



SLIIT

Discover Your Future

IE2022 - Introduction to Cyber Security

Lecture - 08

Asymmetric Encryption Algorithms and PKI

Mr. Amila Senarathne



Asymmetric Encryption Algorithms and Public Key Infrastructure

- ★ Reading Assignment

- CCNA Security Curriculum, Chapter 7: Cryptographic Systems

- ★ Supplementary text

- W. Stallings and L. Brown, “Computer Security, Principles and Practice, 2nd edition, Pearson, 2012, Chapter 2 :Cryptographic Tools.

Topics to be discussed

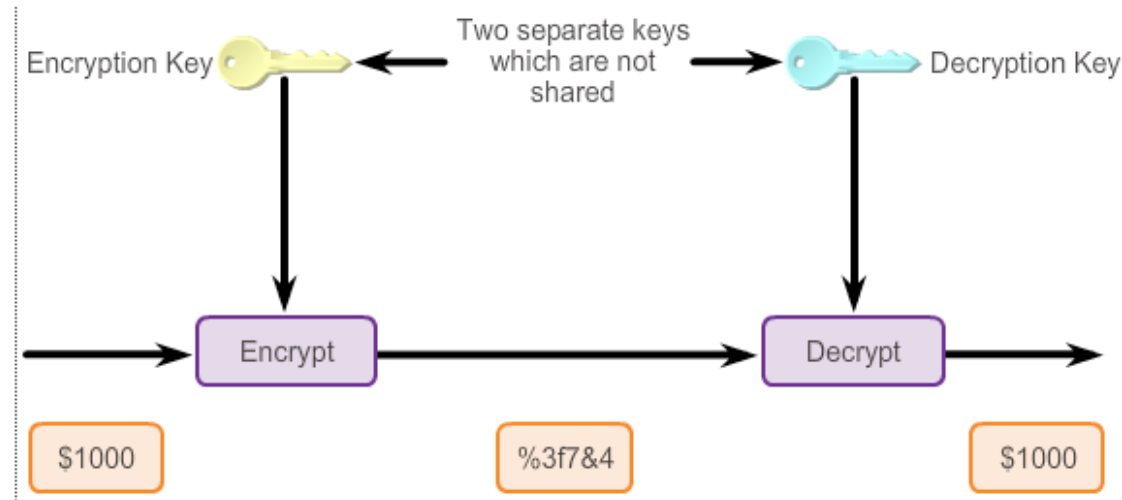
- ★ Basics of Public Key Cryptography (Asymmetric Encryption)
- ★ Digital Signatures
 - Properties of Digital Signature
 - Digital Signature Process
 - Digitally Signed Code
- ★ Diffie-Hellman Key Exchange
- ★ Asymmetric Encryption Algorithms
- ★ Public Key Infrastructure

PUBLIC KEY CRYPTOGRAPHY

Asymmetric Encryption Algorithms

Asymmetric encryption algorithms characteristics include:

- Asymmetric encryption algorithms are best known as public key algorithms.
- The usual key length is 512 to 4,096 bits.
- A sender and receiver do not share a secret key.
- These algorithms are relatively slow, because they are based on difficult computational algorithms.
- Examples: RSA, ElGamal, elliptic curves, and DH.



Asymmetric Key Algorithms

- ★ **Asymmetric algorithms are also called public-key algorithms.**
- ★ Public-key algorithms are asymmetric algorithms based on the use of two different keys, instead of one.
 - **Private key** - This key must be know *only* by its owner.
 - **Public key** - This key is known to everyone (it is *public*).
- ★ The key used for encryption is different from the key used for decryption.
 - However, the decryption key cannot, in any reasonable amount of time, be calculated from the encryption key and vice versa.
- ★ Public-key systems have a clear advantage over symmetric algorithms.
 - There is no need to agree on a common key for both the sender and the receiver.

Asymmetric Key Algorithms Cont.

- ✳ Either key can be used for encryption, but the complementary matched key is required for decryption.
 - If a public key encrypts data, the matching private key decrypts data.
 - If a private key encrypts data, the matching public key decrypts data.

