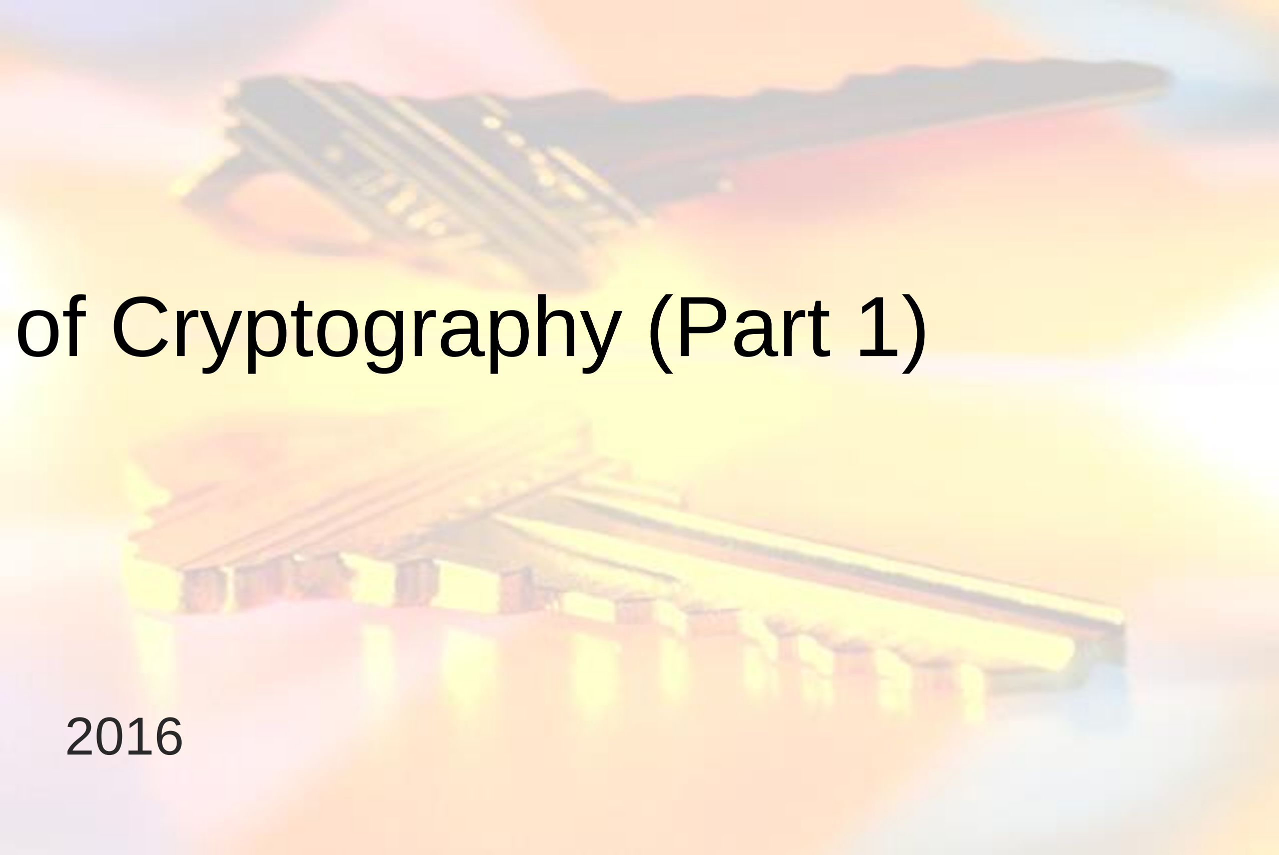
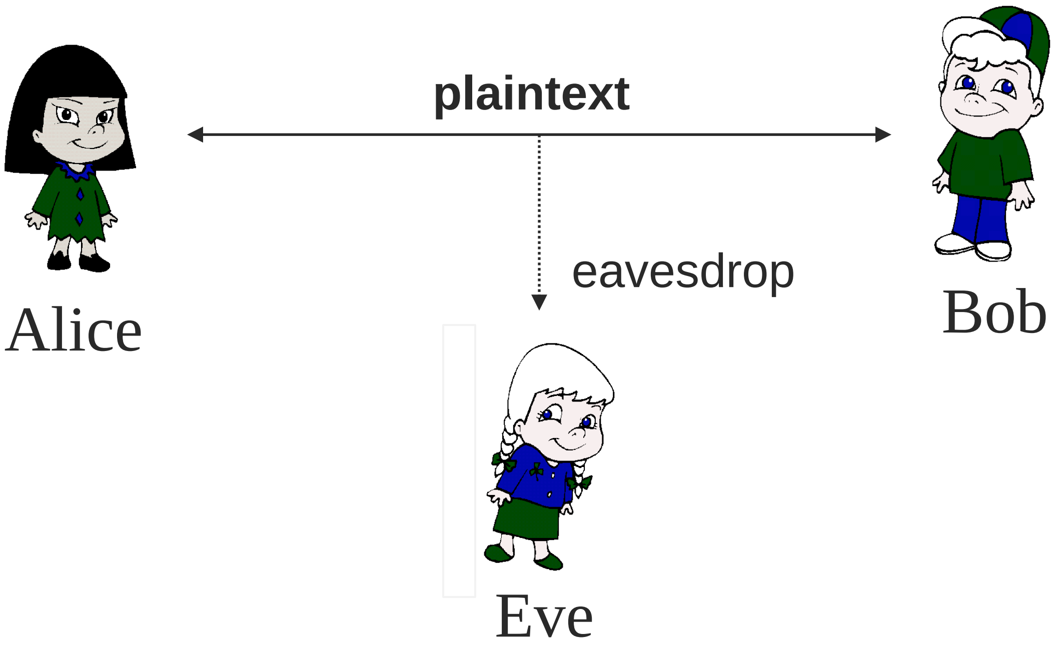
Elements of Cryptography (Part 1)
Lect 5



2016

What is this lecture about?



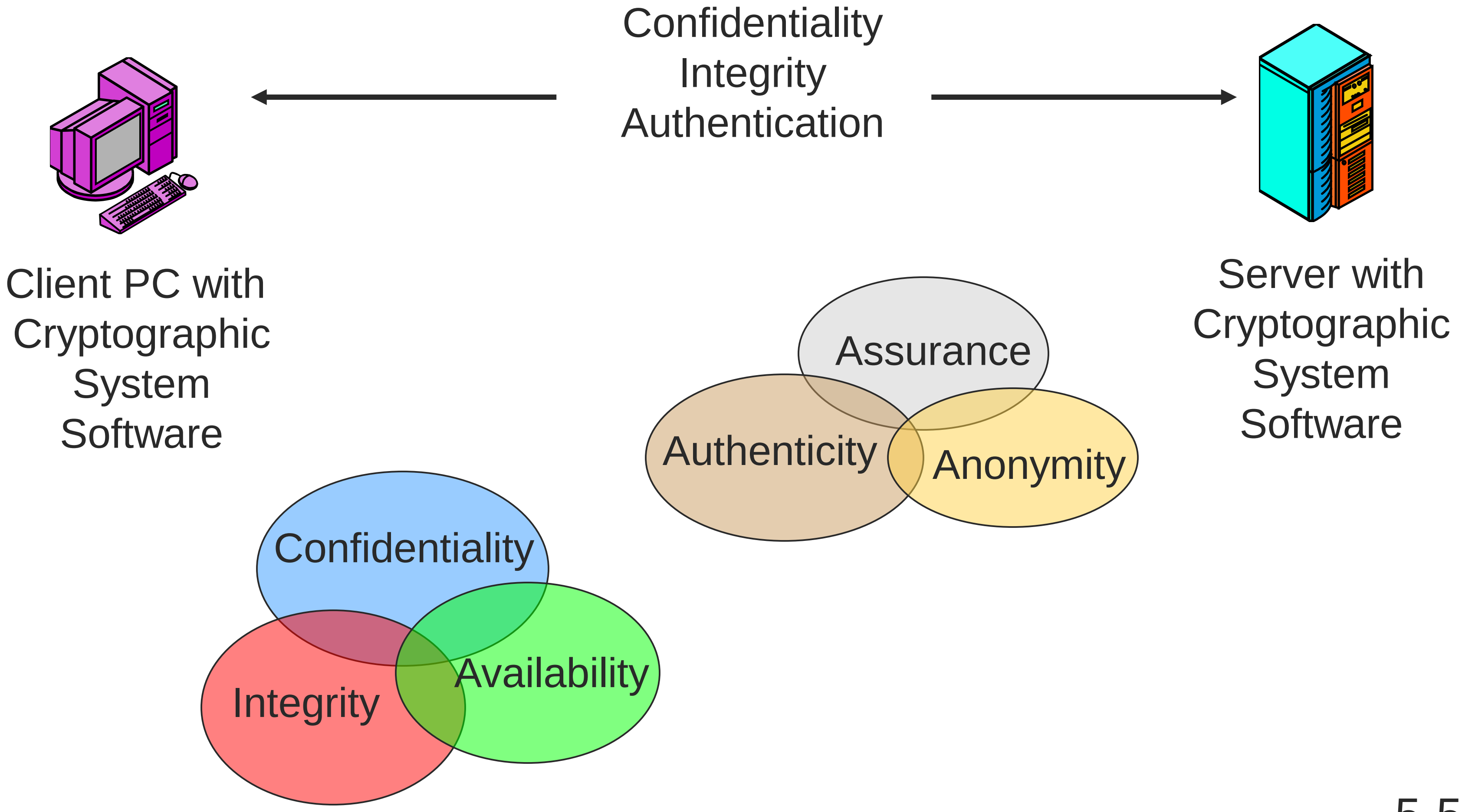
- Alice and Bob wish to communicate securely over a network
  - How can this be achieved?
  - What do we mean with secure communication?

