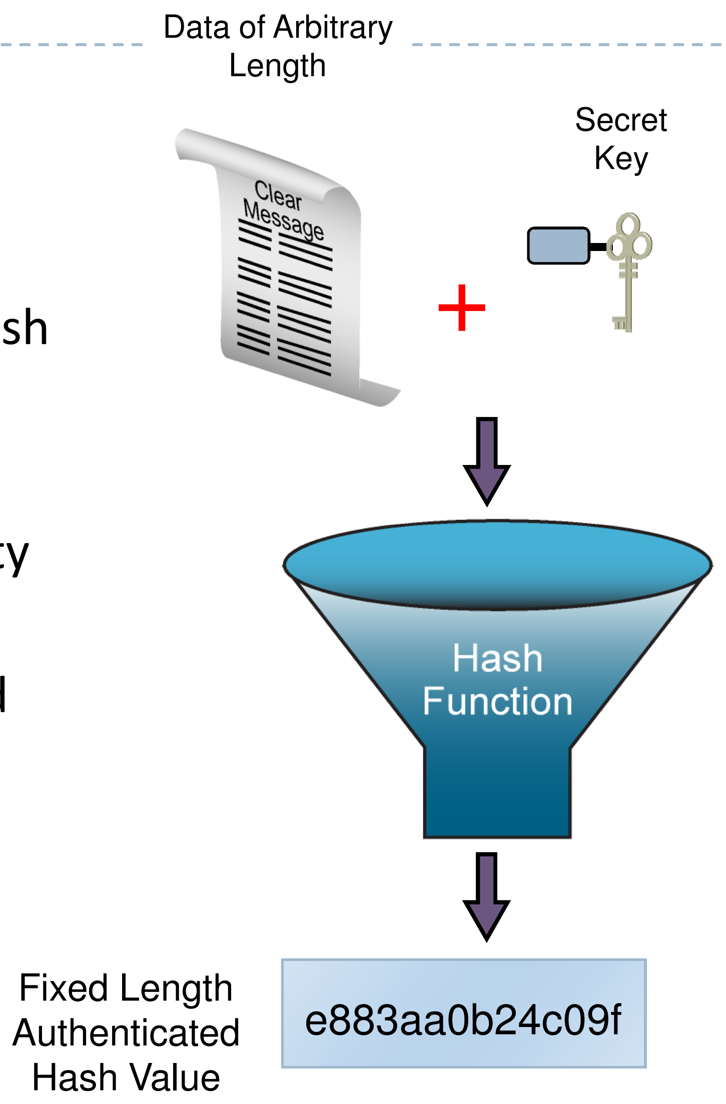
Try hash-it.net



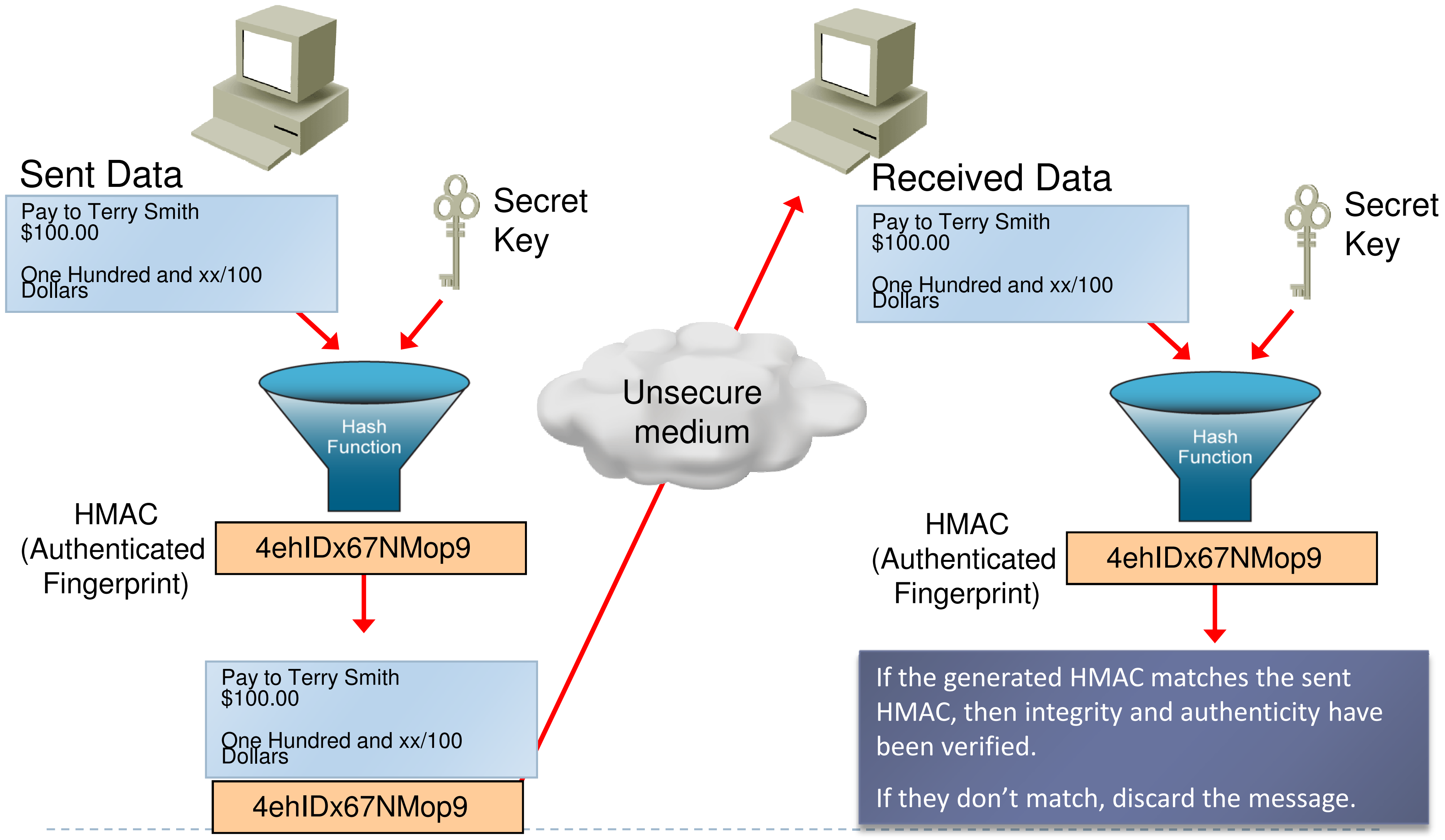
```
text:    this course is so cool!
md5:     dd1fbc73ae781797ebeaa50098418fdc
sha1:    a23b1e4d1a6ba4ea24bfce8a342749e3fdaa4c7d
sha256:  3a9d9dff3877baaf1c153039b94f19cd65cc493c3fa4f9d7097407d0f51a893d
```

HMAC

- HMAC = Hash-based Message Authentication Code
- Calculates a MAC using a cryptographic hash function along with a secret key.
- Validates the integrity AND the authenticity of the message.
- Any message-digest algorithm can be used (MD5, SHA-1).
  - Derived algorithms are:
    - HMAC-MD5
    - HMAC-SHA1



HMAC example

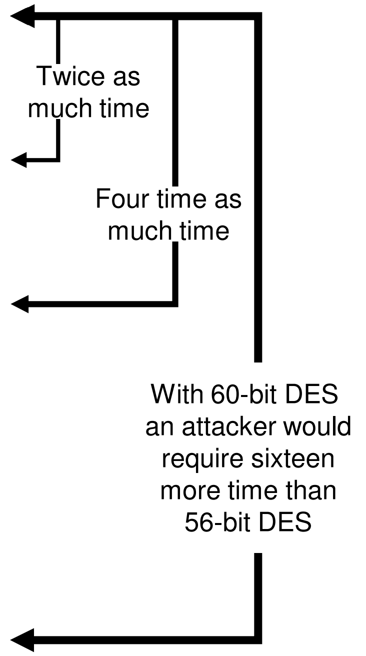


If the generated HMAC matches the sent HMAC, then integrity and authenticity have been verified.

If they don't match, discard the message.

Key space

DES Key	Keyspace	# of Possible Keys
56-bit	2^{56} 11111111 11111111 11111111 11111111 11111111 11111111 11111111	72,000,000,000,000,000
57-bit	2^{57} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 1	144,000,000,000,000,000
58-bit	2^{58} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11	288,000,000,000,000,000
59-bit	2^{59} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 111	576,000,000,000,000,000
60-bit	2^{60} 11111111 11111111 11111111 11111111 11111111 11111111 11111111 1111	1,152,000,000,000,000,000


Twice as much time
Four time as much time
With 60-bit DES an attacker would require sixteen more time than 56-bit DES

- ✓For each bit added to the DES key, the attacker would require twice the amount of time to search the keyspace.
- ✓Longer keys are more secure but are also more resource intensive and can affect throughput.

Types of keys

	Symmetric Key	Asymmetric Key	Digital Signature	Hash
Protection up to 3 years	80	1248	160	160
Protection up to 10 years	96	1776	192	192
Protection up to 20 years	112	2432	224	224
Protection up to 30 years	128	3248	256	256
Protection against quantum computers	256	15424	512	512

- ✓ Calculations are based on the fact that computing power will continue to grow at its present rate and the ability to perform brute-force attacks will grow at the same rate.
- ✓ Note the comparatively short symmetric key lengths illustrating that symmetric algorithms are the strongest type of algorithm.

Key management

Nowadays, an automatic process. Uses random numbers to minimize prediction.

Key Generation

Certain keys are weaker than others. They are regenerated if found (Caesar keys 0 and 25 do not encrypt).

Key Verification

Key Exchange

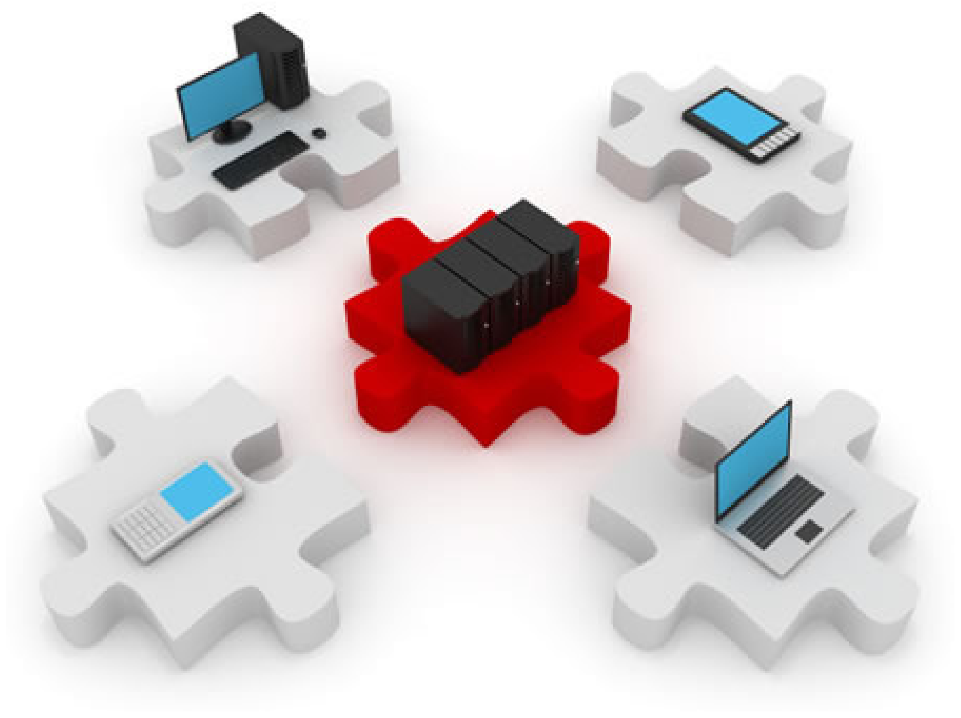
The method used for exchanging keys over an unsecure medium must be secure.

Key Storage

If keys are stored in clear text, they can be sent as hashes. If they are stored as hashes, they must be sent in clear text.

Key Revocation and Destruction

Revocation notifies all interested parties that a certain key has been compromised and should no longer be used.



Data Confidentiality

OSI-layered security approach

- Cryptographic algorithms can be implemented at several layers of the OSI model:
  - Data link layer can be encrypted using proprietary link-encrypting devices.
  - Network layer protocols, like IPsec, provide network layer confidentiality.
  - SSL (Secure Sockets Layer) and TLS (Transport Layer Security) provide confidentiality for the transport and session layers.
  - Application-specific security algorithms protect data at application level.