Asymmetric Key Characteristics

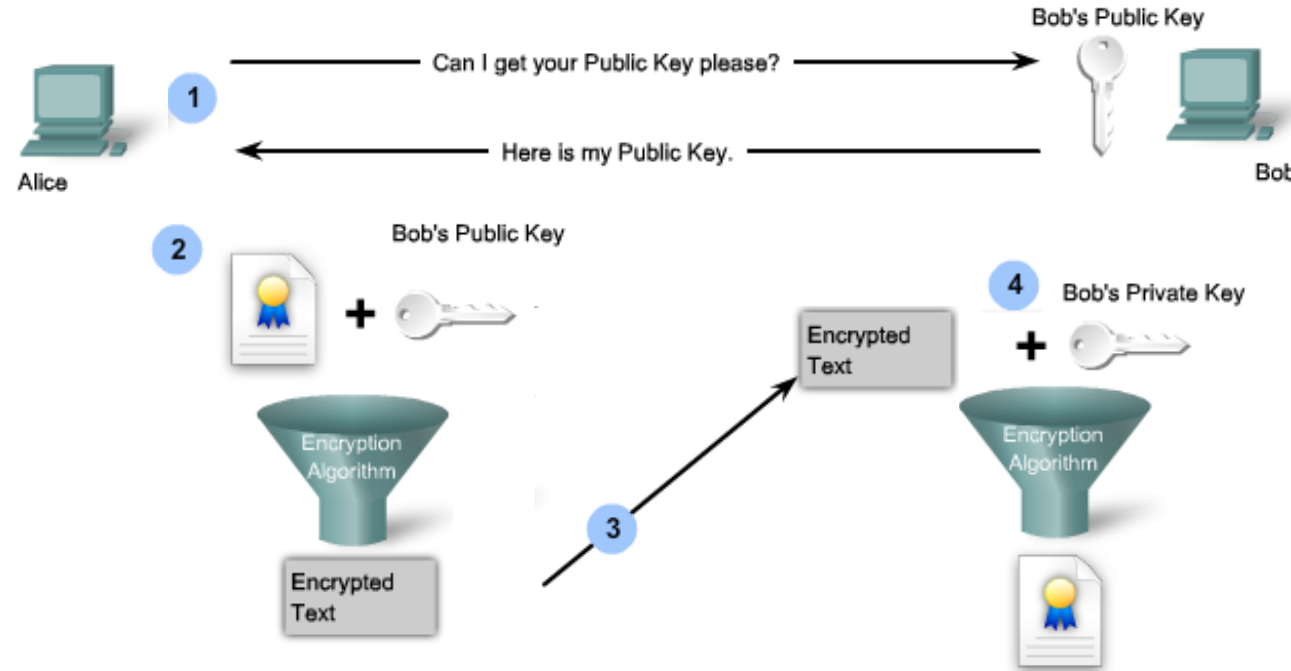


Confidentiality

- ★ Confidentiality is achieved when the encryption process is started with the public key.
- ★ When the public key is used to encrypt the data, the private key must be used to decrypt the data.
 - Only one host has the private key guaranteeing confidentiality.

Asymmetric Algorithms for Confidentiality

Public Key (Encrypt) + Private Key (Decrypt) = Confidentiality



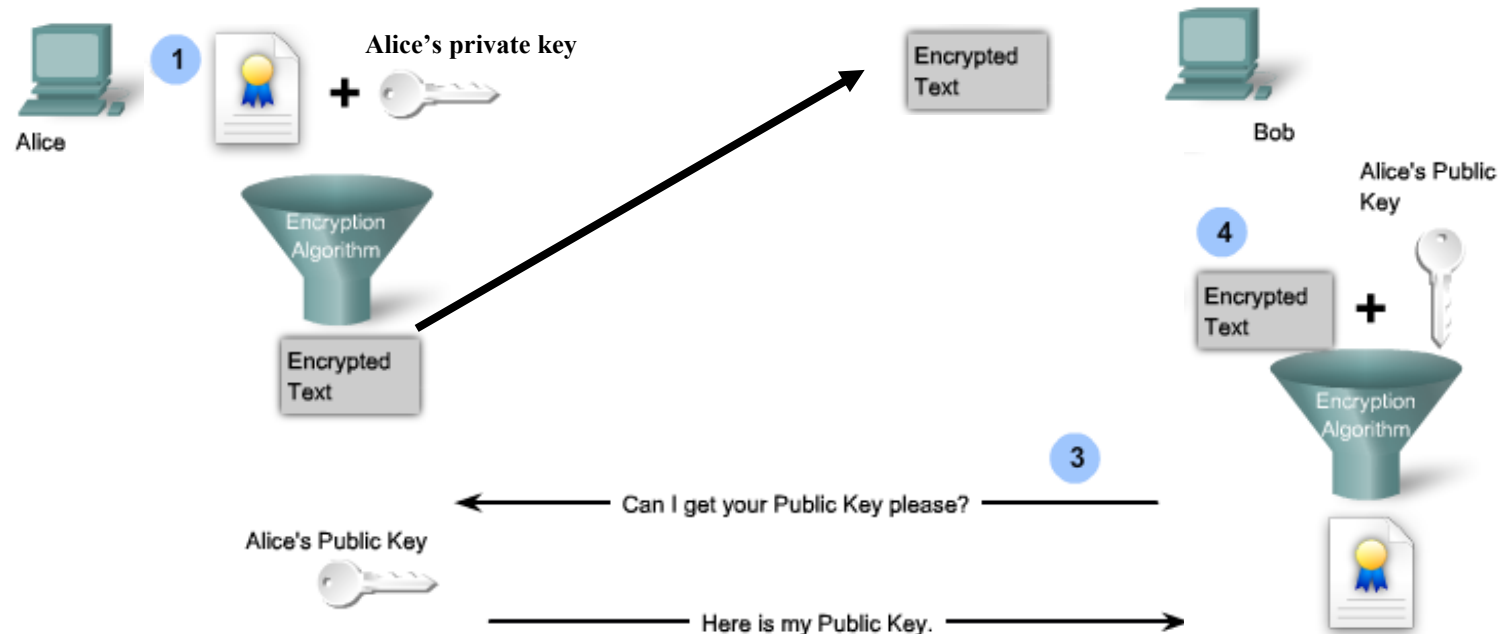
1. Alice asks Bob for his public key and Bob sends it to her.
2. Alice uses Bob's public key to encrypt a message using an agreed-upon algorithm.
3. Alice sends the encrypted message to Bob.
4. Bob uses his private key to decrypt and reveal the message.

Authentication

- ★ Authentication is achieved when the encryption process is started with the private key.
- ★ The corresponding public key must be used to decrypt the data.
- ★ Since only one host has the private key, only that host could have encrypted the message, providing authentication of the sender.

Asymmetric Algorithms for Authentication

Private Key (Encrypt) + Public Key (Decrypt) = Authentication



1. Alice encrypts a message with her private key.
2. Alice transmits the encrypted message to Bob.
3. To verify that the message actually came from Alice, Bob requests and acquires Alice's public key.
4. Bob uses the public key to successfully decrypt the message and authenticate that the message did, indeed, come from Alice.