Outline

- Basic notions
- Symmetric Key Encryption
- Public Key Encryption (asymmetric)
- Digital Signatures
- Digital Certificates
- Public Key Infrastructure

1. Basic Notions

Secure Communication



Confidentiality
Integrity
Authentication

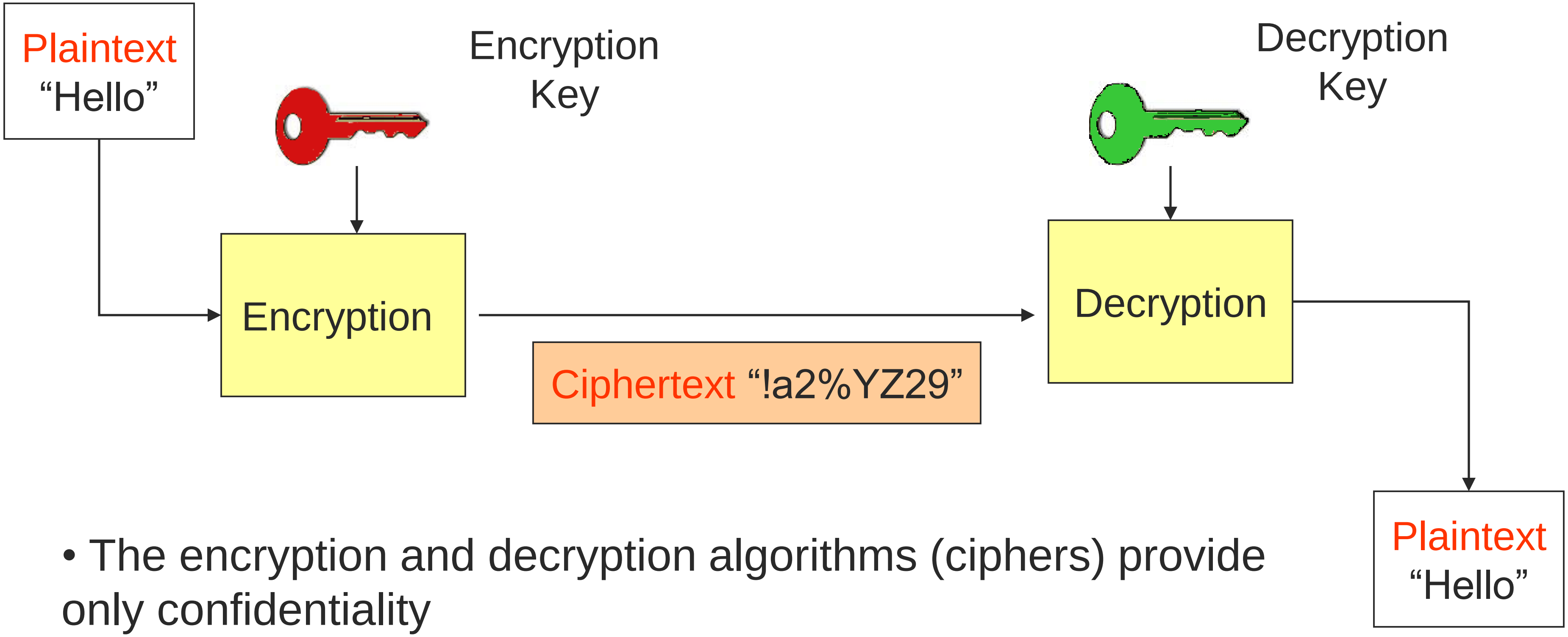
Client PC with Cryptographic System Software

Server with Cryptographic System Software

Assurance
Authenticity
Anonymity

Confidentiality
Integrity
Availability

Basics



Plaintext “Hello”

Encryption Key

Encryption

Ciphertext “!a2%YZ29”

Decryption Key

Decryption

Plaintext “Hello”

- The encryption and decryption algorithms (ciphers) provide only confidentiality

- The overall security of a system depends on more than just the choice of these algorithms

- Encryption = enciphering   Decryption = deciphering

Key Length

Key Length in Bits	Number of Possible Keys
1	2
2	4
4	16
8	256
16	65,536
40	1,099,511,627,776
56	72,057,594,037,927,900
112	5,192,296,858,534,830,000,000,000,000,000,000

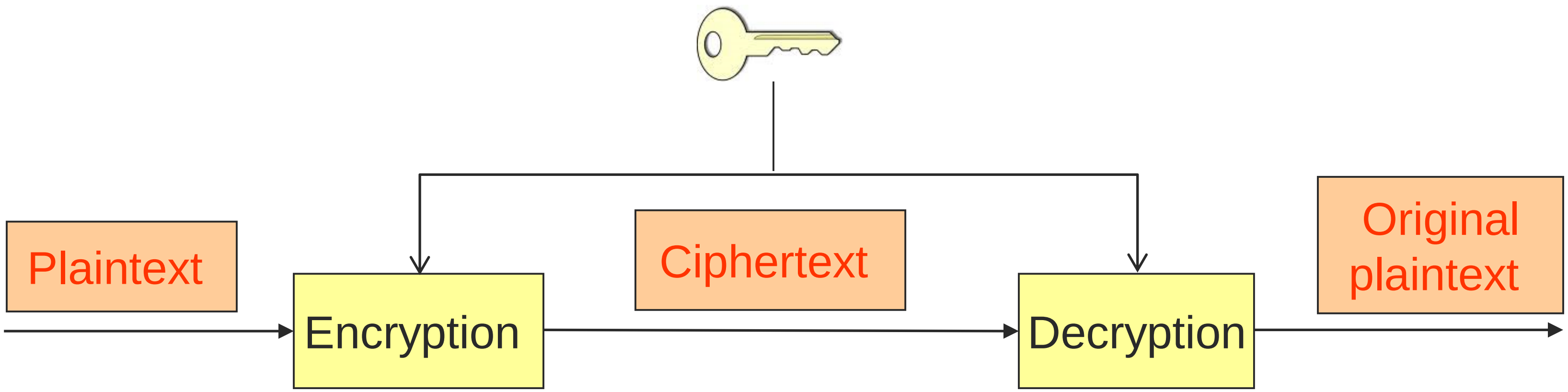
Key Length

Key Length in Bits	Number of Possible Keys
112	5.1923E+33
168	3.74144E+50
256	1.15792E+77
512	1.3408E+154