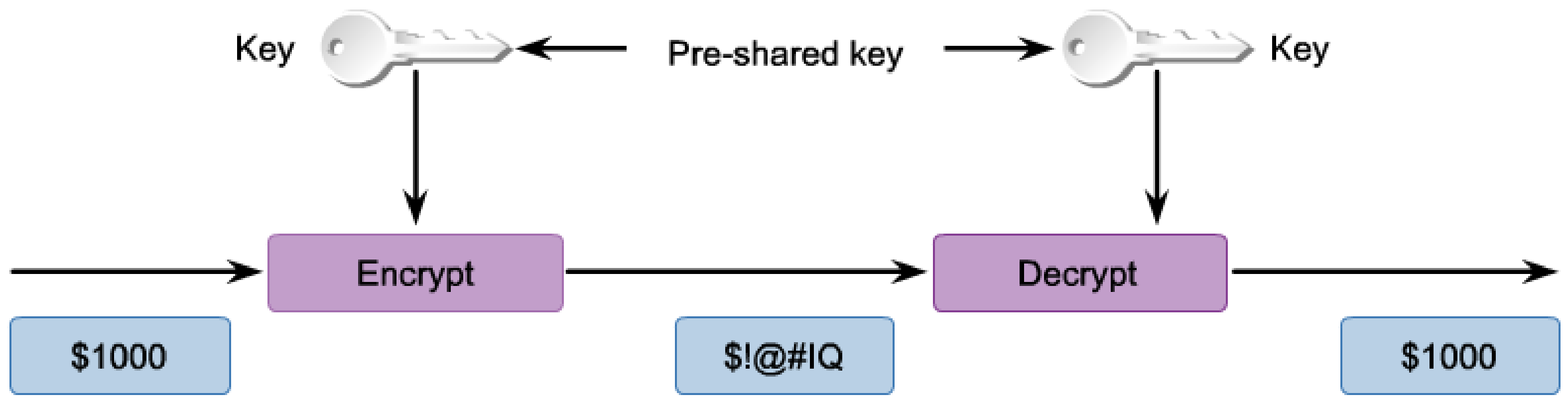
- Regardless the level, keys are involved and overheads, too.
  - Shorter keys offer faster processing but are less secure.
  - Longer keys take longer to process, but are more secure.

Encryption algorithms

- Cryptographic encryption can protect data using one of two methods:
  - Protect the algorithm
  - Protect the keys
- All modern cryptographic algorithms are public.
  - So keys are the ones that should be protected.

- There are two types of “key-protecting” encryption algorithms:
  - Symmetric-key algorithms
    - Use the same key to encrypt and decrypt.
    - The key must be pre-shared.
  - Asymmetric-key algorithms
    - Different keys for encryption and decryption
    - No need for keys to be pre-shared
    - Require very long keys
    - Several thousand times slower than symmetric-key algorithms

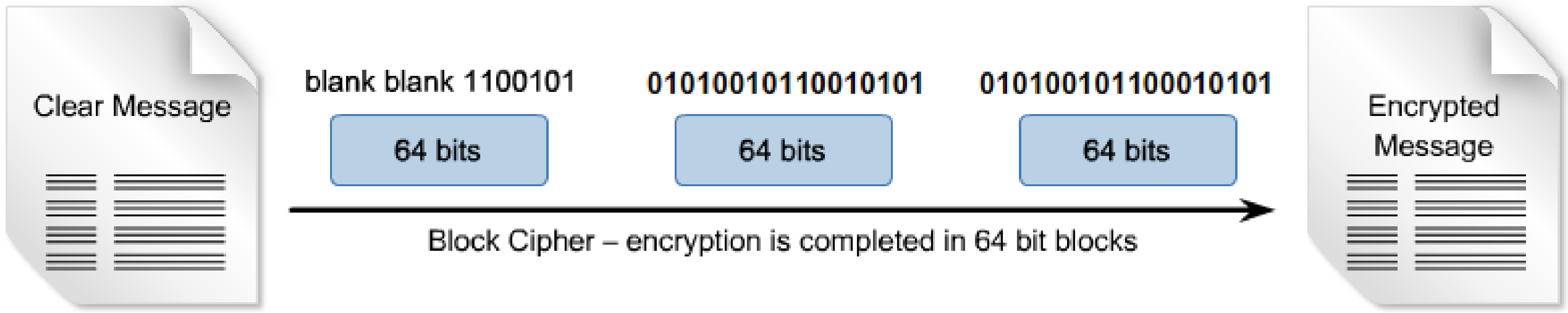
Symmetric algorithm



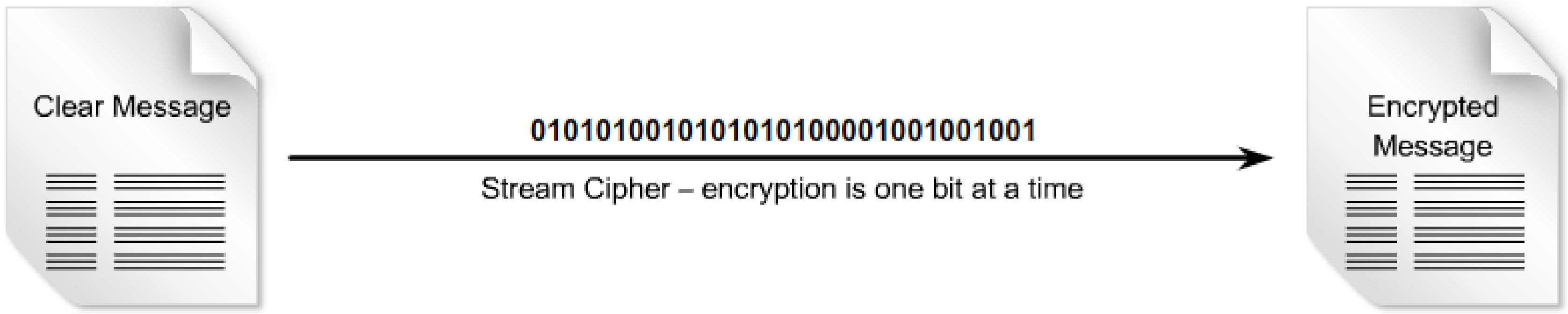
- Identical keys to a single “padlock”.
- The key has to be exchanged prior to sending any secret message (and has to remain secret).
  - The key usually has 80 to 256 bits in length
- Easy to process (less CPU-intensive calculations).
- The more encrypted material available, the easier to break.
  - Key needs to be changed periodically – how do you transmit it?
- Examples: DES, 3DES, AES, IDEA, Blowfish

Symmetric encryption ciphers

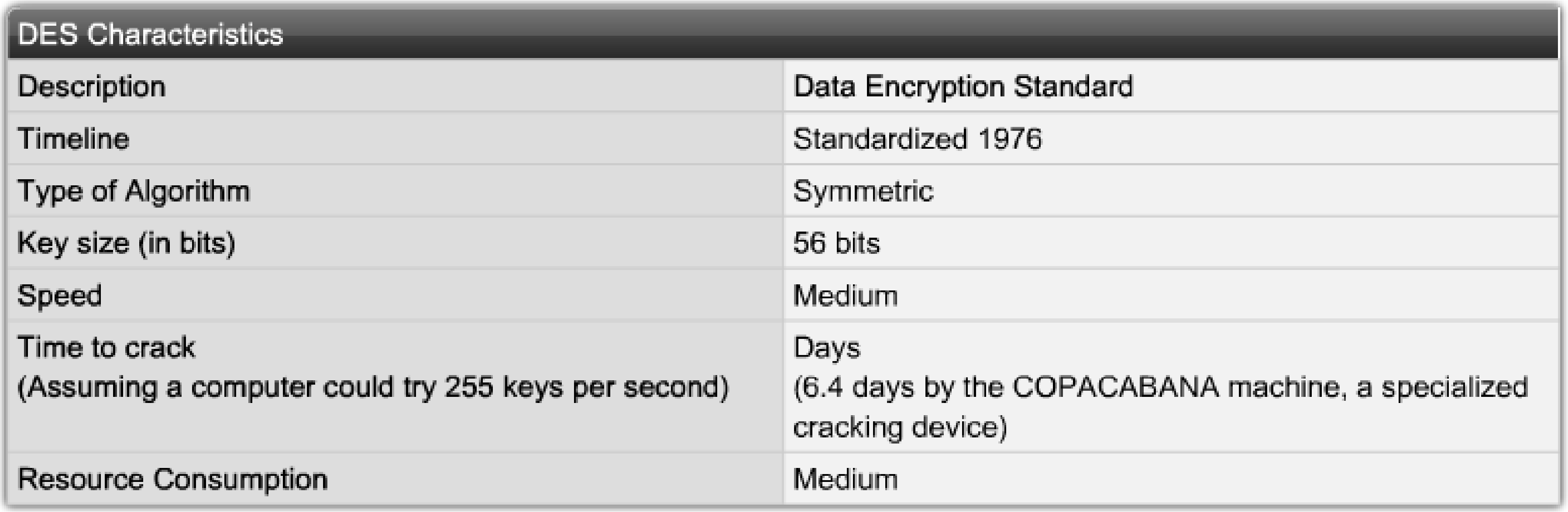
- Block ciphers: DES (64), AES (128)



- Stream ciphers: RC4, A5 (GSM call encryption)



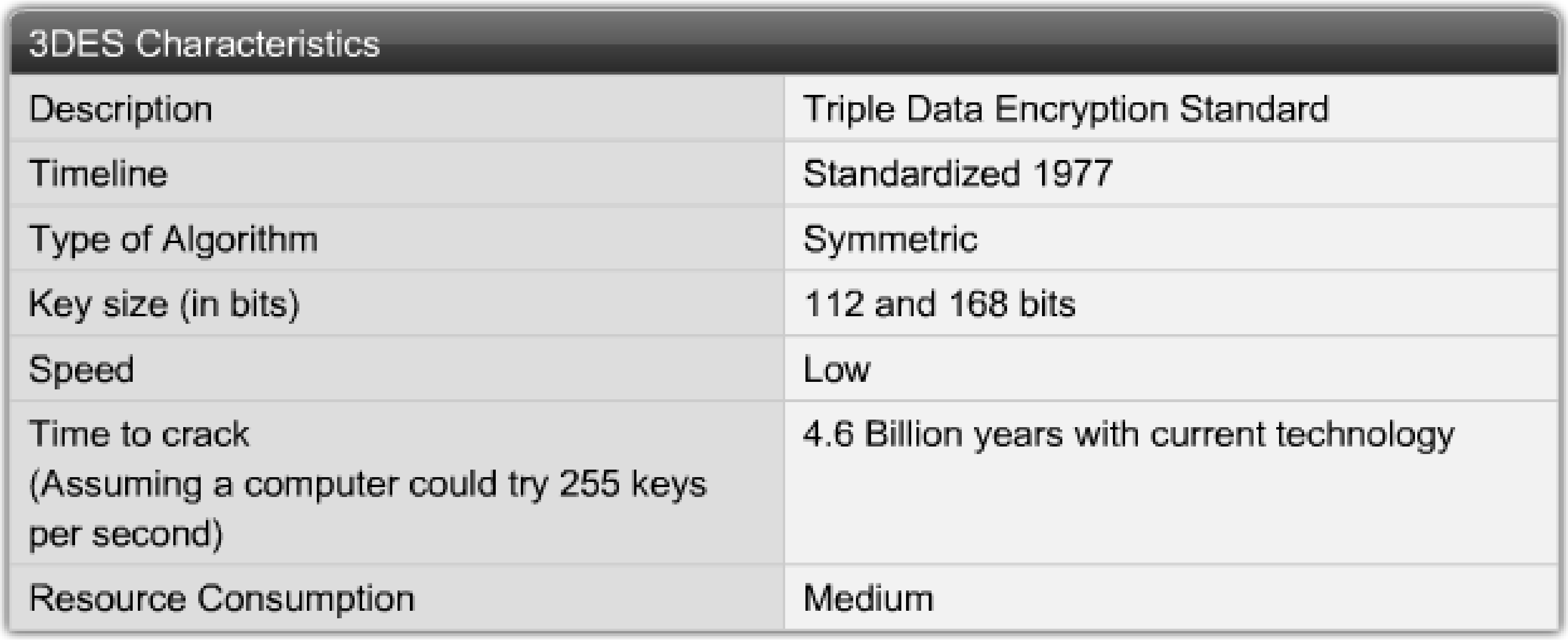
DES



DES Characteristics	
Description	Data Encryption Standard
Timeline	Standardized 1976
Type of Algorithm	Symmetric
Key size (in bits)	56 bits
Speed	Medium
Time to crack (Assuming a computer could try 255 keys per second)	Days (6.4 days by the COPACABANA machine, a specialized cracking device)
Resource Consumption	Medium

- DES uses a fixed-length key of 64 bits.
  - But only 56 bits are used for encryption, the others are used for parity.
- Has been replaced by 3DES because it can be “easily” cracked.
- Usable only if the key is changed very often.