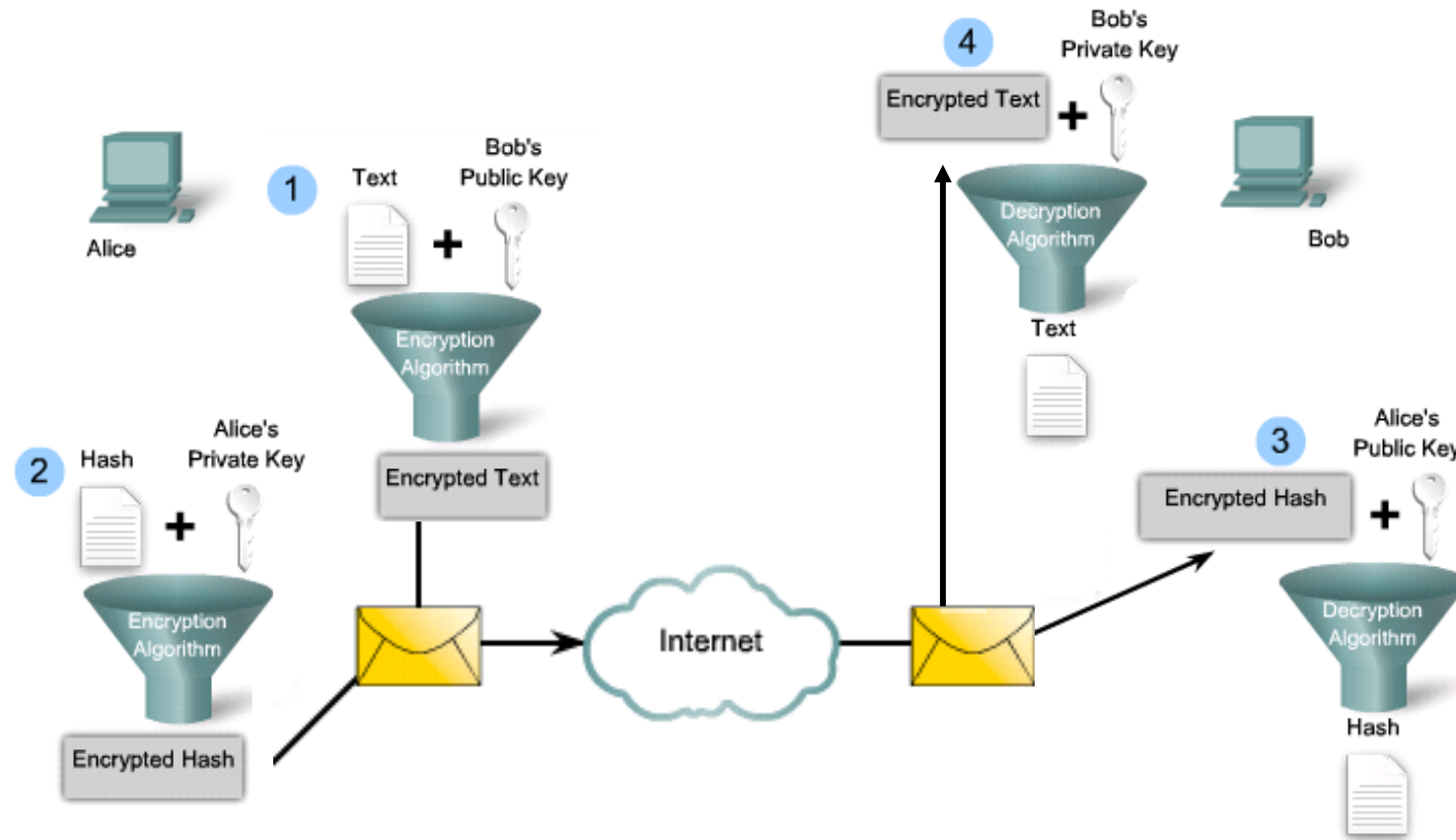
Symmetric Versus Asymmetric Key Algorithms

Asymmetric Algorithms

When sending a message that ensures message confidentiality, authentication and integrity, the combination of two encryption phases is necessary.

- ★ **Phase 1 - Confidentiality**
- ★ **Phase 2 - Authentication and Integrity**

Combining Authentication and Confidentiality



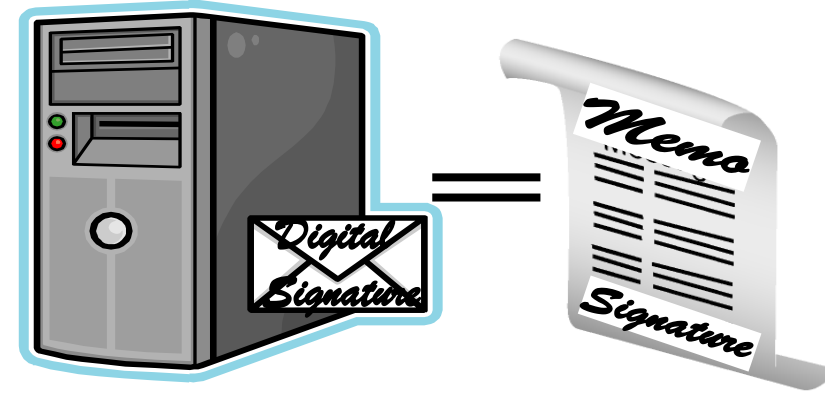
1. Alice encrypts a message using Bob's public key.
2. Alice encrypts a hash of the message using her private key.
3. Bob uses Alice's public key to decrypt and reveal the hash.
4. Bob uses his private key to decrypt and reveal the message.

