UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

Prednáška 7
Cryptography
and endpoint protection

Riešenie bezpečnostných incidentov
(CyberOps Associate v1.02)

Mgr. Jana Uramová, PhD.
Katedra informačných sietí
Fakulta riadenia a informatiky, ŽU

Vytvorené v rámci projektu KEGA 026TUKE-4/2021.

Ktorý výsledok pokrýva táto prednáška

Výsledky vzdelávania

Študent po absolvovaní predmetu získa vedomosti a zručnosti potrebné na úspešné zvládnutie úloh, povinností a zodpovedností bezpečnostného analytika v operačnom centre bezpečnosti.
Študent po absolvovaní predmetu bude vedieť:

- Vysvetliť rolu analytika v rámci kybernetickej bezpečnosti
- Vysvetliť prostriedky operačného systému Windows a Linux a charakteristiky pre podporu analýzy v rámci kybernetickej bezpečnosti
- Analyzovať operácie v rámci sieťových protokolov a služieb
- Vysvetliť operácie sieťovej infraštruktúry
- Klasifikovať rôzne typy sieťových útokov
- Použiť sieťové monitorovacie nástroje na identifikáciu útokov proti sieťovým protokolom a službám
- Použiť rôzne metódy na prevenciu škodlivého prístupu do počítačových sietí, k používateľom a k dátam
- Vysvetliť vplyvy kryptografie v rámci monitorovania bezpečnostných sietí
- Vysvetliť, ako skúmať zraniteľnosti a útoky koncových zariadení
- Identifikovať hlásenia v rámci sieťovej bezpečnosti
- Analyzovať sieťovú prevádzku na overenie potencionálneho zneužitia siete
- Aplikovať reakčné modely na incident, a získať prostriedky na manažovanie sieťových bezpečnostných incidentov

- Prerekvizity:
  - Princípy IKS, Počítačové siete 1, Úvod do OS

Preliminary version of topics for lectures

Planning

Week	CyberOps Modules in lectures	Exam from:
1	Chapter 1 The Danger Chapter 2 Fighters in the War Against Cybercrime Chapter 3: The Windows Operating System	none
2	Chapter 4: Linux Overview Chapter 5 Network Protocols Chapter 6 Ethernet and Internet Protocol (IP) Chapter 7 Connectivity Verification Chapter 8 Address Resolution Protocol Chapter 10 Network Services Chapter 11 Network Communication Devices	1-2
3	Chapter 9 The Transport Layer (+nmap) Chapter 12 Network Security Infrastructure	3-4
4	Chapter 13 Attackers and Their Tools Chapter 14 Common Threats and Attacks	5-10

Week	CyberOps Modules in Lectures	Exam from:
5	Chapter 15 Network Monitoring and Tools *(SIEM, SOAR)* Chapter 16 Attacking the Foundation *(L2, L3 protocols vulnerabilities and attacks)* Chapter 17 Attacking What We Do *(L7 vulnerabilities and attacks)*	11-12
6	Chapter 18 Understanding Defense *(security management)* Chapter 19 Access Control *(AAA)* Chapter 20 Threat Intelligence *(commercials, CVE database)*	13-17
7	Chapter 21 Cryptography Chapter 22 Endpoint Protection	18-20
8	Chapter 23 Endpoint Vulnerability Assessment Chapter 24 Technologies and Protocols	none
9	Chapter 25 Network Security Data Chapter 26 Evaualting Alerts (in Security Onion)	21-23
10	Chapter 27 Working with Network Security Data (Security Onion and ELK) Chapter 28 Digital Forensics and Incident Analysis and Response	24-25
11	Expert talk (invited lecture)	26-28

Obsah dnešnej prednášky

Čo prejdeme spolu na prednáške:

- **Chapter 21 Cryptography**
- **Chapter 22 Endpoint Protection**

Module 21:
Cryptography

Module Objective: Explain how the public key infrastructure (PKI) supports network security.

Topic Title	Topic Objective
Integrity and Authenticity	Explain the role of cryptography in ensuring the integrity and authenticity of data.
Confidentiality	Explain how cryptographic approaches enhance data confidentiality.
Public Key Cryptography	Explain public key cryptography.
Authorities and the PKI Trust System	Explain how the public key infrastructure functions.
Applications and Impacts of Cryptography	Explain how the use of cryptography affects cybersecurity operations.

21.1 Integrity and Authenticity
(throught hashes)

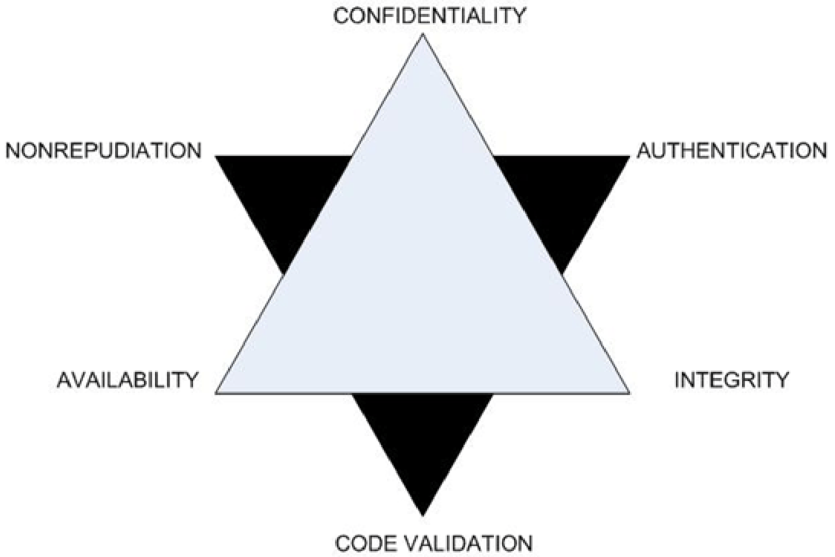
© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Cryptography

Securing Communications

- Dôvernosť (confidentiality)
- Integrita (integrity)
- Dostupnosť (availability)

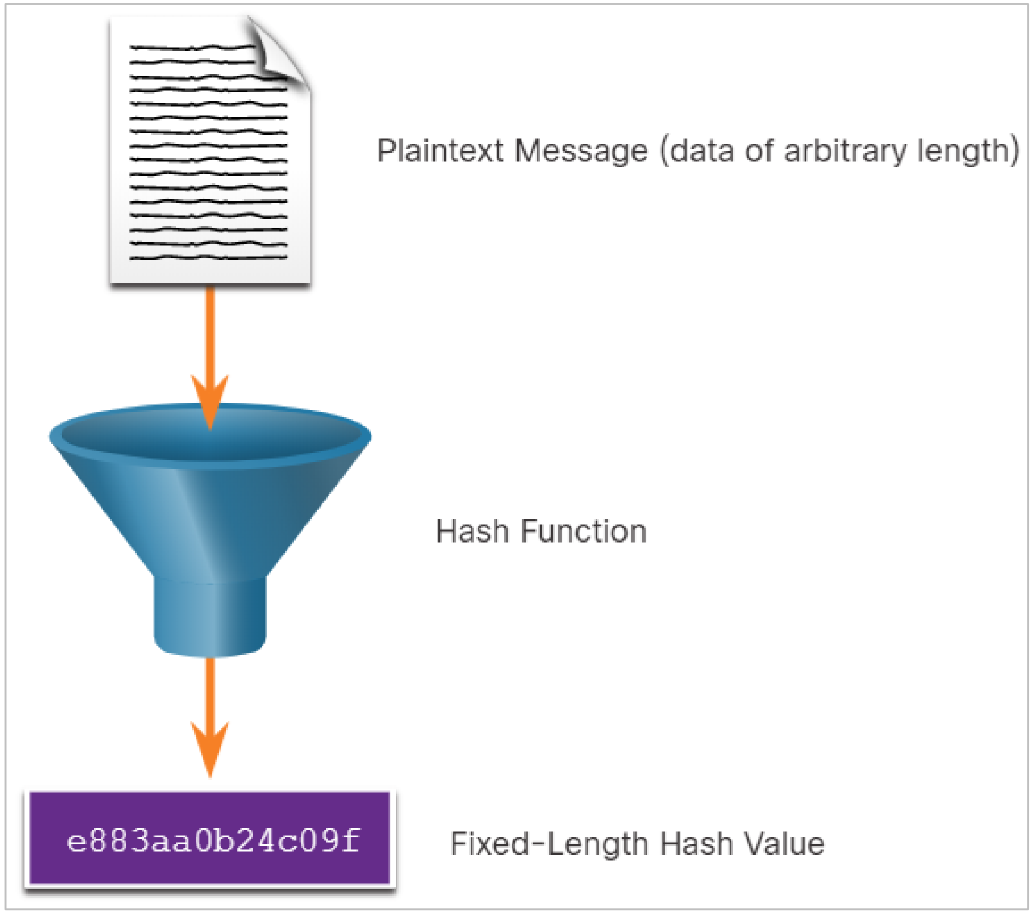
- Autenticita (authenticity)
- Nepopierateľnosť (non-repudiation)
- Správnosť kódu (code validation)



- Organizations must provide support to secure the data **internally** as well as **externally**.
- The four elements of securing communications are:
  - **Data Integrity** - Guarantees that the message was not altered.
  - **Origin Authentication** - Guarantees that the message is not a forgery and it actually comes from whom it states.
  - **Data Confidentiality** - Guarantees that only authorized users can read the message.
  - **Data Non-Repudiation** - Guarantees that the sender cannot repudiate, or refute, the validity of a message sent.

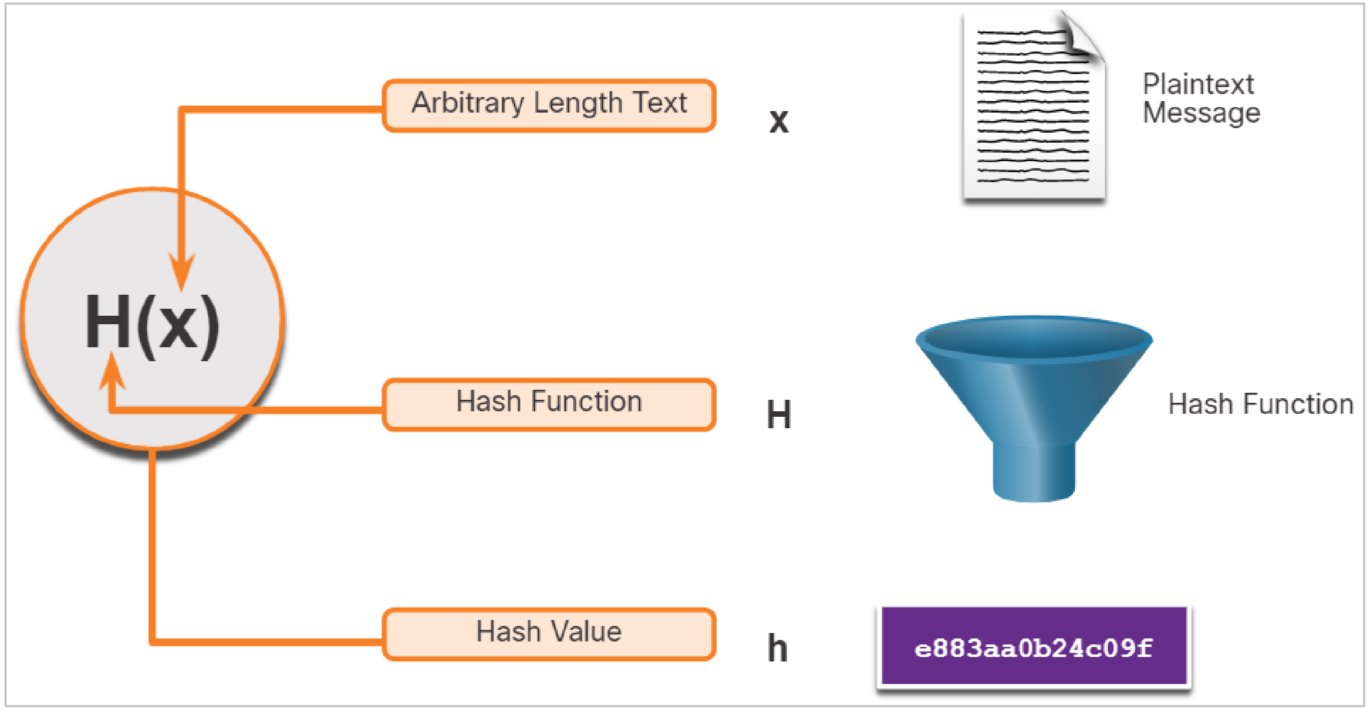
Cryptographic Hash Functions

- Hashes are used to verify and ensure data integrity.
- Hashing is based on a one-way mathematical function that is
  - relatively **easy to compute**,
  - but significantly **harder to reverse.**
- A hash function takes a variable block of **binary data**, called the message, and produces a fixed-length, condensed representation, called the **hash** = message digest, digest, or digital fingerprint.
- it is computationally infeasible *(výpočtovo nemožné)* for 2 different sets of data to come up with the same hash output.
- Every time the data is changed or altered, the hash value also changes.



Cryptographic Hash Functions

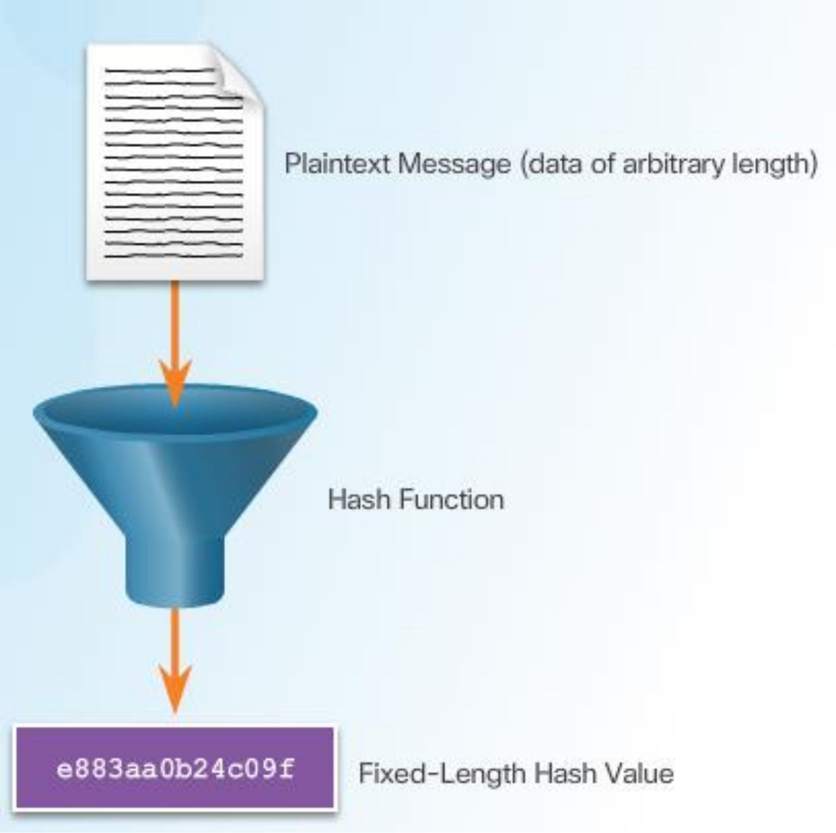
- **Math**
  - *h* = *H*(*x*)
    - *H* takes an input *x* and returns a fixed-size string called the hash value *h*



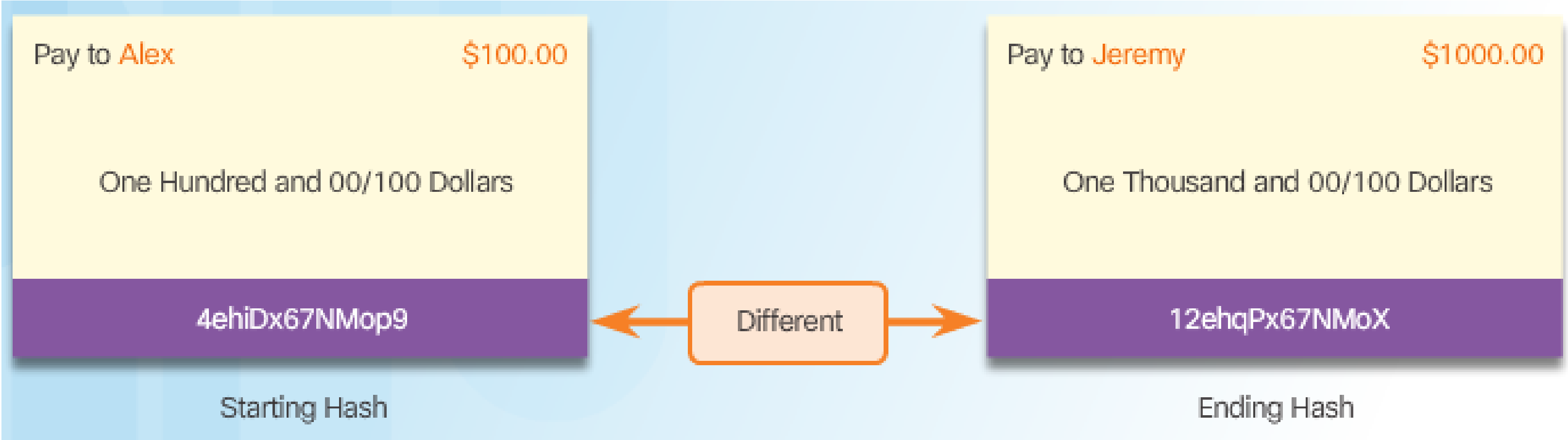
- **Properties**:
  - The input can be any length.
  - The **output** has a fixed length.
  - **H(x)** is relatively easy to compute for given x.
  - **H(x)** is one way and not reversible.
  - **H(x)** is collision free, meaning that two different input values will result in different hash values (hard to find two different input values that result in the same hash value)

Cryptographic Hashes - Hash Function

- Used for
  - Mostly: integrity assurance
  - In some cases: the authenticity
- Application examples
  - Message integrity assurance
    - digitally signed contracts, and public key infrastructure (PKI) certificates, software packages
  - Proof of authenticity
    - when it is used with a symmetric secret authentication key
      - routing protocol auth
      - IPSec
      - PPP CHAP
      - And others



Integrity - Well-Known Hash Functions



- Hash function
  - Helpful: Indicates accidental changes (results of error)
  - Attention: Cannot be used to guard against deliberate changes
    - There is no possibilities to indicate hash originality by hash function itself
    - Anyone may intercept the message, change the content and recalculate and append a new hash
      - MitM attacks
- Well known hash functions
  - MD5 with 128-bit digest
  - SHA-256 with 256-bit digest

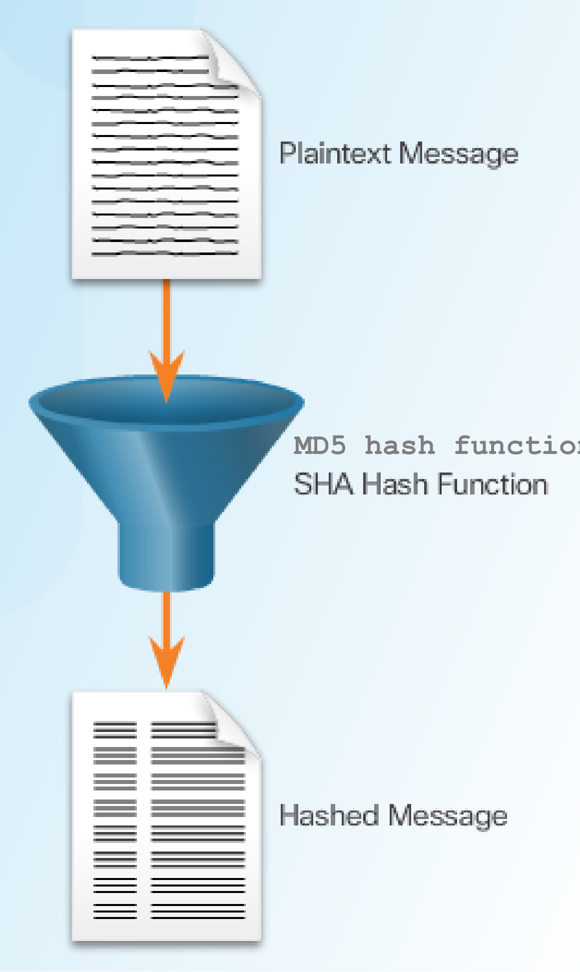
Message Digest 5 Algorithm / Secure Hash Algorithm

MD5

- Developed by Ron Rivest
- Used in a variety of Internet applications today
- Now considered as a legacy algorithm
  - The usage should be avoided
  - Use only when no better alternatives are available