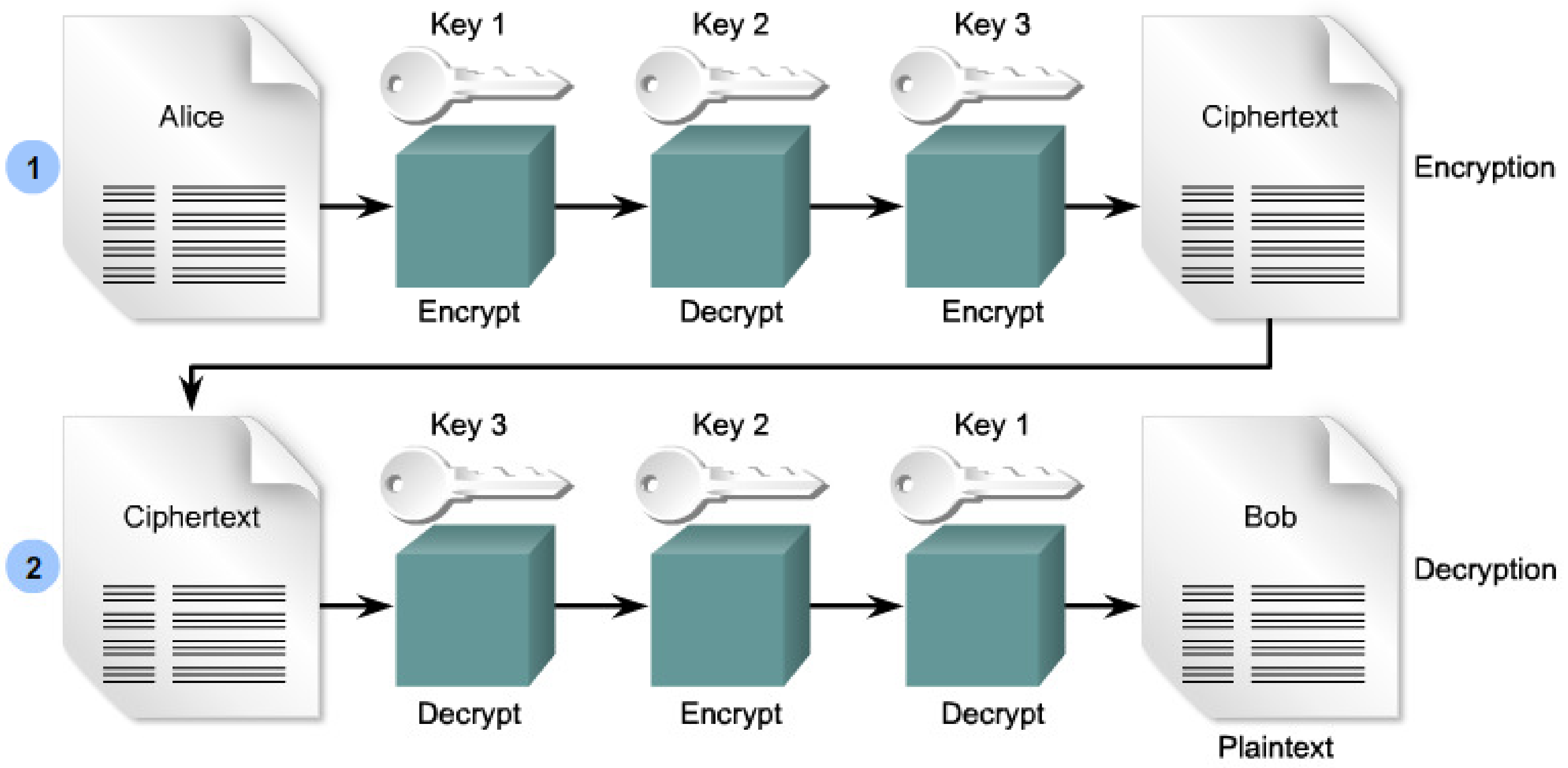
3DES



3DES Characteristics	
Description	Triple Data Encryption Standard
Timeline	Standardized 1977
Type of Algorithm	Symmetric
Key size (in bits)	112 and 168 bits
Speed	Low
Time to crack (Assuming a computer could try 255 keys per second)	4.6 Billion years with current technology
Resource Consumption	Medium

- Consists of applying DES three times in a row to a plaintext block.
- Considered trustworthy, hasn't been cracked in more than 35 years.
- Cisco's IPsec configurations can use DES and 3DES.

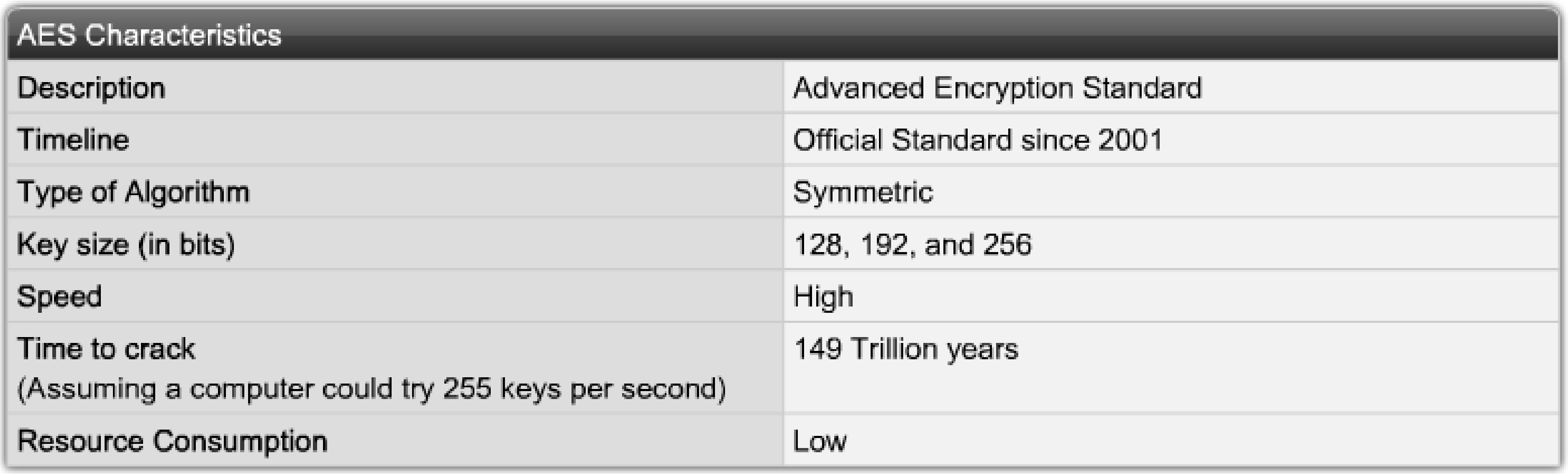
3DES in action



1. The clear text from Alice is encrypted using Key 1. That ciphertext is decrypted using a different key, Key 2. Finally that ciphertext is encrypted using another key, Key 3.

2. When the 3DES ciphered text is received, the process is reversed. That is, the ciphered text must first be decrypted using Key 3, encrypted using Key 2, and finally decrypted using Key 1.

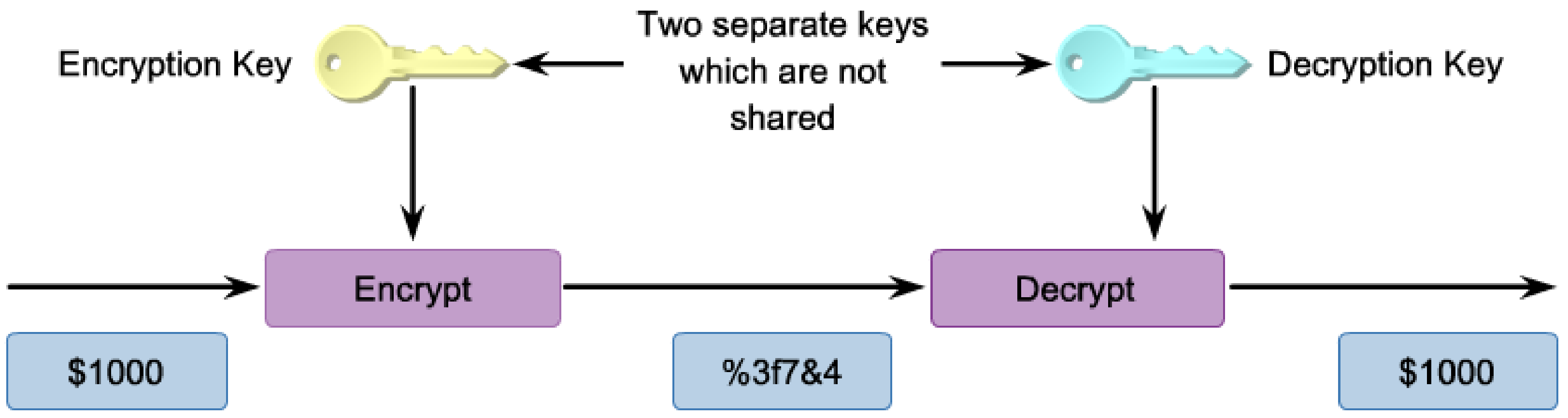
AES



AES Characteristics	
Description	Advanced Encryption Standard
Timeline	Official Standard since 2001
Type of Algorithm	Symmetric
Key size (in bits)	128, 192, and 256
Speed	High
Time to crack (Assuming a computer could try 255 keys per second)	149 Trillion years
Resource Consumption	Low

- The original algorithm was designed to work with any key and block length that were multiple of 32-bits.
- The older an algorithm is, the more it is trusted.
  - That is why 3DES is still “more trusted” than AES.

Asymmetric algorithm



- AKA “public-key” algorithms.
- The sender and the receiver to not share a secret key.
  - The key is usually 512 to 4096 bits in length.
- The decryption key cannot be derived from the encryption key.
- Slow algorithms – complex computations
- Examples: RSA, elliptic curves, Diffie-Helman

How do you “imagine” asymmetric encryption?

- We have Alice and Bob ☺
  - Of course, Alice (always) wants to send a message to Bob.
  - In a secure manner!
- In case of symmetric encryption:
  - Alice locks, Bob unlocks – simple!
  - But Alice has to tell Bob her key and we don’t want that.
- In case of asymmetric encryption:
  - Alice puts the message in a box.
  - Alice puts on her own padlock and sends to Bob.
  - Bob adds his own padlock and sends it to Alice.
  - Alice removes her padlock, sends back to Bob.
  - Bob removes his padlock, reads the message.
- No keys are exchanged, no clear text message is sent.
- Now translate this into mathematics! ☺

Same situation, towards cryptography

Alice, Bob agree on information Y

Alice computes A(Y)
Mails it to Bob

Bob computes B(Y)
Mails it to Alice

Alice computes A(B(Y))

Bob computes B(A(Y))

A(B(Y)) = B(A(Y)) = secret key

“Trudy” is listening and “hears” Y, A(Y), B(Y), but can’t compute the secret key.
A(Y) and B(Y) cannot be combined to create A(B(Y)).

Problem: how do you make A(B(Y)) = B(A(Y))?