Using Digital Signatures

- ★ Authenticity of digitally signed data
 - Digital signatures authenticate a source, proving that a certain party has seen and signed the data in question.
- ★ Integrity of digitally signed data
 - Digital signatures guarantee that the data has not changed from the time it was signed.
- ★ Nonrepudiation of the transaction
 - The recipient can take the data to a third party, and the third party accepts the digital signature as a proof that this data exchange did take place.
 - The signing party cannot repudiate that it has signed the data.

Properties

- ★ **The signature is authentic and not forgeable:** The signature is proof that the signer, and no one else, signed the document.
- ★ **The signature is not reusable:** The signature is a part of the document and cannot be moved to a different document.
- ★ **The signature is unalterable:** After a document is signed, it cannot be altered.
- ★ **The signature cannot be repudiated:** For legal purposes, the signature and the document are considered to be physical things. The signer cannot claim later that they did not sign it.



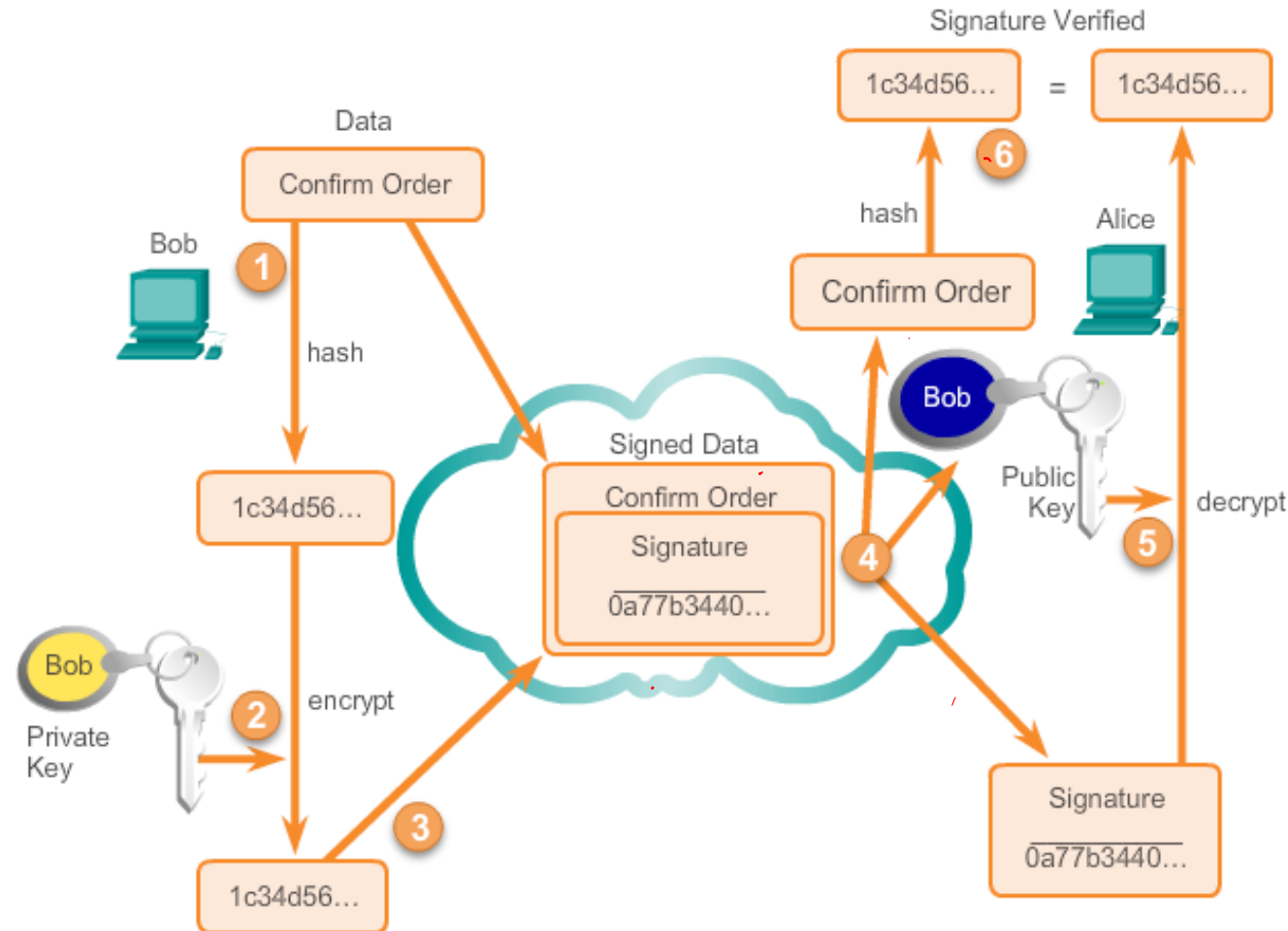
Digital Signature Process

There are six steps to the digital signature process, as shown in the figure (next slide):

1. The sending device, the signer, creates a hash of the document.
2. The sending device encrypts the hash with the private key of the signer.
3. The encrypted hash, known as the signature, is appended to the document.
4. The receiving device, the verifier, accepts the document with the digital signature and obtains the public key of the sending device.
5. The receiving device decrypts the signature using the public key of the sending device. This step unveils the assumed hash value of the sending device.
6. The receiving device makes a hash of the received document, without its signature, and compares this hash to the decrypted signature hash. If the hashes match, the document is authentic; it was signed by the assumed signer and has not changed since it was signed.