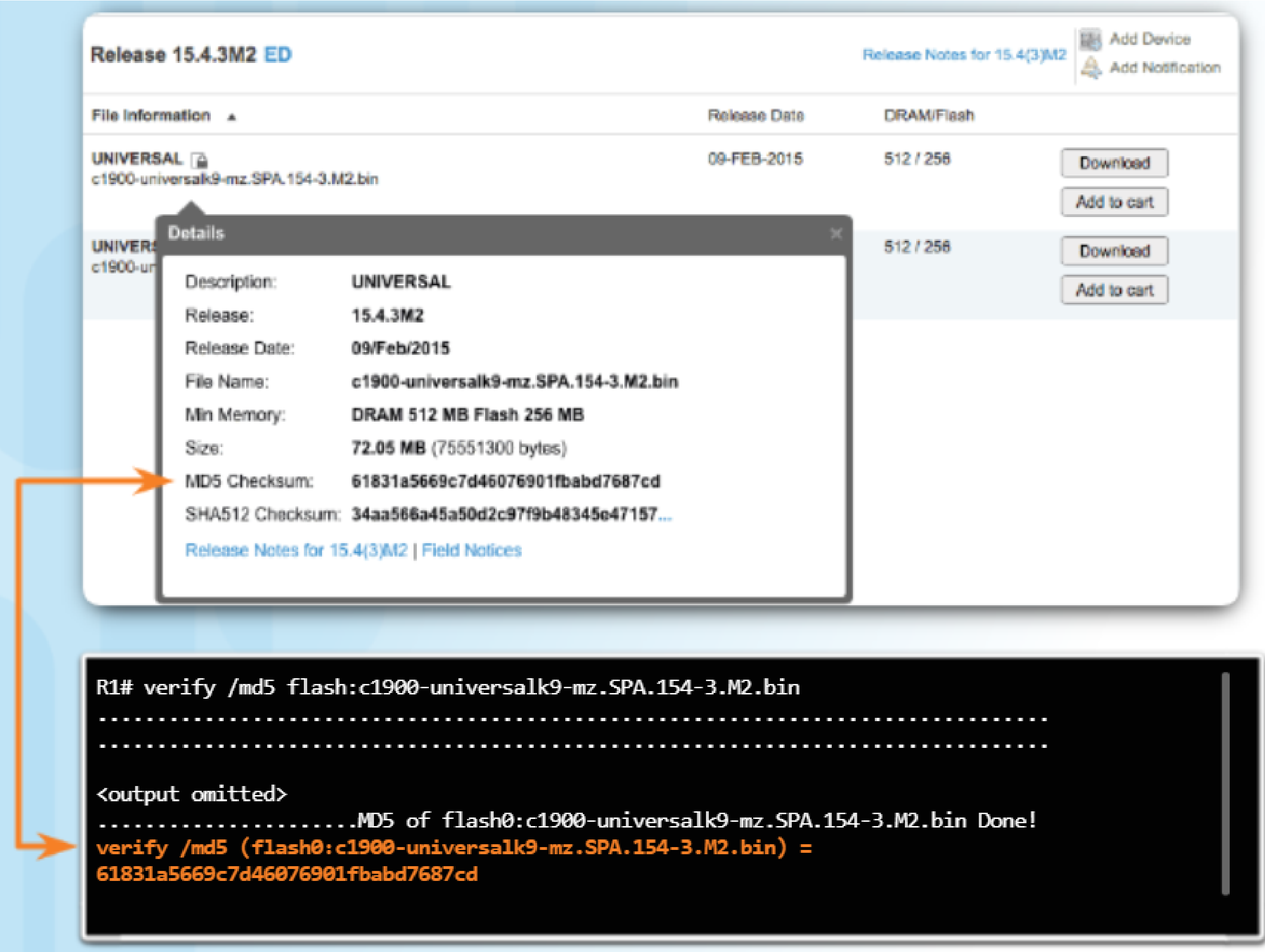
SHA

- Developed by The U.S. National Institute of Standards and Technology (NIST)
- Design is very similar to the MD5
  - But little bit slower
- Two versions
  - SHA-1
    - takes a message of less than 2^64 bits in length and produces a 160-bit message digest
    - Considered legacy, avoid use it
  - SHA-2 family
    - SHA-224 (224 bit)
    - SHA-256 (256 bit)
    - SHA-384 (384 bit)
    - SHA-512 (512 bit)



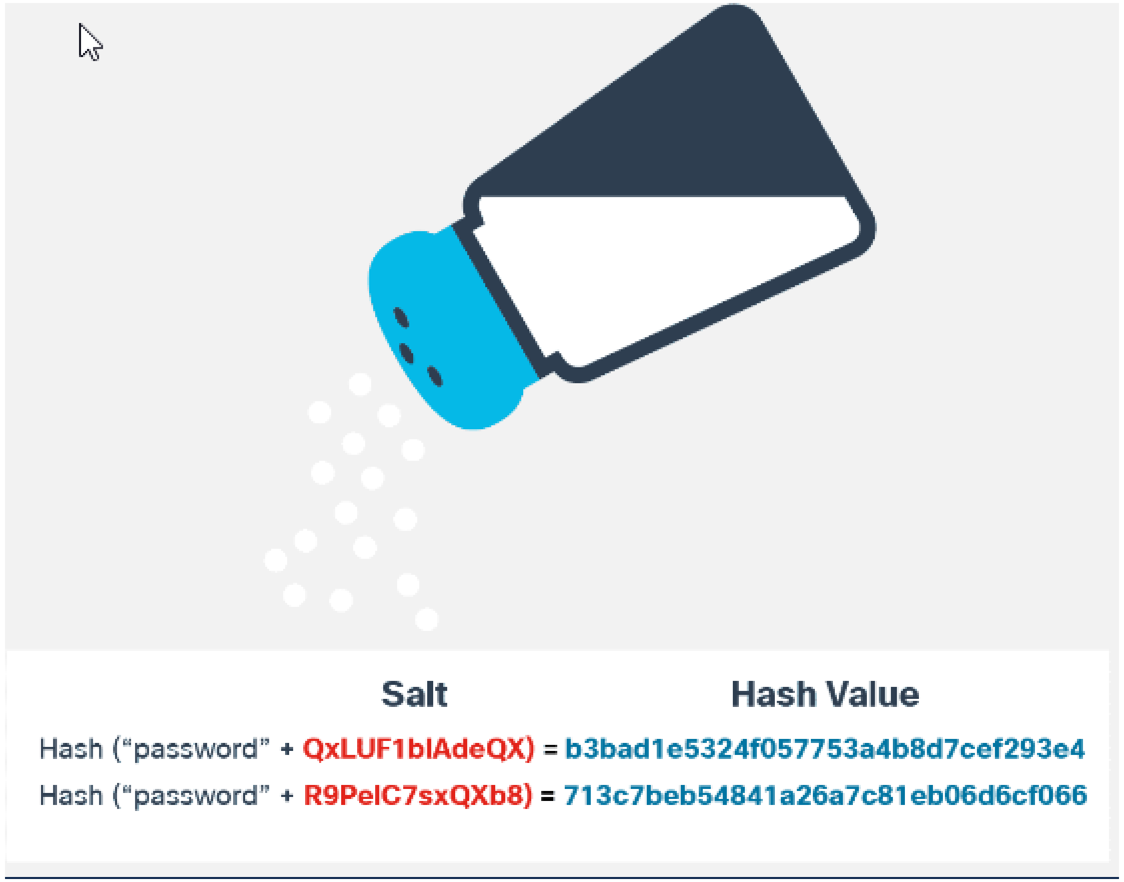
MD5 Versus SHA

- MD5, SHA-1
  - Faster,
  - but security flaws were discovered
  - Not recommended
- Good practise:
  - SHA-256 or higher
- Cisco signs IOS images
  - **`Verify /md5`**



Salting

- Techniques that improve hashing
- Salt
  - Randomly generated string used as an additional input to key
  - Stored in DB with keys/passwd then
  - Good way
    - use cryptographically secure pseudo-random number generator (CSPRNG)
    - Never use the same slat twice

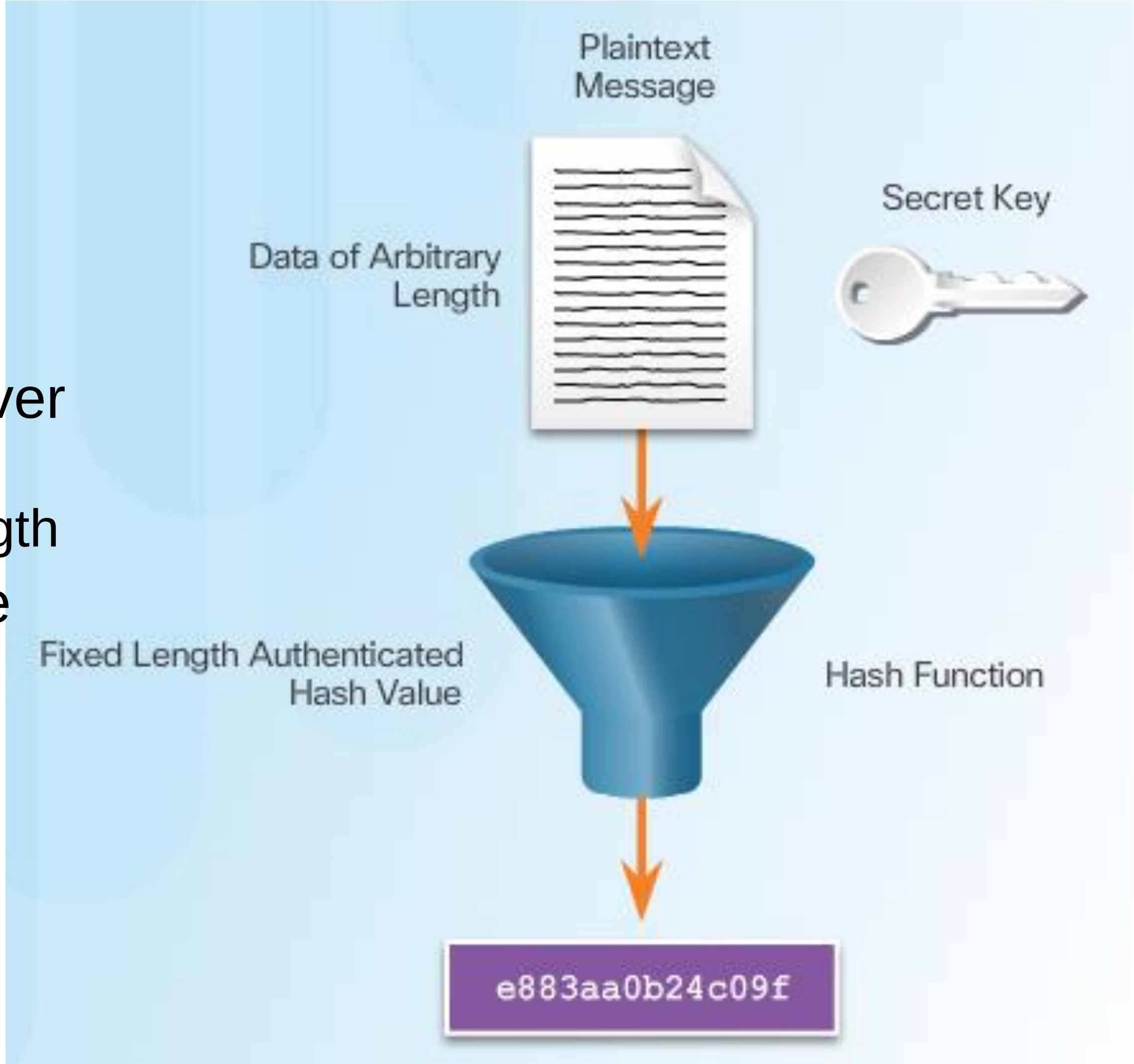


Authenticity with HMAC

HMAC or KHMAC

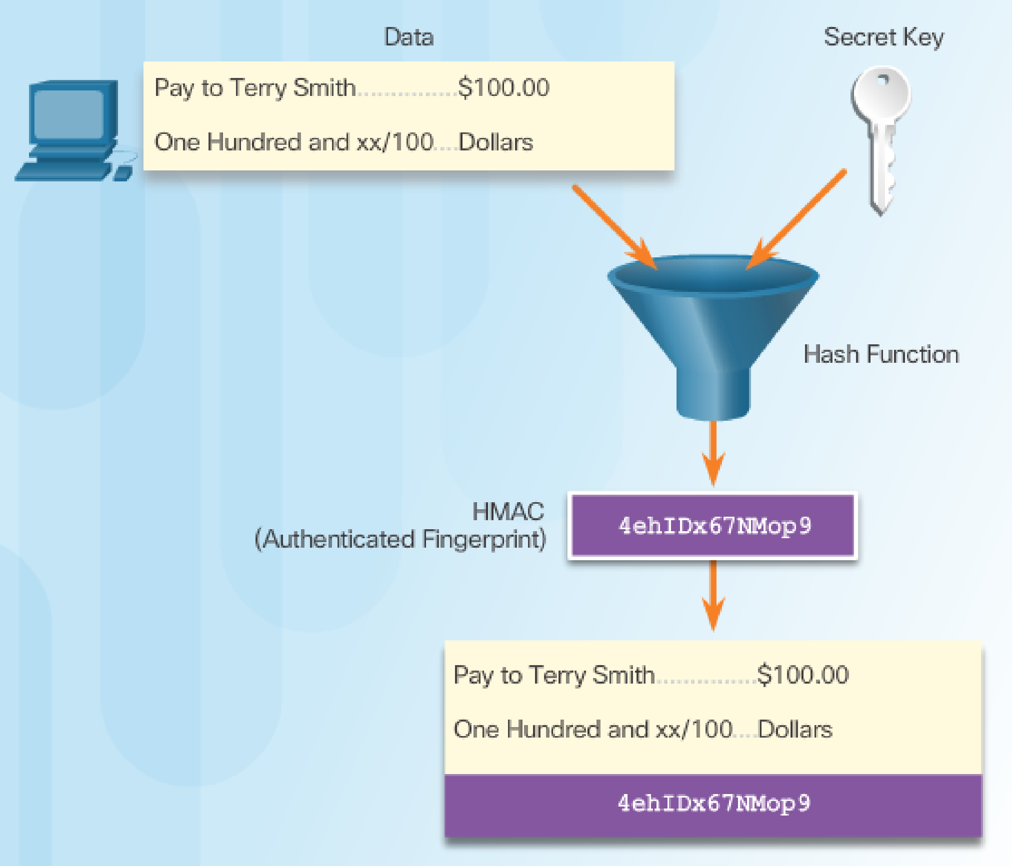
Keyed-Hash Message Authentication Code

- Types of message authentication code (MAC) functions
- Use an additional shared secret key as input to the hash function
  - Key has to be known to sender and receiver
    - Only the same inputs (message + key) produce the same Hash (digest) – fixed length
  - Adds authentication to integrity assurance
    - Protects againt MiTM
- Two well-known HMAC functions
  - Keyed MD5 (HMAC-MD5)
  - Keyed SHA-1 (HMAC-SHA-1)
- Doubts?
  - How to distribute secret keys