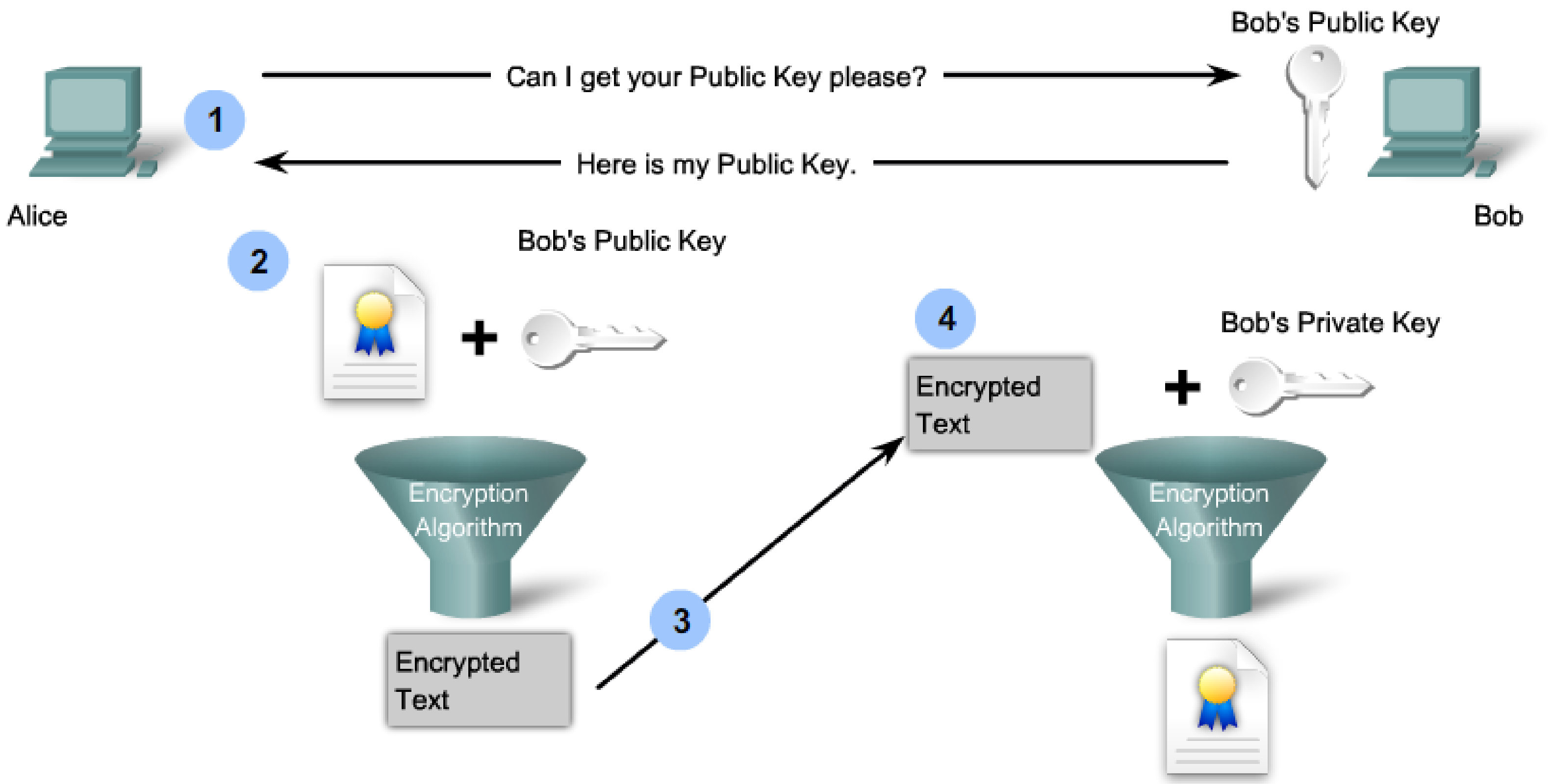
Finally, some math!

- The Diffie-Hellman key exchange protocol
  - Allows Alice and Bob to establish a secret key
- Two numbers have to be exchanged between Alice and Bob: a prime number ($p=23$) and a base ($g=5$).

- Alice chooses a secret integer ($a=6$) and sends Bob $A = g^a \bmod p$
  - $A = 5^6 \bmod 23 = 8$
    - A is Alice's public key and a is her private key.
- Bob chooses a secret integer ($b=15$) and sends Alice $B = g^b \bmod p$
  - $B = 5^{15} \bmod 23 = 2$
    - B is Bob's public key and b is his private key.
- Alice computes secret = $B^a \bmod p$
  - $19^6 \bmod 23 = \mathbf{2}$
- Bob computes secret = $A^b \bmod p$
  - $8^{15} \bmod 23 = \mathbf{2}$

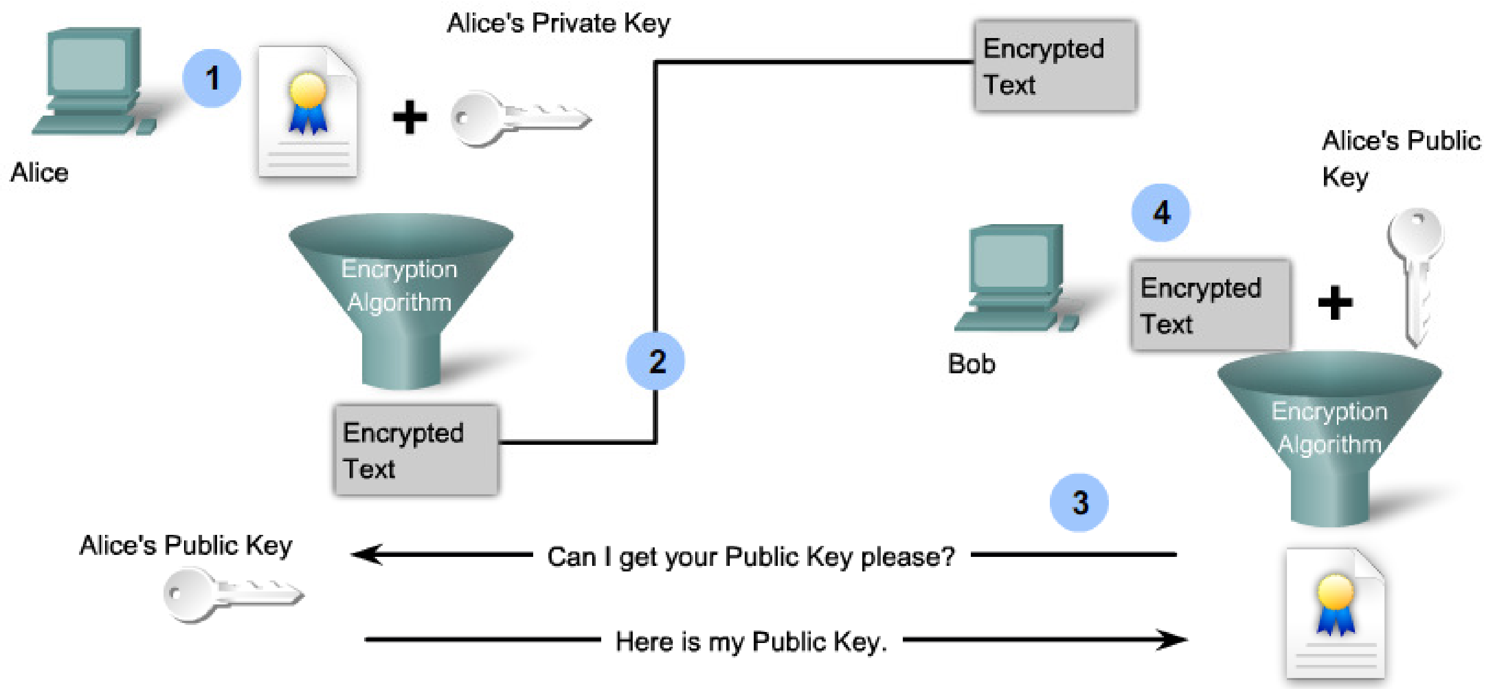
Green = public data
Red = private data

Asymmetric encryption provides confidentiality



- Bob’s public key is given to Alice.
- Alice encrypts a message for Bob with his public key.
- Only Bob’s private key can decrypt the message.

Asymmetric encryption provides authentication

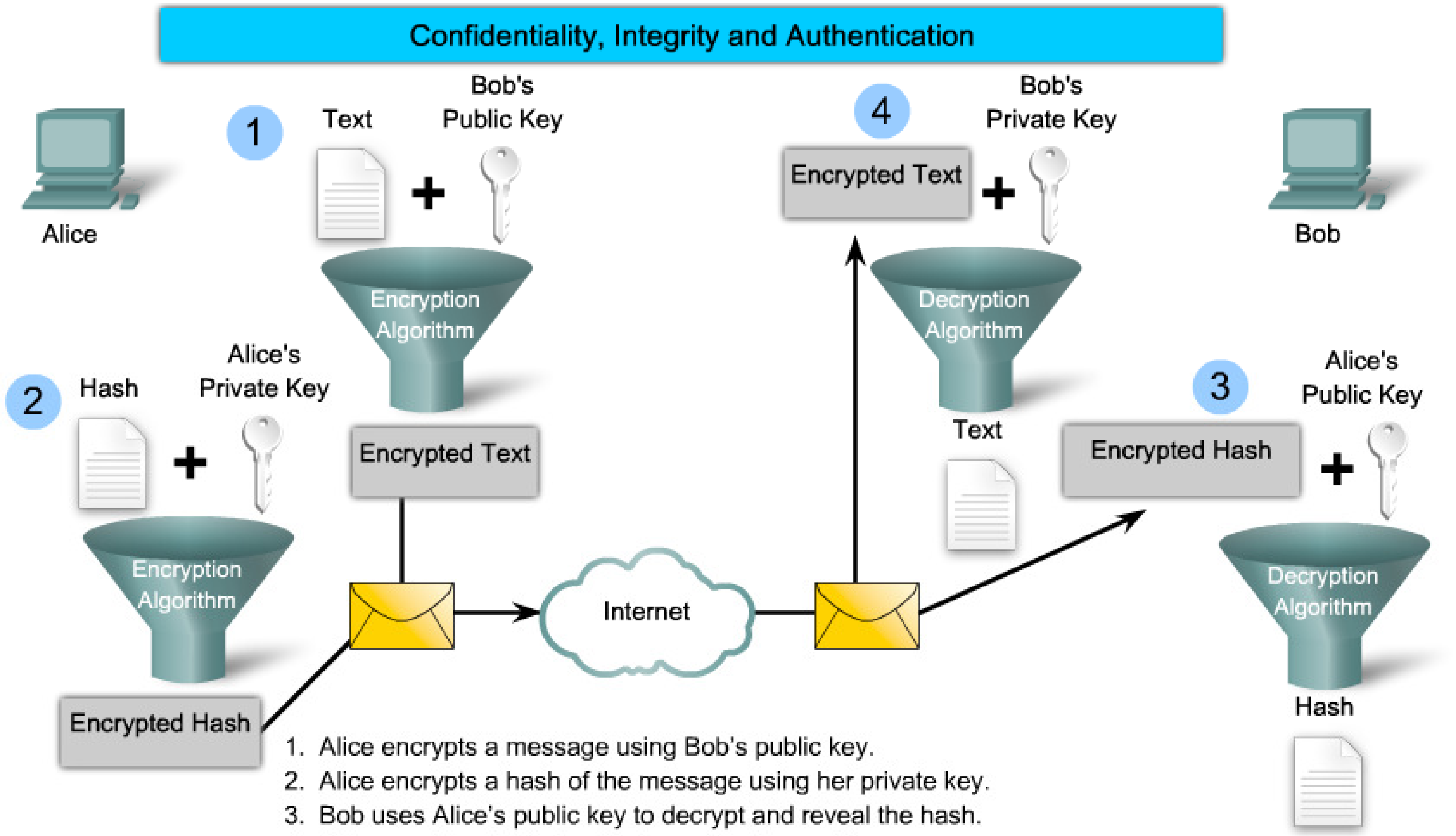


- Alice encrypts a message for Bob with her private key.
- Bob requests Alice's public key.
- If Bob successfully decrypts the message with Alice's public key then he can be sure the message really came from Alice.

Summing it all up – asymmetric encryption

- How does a “secure” data transfer look now?
  - It requires confidentiality, integrity and authentication.
- Alice -> Bob
  - Alice encrypts the message using Bob’s public key
    - Bob decrypts using his private key and thus confidentiality is ensured.
  - Alice calculates a hash of the message and attaches it to the message.
    - Bob calculates the hash again and if the two hashes match, integrity is ensured.
  - Alice encrypts the hash before attaching it, using her private key.
    - Bob decrypts the hash using Alice’s public key, which ensures authentication.