Digital Signature Process Cont.

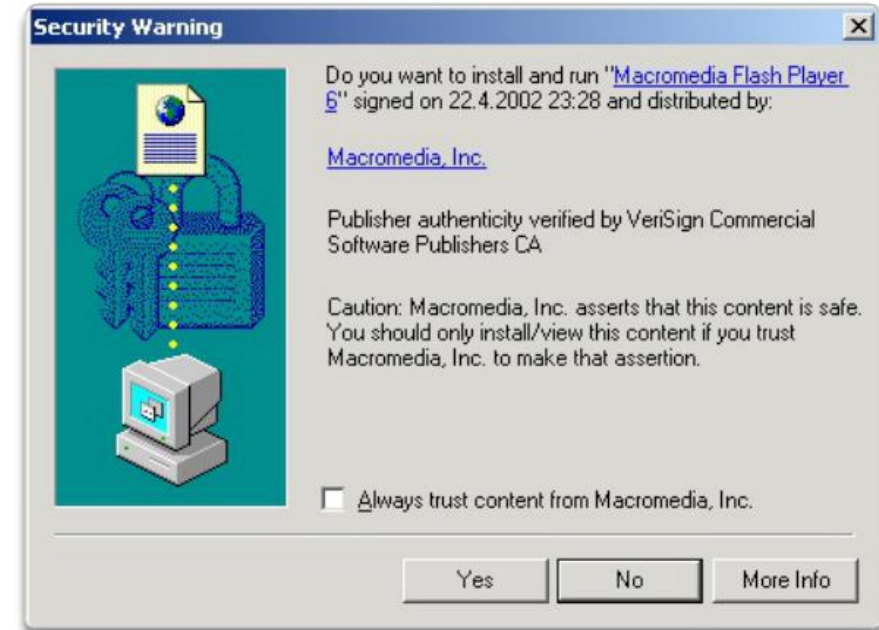
Digital Signature Process



Digitally Signed Code

Digitally signing code provides several assurances about the code:

- The code has not been modified since it left the software publisher.
- The code is authentic and is actually sourced by the publisher.
- The publisher undeniably publishes the code.
- This provides nonrepudiation of the act of publishing.



Symmetric Versus Asymmetric Key Algorithms

Asymmetric Algorithms Cont.

- ★ Well-known asymmetric key algorithms:
 - Diffie-Hellman
 - Digital Signature Standard (DSS), which incorporates the Digital Signature Algorithm (DSA)
 - RSA encryption algorithms
 - ElGamal
 - Elliptical curve techniques

Symmetric Versus Asymmetric Key Algorithms

Asymmetric Algorithms

Algorithm	Key length (in bits)	Description
Diffie-Hellman	512, 1024, 2048	Public key algorithm invented in 1976 by Whitfield Diffie and Martin Hellman that allows two parties to agree on a key that they can use to encrypt messages. Security depends on the assumption that it is easy to raise a number to a certain power, but difficult to compute which power was used given the number and the outcome.
Digital Signature Standard and Digital Signature Algorithm	512 - 1024	Created by NIST and specifies DSA as the algorithm for digital signatures. DSA is a public key algorithm based on the ElGamal signature scheme. Signature creation speed is similar with RSA, but is 10 to 40 times as slow for verification.
RSA encryption algorithms	512 to 2048	Developed by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT in 1977. It is an algorithm for public-key cryptography based on the difficulty of factoring very large numbers. It is the first algorithm known to be suitable for signing and encryption, and is one of the first great advances in public key cryptography. Widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations.
ElGamal	512 - 1024	An asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie-Hellman key agreement. Developed in 1984 and used in GNU Privacy Guard software, PGP, and other cryptosystems. A disadvantage is that the encrypted message becomes very big, about twice the size of the original message, and for this reason, it is only used for small messages, such as secret keys.
Elliptical curve techniques	160	Elliptic curve cryptography was invented by Neil Koblitz in 1987 and by Victor Miller in 1986. Can be used to adapt many cryptographic algorithms, such as Diffie-Hellman or ElGamal. The main advantage of elliptic curve cryptography is that the keys can be much smaller.

Diffie-Hellman Algorithm

- ★ Whitfield Diffie and Martin Hellman invented the Diffie-Hellman (DH) algorithm in 1976.
- ★ The DH algorithm is the basis of most modern automatic key exchange methods and is one of the most common protocols used in networking today.
- ★ DH is not an encryption mechanism
- ★ DH is not typically used to encrypt data.
 - It is a method to securely exchange the keys that encrypt data.
 - This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Diffie-Hellman Algorithm Cont.

- ★ DH is commonly used when data is exchanged using an IPsec VPN, data is encrypted on the Internet using either SSL or TLS, or when SSH data is exchanged.
- ★ It is not an encryption mechanism and is not typically used to encrypt data, because it is extremely slow for any sort of bulk encryption.
- ★ It is common to encrypt the bulk of the traffic using a symmetric algorithm and use the DH algorithm to create keys that will be used by the encryption algorithm.