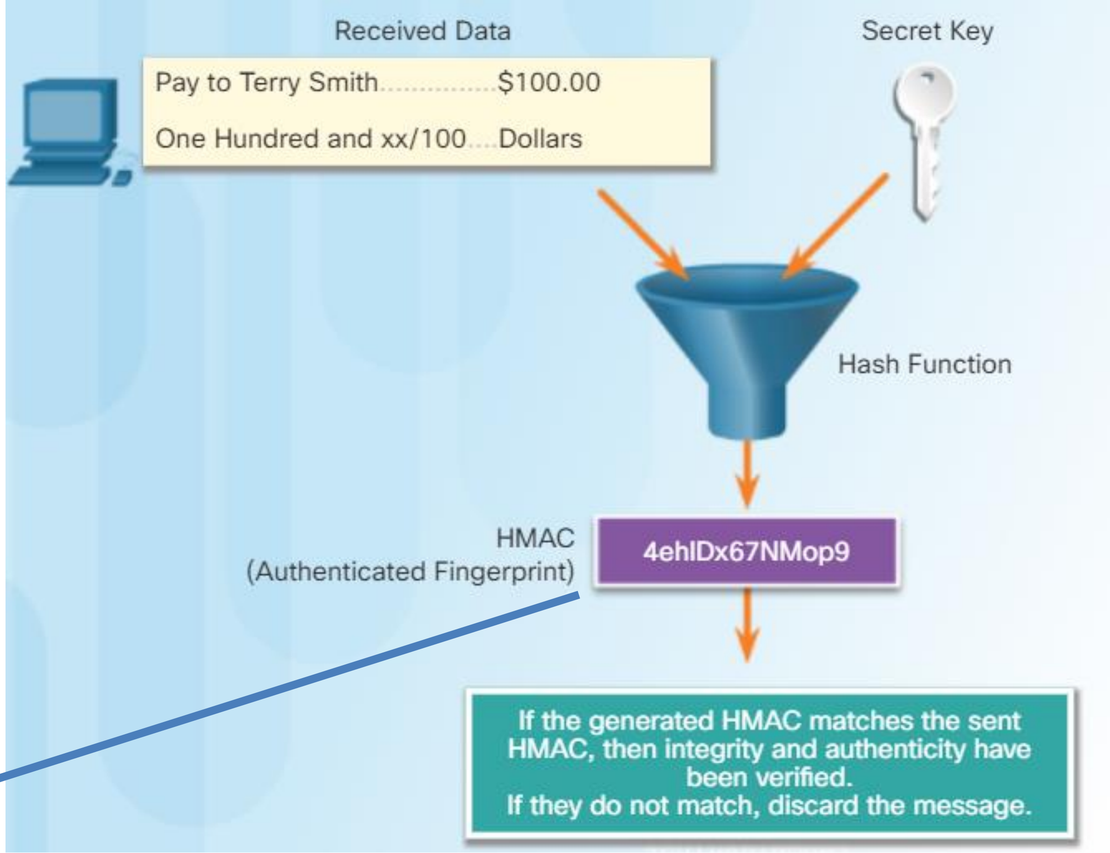


HMAC Operation

Sender: Creating

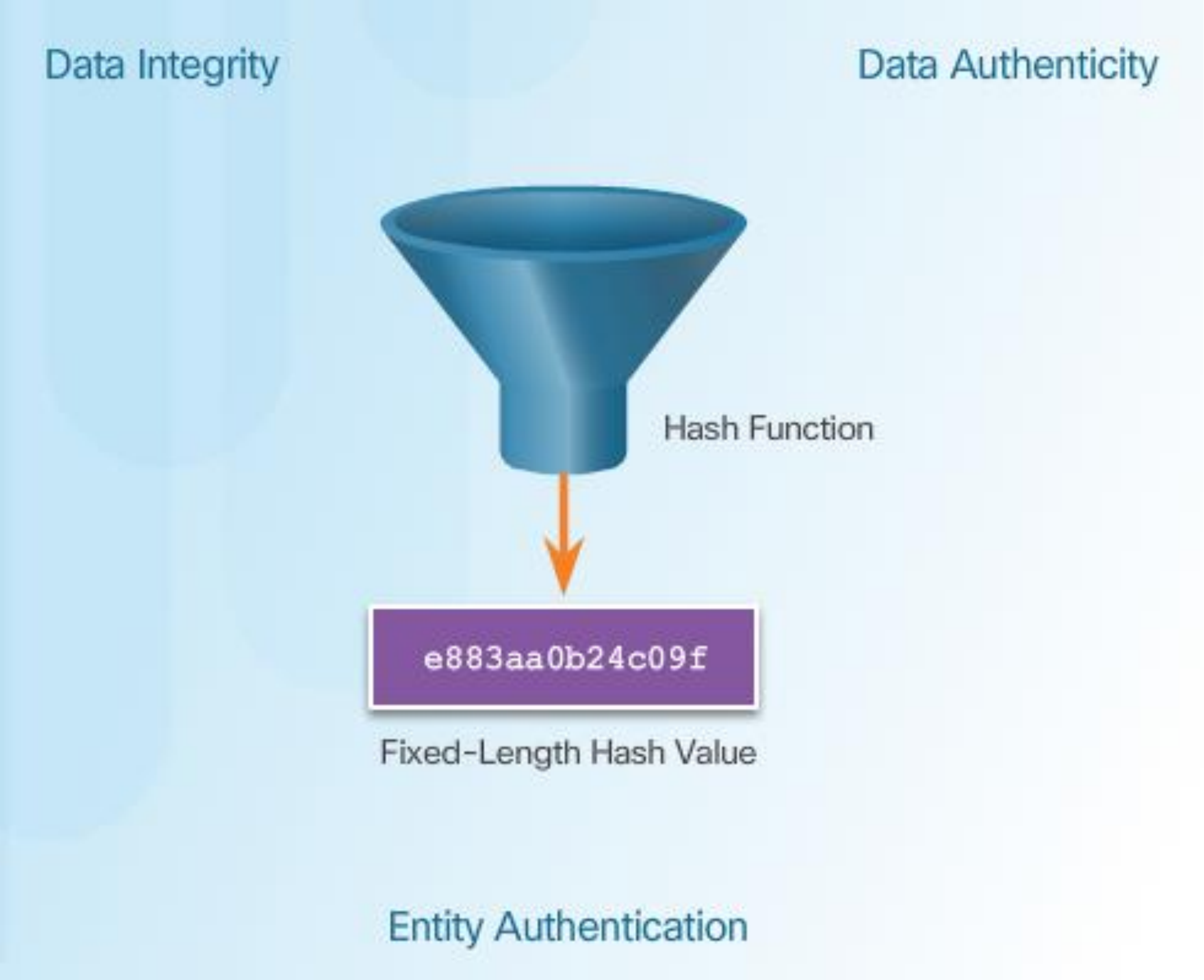


Receiver: Verifying



Hashing in products

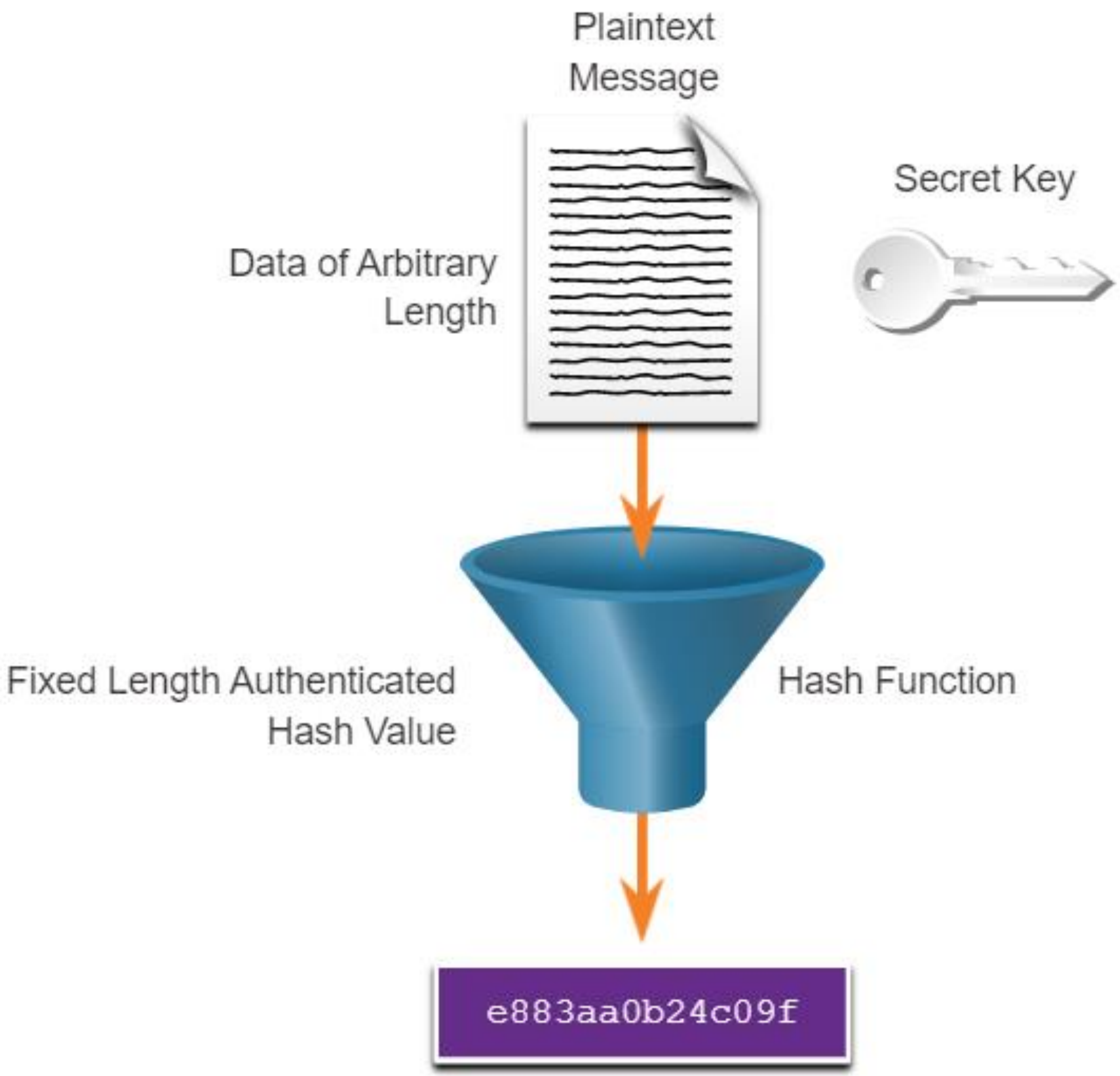
- Cisco uses hashing for entity authentication, data integrity, and data authenticity in products:
  - Router Cisco IOS
    - Authenticity of routing protocol updates
      - HMAC like MD5
      - RIPv2/ng, EIGRP, OSPF
  - IPSec gateways and clients
    - packet integrity and authenticity (MD5 and SHA-1)
  - Cisco IOS images verification
- Vendor neutral:
  - SSL, IPsec, SSH



Origin Authentication

HMAC Hashing Algorithm

- An HMAC is calculated using any cryptographic algorithm that combines a cryptographic hash function with a secret key.
- Only the sender and the receiver know the secret key, and the output of the hash function depends on the input data and the secret key.
- Only parties who have access to that secret key can compute the digest of an HMAC function.
- If two parties share a secret key and use HMAC functions for authentication, a properly constructed HMAC digest of a message that a party has received indicates that the other party was the originator of the message.



Origin Authentication

Cisco Router HMAC Example

A. In the figure, HMACs are used by Cisco routers that are configured to use Open Shortest Path First (OSPF) routing authentication.

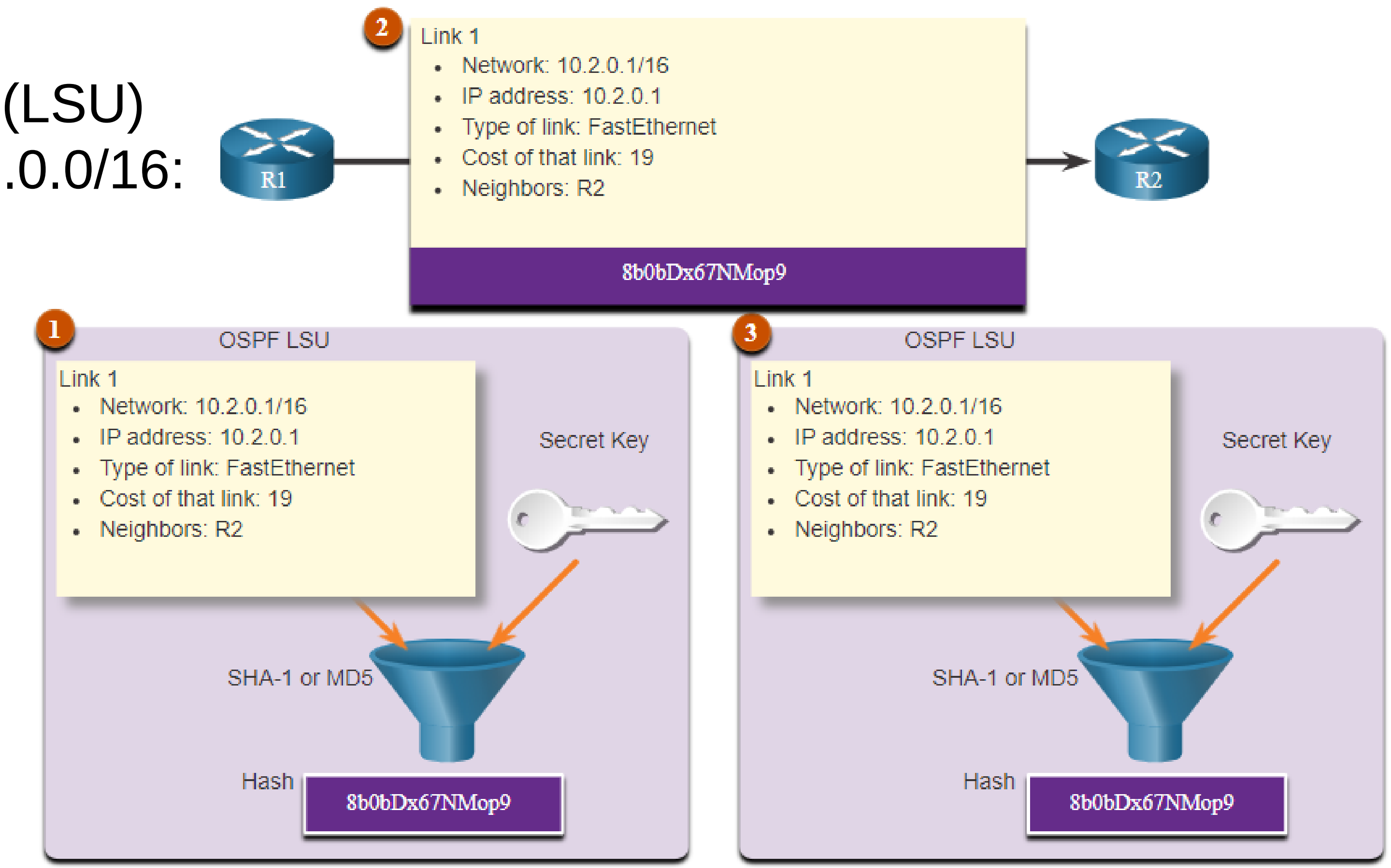
B. R1 is sending a link state update (LSU) regarding a route to network 10.2.0.0/16:

   a. R1 calculates the hash value using the LSU message and the secret key.

   b. The resulting hash value is sent with the LSU to R2.

   c. R2 calculates the hash value using the LSU and its secret key. R2 accepts the update if the hash values match. If they do not match, R2 discards the update.

LSA Type 1:	Router LSA
LSA Type 2:	Network LSA
LSA Type 3:	Summary LSA
LSA Type 4:	Summary ASBR LSA
LSA Type 5:	Autonomous system external LSA
LSA Type 6:	Multicast OSPF LSA
LSA Type 7:	Not-so-stubby area LSA
LSA Type 8:	External attribute LSA for BGP



KIS FRI UNIZA

21.2 Confidentiality
(throught encryption)

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Classes of Encryption Algorithms

- Two approaches to ensure security using encryption
  - Protect **algorithm**
    - Secrecy of the algorithm itself; reveal of algorithm ==> need to change algorithm
  - Protect **keys**
    - Algorithms are publicly available and well know
- 2 classes of encryption which differ in how they use keys:

Symmetric Encryption Algorithm	Asymmetric Encryption Algorithm
Best known as shared-secret key algorithms.	Best known as public key algorithms.
The usual key length is 112 to 256 bits.	The usual key length is 512 to 4,096 bits.
A sender and receiver must share a secret key.	A sender and receiver do not share a secret key.
Algorithms are usually quite fast (wire speed) because they are based on simple mathematical operations.	Algorithms are relatively slow because they are based on difficult computational algorithms.
Examples include DES, 3DES, AES, IDEA, RC2/4/5/6, and Blowfish.	Examples include RSA, ElGamal, elliptic curves, and DH. and PKI

classified as a block cipher or a stream cipher
for encryption bulk data such as in VPN traffic

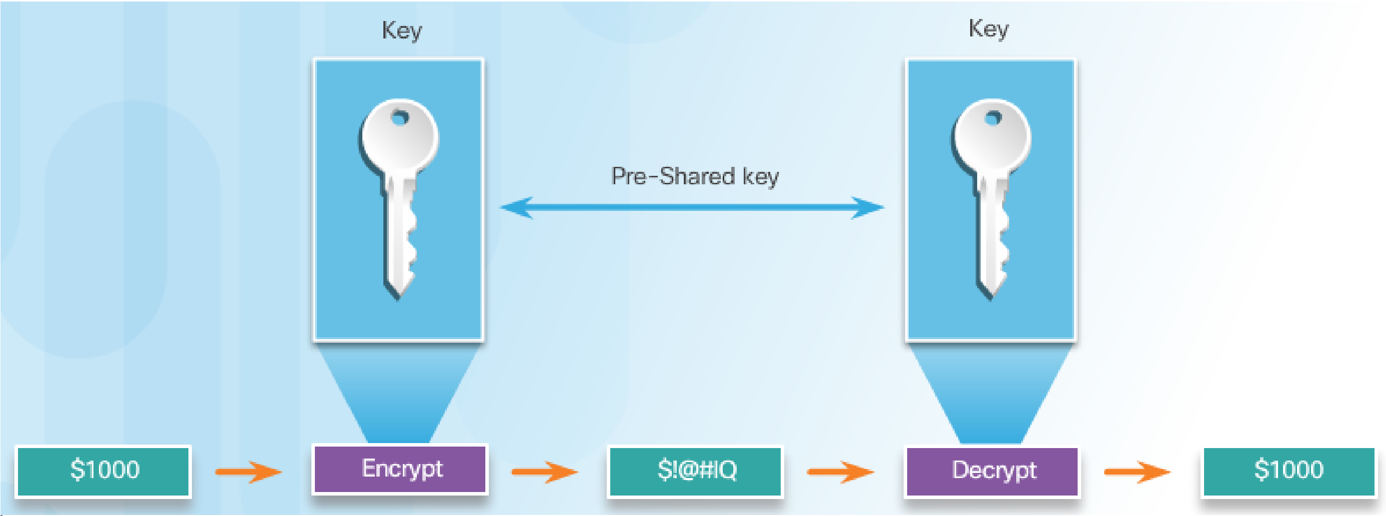
for quick data transmissions such as HTTPs

KIS FRI UNIZA

Symmetric Encryption Algorithm

Symmetric Encryption

- Or secret key encryption
  - Encryption and decryption **use the same key**
- Popular and commonly used
  - Fast and speedy
    - Simpler math operations
    - Shorter keys
    - Easily accelerated in HW
    - Almost wire speed encryption
      - Useful for VPN and VoIP for example
  - Exists many of symmetric encryption algorithms
- Problems
  - Key distribution
    - Sender and the receiver must exchange keys somehow
    - Ideally using secured channel
  - Security of keys is the point

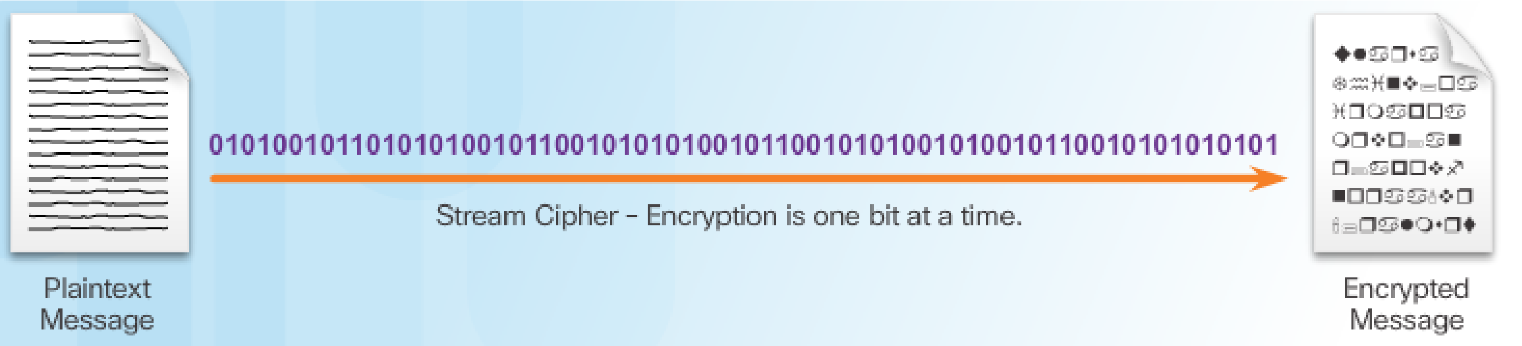


Symmetric Encryption Algorithm	Key Length (in bits)
DES	56
3DES	112 and 168
AES	128, 192, and 256
Software Encryption Algorithm (SEAL)	160
The RC series	RC2 (40 and 64) RC4 (1 to 256) RC5 (0 to 2040) RC6 (128, 192, and 256)

Symmetric Encryption Algorithm

Symmetric Block Ciphers and Stream Ciphers

- Are the most commonly used techniques of symmetric algorithm
- **Block cipher**
  - Transforms a fixed-length block of plaintext (chunks) into a common block of ciphertext
    - 64 or 128 bits
  - Block is encrypted at one time
  - Increase data size
    - Output is bigger as input (multiples of block size)
- **Stream Cipher**
  - Encrypts message per bit/byte
    - *„Block cipher with a block size of 1 bit (or byte)“*
    - Faster than block ciphers, continuous



- Much faster then block ciphers
- Do not increase output size
  - Example is RC4 and A5 – used to encrypt GSM cell phone communication, DES, … Vigenère cipher
- Periodic
  - the key is of finite length
  - Then is repeating (if message is bigger)

Choosing an Encryption Algorithm

- Administrator decision
- Criteria
  - The algorithm is **trusted** by the cryptographic community
    - Older and mature algorithm are more trusted
  - The algorithm adequately **protects against brute-force** attacks
  - The algorithm supports **variable and long key** lengths and scalability
  - The algorithm does not have **export or import restrictions**
    - For example outside of U.S.

	DES	3DES	AES
The algorithm is trusted by the cryptographic community.	Replaced by 3DES	Yes	Ongoing Evaluation
The algorithm adequately protects against brute-force attacks.	No	Yes	Yes

DES Symmetric Encryption

Parity bit examples

sequence of seven bits	with eighth even parity bit:	with eighth odd parity bit:
0100010	01000100	01000101
1000000	10000001	10000000

ComputerHope.com

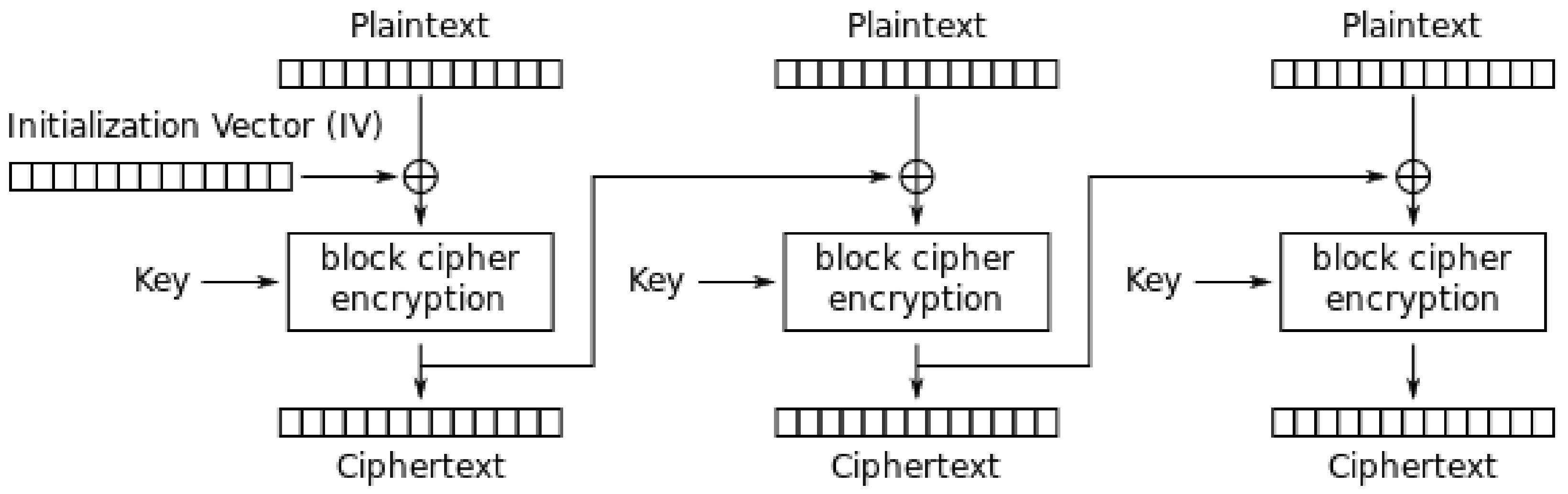
- Well-known, legacy (1976), symmetric, block cipher (64bit), fixed key size of **64bit**
  - Too insecure for most of current applications
    - Feasilibity of brute-force attacks (or faster types as differential cryptanalysis (DC), linear cryptanalysis, and staticitical cryptanalysis - Davies' attack)
  - Stronger encryption: 56 bits is used for encryption, 8 bits for odd parity of key integrity
  - Weaker encryption: 40bit key, 16 known bit
- Essentially a sequence (16 rounds) of permutations and substitutions of data bits combined with an encryption key

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

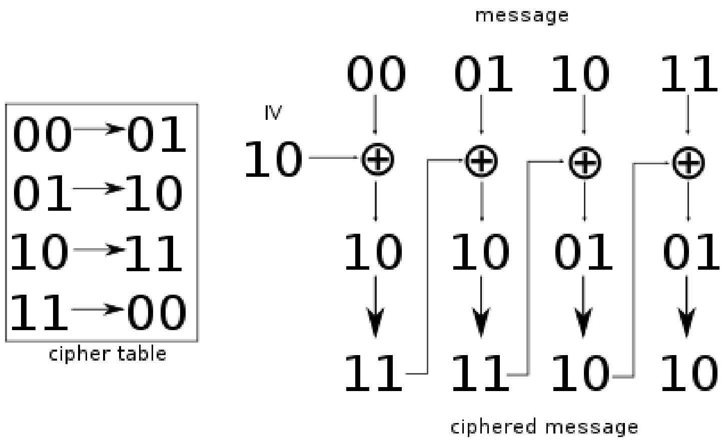
- Summary
  - Insecure, not recommended
  - But if used
    - Change keys frequently
    - Test for a weak key
      - It has 4 weak keys and 12 semi-weak keys
    - Use secure channel for key exchange
    - Use DES in CBC mode of operation