- Cryptanalysts try to crack keys
- Exhaustive search is thwarted by having long keys (> 100 bits)
- Change keys frequently

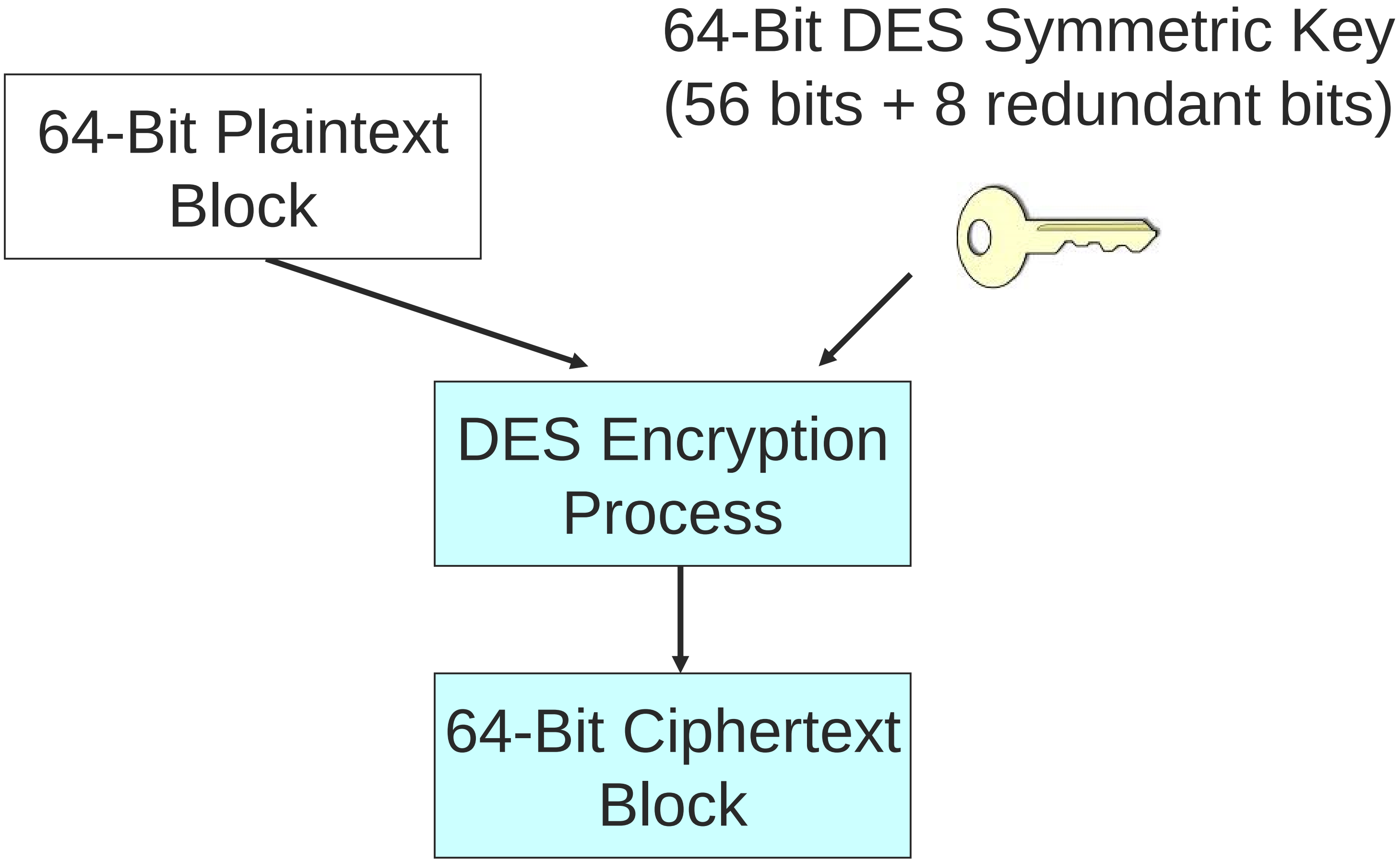
2. Symmetric Key Encryption

Symmetric Key Encryption



- Stream ciphers
  - encrypt the digits (typically bytes) of a message one at a time
- Block ciphers
  - take a number of bits and encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size
  - blocks of 64 bits are typically used

Data Encryption Standard (**DES**)



64-Bit DES Symmetric Key
(56 bits + 8 redundant bits)

64-Bit Plaintext Block

DES Encryption Process

64-Bit Ciphertext Block

About **DES** …

- Block cipher
- Based on the Lucifer system proposed by IBM to NBS in 1977
- No longer in use since 1997
- In July 1997, with the help of 14,000 computers on the Internet, it was possible to break **DES** key in 90 days
- Within 6 months, the time to break the **DES** key was reduced to 39 days
- The successful attacks on DES lead to the development of cryptanalysis

Triple DES (**3DES**)

168-Bit Encryption with Three 56-Bit Keys

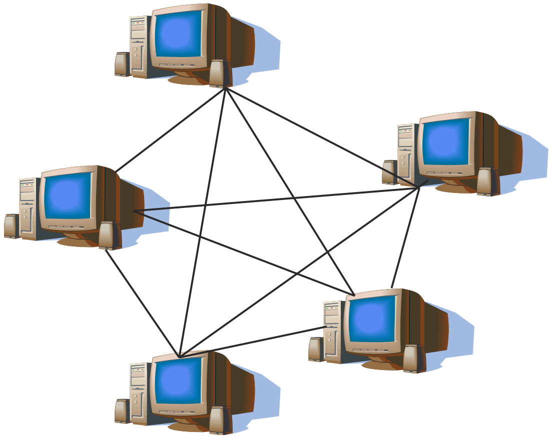
Sender	**Receiver**
Encrypts plaintext with the 1st key	Decrypts ciphertext with the 3d key
Decrypts output of first step with the 2nd key	Encrypts output of the first step with the 2nd key
Encrypts output of second step with the 3d key; gives the ciphertext to be sent	Decrypts output of second step with the 1st key; gives the original plaintext

DES, **3DES**, and **AES**

	DES	**3DES**	**AES**
Key Length (bits)	56	112 or 168	128, 192, 256
Strength	Weak	Strong	Strong
Processing Requirements	Moderate	High	Modest
RAM Requirements	Moderate	High	Modest

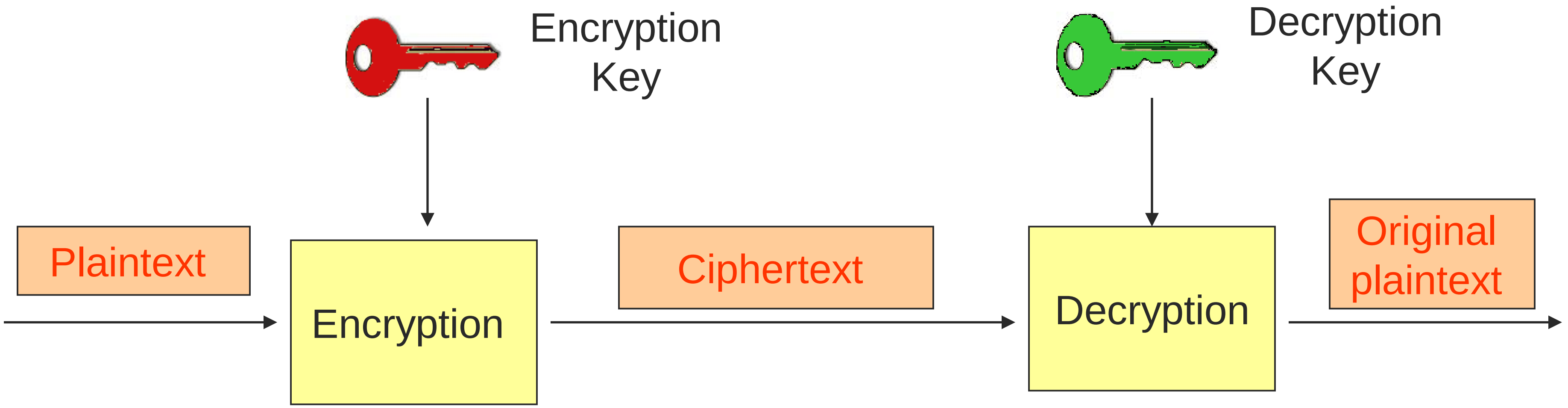
Drawback of Symmetric Key Encryption

- The same key is used for encryption and decryption
- Each pair of users must agree on a common secret key
  - **How to securely exchange keys?**
- If Bob sends messages to several different people (Alice, Paul, Eve, …) then he must have a different key for each of them
  - **Large number of keys must be managed over a network of users**