Let’s draw a complicated picture of it

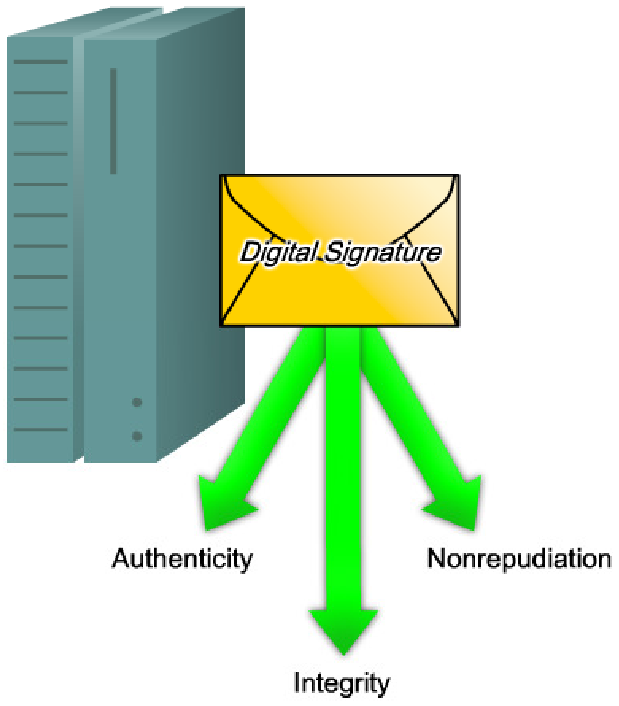


1. Alice encrypts a message using Bob's public key.
2. Alice encrypts a hash of the message using her private key.
3. Bob uses Alice's public key to decrypt and reveal the hash.
4. Bob uses his private key to decrypt and reveal the message.

Digital signatures

Digital signatures

- A digital signature provides three services:
  - Authenticity – they prove that a certain party has signed the data in question.
  - Integrity – they guarantee that the data has not changed from source to destination.
  - Non-repudiation – Nobody except the signing party can sign the message.
- Digital signatures algorithms: DSA, RSA



Nonrepudiation vs. HMAC

- With HMAC, authenticity and integrity is provided, too.
  - Integrity: a hash is attached to the message, which proves that the message was not modified in transit.
  - Authenticity: the hash is calculated using a shared secret known only by some.
  - Taking the data to a third party <u>does not</u> prove that the third party did not sent the message (the secret is shared).

- Non-repudiation is only provided by digital signatures.
  - A hash is calculated from the data and a private secret.
  - Taking the data to a third party <u>does</u> prove that the third party did not send the message (the secret is not shared).
  - The real sender of the message cannot deny (repudiate) the fact that it sent the message in the first place.

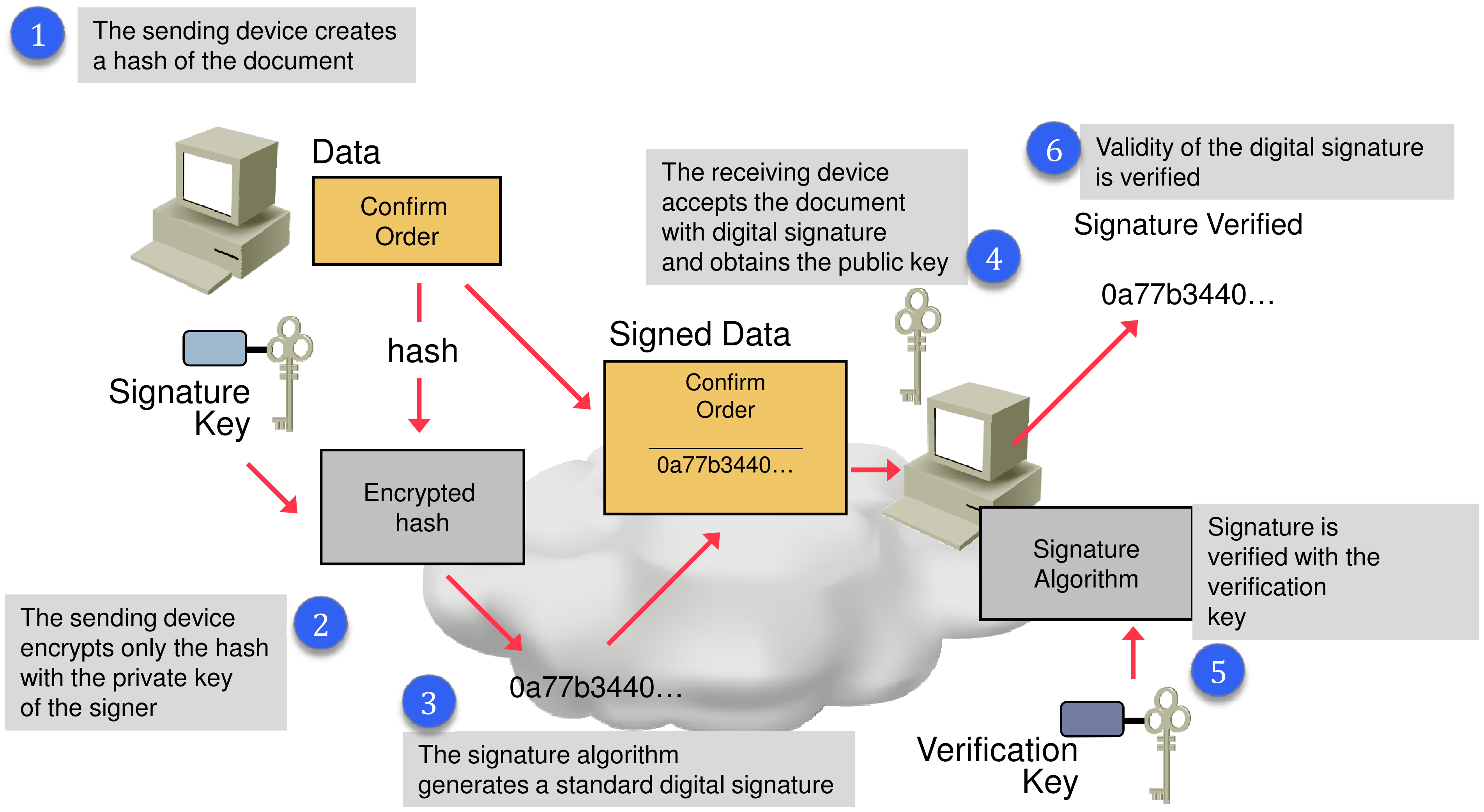
Properties of digital signatures

- Authentic and non-forgeable
  - No one else could have signed the document.
- Not reusable
  - The signature is part of the document and cannot be used with another document.
- Unalterable
  - After a document is signed, it cannot be altered.
- Cannot be repudiated
  - The signer cannot claim later that it did not sign it.

How does a digital signature work?

- The sending device (signer) creates a hash of the document.
- The sending device encrypts the hash with its own private key.
- The encrypted hash is appended to the document.
  - This hash is the signature.
- The receiving device (verifier) accepts the document and receives the public key of the sender.
- The receiving device decrypts the signature using the public key of the sending device.
- The receiving device calculates its own hash of the message and compares with the decrypted one.
  - If these hashes match, the document is authentic.
  - It has not been altered and has been signed by the assumed signer.

Still, HOW does a digital signature work?


1
The sending device creates a hash of the document
Data
Confirm Order
hash
Signature Key
Encrypted hash
The sending device encrypts only the hash with the private key of the signer
2
3
0a77b3440…
The signature algorithm generates a standard digital signature
Signed Data
Confirm Order
0a77b3440…
The receiving device accepts the document with digital signature and obtains the public key
4
6
Validity of the digital signature is verified
Signature Verified
0a77b3440…
Signature Algorithm
Signature is verified with the verification key
5
Verification Key

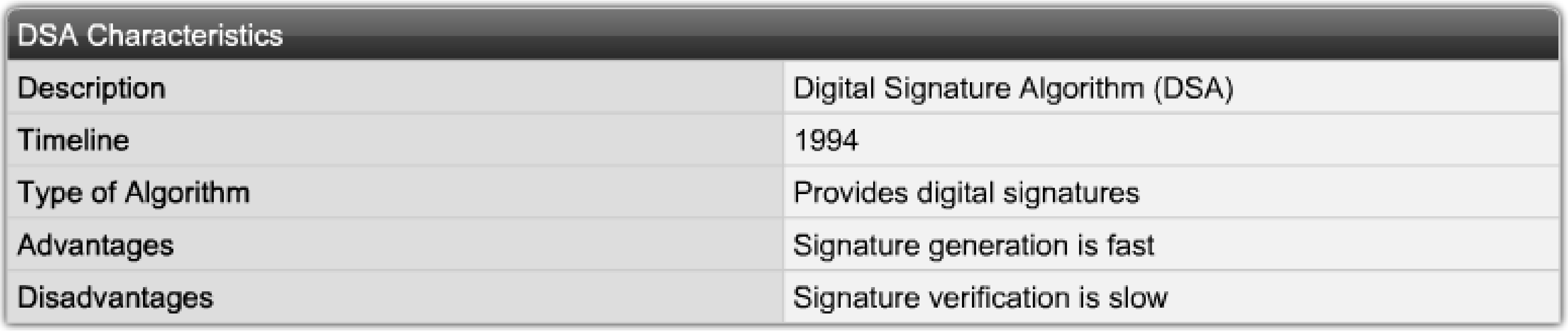
Digital signatures on software applications

- A signature of the package/application/executable can be checked before the application is installed/executed.

- Signing a code provides the following assurances:
  - The code has not been modified since released by the publisher.
  - The code is actually issued by the publisher (and not someone else).
  - The publisher can be held accountable for the software (this is non-repudiation).

- The only way to forge a digital signature is to steal the publisher’s private key.
- The user must obtain a public key to validate the signature.
  - Public keys are given to anyone (hence “public”…).

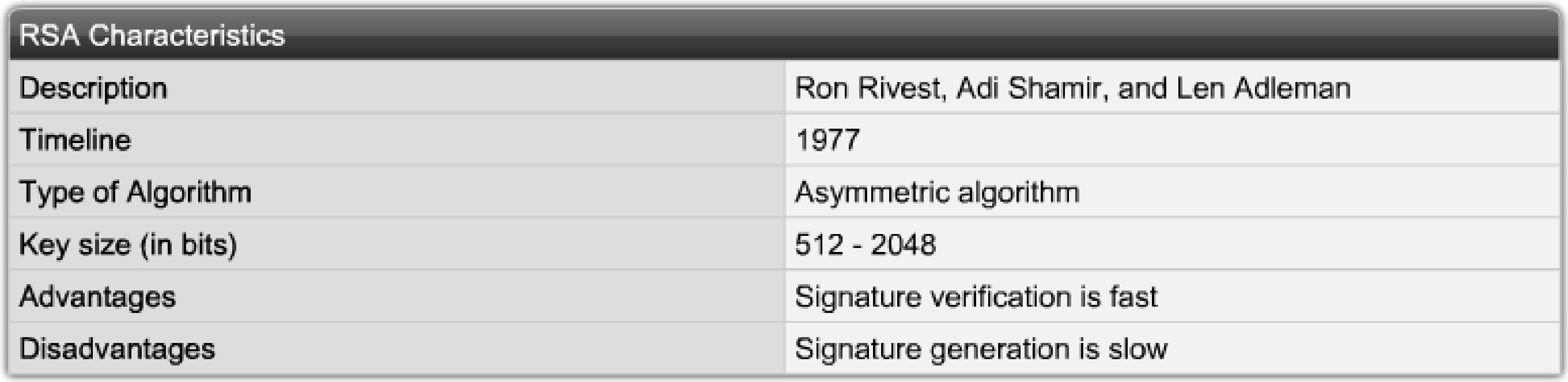
DSA



DSA Characteristics	
Description	Digital Signature Algorithm (DSA)
Timeline	1994
Type of Algorithm	Provides digital signatures
Advantages	Signature generation is fast
Disadvantages	Signature verification is slow