Weak keys for DES: https://crypto.stackexchange.com/questions/12214/can-you-explain-weak-keys-for-des

DES in CBC mode of operation



Cipher Block Chaining (CBC) mode encryption



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

$A \oplus B = Q$

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_block_chaining_(CBC

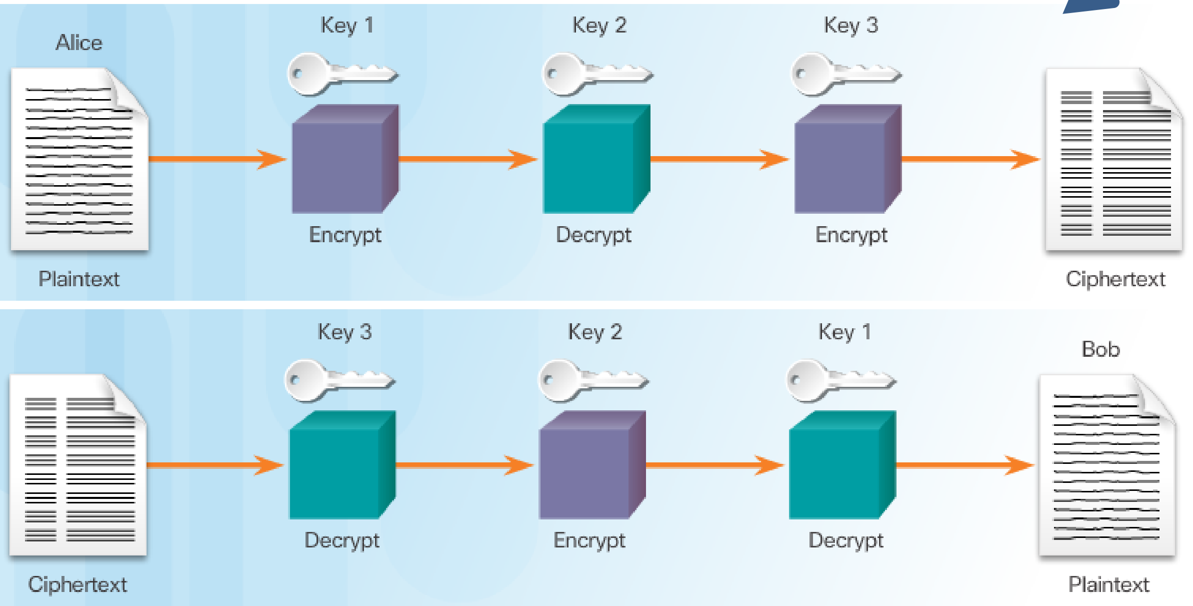
Symmetric Encryption Algorithm

Improved DES => 3DES

- 3DES – improvement of DES algorithm (1977)
  - Called 3DES-Encrypt-Decrypt-Encrypt, Triple DES (TDES), officially the Triple Data Encryption Algorithm (TDEA or TripleDEA),
  - It applies DES three times in a row to a plaintext block
  - no longer considered adequate in the face of modern cryptanalytic techniques and supercomputing power (35years of testing DES and 3DES)
    - CVE released in 2016, *CVE-2016-2183*
    - CVE, combined with the inadequate key size of DES and 3DES => NIST has deprecated DES and 3DES for *new* applications in 2017, and for *all* applications by the end of 2023

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

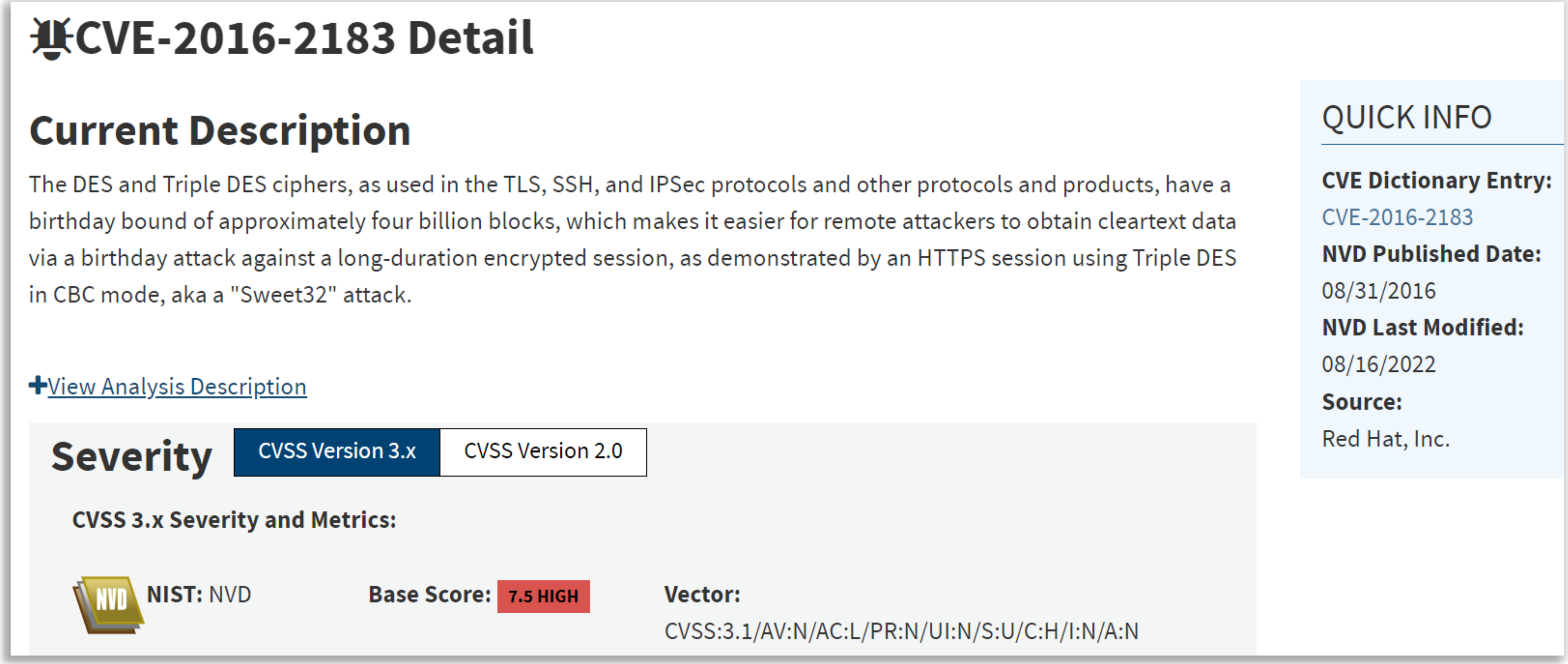
- Key size is 64 (56 bits), but
- Uses several keys K1, K2, K3 and three options
  - **Keying option 1,** Key size 168 (3*56bits)
    - All three keys are independent (3TDEA), best security
  - **Keying option 2,** Key size 112 (2*56bits)
    - K_1 and K_2 are independent, and $K_3 = K_1$ (2TDEA)
  - **Keying option 3,** Key size 56 (1*56bits)
    - All three keys are identical, worst security



Symmetric Encryption Algorithm

3DES

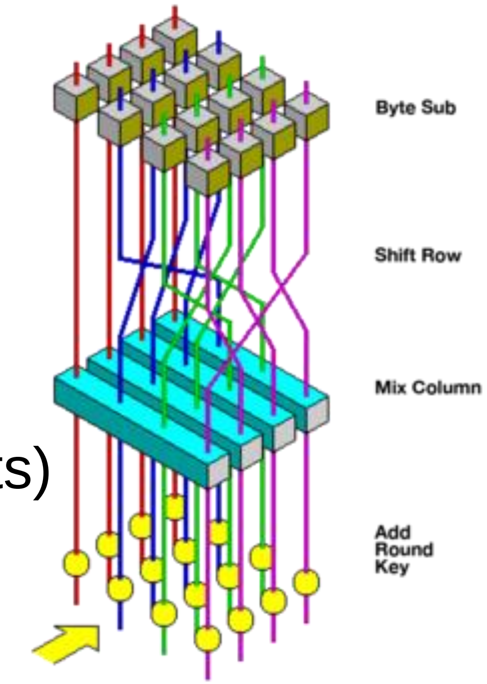
- no longer considered adequate in the face of modern cryptanalytic techniques and supercomputing power (35years of testing DES and 3DES)
  - CVE released in 2016, *CVE-2016-2183*
  - CVE, combined with the inadequate key size of DES and 3DES => NIST has deprecated DES and 3DES for *new* applications in 2017, and for *all* applications by the end of 2023



It has been replaced with the more secure, more robust AES

Symmetric Encryption Algorithm

Advanced Encryption Standard (AES)



- Proposed as a replacement for DES in 1998
  - Based on Rijndael iterated block cipher
- Uses variable block length and key length (both of 128, 192 or 256-bits)
  - But can be extended in multiples of 32 bits
- As secure (or better) as 3DES, and much faster
  - Support longer keys
  - Supports efficient implementation in hardware or software on a range of processors
    - Suitable for high-throughput, low-latency environments (especially sw. based)
  - But is **younger as DES** ! (less trustworthy)
- Used worldwide
  - Some attack have been published
    - But are not yet computationally feasible

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

Known attacks

- a **cryptographic "break"** is anything faster than a **brute-force** attack (i.e., performing one trial decryption for each possible key in sequence)
  - A break can thus include results that are infeasible (neuskutočiteľné) with current technology
  - Despite being impractical, theoretical breaks can sometimes provide insight into vulnerability patterns (for AES several, but no real/practical break)
- **126-bit key** would still take **billions of years** to brute force on current and foreseeable hardware
- In November 2009, the first known-key distinguishing attack
  - authors calculate the best attack using their technique on AES with a 128-bit key requires storing 2^{88} **bits of data**
    - that works out to about 38 trillion terabytes of data, which is more than all the data stored on all the computers on the planet in 2016 ☺
      - as such, there are no practical implications on AES security

Know-key (choosen-key) distinguishing attacks

- These attacks do not directly compromise the confidentiality of ciphers, because in a classical scenario, the key is unknown to the attacker.
- They apply in the "open key model" instead
- They are known to be applicable in some situations where block ciphers are converted to hash functions, leading to practical collision attacks against the hash

- At present, there is no known practical attack that would allow someone without knowledge of the key to read data encrypted by AES when correctly implemented

Alternate Encryption Algorithms

Software-Optimized Encryption Algorithm (SEAL)

- Alternative to DES/3DES/AES
- Stream cipher with a 160 bit key
- Fast with low CPU overhead
  - But slower initialization phase
- SEAL restrictions for Cisco routers
  - Router and the peer must support IPsec
  - Router and the other peer must run an IOS image that supports encryption
  - Router and the peer must not have hardware IPsec encryption

Description	Software-Optimized Encryption Algorithm
Timeline	First published in 1994, current version is 3.0 (1997)
Type of Algorithm	Symmetric
Key Size (in bits)	160
Speed	High
Time to Crack (Assuming a computer could try 255 keys per second)	Unknown but considered very safe
Resource Consumption	Low

RC Algorithms

- Provides very good speed and variable key-length capabilities
- Several RC (Rivest Cipher) algorithms
  - RC2:
    - 64-bit block cipher with a variable size key
    - replacement for DES
    - 1996, source code for RC2 was anonymously posted to the Internet => suggest that it had been reverse engineered
    - 1998 Ron Rivest authored an RFC publicly describing RC2 himself
  - RC4:
    - Stream cipher, variable length key, widely used
    - Before 2015 widely used: for WEP in 1997, WPA in 2003/4 for wireless cards, SSL in 1995, TLS in 1999
    - multiple vulnerabilities have been discovered => insecure
    - it was prohibited for all versions of TLS by RFC 7465 in 2015
  - RC6
    - AES opponent
    - in August 2016, code for various network security devices was disclosed (reputed to be Equation Group or NSA "implants")

Ron Rivest

Equation Group - is a highly sophisticated threat actor suspected of being tied to the Tailored Access Operations (TAO) unit of the United States National Security Agency (NSA), i.e. cyber-warfare intelligence-gathering unit (now: Computer Network Operations)

RC Algorithms	Timeline	Type of Algorithm	Key Size in Bits
RC2	1987	Block cipher	40 and 64
RC4	1987	Stream cipher	1 to 256
RC5	1994	Block cipher	0 to 2048
RC6	1998	Block cipher	128, 192, or 256

AES process from 1997 to 2000

- On January 2, **1997**, NIST announced that they **wished** to choose a successor to DES to be known as AES
- call for new algorithms on September 12, 1997
  - algorithms were all to be **block ciphers**, supporting a block size of **128 bits** and **key sizes** of **128, 192, and 256** bits
- **15 designs** were created and submitted from several countries
  - CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, Rijndael, SAFER+, Serpent, and Twofish
  - candidates were assessed not only on security, but also on performance in a variety of settings (PCs of various architectures, smart cards, hardware implementations) and on their feasibility in limited environments (smart cards with very limited memory, low gate count implementations, FPGAs)
- **5 AES finalists**: MARS, RC6, Rijndael, Serpent, and Twofish
- **Winner**: on October 2, **2000**, NIST announced that Rijndael had been selected as the proposed AES
- **FIPS**: On November 26, **2001**, NIST announced that AES was approved as FIPS PUB 197

Federal Information Processing Standards (FIPS) of the US

- are a set of publicly announced standards that the National Institute of Standards and Technology (NIST) has developed for use in computer systems of non-military, American government agencies and contractors.
- FIPS standards establish requirements for ensuring computer security and interoperability, and are intended for cases in which suitable industry standards do not already exist.
- Many FIPS specifications are modified versions of standards the technical communities use, such as the American National Standards Institute (ANSI), the Institute of Electrical and Electronics Engineers (IEEE), and the International Organization for Standardization (ISO).

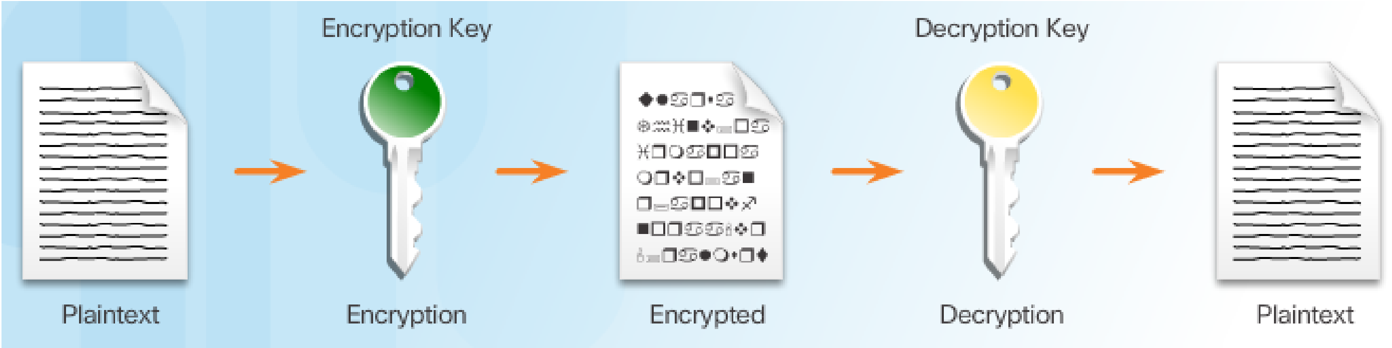
Summarization

Symmetric Encryption Algorithms	Description
Data Encryption Standard (DES)	This is a legacy algorithm. It uses a short key length that makes it insecure.
3DES (Triple DES)	This is the replacement for DES and repeats the DES algorithm three times. It should be avoided as it is scheduled to be retired in 2023. If implemented, use very short key lifetimes.
Advanced Encryption Standard (AES)	It offers combinations of 128-, 192-, or 256-bit keys to encrypt 128, 192, or 256 bit-long data blocks.
Software-Optimized Encryption Algorithm (SEAL)	It is a stream cipher that uses a 160-bit encryption key and has a lower impact on the CPU compared to other software-based algorithms.
Rivest ciphers (RC) series algorithms	RC4 is a stream cipher that was used to secure web traffic. It has been found to have multiple vulnerabilities which have made it insecure. RC4 should not be used.

Asymmetric Key Algorithms

- Called public-key algorithms
- Uses separated but matching keys
  - Public key (publicly available)
    - For encryption
  - Private key (kept secure)
    - For decryption
  - Works also in opposite direction
  - There is no option from one key to calculate the other one
  - Typical key length is 512 to 4,096 bits
    - <1024: considered unreliable
    - >1024: considered trusted
- achieve confidentiality and authenticity by using this process.
- algorithms are substantially slower than symmetric algorithms

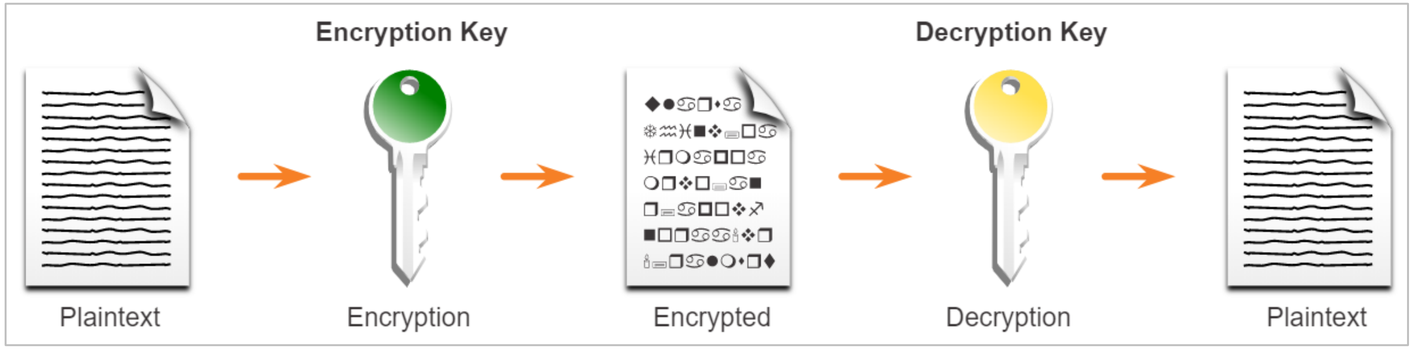
- Four well-known protocols that use asymmetric key algorithms:
  - Internet Key Exchange (IKE)
    - for IPSec
  - Secure Socket Layer (SSL) / Transport Layer Security (TLS)
  - Secure Shell (SSH)
  - Pretty Good Privacy (PGP)



Confidentiality

Asymmetric Encryption

- also called public-key algorithms, are designed in a way that the encryption and the decryption keys are different.
- use a public key and a private key (separated but matching keys)
  - both keys are capable of the encryption process (mostly public key), but the complementary paired key is required for decryption (mostly private key).
  - the process is also reversible. Data that is encrypted with the public key requires the private key to decrypt.
- achieve confidentiality and authenticity by using this process.
- can use key lengths between 512 to 4,096 bits.
- algorithms are substantially slower than symmetric algorithms



Diffie-Hellman Key Exchange (for symmetric ciphers)