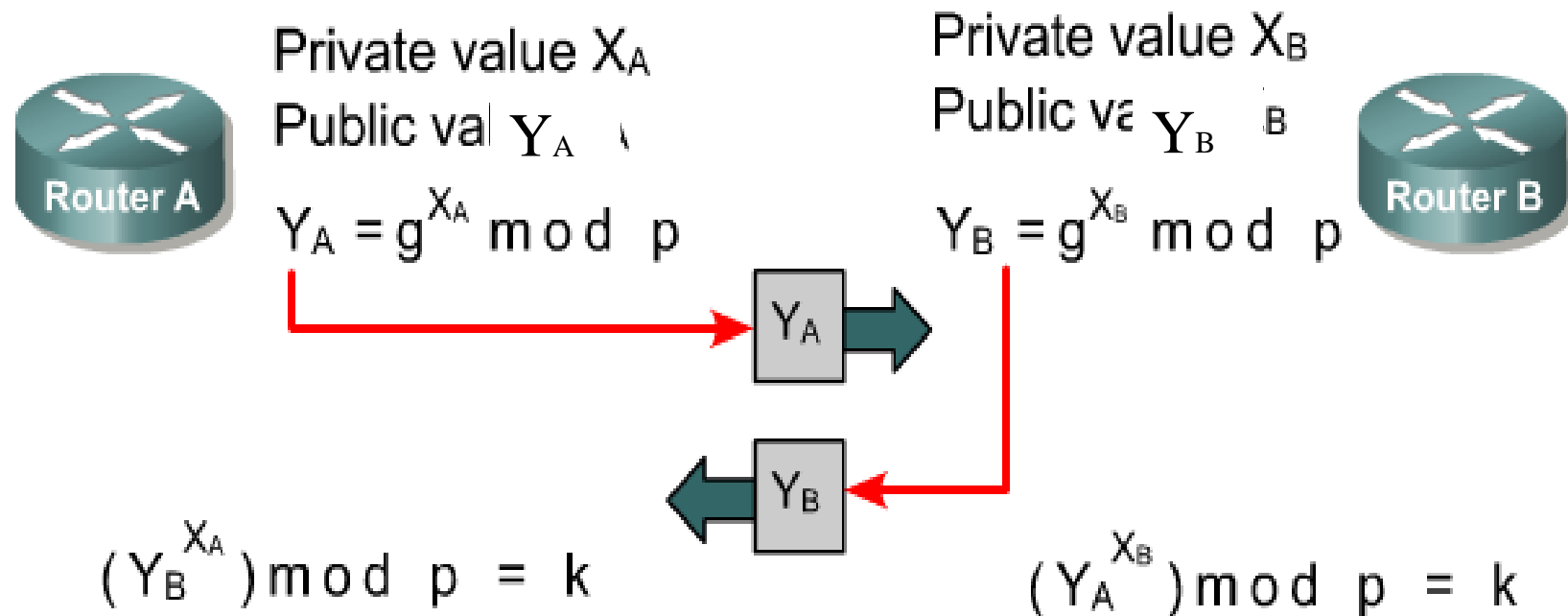
Diffie-Hellman Algorithm Cont.

DH Characteristics

Description	Diffie-Hellman Algorithm
Timeline	1976
Type of Algorithm	Asymmetric
Key size (in bits)	512, 1024, 2048
Speed	Slow
Time to crack (Assuming a computer could try 255 keys per second)	Unknown but considered very safe
Resource Consumption	Medium

DH Operation


Performs authenticated key exchange



DH Operation Cont.

Alice and Bob DH Key Exchange

Alice			Bob		
Shared	Secret	Calc	Shared	Secret	Calc
5, 23			5, 23		
	6	$5^6 \bmod 23 = 8$			



- Bob and Alice agree to use a base number $g=5$ and prime number $p=23$.
- Alice chooses a secret integer $a=6$.
- Alice sends Bob $(g^a \bmod p)$ or $5^6 \bmod 23 = 8$.

DH Operation Cont.

Modulo

- In computing, the modulo operation finds the remainder of division of one number by another.
- Given two numbers, **X** and **Y**, a modulo **N** (abbreviated as a mod N) is the remainder, on division of a by **N**.
- For instance:
 - "**8** mod **3**" would evaluate to **2**.
 - "**9** mod **3**" would evaluate to **0**.

DH Operation Cont.

Alice and Bob DH Key Exchange

Alice			Bob		
Shared	Secret	Calc	Shared	Secret	Calc
5, 23			5, 23		
	6	$5^6 \bmod 23 = 8$			
				15	$5^{15} \bmod 23 = 19$
		$19^6 \bmod 23 = 2$			$8^{15} \bmod 23 = 2$

- Meanwhile Bob chooses a secret integer $b = 15$.
- Bob sends Alice $(g^a \bmod p)$ or $5^{15} \bmod 23 = 19$.
- Alice computes $(x^a \bmod p)$ or $19^6 \bmod 23 = 2$.
- Bob computes $(x^a \bmod p)$ or $8^6 \bmod 23 = 2$.

DH Operation Cont.

Alice and Bob DH Key Exchange

Alice			Bob		
Shared	Secret	Calc	Shared	Secret	Calc
5, 23			5, 23		
	6	$5^6 \bmod 23 = 8$			
				15	$5^{15} \bmod 23 = 19$
		$19^6 \bmod 23 = 2$			$8^{15} \bmod 23 = 2$

- The result (2) is the same for both Alice and Bob.
- They will now use this as the secret key for encryption.

- The initial secret integer used by Alice (**6**) and Bob (**15**) are very, very large numbers (**1,024** bits).
- **8 bits = 10101010**
- **1,024 bits =**



Digital Signature Algorithm

- ★ Well-known asymmetric algorithms, such as RSA or Digital Signature Algorithm (DSA), are typically used to perform digital signing.
- ★ In 1994, the U.S. NIST selected the DSA as the DSS. DSA is based on the discrete logarithm problem and can only provide digital signatures.
- ★ A network administrator must decide whether RSA or DSA is more appropriate for a given situation.
 - DSA signature generation is faster than DSA signature verification.
 - RSA signature verification is much faster than signature generation.