3. Public Key Encryption

Public Key Encryption



Encryption Key

Decryption Key

Plaintext

Encryption

Ciphertext

Decryption

Original plaintext

- Introduced mid 1970s
- Keys are related mathematically
- Private key cannot be feasibly derived from public key
- The two keys are cryptographically inverse

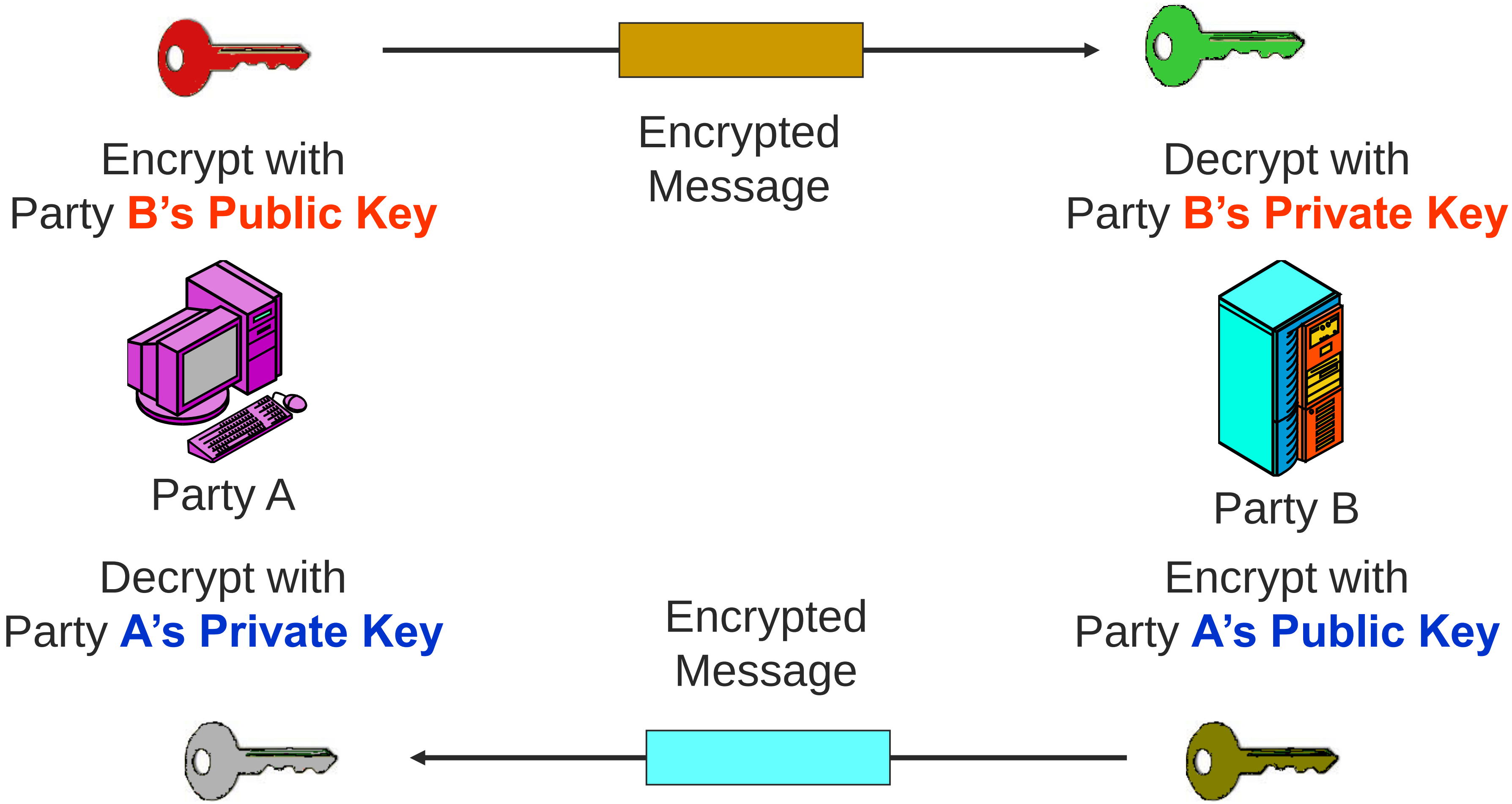
Public Key Encryption

- One key is made publically available
  - Like having a catalogue of keys

↓

- Reduces the number of keys
- Distribution and management of keys becomes easier