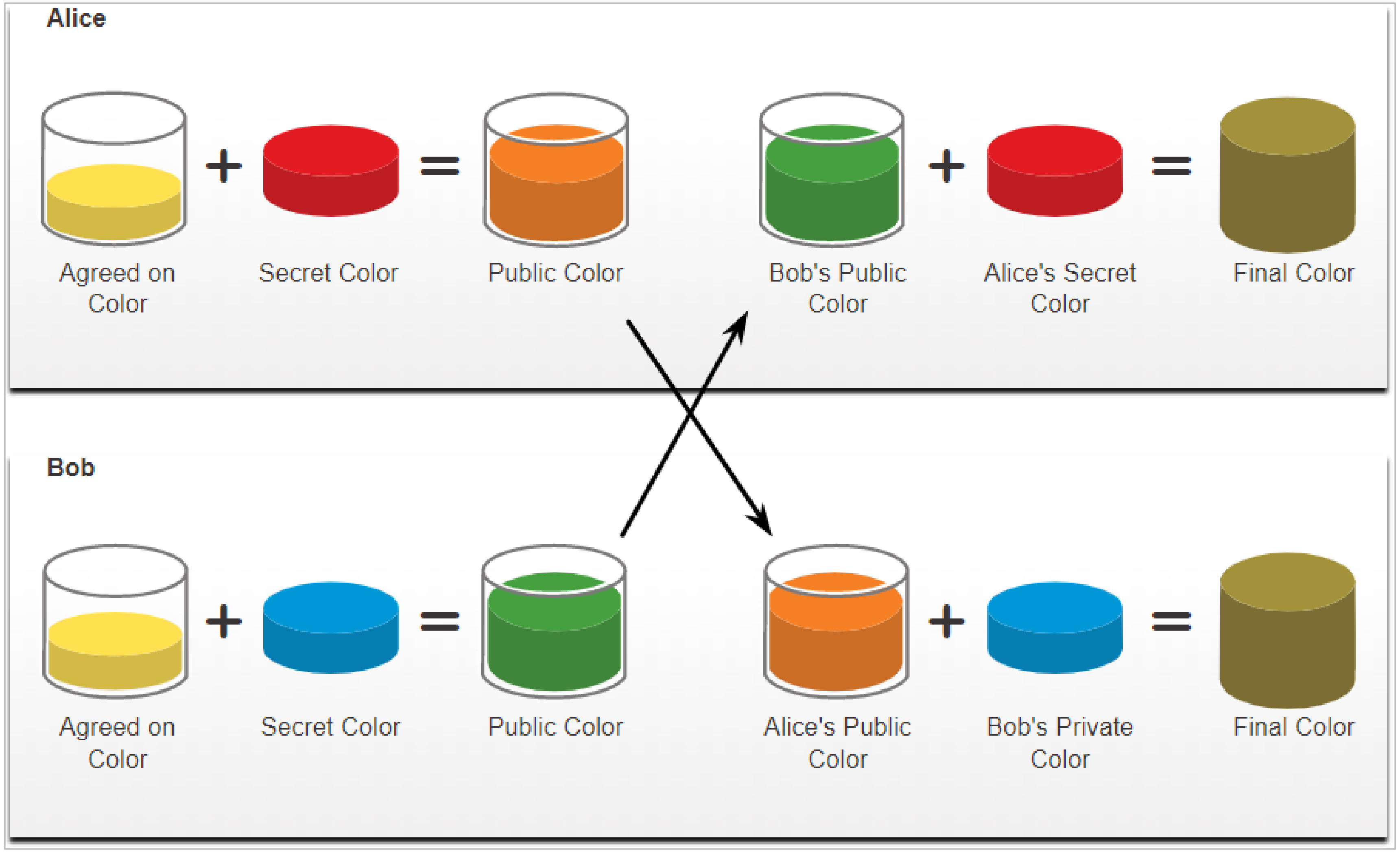
Diffie-Hellman (DH) Algorithm

- Remember? The problem (challenge) of symmetric encryption algorithms …
  - … is “*How to exchange the key over an untrusted environment?*”
- Invented Whitfield Diffie and Martin Hellman in 1976
- It is not a encryption mechanism
- Provides automatic and secure key “exchange” method
  - But the shared key is never exchanged over the net
  - Mechanism allows two computers to generate an identical shared secret on both systems (using math of large numbers)
    - Without having communicated before
  - Is asymmetric, and slow
- Commonly used for
  - IPSec VPN, SSL, TLS, SSH
    - Initial key handshake using slow and asymmetric DH
    - Encryption then uses fast symmetric algorithm

Description	Diffie-Hellman Algorithm
Timeline	1976
Type of Algorithm	Asymmetric
Key Size (in bits)	512, 1024, 2048, 3072, 4096
Speed	Slow
Time to Crack (Assuming a computer could try 255 keys per second)	Unknown but considered safe using keys of 2048 or higher
Resource Consumption	Medium

DH Operation

- The new shared key (Final Color) is never actually exchanged between the sender and receiver



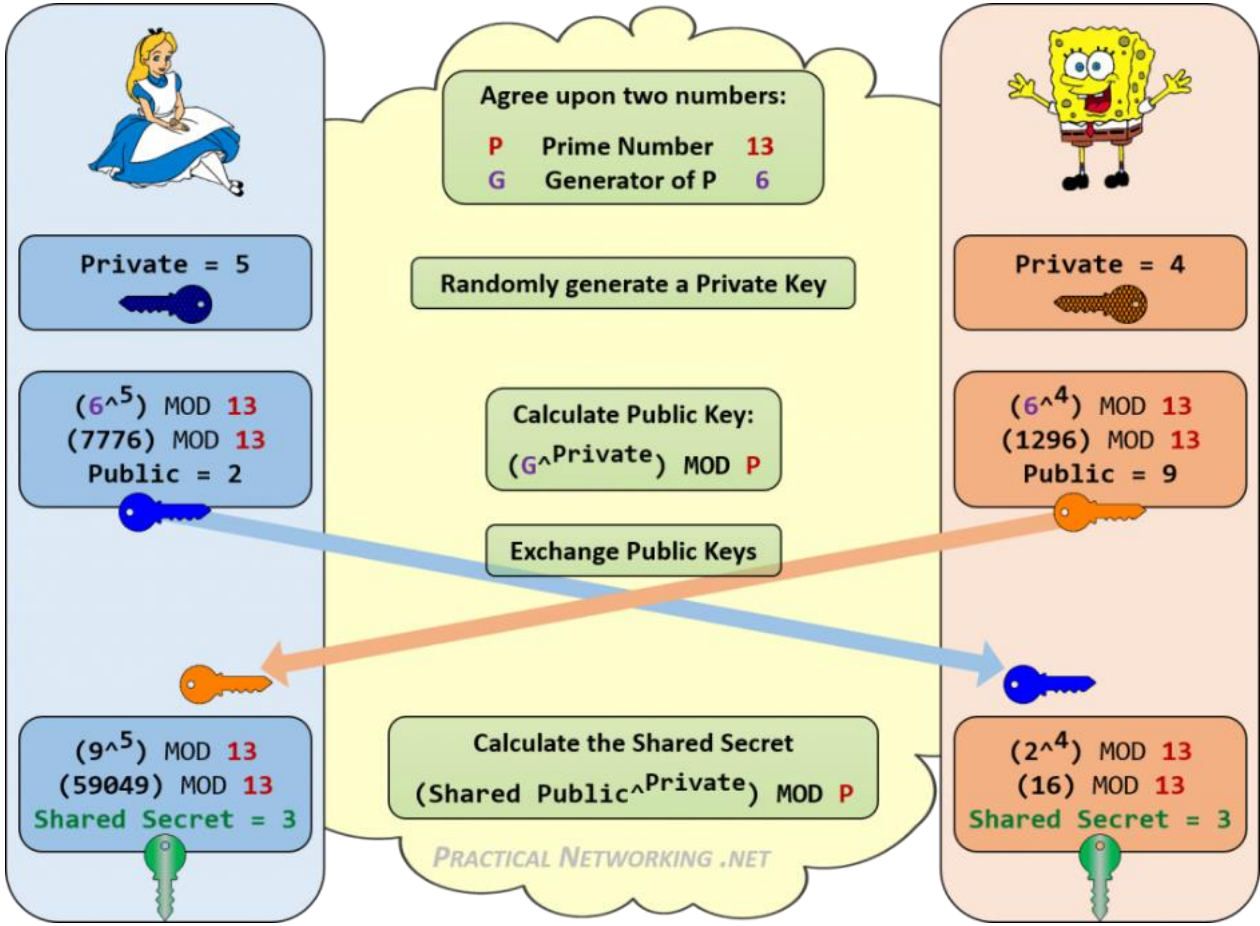
DH Operation

Diffie-Hellman Key Exchange

- Several steps
  1. Both sites must agree on two shared and non-secret numbers
     - **P** as a prime number (modulus), typically <u>large</u>
     - **G** as a base number (generator), typically <u>small</u>
  2. Both sites generates one own private number PRIVATE
  3. Both sites using **G**, **P** and the private number generates a Public number (key)
     - $(G^{PRIVATE})$ MOD **P** = SHARED PUBLIC KEY
  4. Both sites will exchange theirs public keys unencrypted over the net
  5. Each site using **G**, **P** and an opposite public key to generate shared secret
     - $(SHARED\_PUBLIC^{PRIVATE})$ MOD **P** = **SECRET KEY**


$$(6^{4} \bmod 13)^{5} \bmod 13 = (6^{5} \bmod 13)^{4} \bmod 13$$

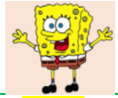
Shared secret



http://www.practicalnetworking.net/series/cryptography/diffie-hellman/

DH operation

DH groups



$(6^{4} \bmod 13)^{5} \bmod 13 = (6^{5} \bmod 13)^{4} \bmod 13$

Shared secret

- Diffie-Hellman uses different DH groups to determine the strength of the key that is used in the key agreement process.
  - The higher group numbers are more secure, but require additional time to compute the key.
- The following identifies the DH groups and their associated **prime number value**:
  - DH Group 1: 768 bits
  - DH Group 2: 1024 bits
  - DH Group 5: 1536 bits
  - DH Group 14: 2048 bits
  - DH Group 15: 3072 bits
  - DH Group 16: 4096 bits
  - DH Group 19: 256 bits
  - DH Group 20: 384 bits
  - DH Group 21: 521 bits
  - DH Group 24: 2048 bit

Do not use for FIPS 140 mode

Modular exponential group with 'X'-bit modulus

Random 'Y'-bit elliptic curve group

Modular exponential group with 2048-bit modulus and 256-bit prime order subgroup

Types of Asymmetric Algorithms (1)

- Common examples of asymmetric encryption algorithms are described in the table.

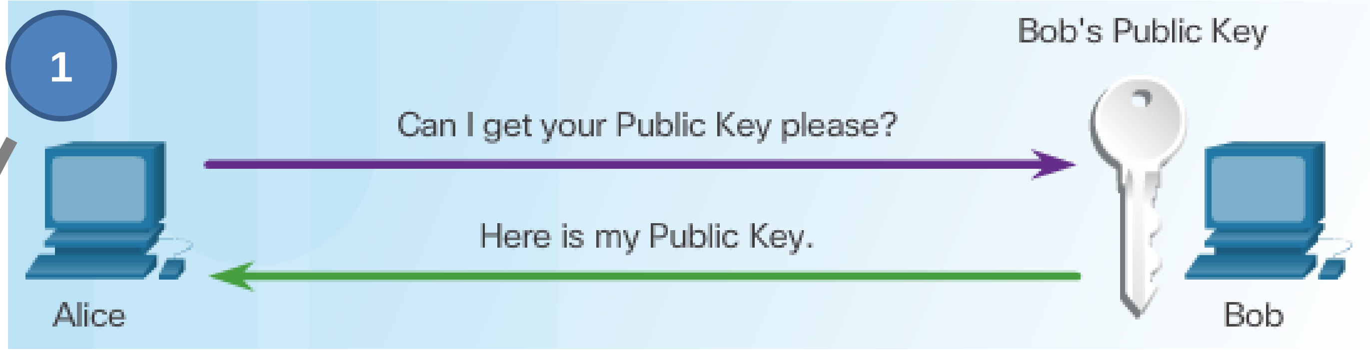
Asymmetric Encryption Algorithms	Key Length	Description	Enc / Dec	DS	Key Ex
Diffie-Hellman (DH)	512, 1024, 2048, 3072, 4096	This algorithm allows two parties **to agree on a key** that they can use to encrypt messages they want to send to each other. The security depends on the assumption that it is easy to raise a number to a certain power, but difficult to compute which power was used, given the number and the outcome.	N	N	Y
Digital Signature Standard (DSS) and Digital Sign. Algorithm (DSA)	512 – 1024	It specifies DSA as the algorithm **for digital signatures (only)**, it was created by **NIST**. DSA is a public key algorithm **based on the ElGamal signature scheme**. Signature creation speed is similar to RSA, but is 10 to 40 times slower for verification.	N	Y	N
Elliptic curve techniques	224 or higher	Elliptic curve cryptography 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.	Y	Y	Y

Types of Asymmetric Algorithms (2)

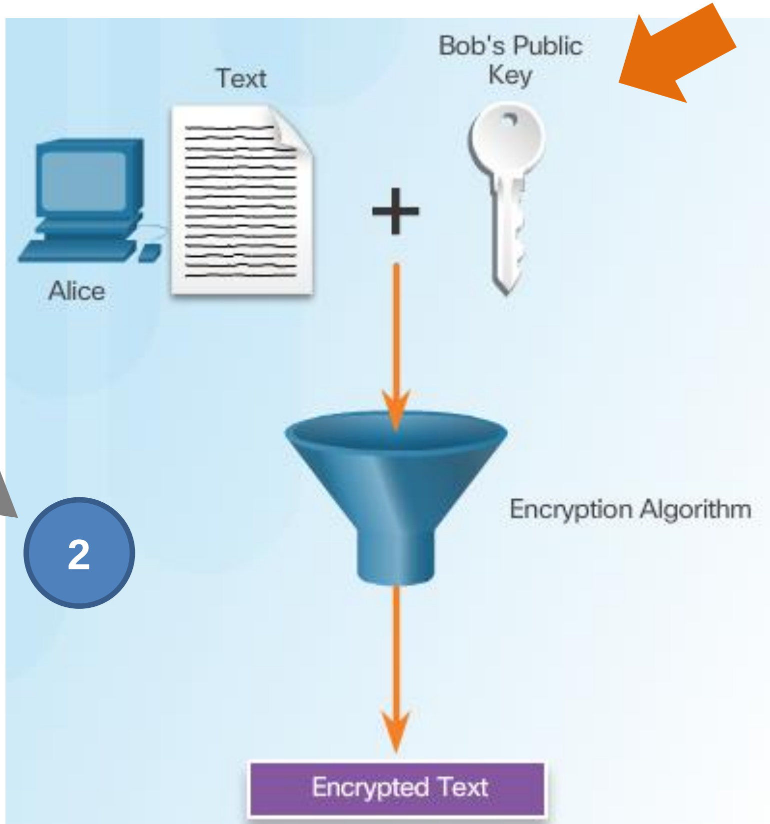
Asymmetric Encryption Algorithm	Key Length (in Bits)	Description	Enc / Dec	DS	Key Ex
RSA encryption algorithms	512 to 2048	Developed by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT in 1977. It is an algorithm for public-key cryptography that is based on the current difficulty of factoring very large numbers. It is the first algorithm known to be suitable for signing as well as encryption, and 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.	Y	Y	Y
ElGamal	512 - 1024	An asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie-Hellman key agreement. Described by Taher ElGamal in 1984 and is used in GNU Privacy Guard software, PGP, and other cryptosystems. A disadvantage of the ElGamal system 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.	Y (secret key)	N	N

Asymmetric Key Algorithms – How to achieve Confidentiality

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

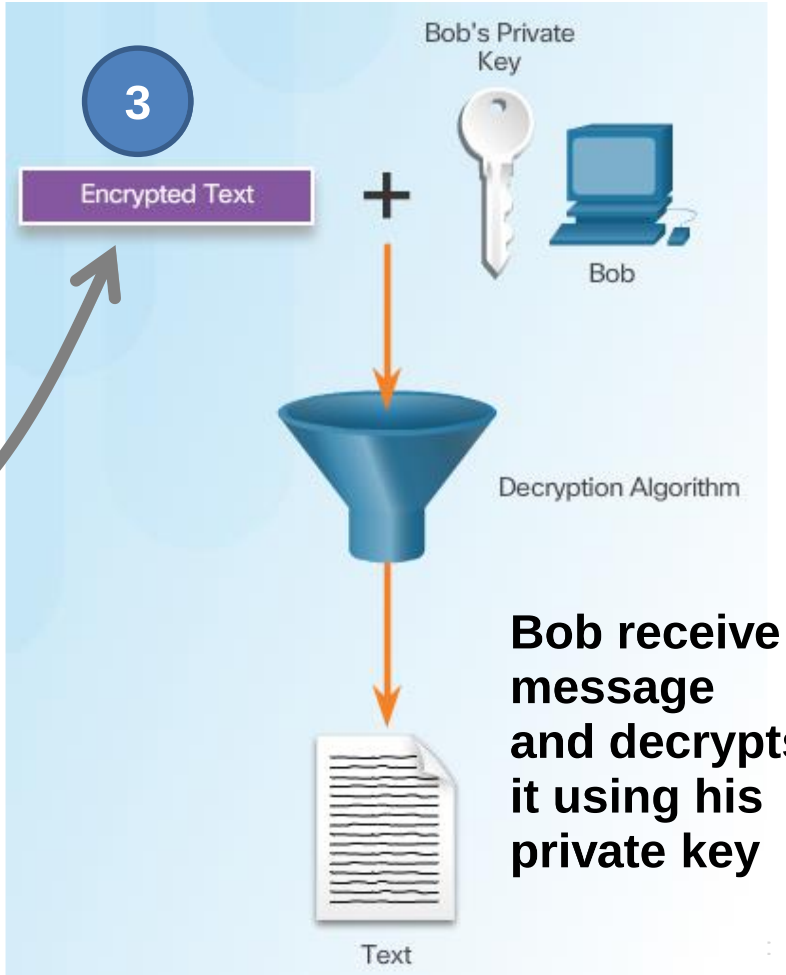


Alice acquires Bob's public key



Alice encrypts message using Bob's public key

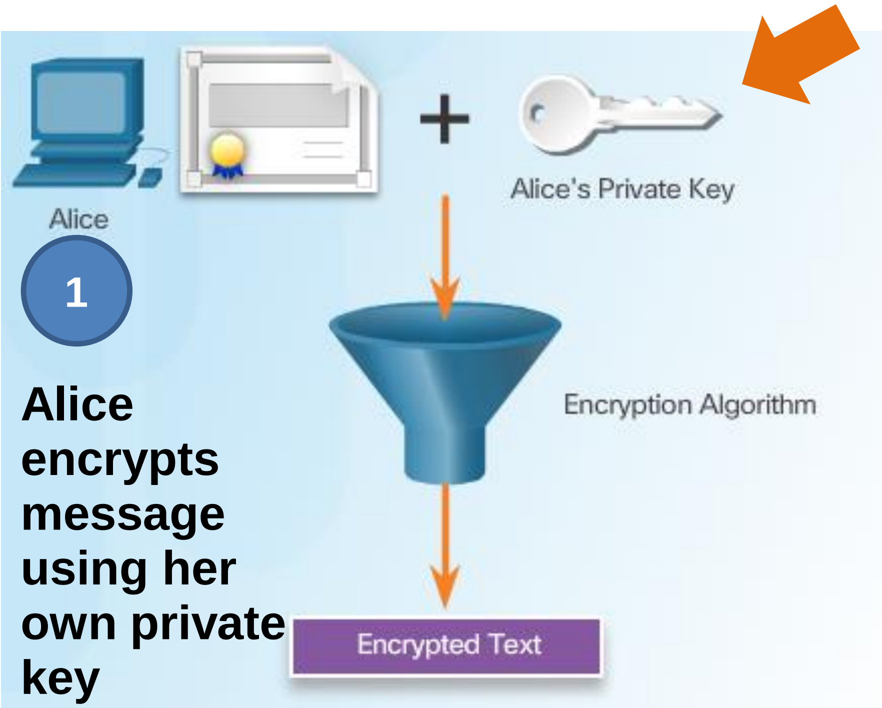
Alice send the message to Bob



Bob receive message and decrypts it using his private key

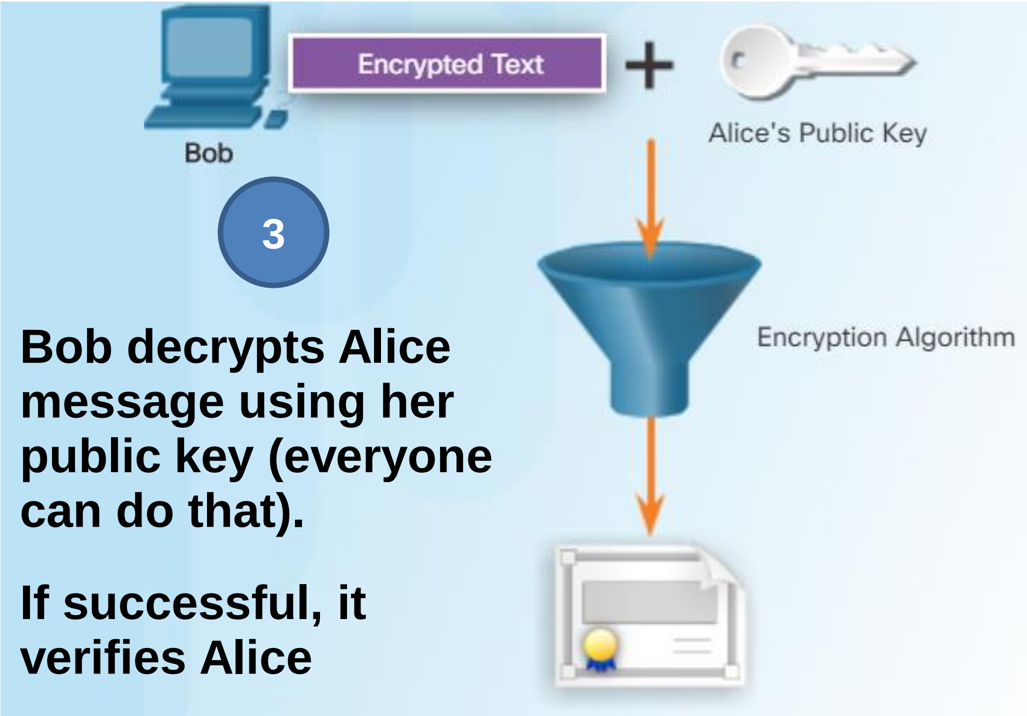
UNIZA

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



Alice encrypts message using her own private key

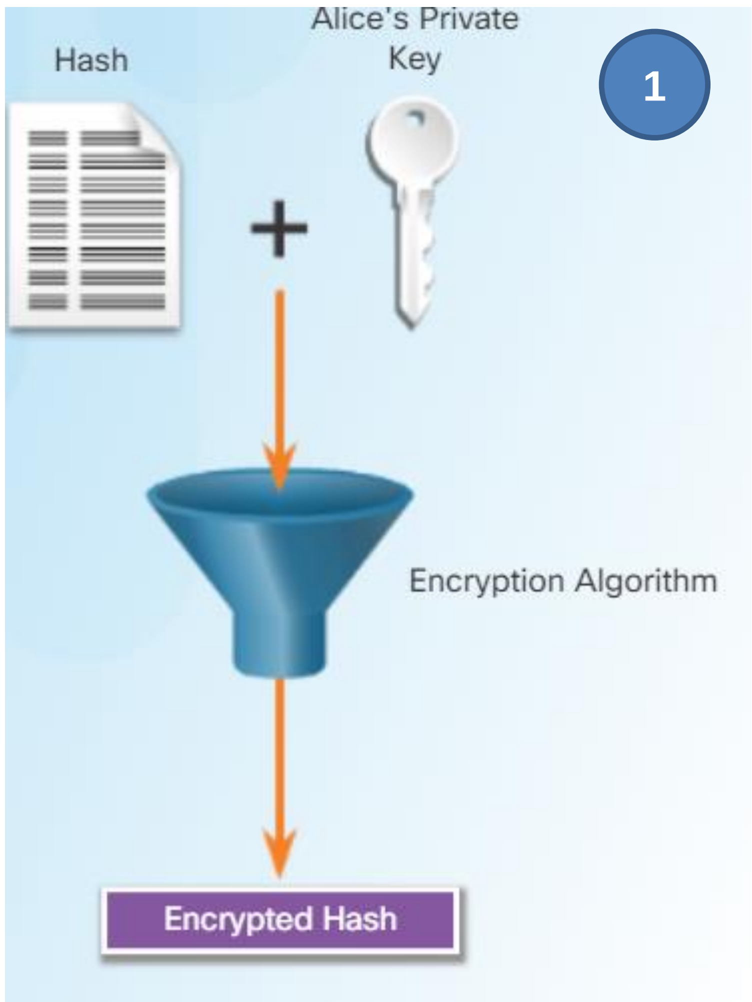
Bob needs Alice's public key to verify Alice. He will request it.