Digital Signature Algorithm Cont.

DSA Scorecard

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

RSA Asymmetric Algorithm

- ★ RSA is one of the most common asymmetric algorithms.
- ★ Ron Rivest, Adi Shamir, and Len Adleman invented the RSA algorithm in 1977.
- ★ Patented public-key algorithm.
 - The patent expired in September 2000.
 - The algorithm is now in the public domain.

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

RSA Summary

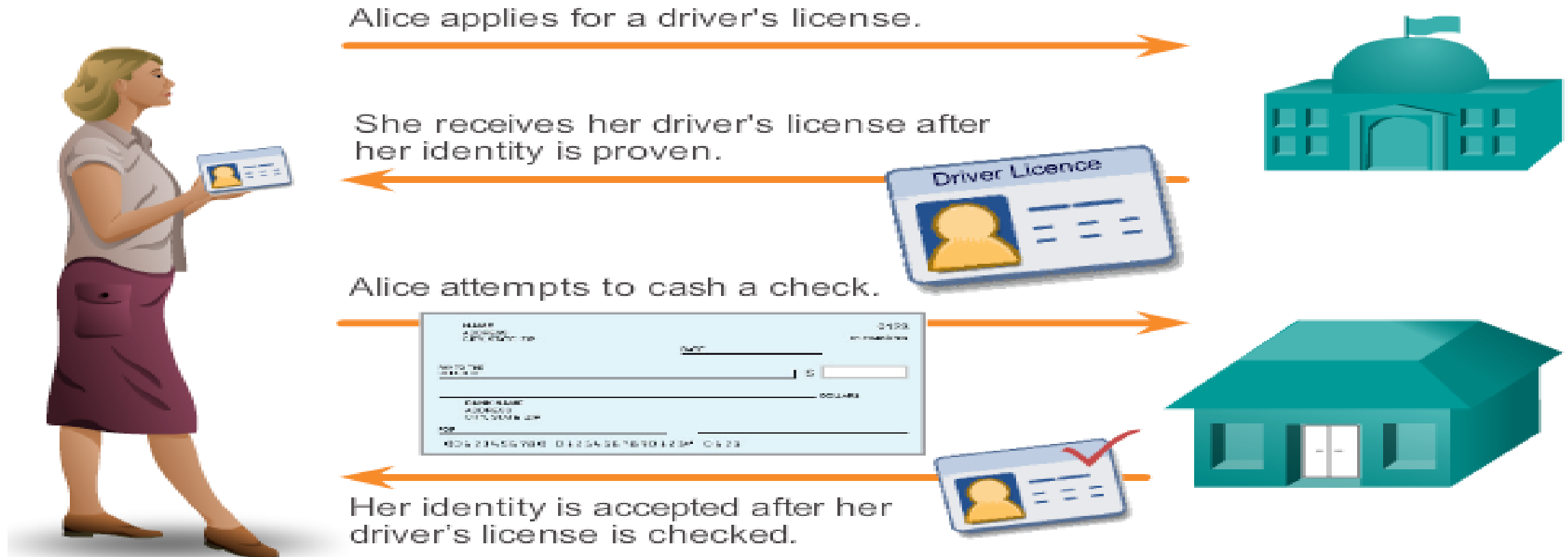
- ★ RSA is about 100 times slower than DES in hardware.
- ★ RSA about 1,000 times slower than DES in software. This performance problem is the main reason that RSA is typically used only to protect small amounts of data.
- ★ RSA is mainly used to ensure confidentiality of data by performing encryption, and to perform authentication of data or nonrepudiation of data, or both, by generating digital signatures.

Public Key Infrastructure Overview

- ★ PKI is the service framework needed to support large-scale public key-based technologies. Scalable solutions that are an extremely important authentication solution for VPNs.
- ★ PKI is a set of technical, organizational, and legal components that are needed to establish a system that enables large-scale use of public key cryptography to provide authenticity, confidentiality, integrity, and nonrepudiation services.
- ★ The PKI framework consists of the hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates.

Public Key Infrastructure Overview

Driver License PKI Analogy



PKI Framework

- ★ PKI Certificates—are published public information containing the binding between the names and public keys of entities.
- ★ PKI Certificate Authority (CA)
 - A trusted third-party entity that issues certificates.
 - A CA always signs the certificate of a user.
 - Every CA also has a certificate containing its public key, signed by itself.
 - This is called a CA certificate or, more properly, a self-signed CA certificate.

Components of a PKI

- ✦ Building a large PKI involves a huge amount of organizational and legal work.
- ✦ There are five main components of a PKI:
 - PKI users, such as people, devices, and servers
 - CAs for key management
 - Storage and protocols
 - Supporting organizational framework, known as practices and user authentication using Local Registration Authorities (LRAs)
 - Supporting legal framework

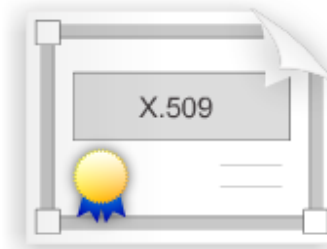
Components of a PKI Cont.