Public Key Encryption for Confidentiality



Encrypt with
Party **B’s Public Key**

Encrypted
Message

Decrypt with
Party **B’s Private Key**

Party A

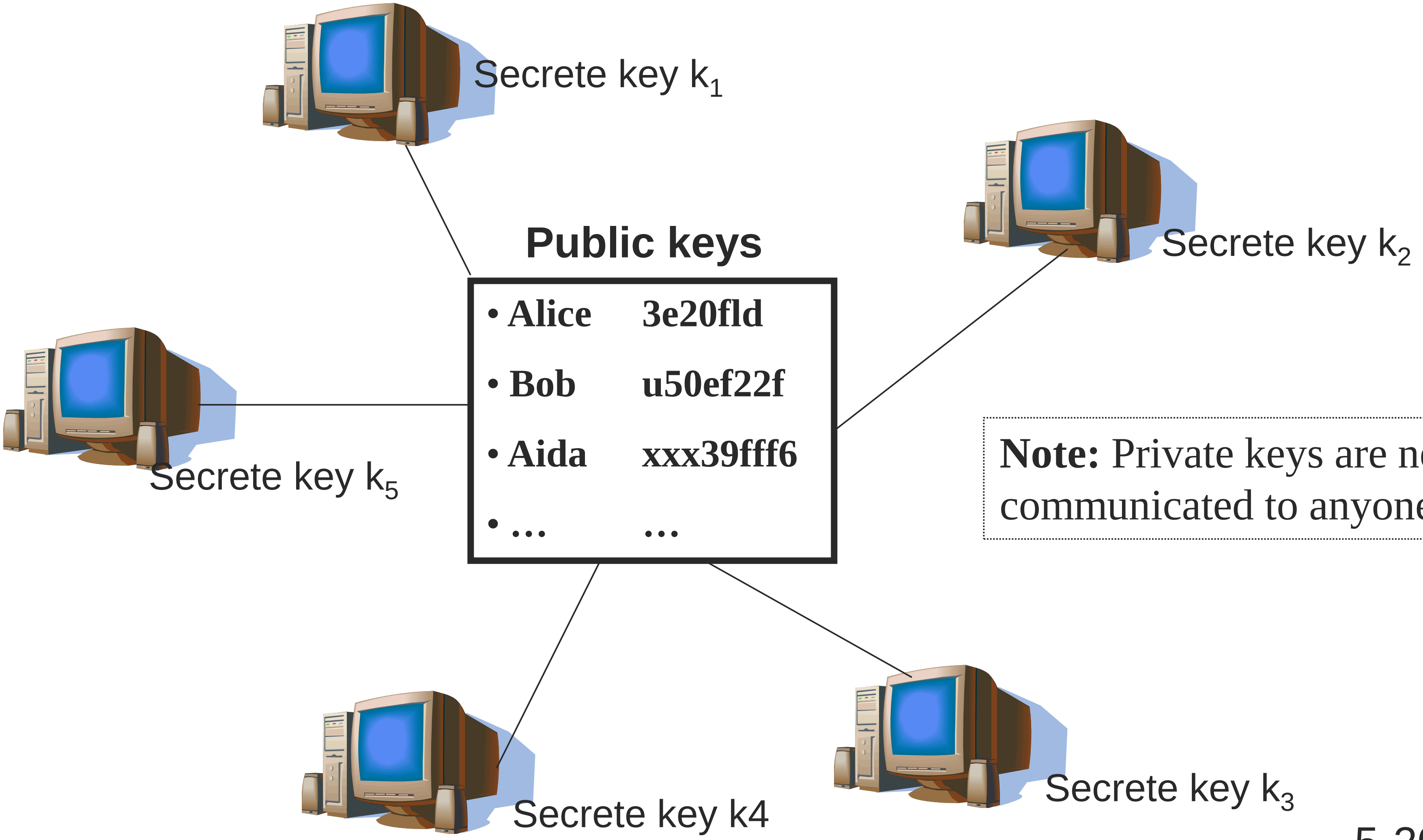
Party B

Decrypt with
Party **A’s Private Key**

Encrypted
Message

Encrypt with
Party **A’s Public Key**

Public Key Encryption



Secrete key k_1

Secrete key k_2

Secrete key k_3

Secrete key k4

Secrete key k_5

Public keys

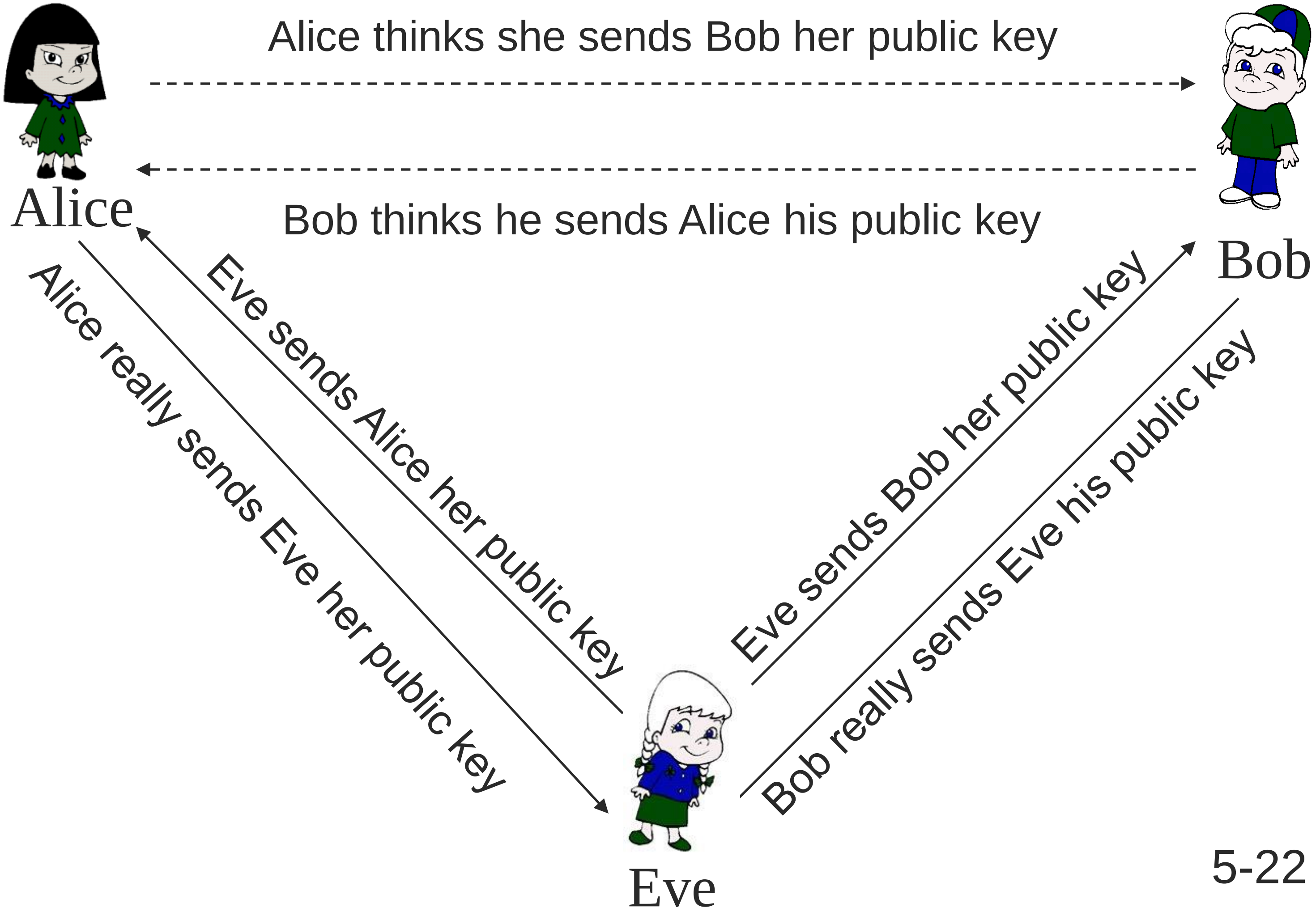
• **Alice**	**3e20fld**
• **Bob**	**u50ef22f**
• **Aida**	**xxx39fff6**
• …	…

Note: Private keys are never communicated to anyone

Public Key Encryption Algorithms

- RSA
  - Popular public key encryption today
  - 1,024 or 2,048 bits to be strong today
- Elliptic curve cryptosystem (ECC)
  - 512 bits to be strong today (more efficient than RSA)
- In contrast, symmetric key methodologies only need key lengths of 100 bits to be strong today
- To be used only with short plaintext messages

Man-in-the-Middle Attack



Alice

Bob

Eve

Alice thinks she sends Bob her public key

Bob thinks he sends Alice his public key

Alice really sends Eve her public key

Eve sends Alice her public key

Eve sends Bob her public key

Bob really sends Eve his public key

Man-in-the-Middle Attack

- Encrypting messages is not enough to guarantee a secure communication
  - Authenticity is not guaranteed
  - Message integrity is not guaranteed
- In e-commerce it is also essential that messages satisfy the same requirement as formal signed documents (non-repudiation)

Digital signatures and certificates

4. Digital Signatures

Hashing

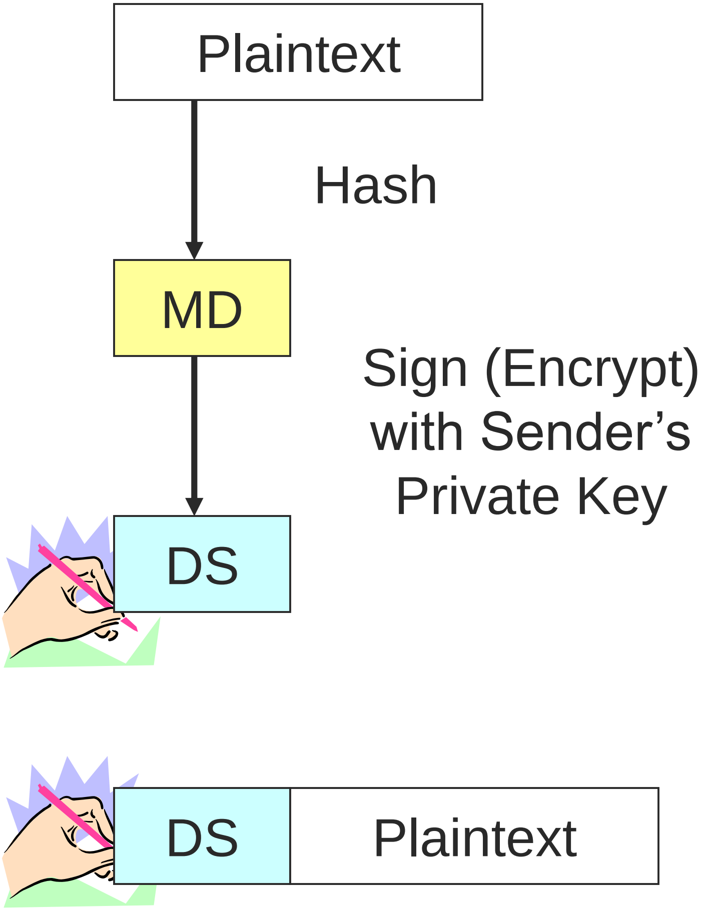
- Useful in checking message integrity
- Hashing produces a result (hash) that has always the same length regardless of the length input
- Hashing is repeatable: given the same bit string, will always give the same hash
  - No key
- Hashing is irreversible
- Hashing is NOT encryption
- Collisions should not occur easily