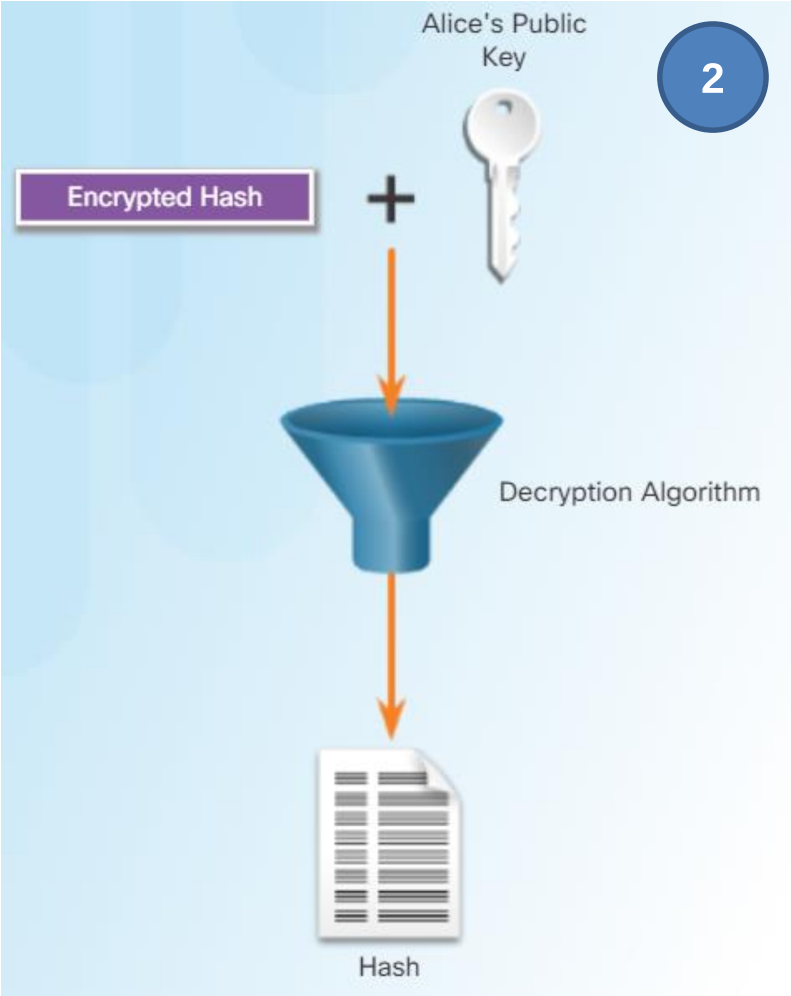
Bob decrypts Alice message using her public key (everyone can do that).

If successful, it verifies Alice

Asymmetric Algorithms - Integrity



- Alice calculates hash across the message
- Encrypt hash using her private key
- Alice attach hash to the message (encrypted or not)



- Bob decrypt Alice hash using her public key
- Bob calculates locally the hash over received message
- If hashes are the same it verifies that message was not modified
- It also verifies Alice

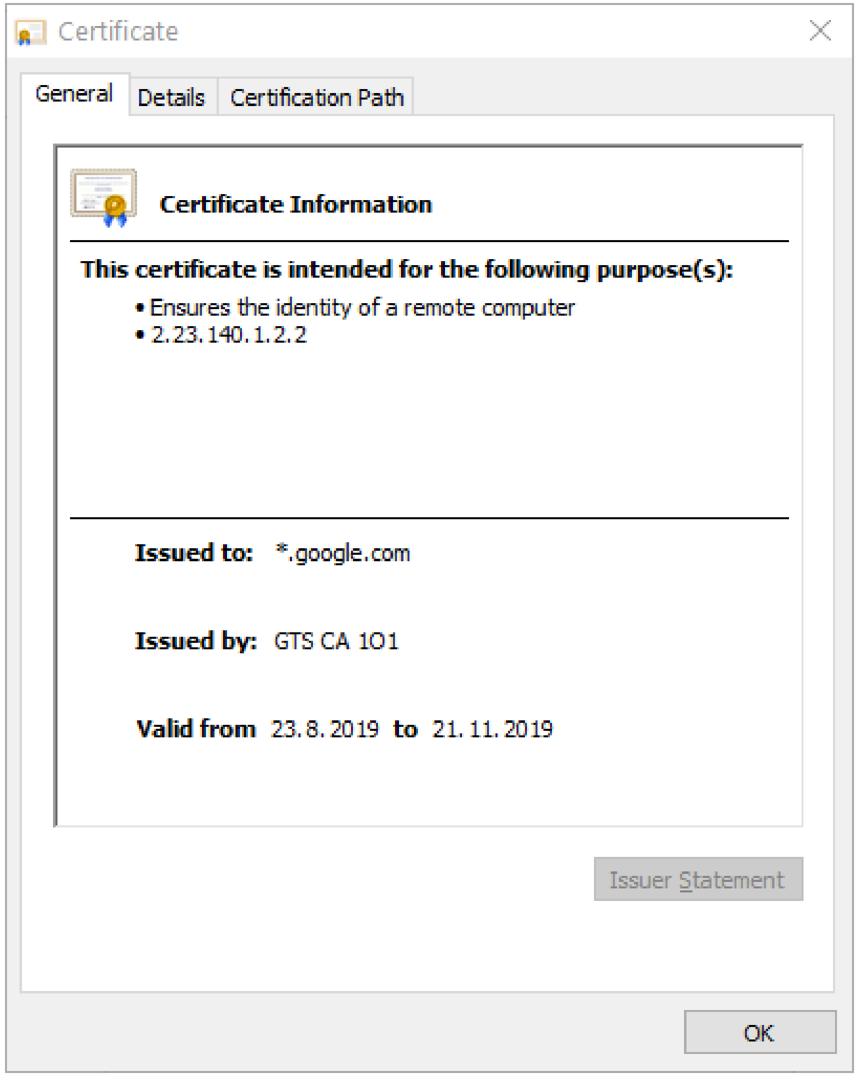
21.3 Public Key Cryptography (Using Digial Signatures)

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Using Digital Signatures (1)

- Integrity and authenticity features => allows service called
  Digital Signature (DS)
- DS uses asymetric cryptography
- Digital signature **properties**:
  - Signature is authentic
    - Provides proof of signer
  - Signature is unalterable (integrity)
    - Signed document can not be altered
  - Signature is not reusable
    - is part of a document and can not be moved to another one
  - Signature is non-repudiated *(podpis je nepopierateľný)*

- Used as a proof of authorship of the content of a document
- Commonly used in these situations:
  - Verifies the identity of an organization or individual
  - Authenticates a vendor websites, email senders, …
  - Code signing



Code signing

- Digitally signing code provides several assurances about the code:
  - AUTHENTICITY: The code is authentic and is actually sourced by the publisher.
  - INTEGRITY: The code has not been modified since it left the software publisher.
  - NON-REPUDIATION: The publisher undeniably published the code.
- which allows the end user to verify the signature before installing the software
- Cisco also provides digitally signed IOS images
  - ISR **image** naming conventions **includes** “SPA”
    - For example: c1900-universalk9-mz.SPA.154-3.M2.bin
- Each character of **SPA (SSA)** has the following meaning:
  - **S**: Stands for digitally signed software.
  - **P**: Stands for a production image, or **S**: special development image
  - **A**: Indicates the key version used to digitally sign the image (A, B, C ...)
    - Key type and key version are stored as part of the key record in the key storage of the device
- FIPS 140: standard requires software to be digitally signed and to be verified for authenticity and integrity prior to load and execution

Example of Code Signing

Identifying Digitally Signed Cisco SW

Device# **show version**
Cisco IOS Software, C3900 Software (C3900-UNIVERSALK9-M),
12.4(20090904:044027) [i12 577]
Copyright (c) 1986-2009 by Cisco Systems, Inc.
Compiled Fri 04-Sep-09 09:22 by xxx
ROM: System Bootstrap, Version 12.4(20090303:092436)
C3900-2 uptime is 8 hours, 41 minutes
System returned to ROM by reload at 08:40:40 UTC Tue May 21 1901!
System image file is "c3900-universalk9-mz.SSA„

Device# **show software authenticity file flash0:c3900-universalk9-mz.SSA**

```
File Name                  : flash0:c3900-universalk9-mz.SSA
Image type                 : Development
    Signer Information
        Common Name           : xxx
        Organization Unit     : xxx
        Organization Name     : xxx
    Certificate Serial Number : xxx
    Hash Algorithm            : SHA512
    Signature Algorithm       : 2048-bit RSA
    Key Version               : A
```

Device# **show software authenticity running**

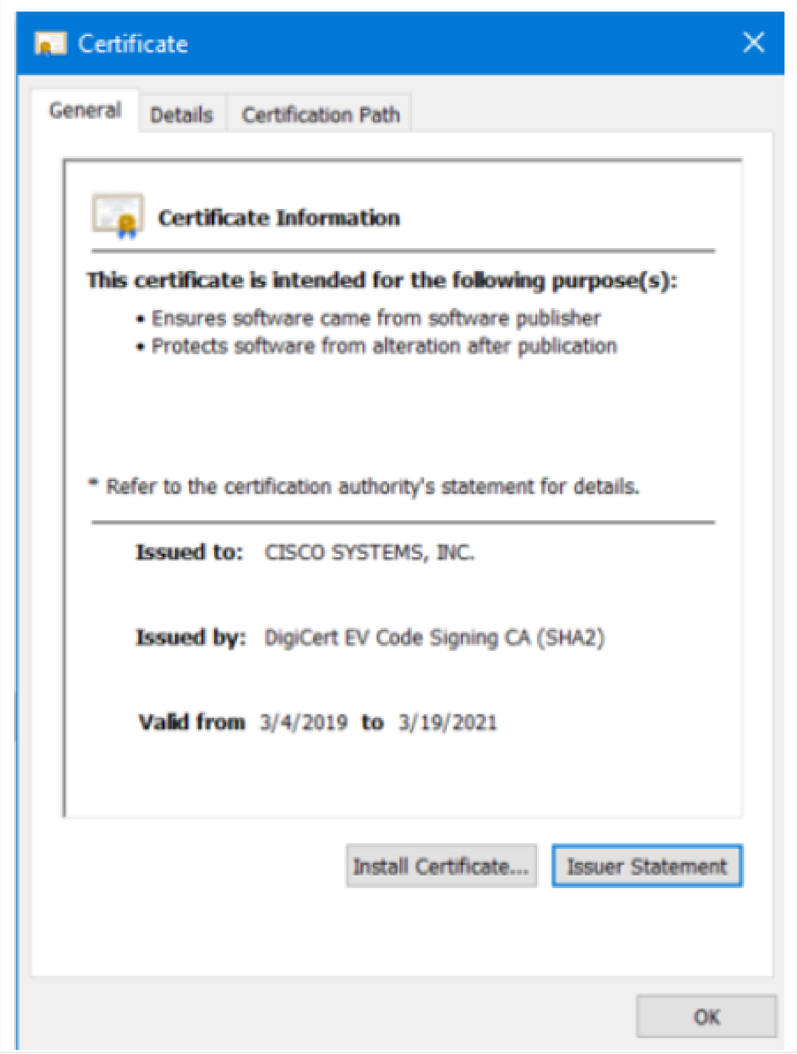
```
SYSTEM IMAGE
-------------------
Image type                 : Development
    Signer Information
        Common Name           : xxx
        Organization Unit     : xxx
        Organization Name     : xxx
    Certificate Serial Number : xxx
    Hash Algorithm            : xxx
    Signature Algorithm       : 2048-bit RSA
    Key Version               : xxx

    Verifier Information
        Verifier Name         : ROMMON 2
        Verifier Version      :  System Bootstrap, Version 12.4(20090409:084310)
ROMMON 2
---------------
Image type                 : xxx
    Signer Information
        Common Name           : xxx
        Organization Unit     : xxx
        Organization Name     : xxx
    Certificate Serial Number : xxx
    Hash Algorithm            : xxx
    Signature Algorithm       : 2048-bit RSA
    Key Version               : xx

    Verifier Information
        Verifier Name         : ROMMON 2
        Verifier Version      :  System Bootstrap, Version 12.4(20090409:084310) [
```

https://www.cisco.com/c/en/us/td/docs/ios/fundamentals/configuration/guide/TIPs_Conversion/lmsi_15_1s_book/cf_dgtly_sgnd_sw.html

Digital Signatures for Code Signing

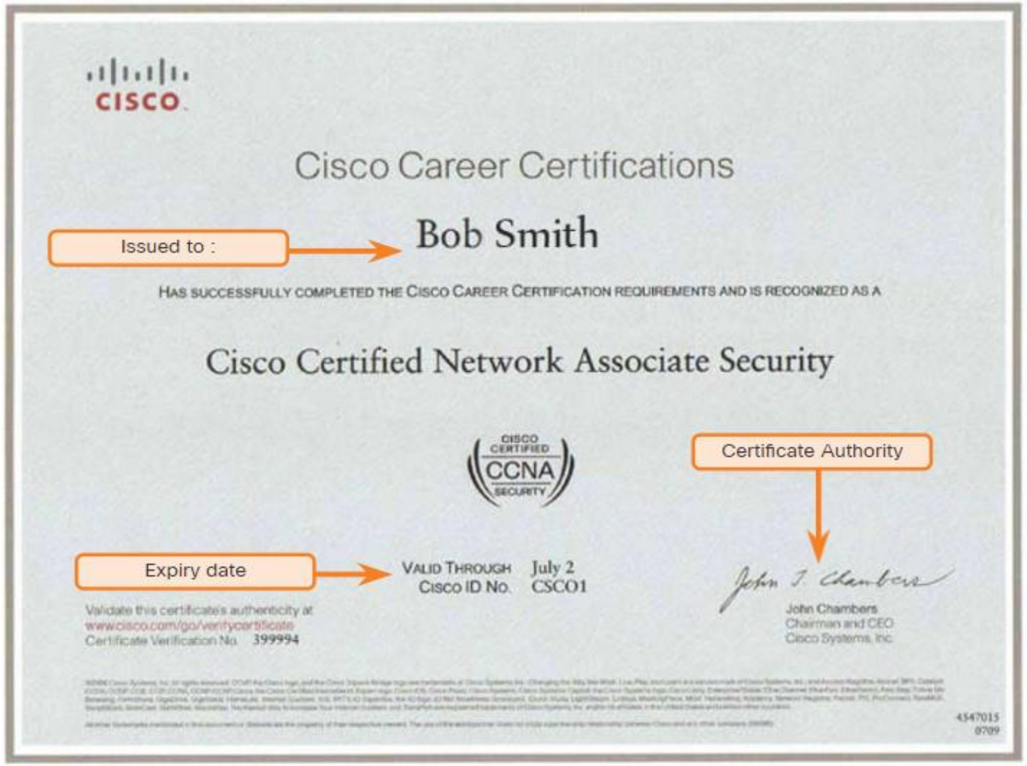
The properties of a file that has a digitally signed certificate are as follows:

Properties	Description
File Properties	This executable file was downloaded from the internet and it contains a software tool from Cisco Systems.
Digital Signatures	This tab reveals that the file is from a trusted organization, Cisco Systems Inc.
Digital Signatures Details	This window reveals that the file was signed by Cisco Systems, Inc mentioning the given year, month and time.
Certificate Information	The **General** tab provides information such as who the certificate was issued to, and who issued the certificate. It also displays the period for which the certificate is valid.
Certificate Path	In this tab, you can see the file was signed by Cisco Systems, as verified to DigiCert.



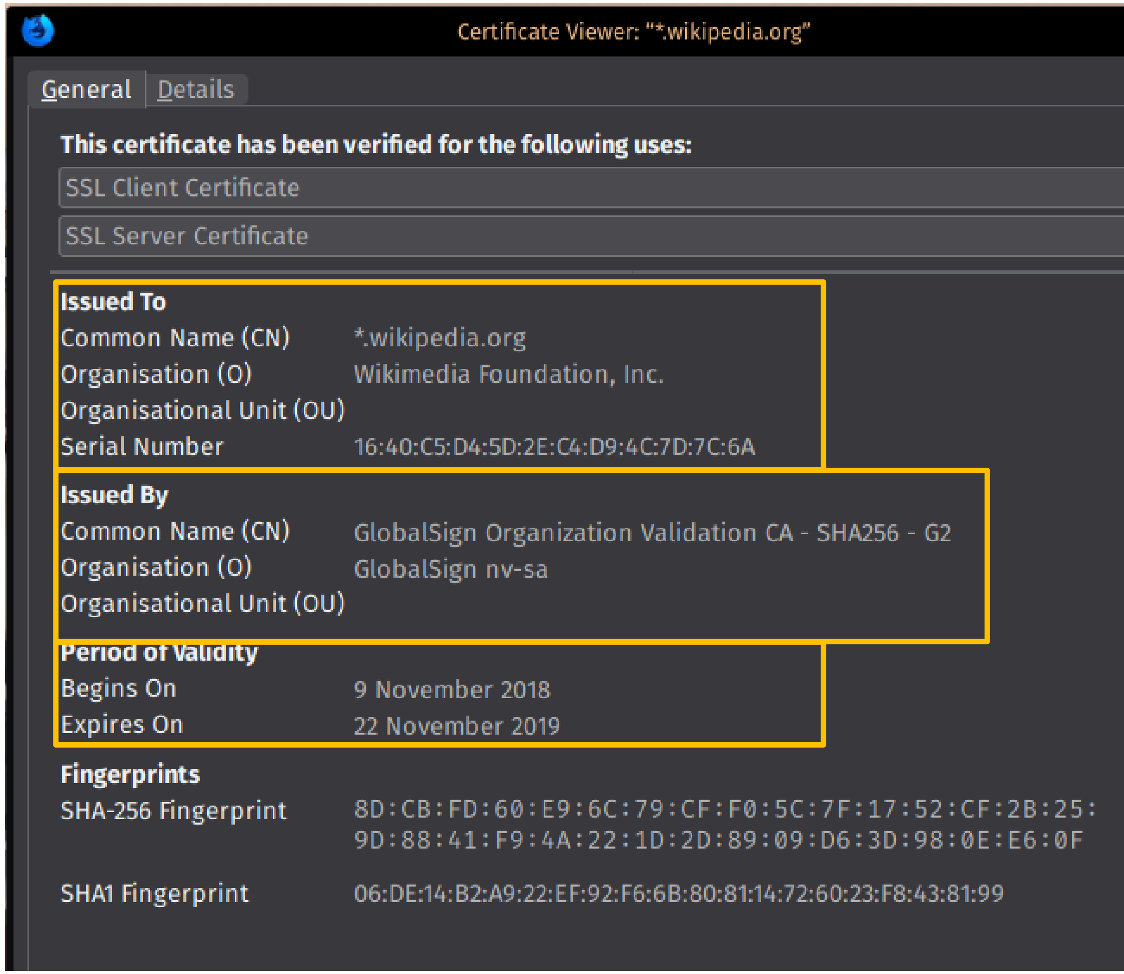
Digital Signatures for Digital Certificates

- A digital certificate enables users, hosts, and organizations **to securely exchange information over the Internet**.
- It is used to **authenticate** and **verify** that a **user** who is sending a message is who they claim to be.
- Digital certificates can also be used to provide **confidentiality for the receiver** with the means to **encrypt a reply**.
- Digital certificates are similar to physical certificates.
- Digital certificate **independently** verifies an identity.
- In other words:
  - a certificate verifies an identity
  - a signature verifies information coming from an identity.