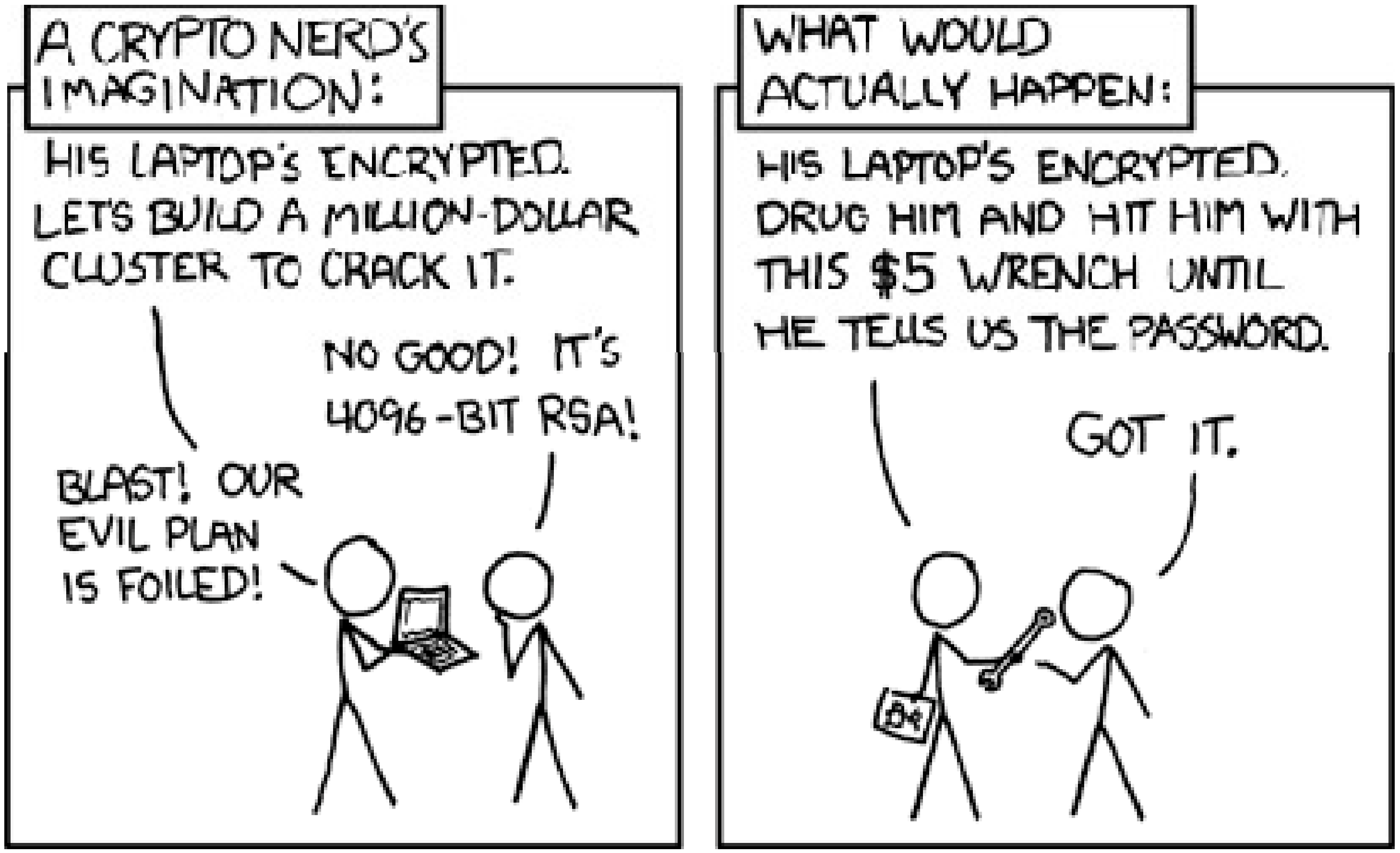
- United States Federal Government standard for digital signatures.
- Criticized because its slow signature verification algorithm.
- DSA signature generation is faster than DSA signature verification.

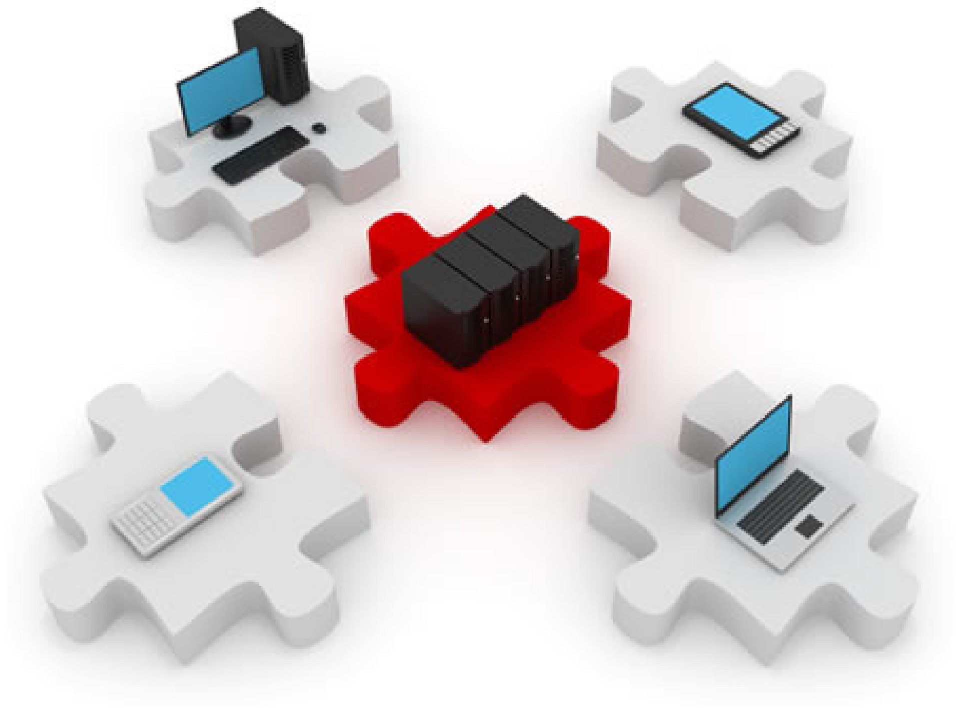
RSA



RSA Characteristics	
Description	Ron Rivest, Adi Shamir, and Len Adleman
Timeline	1977
Type of Algorithm	Asymmetric algorithm
Key size (in bits)	512 - 2048
Advantages	Signature verification is fast
Disadvantages	Signature generation is slow

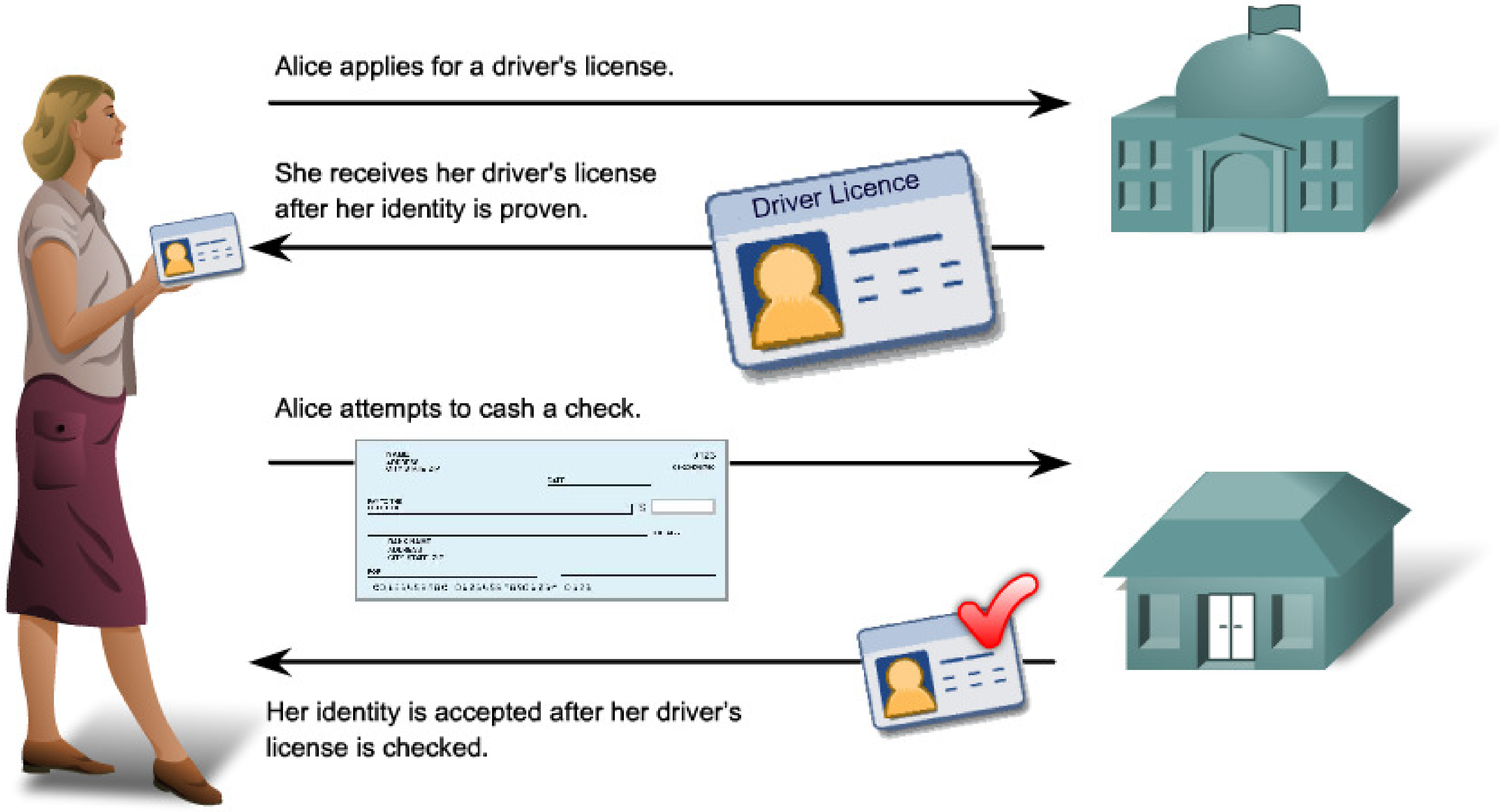
- Asymmetric algorithm used for signatures and encryption as well.
- Widely use in electronic commerce, flexible due to its variable key length.
- Faster signature verification than generation.
- Much slower than DES (symmetric encryption)
  - 100 times slower than DES in software
  - 1000-10000 times slower than DES in hardware
  - 15000 times slower than DES on Cisco routers


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.



Public Key Infrastructure (PKI)

Public key infrastructure similarity


Alice applies for a driver's license.
She receives her driver's license after her identity is proven.
Driver Licence
Alice attempts to cash a check.
Her identity is accepted after her driver's license is checked.