Digital Signature

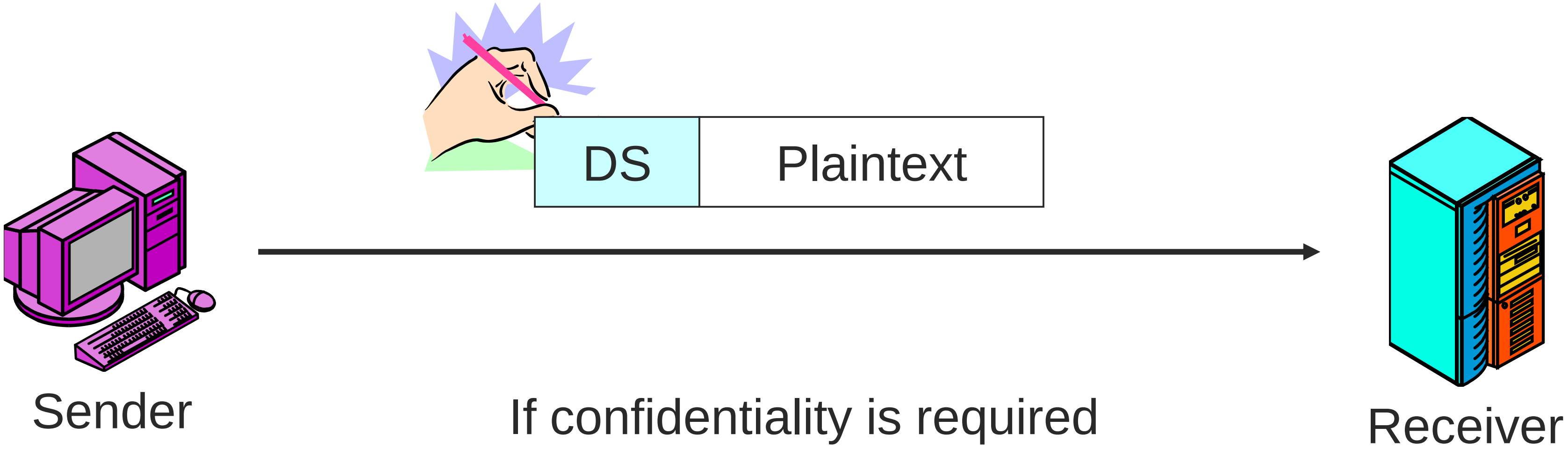
To Create the Digital Signature:

1. Hash the plaintext to create a brief message digest (MD)

2. Sign (encrypt) the message digest with the sender's private key to create the digital signature (DS)



Digital Signature



DS

Plaintext

Sender

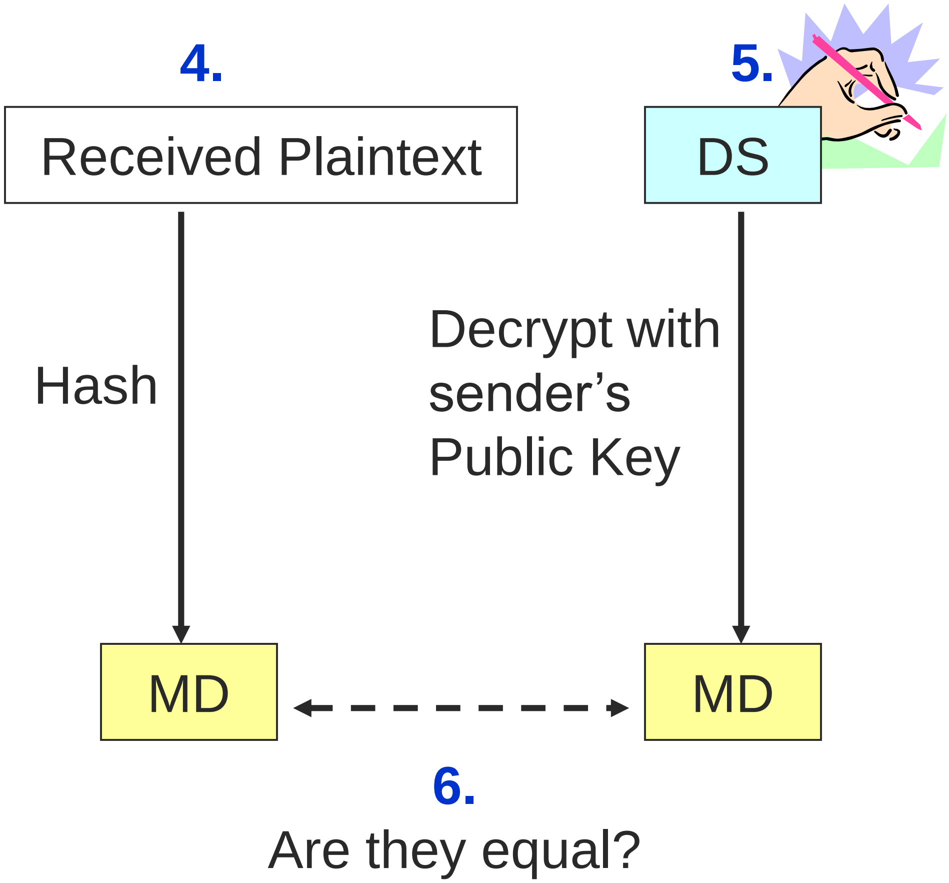
Receiver

If confidentiality is required

↓

encrypt DS and plaintext

Digital Signature



4\.

Received Plaintext

Hash

MD

5\.

DS

Decrypt with sender’s Public Key

MD

6\.

Are they equal?

To Test the Digital Signature

4. Hash the received plaintext with the same hashing algorithm the sender used. This gives the message digest.

5. Decrypt the digital signature with the sender’s public key. This also should give the message digest.

6. If the two match, the message is authenticated.

Digital Signatures are NOT enough

- Digital signatures only guarantee authentication and message integrity, if we really have the party's public key
- Digital signatures alone do not prevent the man-in-the-middle attack

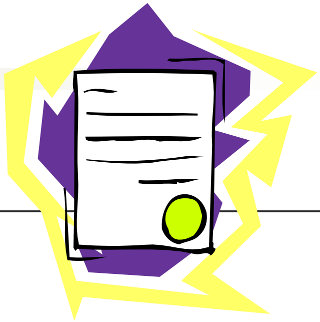
Digital Certificates

5. Digital Certificates

Digital Certificates

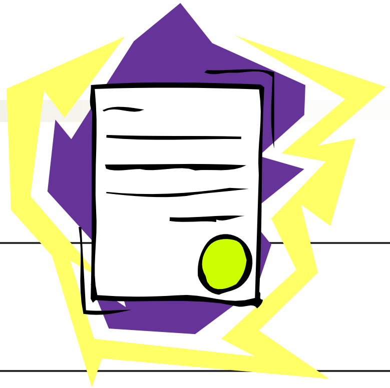
- Trustable document that binds a public key to a owner
- Issued by a certification authority (CA)
  - Independent and trusted source of information about the public keys owners
- A certificate contains the digital signature of the CA that issued it
  - To provide authentication and certificate integrity
  - Hence, to verify the digital signature in the certificate, we need the public key of the CA

X.509 Digital Certificate Fields



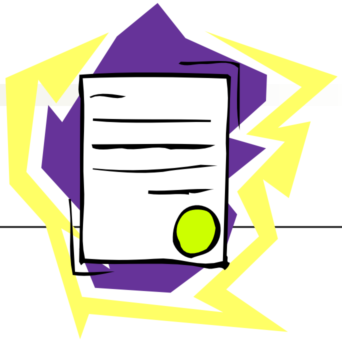
Field	Description
Version Number	Version number of the X.509. Most certificates follow Version 3. Different versions have different fields. This figure reflects the Version 3 standard.
Issuer	Name of the Certificate Authority (CA).
Serial Number	Unique serial number for the certificate set by the CA.