Digital Certificate (DC)

- An electronic document used to prove the ownership of a public key
- Includes
  - information about the **key**
  - information about the **identity of its owner** (called the subject)
    - A person, computer/server
  - digital signature of an entity that has verified the certificate's contents (called the **issuer**)
  - If DC is valid and we trust issuer's we may use it
- Many DC types
  - **TLS/SSL server** certificate
  - **TLS/SSL client** certificate
  - **Email** certificate
  - **Code** signing certificate (to validate signatures on programs)
  - **Qualified** certificate (for electronic signature of **individuals**)
  - **Root** certificate (a self-signed certificate used to sign other certificates)
  - **Intermediate** certificate (used to sign other certs)
  - **End-entity or leaf** certificate (cannot be used to sign other certificates)
  - **Self-signed** certificate (a subject matches its issuer)



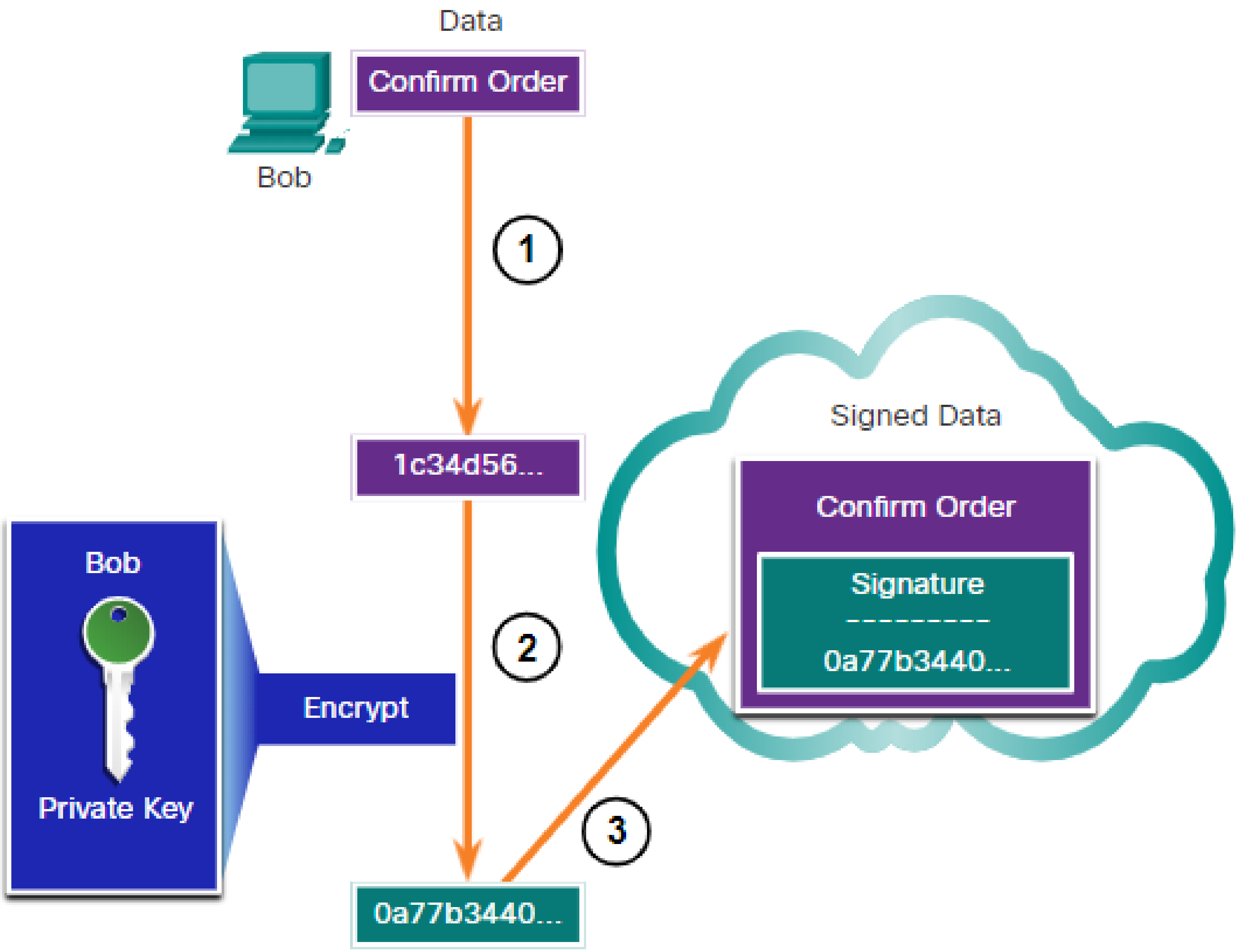
Wiki client/server cert

Using Digital Certificates

Sending Digital Ceritificate – confirmation of order

Scenario - how a DS is used:

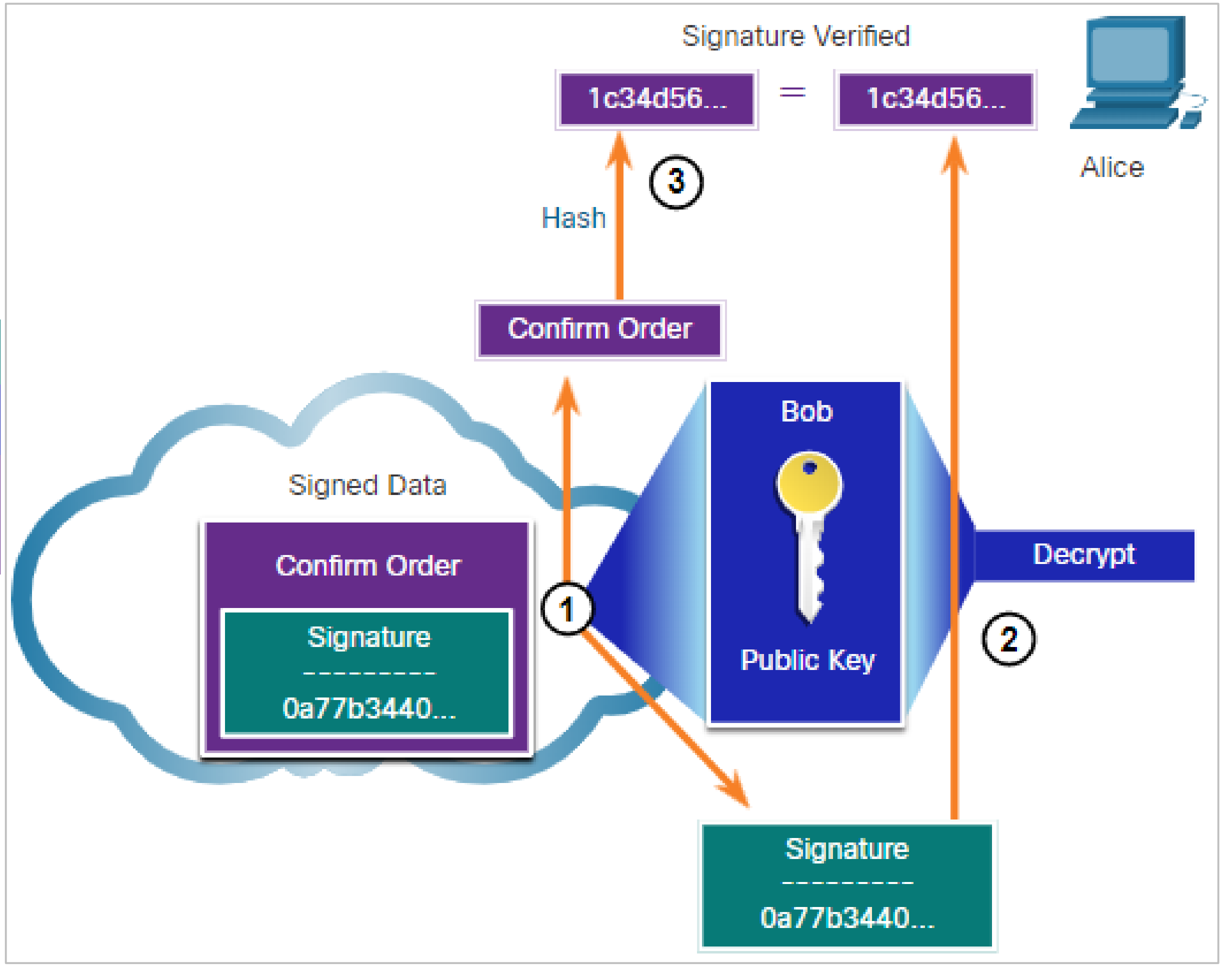
- Bob is confirming an order with Alice, which she is ordering from Bob’s website.
- Bob confirms the order and his computer creates a hash of the confirmation.
- The computer encrypts the hash with Bob's private key.
- The encrypted hash, which is the digital signature, is added to the document.
- The order confirmation is then sent to Alice over the internet.



Receiving Digital Ceritificate – verifiyng of confirmation

When Alice receives the digital signature, the following process occurs:

- Alice's receiver accepts the order confirmation with the digital signature and obtains Bob's public key.
- Alice's computer then **decrypts** the signature using Bob's **public key** which **reveals** the assumed **hash** value of the sending device.
- Alice's computer **creates** a **hash** of the received document, **without its signature**, and **compares** this hash to the decrypted hash.
- If the hashes **match**, the document is **authentic**. This means the confirmation was sent by Bob and has not changed since signed.



Digital Signature Algorithms

- Three algorithms
  - **Digital Signature Algorithm (DSA)**
    - Original standard for
      - **generating public and private key pairs**
      - generating and verifying digital signatures
    - Oldest one
  - **Rivest-Shamir Adelman Algorithm (RSA)** digital signature algorithm
    - asymmetric algorithm for generating and verifying digital signatures.
  - **Elliptic Curve Digital Signature Algorithm (ECDSA)**
    - A newer variant of DSA
    - Provides
      - digital signature authentication and non-repudiation
      - computational efficiency
      - small signature sizes
      - minimal bandwidth

DSA Characteristics

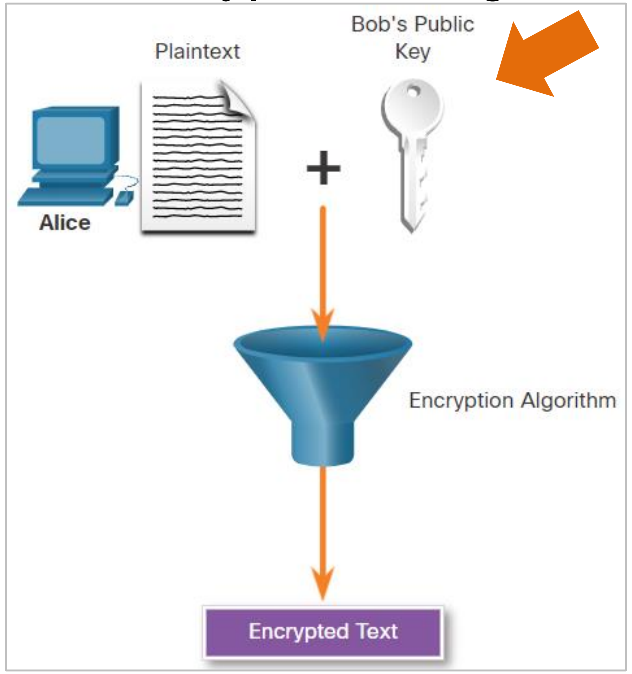
Description	Digital Signature Algorithm (DSA)
Timeline	1994
Type of Algorithm	Provides digital signatures
Advantages	Signature generation is fast
Disadvantages	Signature verification is slow

RSA Characteristics

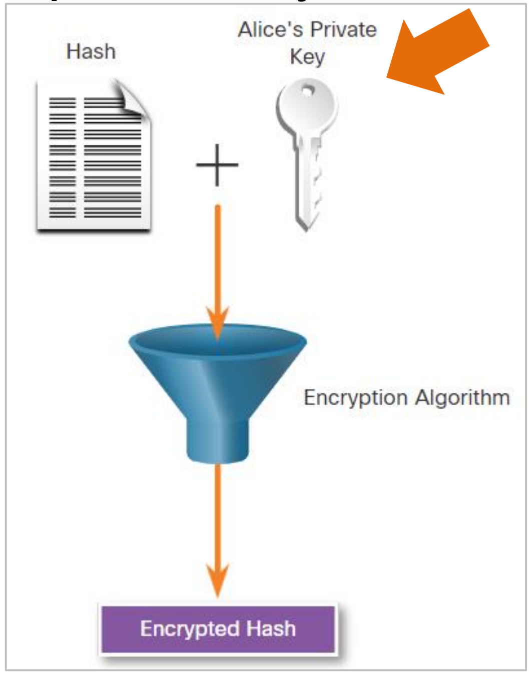
Description	Ron Rivest, Adi Shamir, and Len Adleman
Timeline	1977
Type of Algorithm	Asymmetric algorithm
Key size (in bits)	512 - 2048
Advantages	Signature verification is fast
Disadvantages	Signature generation is slow

Combining 2 asymetric encryption processes

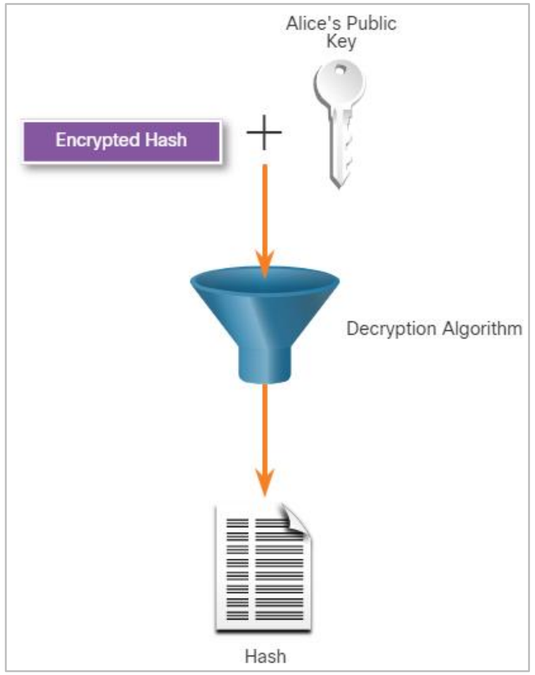
- To provides message confidentiality, authentication, and integrity - in this example, a message will be ciphered using Bob’s public key and a ciphered hash will be encrypted using Alice’s private key.



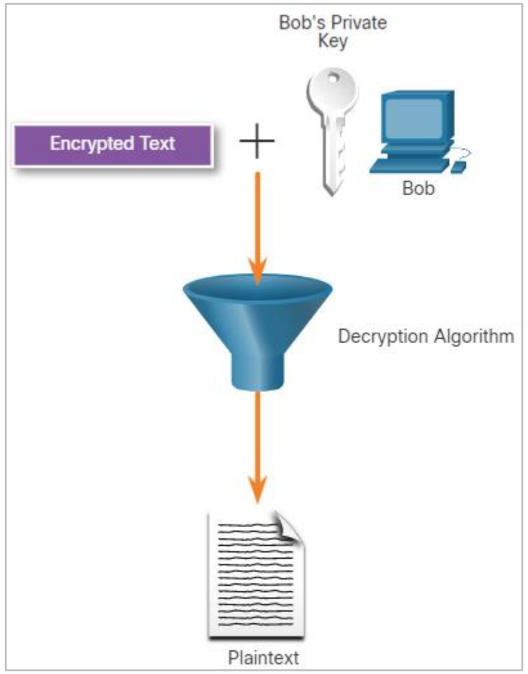
Alice uses Bob’s Public Key to encrypt a message/text



Alice encrypts a hash (from encrypted messafe) using her private key



Bob uses Alice’s public key to decrypt the hash



Bob uses his private key to decrypt the message

21.4 Authorities and the PKI Trust System

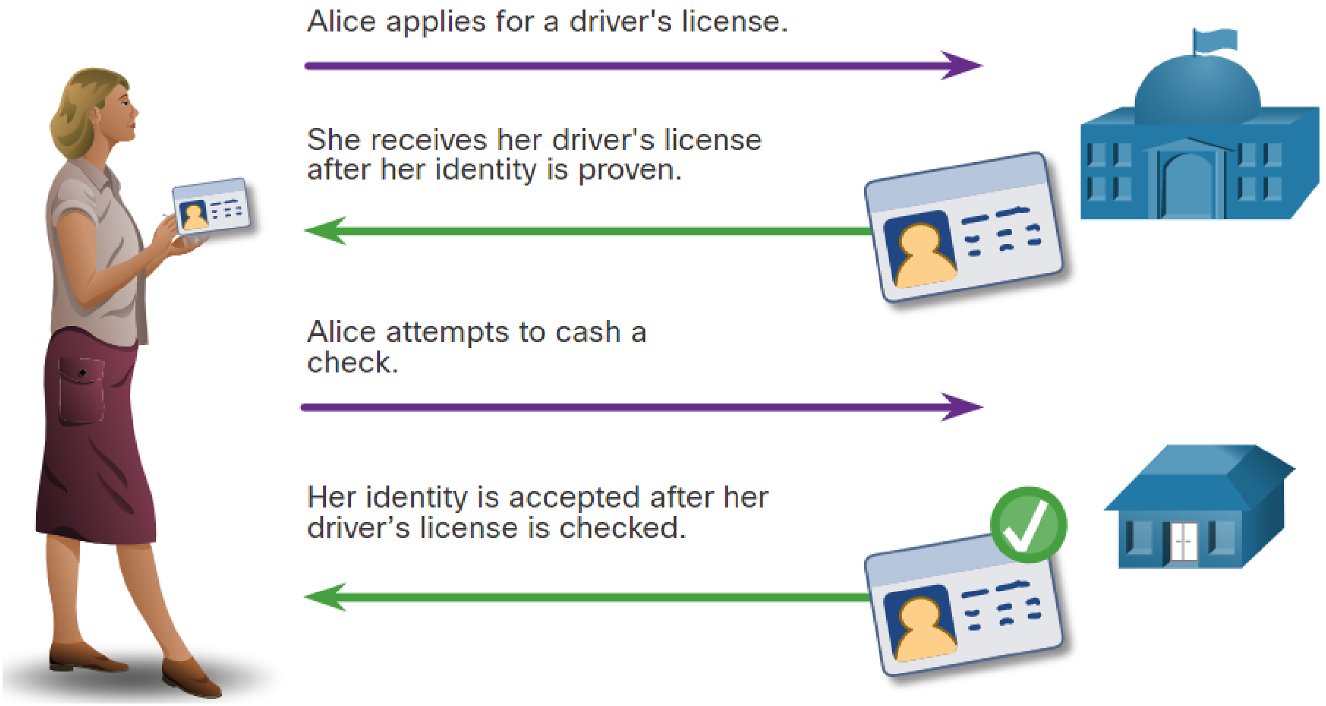
© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

PKI Overview

Public Key Infrastructure

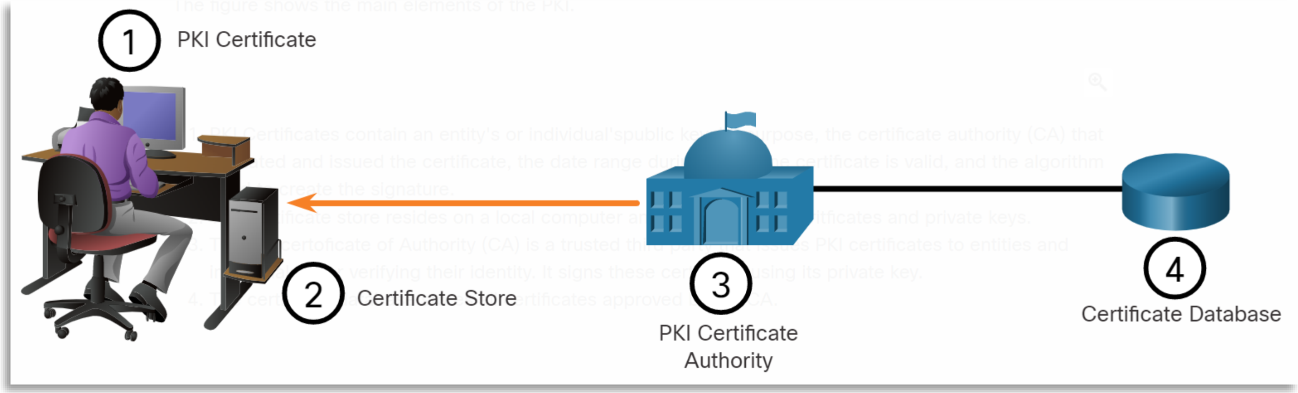
- PKI
  - Solve the problem of secure exchange of identity information
  - Using concept of authority
    - Neutral, commonly trusted third party, a.k.a ***certificate authority***
      - Others accept its credentials
    - It does in-depth investigation of authenticity
    - And it issues digital certificates
- PKI is the framework inspired by legacy authenticity procedures
  - Support large-scale deployment
  - Consists of:
    - hardware, software, people, policies, and procedures
  - needed to
    - create, manage, distribute, use, store and revoke digital certificates