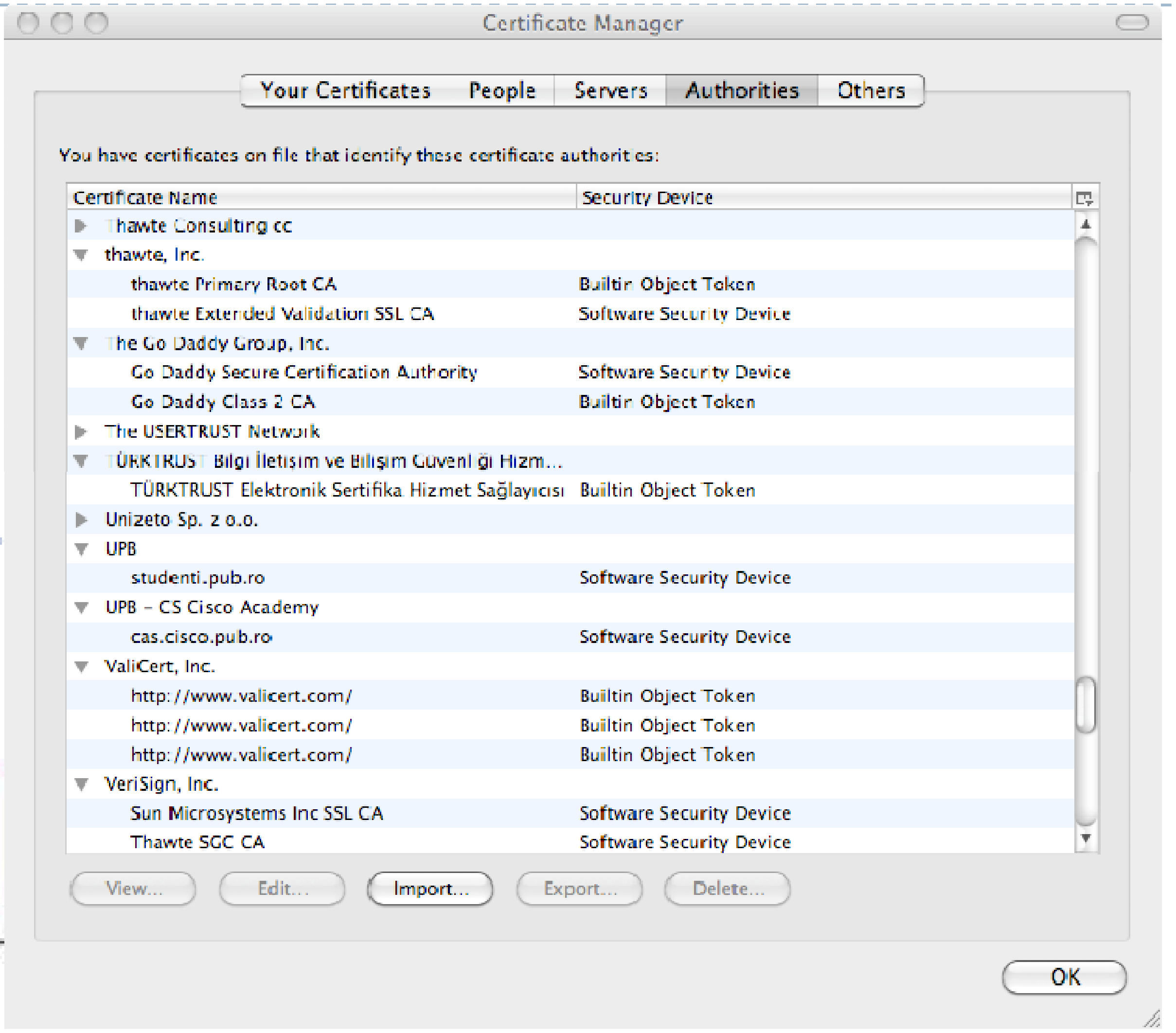
PKI facts

- Difficult to exchange authentication information between all peers, for every communication attempt.
  - If 10 individuals need to validate each other, 90 validations are required.
  - Adding an 11th member to the group would require another 20 validations.
- Peers can agree to accept a third, neutral party’s “opinion”.
- Peers only need to authenticate themselves with the “trusted” institution.

PKI terminology

- PKI
  - A service framework needed to support large scale public key-based technologies.
  - Consists of hardware, software, people, procedures, policies.
- Certificate
  - A document that binds together the identity of an entity (person or company) and its public key.
  - It has to be signed by a CA.
- CA
  - The Certificate Authority is a third party that signs the public keys of entities in a PKI-based system.

CA vendors – Firefox example



You have certificates on file that identify these certificate authorities:

Certificate Name	Security Device
▶ Thawte Consulting cc	
▼ thawte, Inc.	
thawte Primary Root CA	Builtin Object Token
thawte Extended Validation SSL CA	Software Security Device
▼ The Go Daddy Group, Inc.	
Go Daddy Secure Certification Authority	Software Security Device
Go Daddy Class 2 CA	Builtin Object Token
▶ The USERTRUST Network	
▼ TÜRKTRUST Bilgi İletişim ve Bilişim Güvenliği Hizm...	
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı	Builtin Object Token
▶ Unizeto Sp. z o.o.	
▼ UPB	
studenti.pub.ro	Software Security Device
▼ UPB – CS Cisco Academy	
cas.cisco.pub.ro	Software Security Device
▼ ValiCert, Inc.	
http://www.valicert.com/	Builtin Object Token
http://www.valicert.com/	Builtin Object Token
http://www.valicert.com/	Builtin Object Token
▼ VeriSign, Inc.	
Sun Microsystems Inc SSL CA	Software Security Device
Thawte SGC CA	Software Security Device

Levels of trust

- Certificates can be issued with different levels of trust:
  - Class 0: testing purposes, no checks made
  - Class 1: for individuals, mainly for authenticating e-mails
  - Class 2: for organizations, to prove identity
  - Class 3: servers and software signing
  - Class 4: online business transactions between companies
  - Class 5: private organizations and governmental security

- For example, a class 1 certificate might be issued simply by validating an e-mail address.
- A class 3 or 4 certificate might be issued only after the clients identify themselves in person, showing identification documents.

PKI keys

- Each entity can have two key pairs.
- The first pair is intended for encryption operations
  - A pair consists of a public and a private key.
  - The public key encrypts, the private key decrypts.
- The second pair is intended for signing digital operations
  - The private key signs, the public key verifies authenticity

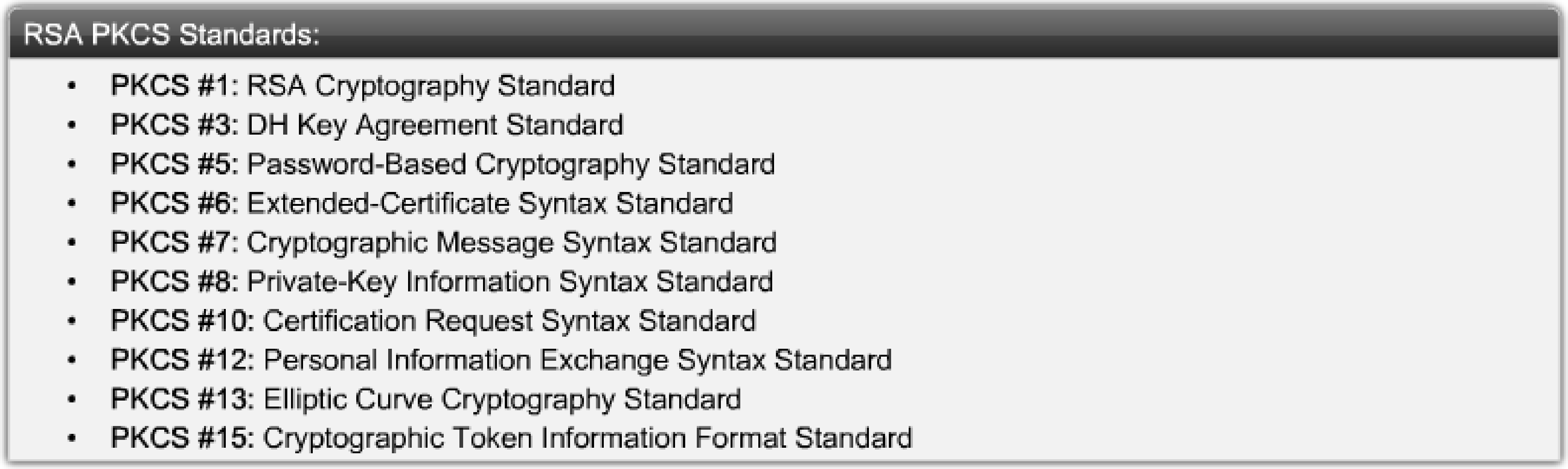
PKI standards

- The X509v3 standard defines the format of a digital certificate.

- Various applications implement certificates that adhere to the X509.v3 standard:
  - Web servers, for website authentication in TLS and SSL
  - Web browsers use it to implement HTTPS client certificates in SSL.
  - SMTP, POP3, LDAP were modified to support SSL and X509v3 certificates.
  - IPsec VPNs use certificates for the public key distribution mechanism
  - PGP (Pretty Good Privacy)

- Certificates can be used at the network layer or the application layer by Cisco routers, VPN concentrators, PIX firewalls, to authenticate IPsec peers.
- Cisco switches can use certificates to authenticate devices on LAN ports.

Other PKI standards



RSA PKCS Standards:

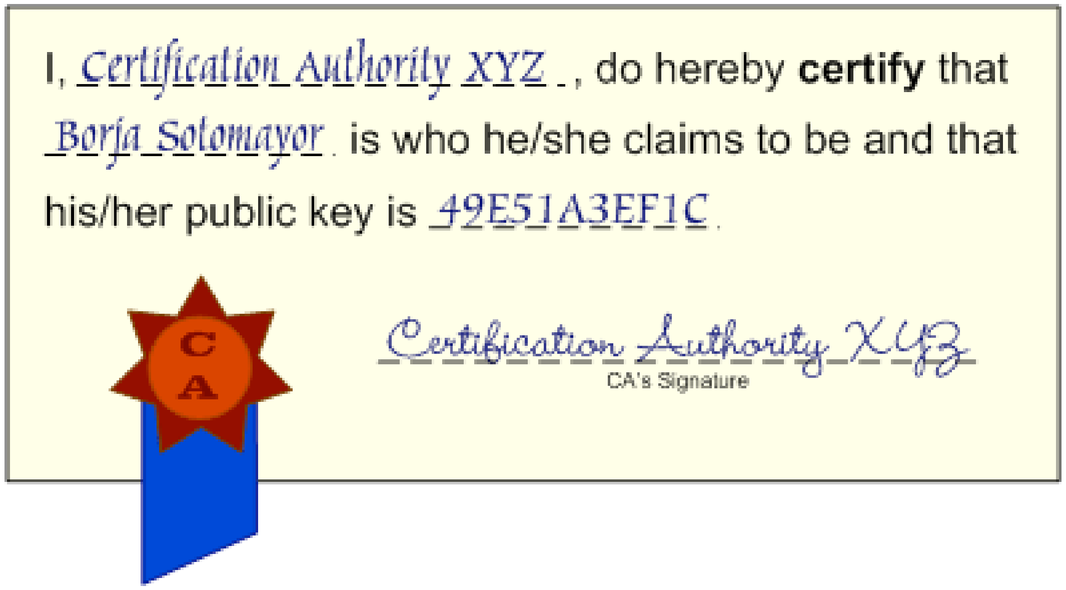
- **PKCS #1:** RSA Cryptography Standard
- **PKCS #3:** DH Key Agreement Standard
- **PKCS #5:** Password-Based Cryptography Standard
- **PKCS #6:** Extended-Certificate Syntax Standard
- **PKCS #7:** Cryptographic Message Syntax Standard
- **PKCS #8:** Private-Key Information Syntax Standard
- **PKCS #10:** Certification Request Syntax Standard
- **PKCS #12:** Personal Information Exchange Syntax Standard
- **PKCS #13:** Elliptic Curve Cryptography Standard
- **PKCS #15:** Cryptographic Token Information Format Standard

- PKCS = Public Key Cryptography Standards

- A series of (sub)standards that define the low-level formats for secure exchange of data.

Certificate authorities

- PKIs form different topologies of trust:
  - Single-root CA topologies
  - Hierarchical CA topologies
  - Cross-certified CA topologies

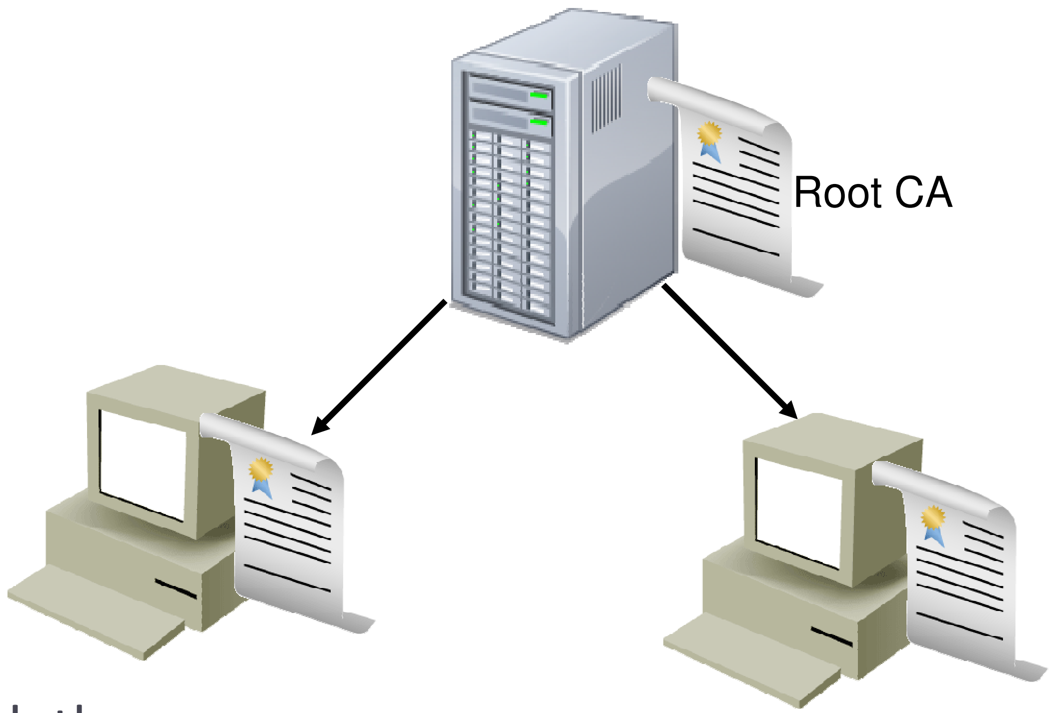


I, Certification Authority XYZ, do hereby **certify** that Borja Sotomayor is who he/she claims to be and that his/her public key is 49E51A3EF1C.