- ★ The trust in the certificate is usually determined by how rigorous the procedure was that verified the identity of the holder when the certificate was issued:
 - Class 0 – Used for testing purposes in which no checks have been performed.
 - Class 1 - Used for individuals with a focus on email.
 - Class 2 - Used for organizations for which proof of identity is required.
 - Class 3 - Used for servers and software signing for which independent verification and checking of identity and authority is done by the issuing certificate authority.
 - Class 4 - Used for online business transactions between companies.
 - Class 5 - Used for private organizations or governmental security.

Interoperability of Different PKI Vendors

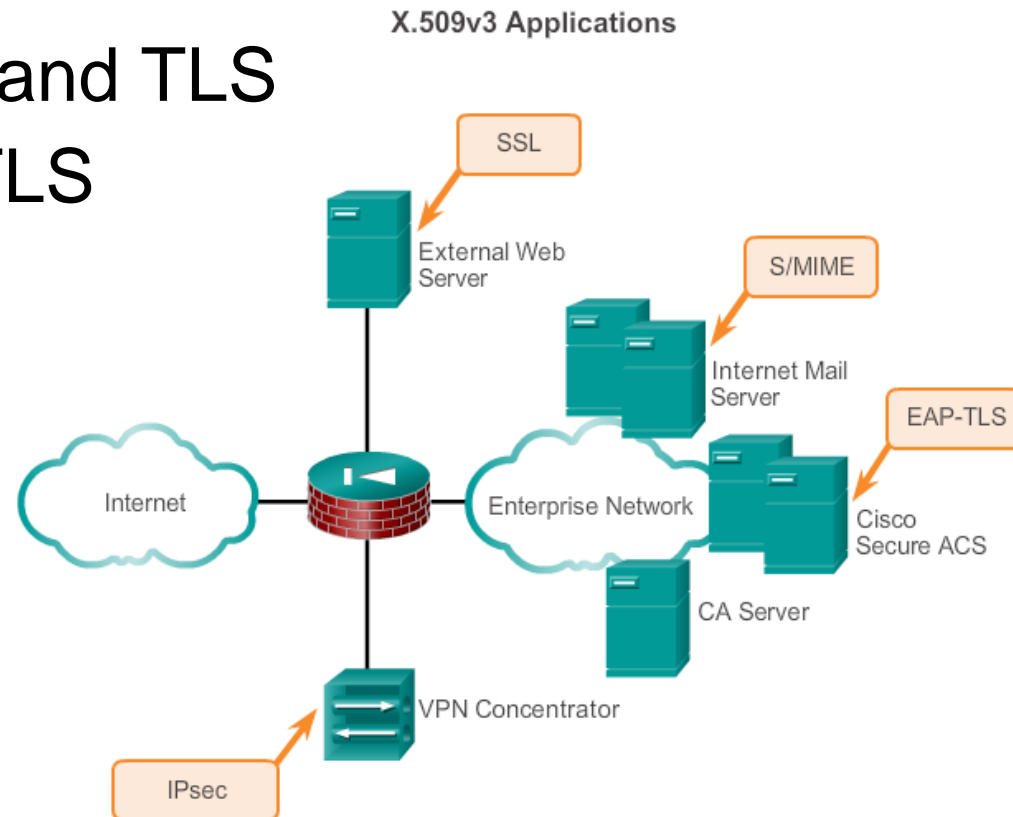
- ✦ Interoperability between different PKI vendors is still an issue.
- ✦ To address this interoperability concern, the IETF formed the Public-Key Infrastructure X.509 (PKIX) workgroup, that is dedicated to promoting and standardizing PKI in the Internet.
- ✦ This workgroup has published a draft set of standards, X.509, detailing common data formats and PKI-related protocols in a network.

IETF PKIX Workgroup



X.509 Standard

- ★ Defines basic PKI formats, such as the certificate and certificate revocation list (CRL) format to enable basic interoperability.
- ★ Widely used for years:
 - Secure web servers: SSL and TLS
 - Web browsers: SSL and TLS
 - Email programs: S/MIME
 - IPsec VPN: IKE



PKI Summary

- ✱ PKI as an authentication mechanism has several characteristics:
 - To authenticate each other, users must obtain the certificate of the CA and their own certificate.
 - Public-key systems use asymmetric keys in which one is public and the other one is private.
 - One of the features of these algorithms is that whatever is encrypted using one key can only be decrypted using the other key.
 - This provides nonrepudiation.
 - Key management is simplified, because two users can freely exchange the certificates.
 - The validity of the received certificates is verified using the public key of the CA, which the users have in their possession.
 - Because of the strength of the algorithms involved, administrators can set a very long lifetime for the certificates, typically a lifetime that is measured in years.

Summary

- ★ Secure communications employs cryptographic methods to protect the confidentiality, integrity, authentication and nonrepudiation of network traffic when traversing the public Internet.
- ★ Cryptology is the combination of:
 - **Cryptography** - Related to the making and using of encryption methods.
 - **Cryptanalysis** - Related to the solving or breaking of a cryptographic encryption method.
- Cryptographic hashes play a vital role when securing network traffic. For example:
 - Integrity is provided by using the MD5 algorithm or the SHA-1 algorithm.
 - Authenticity is provided using HMAC.
 - Confidentiality is provided using various encryption algorithms.

Summary Cont.

- * Encryption can be implemented using a:
 - **Symmetric algorithm** - Various symmetric encryption algorithms can be used, including DES, 3DES, AES, or SEAL.
 - Each option varies with regard to the degree of protection and the ease of implementation.
 - DH is used to support DES, 3DES, and AES.
 - **Asymmetric algorithm** - These can use digital signatures, such as the RSA algorithm, to provide authentication and confidentiality. Asymmetric encryption is usually implemented using PKI.

Questions..?

End of Lecture 8