X.509 Digital Certificate Fields



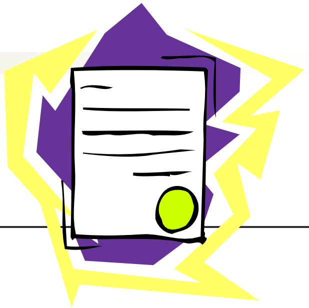
Field	Description
Subject	The name of the person, organization, computer, or program to which the certificate has been issued
Public Key	The public key of the subject
Public Key Algorithm	The algorithm the subject uses to sign messages with digital signatures

X.509 Digital Certificate Fields



Field	Description
Valid Period	The period before which and after which the certificate should not be used. Certificate may be revoked before the end of this period.
Digital Signature	The digital signature of the certificate, signed by the CA with the CA's own private key. Provides authentication and certificate integrity User must know the CA's public key independently.

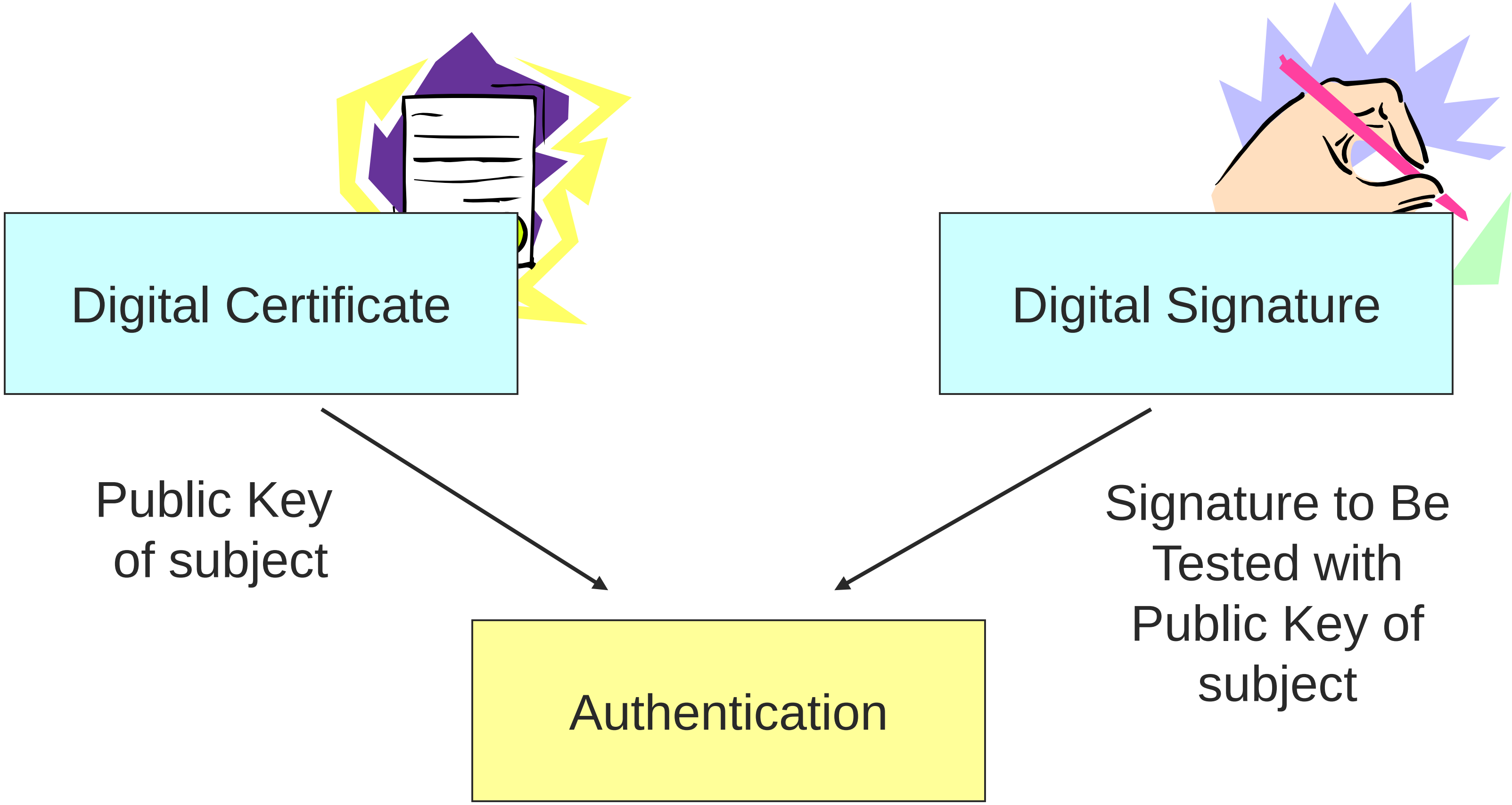
X.509 Digital Certificate Fields



Field	**Description**
Signature Algorithm Identifier	The digital signature algorithm the CA uses to sign its certificates.

- Web browsers know the public keys of major **CA**s (by having the CAs digital certificates)
- Who signs the digital certificate of a **CA**?