C
A
Certification Authority XYZ
CA's Signature

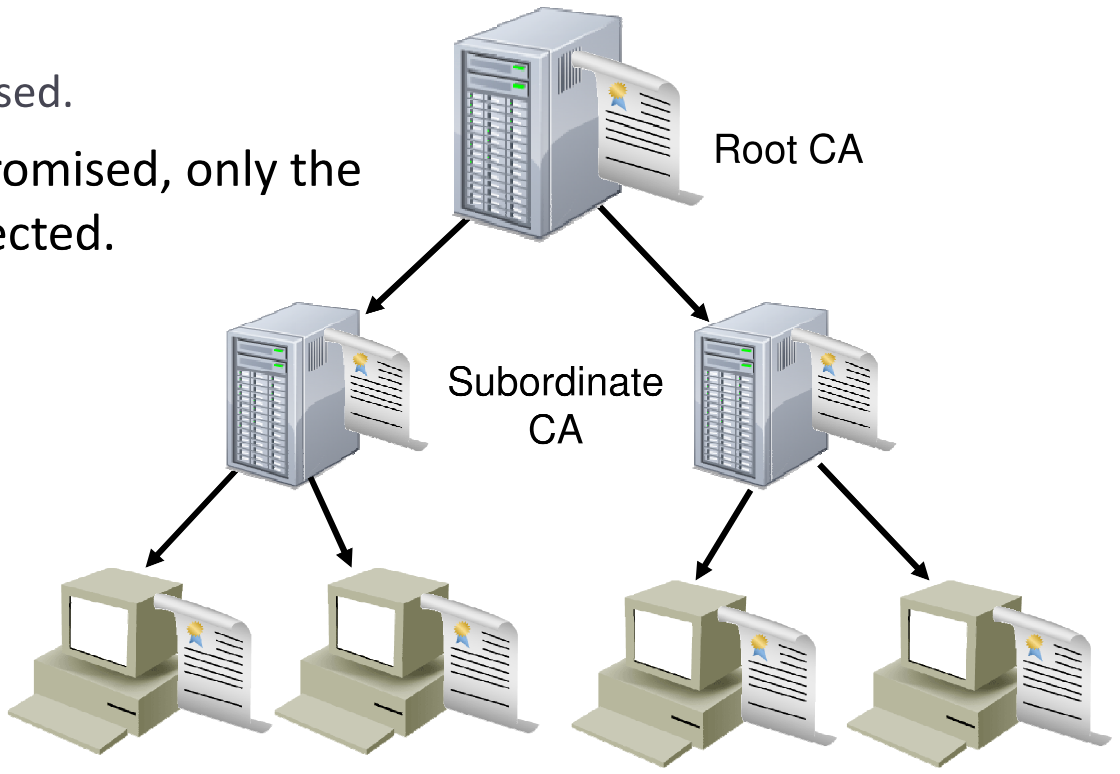
CA topologies – Single-root CA

- Does not scale well.
- Strictly centralized administration.
- A single private and vulnerable key that is used for signing.
  - If this key is compromised, the whole PKI can no longer be trusted.
  - Single point of failure

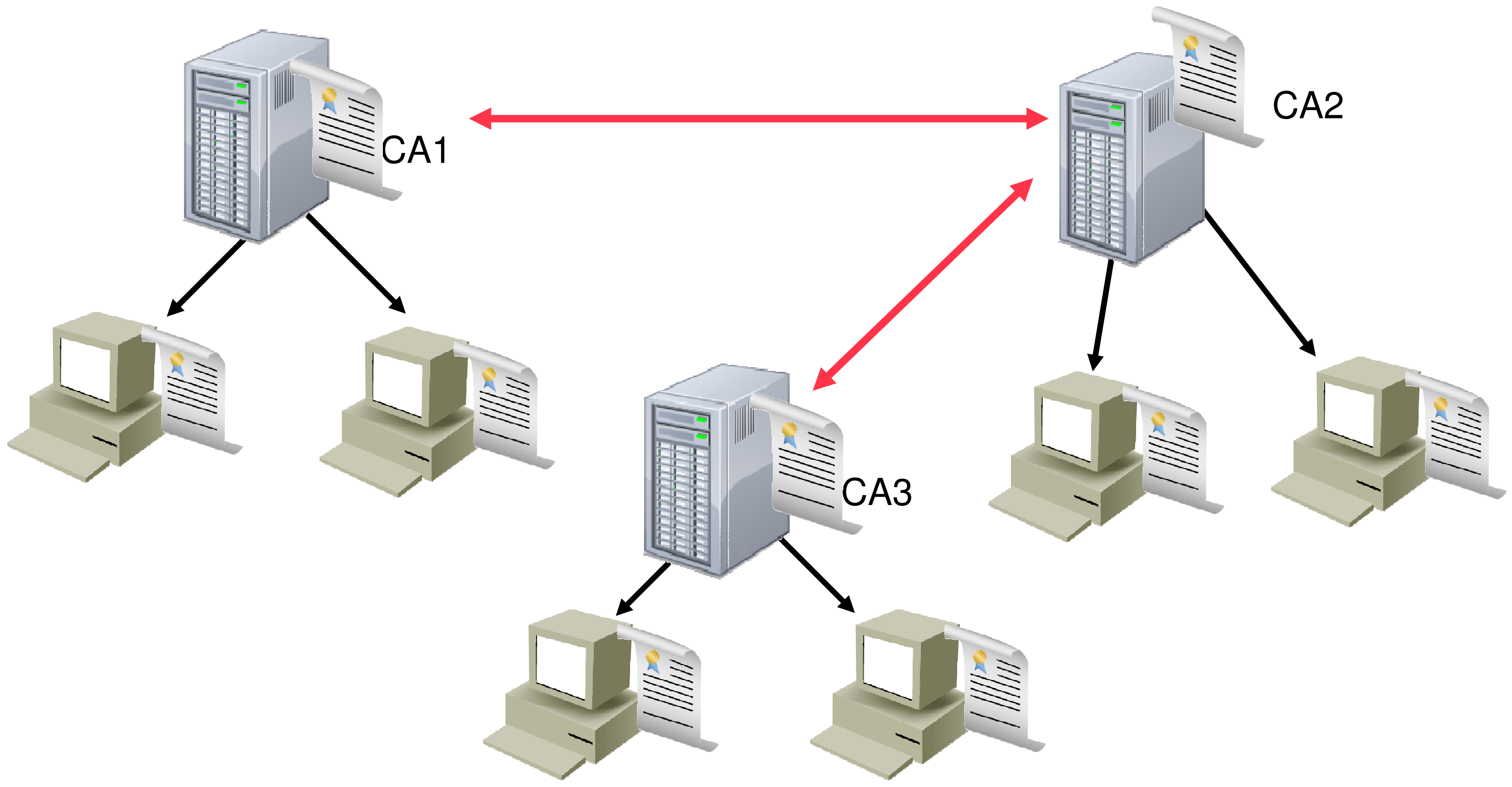


CA topologies – Hierarchical CA topology

- Trust can be delegated to other subordinate CAs.
- Increased stability and manageability.
- The root CA is only used to enroll the subordinate CAs.
  - The root becomes less exposed.
- If a subordinate CA is compromised, only the corresponding branch is affected.



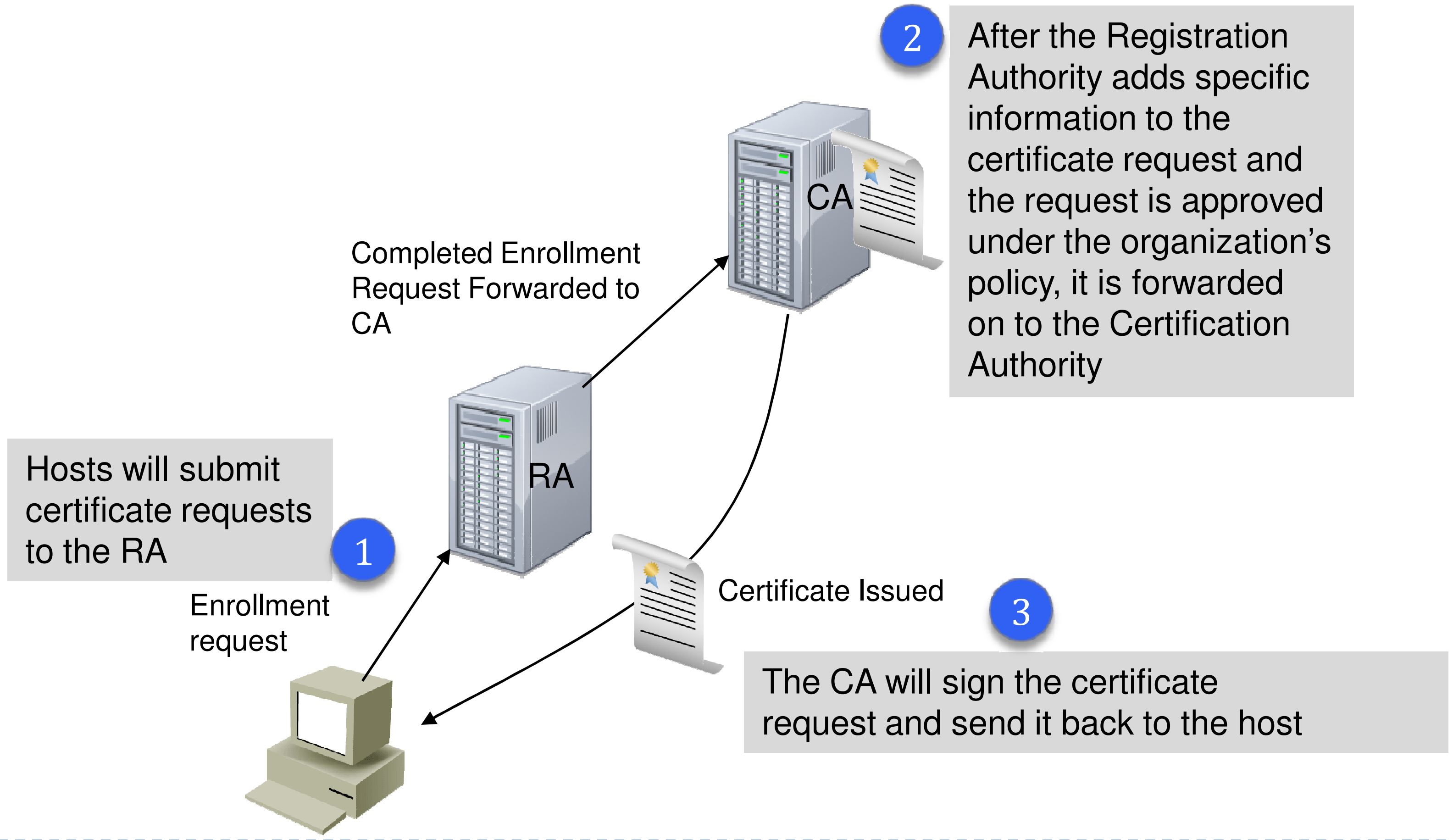
CA topologies – Cross-certified CA topology



- CAs validate each other's root certificates.

Registration authorities

- The task of enrolling with a CA is often delegated to an RA.


2
After the Registration Authority adds specific information to the certificate request and the request is approved under the organization's policy, it is forwarded on to the Certification Authority
CA
Completed Enrollment Request Forwarded to CA
RA
Hosts will submit certificate requests to the RA
1
Enrollment request
Certificate Issued
3
The CA will sign the certificate request and send it back to the host

Final thoughts

- Security, no matter how strong, can always be badly implemented.

- Encryption does not guarantee security.
- There are many ways to break a cryptographic system without cryptanalysis.
  - Viruses, worms, hackers, social engineering, etc.
  - Unauthorized physical access to private keys.

- Cryptography is only one step in the process of computer security.

Privacy

"If McDonalds offered a free Big Mac in exchange for a DNA sample, there'd be lines around the block."

Bruce Schneier

Hackers Night Out – 14 December 19:00