To Obtain Authentication



Digital Certificate

Digital Signature

Public Key
of subject

Signature to Be
Tested with
Public Key of
subject

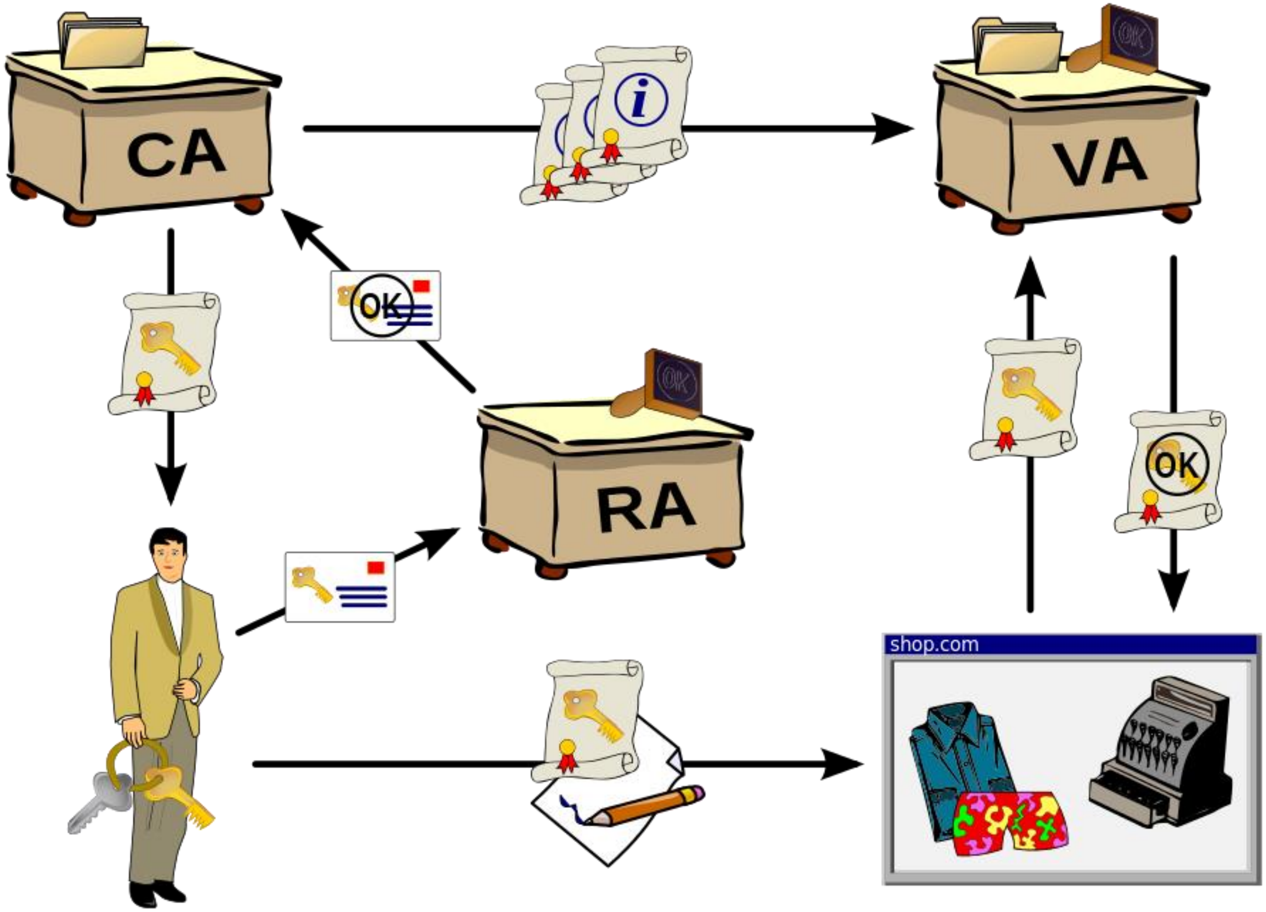
Authentication

6. Public Key Infrastructure (PKI)

Public Key Infrastructure

- To create public/private key pairs
- To create digital certificates
- To distribute securely private keys
- To verify that a certificate is still valid
  - Maintain certificate revocation list (CRL)
- Managed by a CA

Public Key Infrastructure with a CA



Certificate Authority

- **CA**s are not regulated in any country today
  - Anyone can be a **CA**
  - Even an organized crime syndicate
  - Some, such as **VeriSign**, are widely trusted
- Companies can act as their own **CA**s
  - Assign keys and certificates to their internal computers
  - This gets around the need to trust public **CA**s

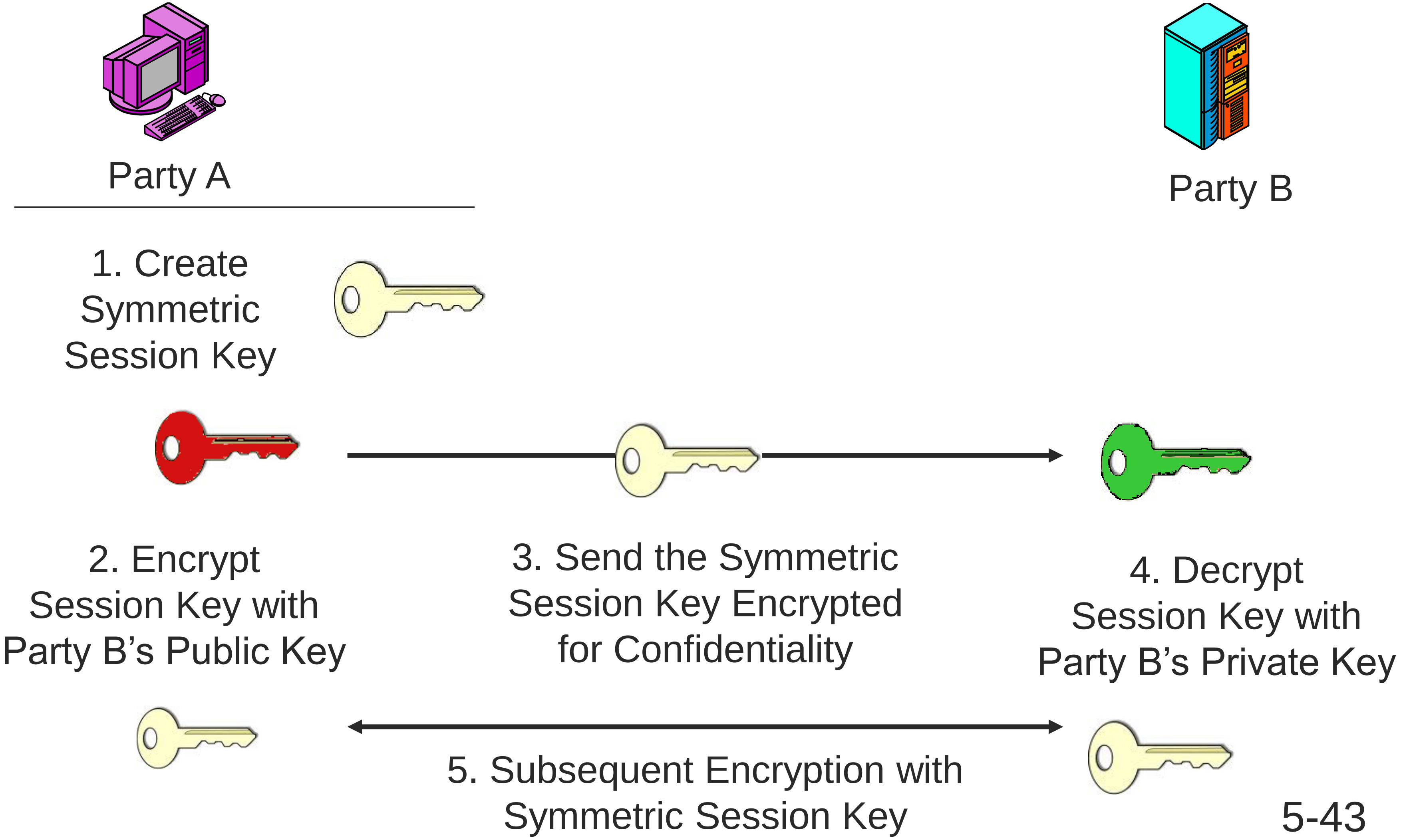
VeriSign

- Uses the concept of classes for different types of digital certificates:
  - Class 1 for individuals (intended for emails)
  - Class 2 for organizations for which proof of identity is required
  - Class 3 for servers and software signing, for which independent verification and checking of identity and authority is done by the issuing certificate authority
  - Class 4 for online business transactions between companies
  - Class 5 for private organizations or governmental security

Symmetric Key Exchange

- Public key algorithms can only be used to encrypt short plaintext messages
- Long plaintext messages are encrypted with symmetric key algorithms
- Public key encryption can be used to distribute securely symmetric keys between two parties
  - Diffie-Hellman key agreement can also be used (Lec6)

Public Key Distribution for Symmetric Session Keys



Party A

Party B

1. Create Symmetric Session Key

2. Encrypt Session Key with Party B's Public Key

3. Send the Symmetric Session Key Encrypted for Confidentiality

4. Decrypt Session Key with Party B's Private Key

5. Subsequent Encryption with Symmetric Session Key

Summary

- Cryptographic Systems
  - Provide protections to dialog sessions automatically
  - Secure communication guarantees:
    - Confidentiality
    - Integrity
    - Authentication

Summary

- Public Key Encryption
  - Each party has a secret private key and a public key
  - Sender uses the receiver's public key to encrypt for confidentiality
  - Receiver uses the receiver's private key to decrypt messages

Summary

- Symmetric key algorithms
  - Used with long plaintext messages
- Public key algorithms
  - Used with short plaintext messages (example: exchange of symmetric session keys)
  - Easier to manage keys
  - Digital signatures
  - Symmetric key exchange

Summary

- Symmetric keys
    - Limited damage if cracked
    - Changed frequently
    - Shorter keys
- Public keys
    - Serious damage if cracked
    - Rarely changed
    - Longer keys

Summary

- Digital Signatures
  - Used in message-by-message authentication
  - Fundamental in e-commerce
  - Verifier uses the subject’s public key to test the digital signature

Summary

- Digital Certificates
  - Verifier uses the subject’s public key to test the digital signature
  - Where does the verifier get the subject’s public key?
  - Digital certificates give the subject’s name and public key
  - Note that both a digital signature and a digital certificate are needed in authentication. Neither alone is enough.

Summary

- Public Key Infrastructure
  - Digital certificates
    - Follow the X.509 standard
  - PKI Server
    - Distributes private keys securely
    - Distributes public keys in digital certificates
    - Provides Certification Revocation List to ensure that digital certificate is still valid

Summary

- Certificate Authorities
  - Manage the PKI
  - If the CA is set up by an attacker, cannot trust its digital certificates
  - Not regulated