Ilustration - how a driver's license is analogous to a digital certificate



- Without PKI
  - We can achieve confidentiality
  - But not authenticity

PKI Framework components



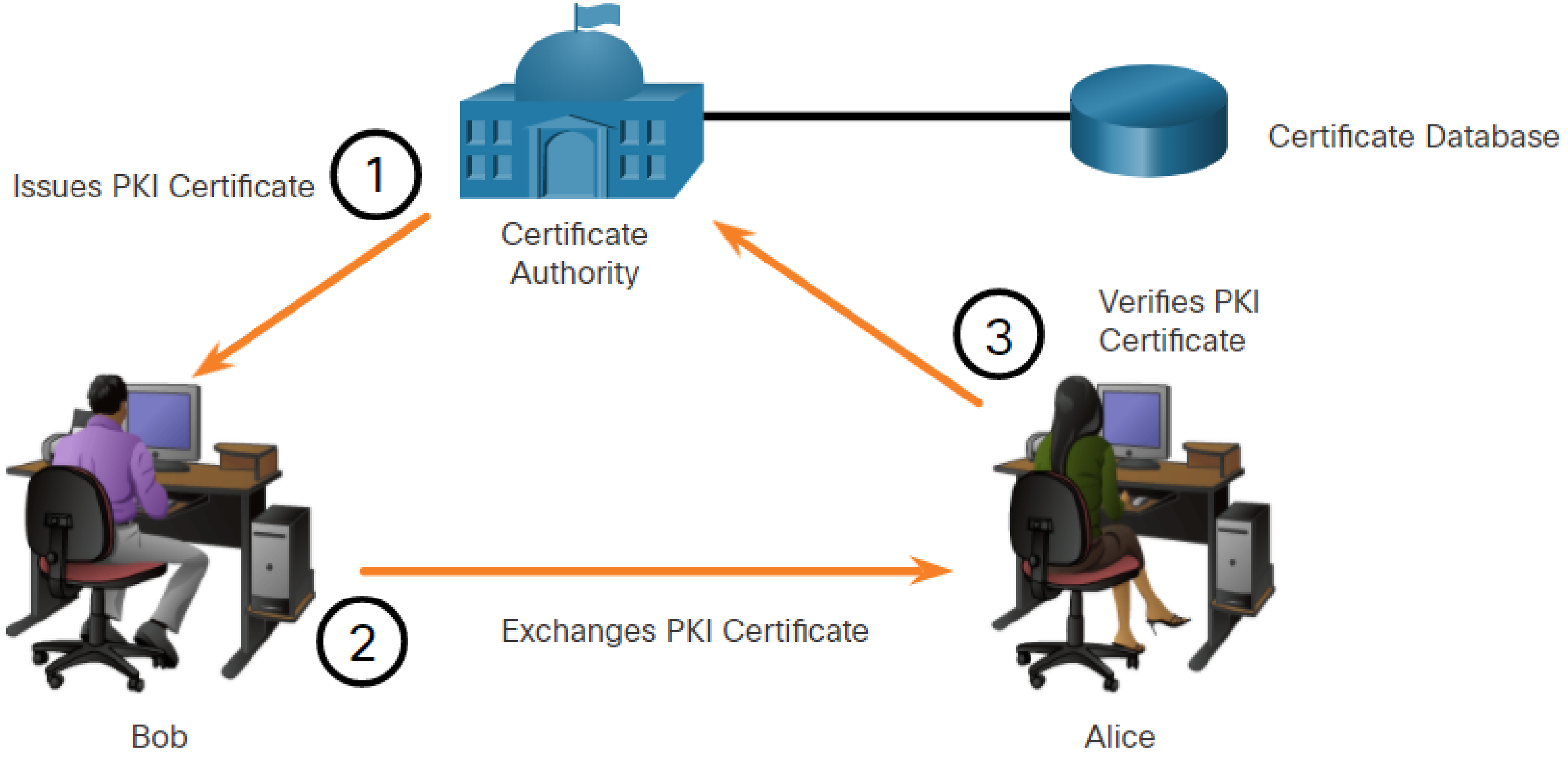
- **Certificate store**
  - Resides on a local computer
  - Store issued certificates and private keys
- **Certificate authority (CA)**
  - Globally trusted third party (company)
  - Locally trusted (enterprise or state)
  - Verifies identity
  - Issues PKI certificates to entities and individuals
  - Digitally signs these certificates
    - using its private key (private key of CA)
  - Examples
    - IdenTrust, DigiCert, Sectigo, GlobalSign, and GoDaddy, Let's Encrypt
  - Some CA public keys are preloaded in some OS
- **Registration authority (RA)**
  - a subordinate CA *(podriadená CA)*
  - certified by a root CA to offload some CA activities (to issue certificates for specific uses, ..)
- **Certification database**
  - Stores all certificates approved by CA

- PKI certificates contain
  - an entity's or individuals's public key, its purpose
  - the CA that validated and issued the certificate
  - the date range during which the certificate is valid
  - the algorithm used to create the signature

- PKI is needed to support **large-scale distribution** and **identification** of **public encryption keys**
- The PKI framework facilitates a **highly scalable trust relationship**.

Interoperation between elements of PKI infrastructure

- Not all PKI certificates are directly received from a CA
- Registration Authority (RA) is a subordinate CA and is certified by a root CA to issue certificates for specific uses.



1. **Issues PKI Certificate**. Bob initially requests a certificate from the CA. The CA authenticates Bob and stores Bob's PKI certificate in the certificate database.
2. **Exchanges PKI Certificate**. Bob communicates with Alice using his PKI certificate.
3. **Verifies PKI Certificate**. Alice communicates with the trusted CA using the CA's public key. The CA refers to the certificate database to validate Bob's PKI certificate.

The PKI Authorities System

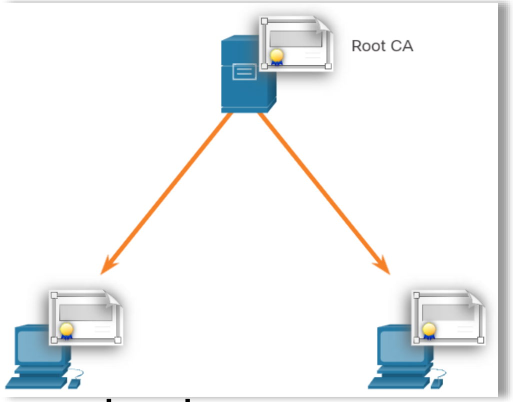
- Many vendors provide CA servers as a managed service or as an end-user product.
- Organizations may also implement private PKIs using Microsoft Server or Open SSL
  - to authenticate employees who are accessing the network
  - enterprise is its own CA
- CAs issue certificates based on classes
  - classess determine how trusted a certificate is (↑number = ↑ trust)
  - resp. how rigorous the procedure of identity verification was
- Some CA public keys are preloaded, such as those listed in web browsers.

Class	Description
0	Used for testing in situations in which no checks have been performed.
1	Used by individuals who require verification of email.
2	Used by organizations for which proof of identity is required.
3	Used for servers and software signing.
4	Used for online business transactions between companies.
5	Used for private organizations or government security.

PKI deployment models

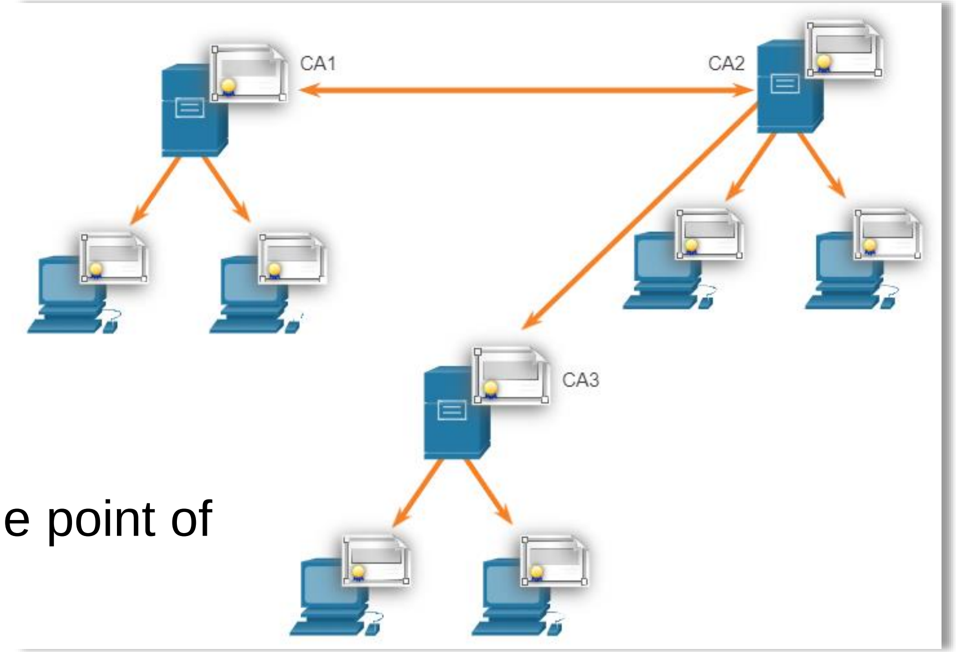
The PKI Trust System

- PKI forms topologies of trusts
  - **Single-Root PKI Topology**
    - The simplest one: usually one organization with Single CA (Root CA): Issues all certs
    - Benefit - simple
    - Minus: no one trust its certs, not scalable solution, Single point of Failure

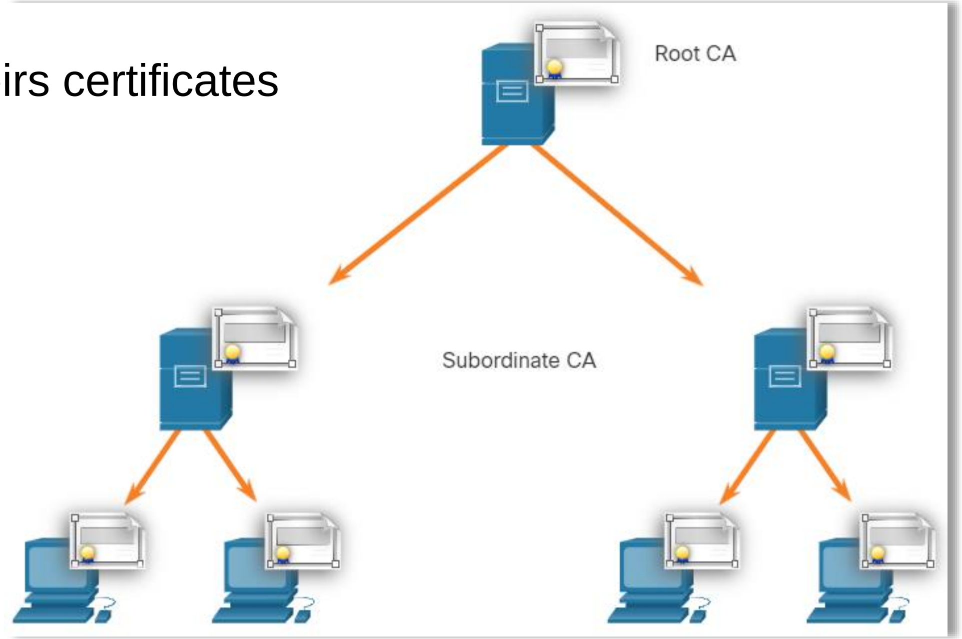


HYBRID - combination

  - **Cross-certified CA topologies**
    - Peer-to-peer model
      - => CA establish trust with other CAs: by cross-certifying theirs certificates
      - Trusted CAs trust each other and theirs certs
    - Solution provides redundancy, no Single PoF



  - **Hierarchical CA topologies**
    - CA organized on levels
      - Root CA: highest CA
        - Issues certs to end users and subordinate CAs
        - Establish community of trust
      - Benefits: increased scalability and manageability
      - Minus: chain of signing CA can be sometimes difficult



PKI interoperability

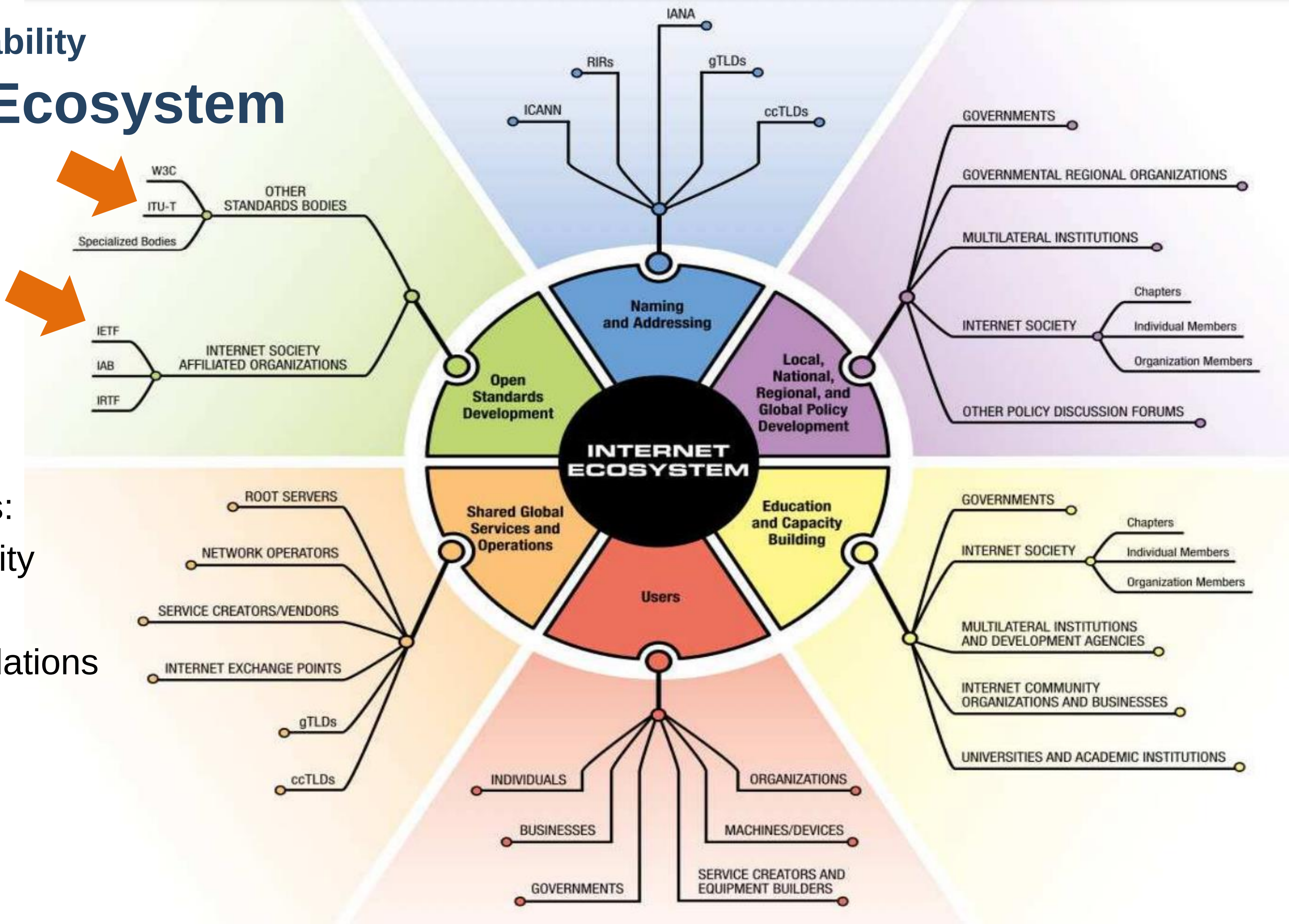
Internet Ecosystem

Also PKI needs:

- Interoperability

How?

- Recommendations
- Standards

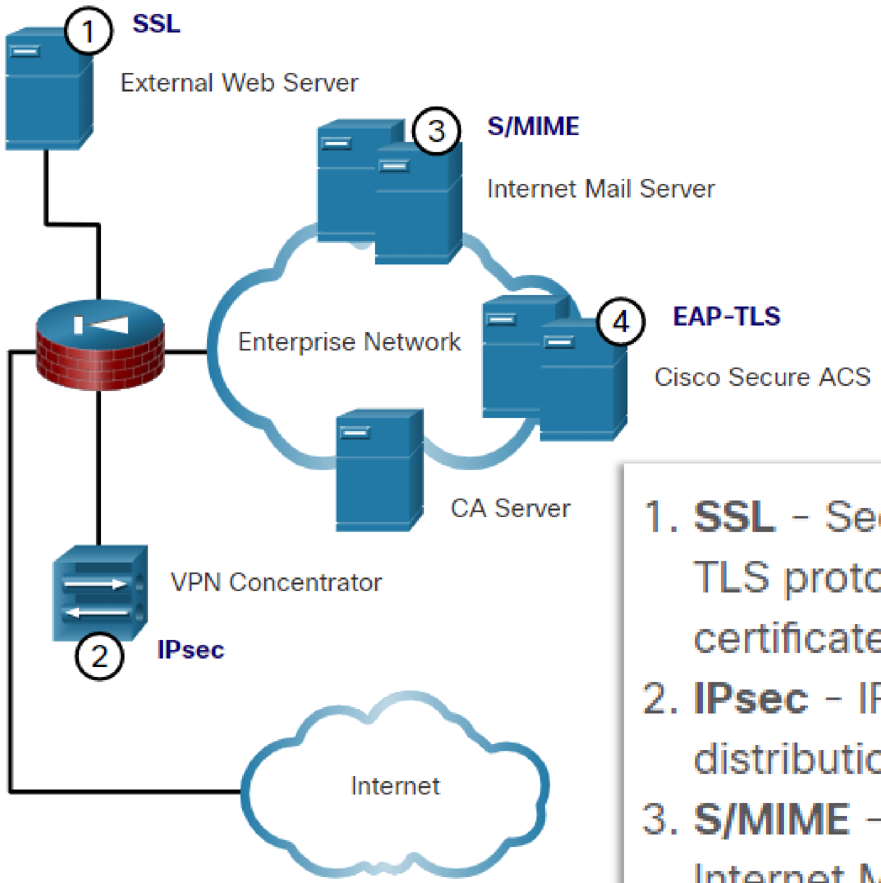


Interoperability of Different PKI Vendors

- PKI infrastructure
- Can be deployed using different vendors entities
  - Requires **interconnection** with different supporting services (LDAP, X.500)
  - Issues of **interoperability** emerged
- ITU reacted with recommendation: Public-key and attribute certificate frameworks (*1988, last update 10/21) https://www.itu.int/rec/T-REC-X.509
- IETF reacted with RFC 2527 standard in 1999 – now obsoleted by RFC 3647 (nov 2003) https://www.rfc-editor.org/rfc/rfc3647
  - Internet X.509 v3 Public Key Infrastructure Certificate Policy and Certification Practices Framework

- **X.509 version 3 (X.509 v3) standard**
  - promotes and standardizes PKI on the Internet
  - **defines the format of a digital certificate**
    - as the certificate and certificate revocation list (CRL) format
      - RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
    - X.509 certificate format is already extensively used and supported
      - IPSec VPN authentication
      - router/switches auth
      - secure web servers supports X509 SSL/TLS
      - email agents …

X.509v3 Applications



Note: LDAP and X.500 are protocols that are used to query a directory service, such as Microsoft Active Directory, to verify a username and password.

1. **SSL** - Secure web servers user X.509.v3 for website authentication in the SSL and TLS protocols, while web browsers use X.509v3 to implement HTTPS client certificates. SSL is the most widely used certificate-based authentication.
2. **IPsec** - IPsec VPNs use X.509 when certificates can be used as a public key distribution mechanism for internet key exchange (IKE) RSA-based authentication.
3. **S/MIME** - User mail agents that support mail protection with the Secure/Multipurpose Internet Mail Extensions (S/MIME) protocol use X.509 certificates.
4. **EAP-TLS** - Cisco switches can use certificates to authenticate end devices that connect to LAN ports using 802.1.x between the adjacent devices. The authentication can be proxied to a central ACS via the Extensible Authentication Protocol with TLS (EAP-TLS).

Signature validation

- Check validation information
  - Validity date range
    - X.509v3 certificates specify "not before" and "not after" dates
- **X.509** is an ITU standard defining the format of public key certificates
  - Used in many Internet protocols (TLS/SSL, electronic signatures, ..)
  - X.509 certificate binds an identity to a public key using a digital signature
  - certificate contains an identity (a hostname, or an organization, or an individual) and a public key (RSA, DSA, ECDSA, ed25519, etc.), and is either signed by a certificate authority or is self-signed
  - someone holding that certificate can use the public key it contains to
    - **establish secure communications** with another party
    - **or validate documents** digitally signed by the corresponding private key

self-signed root certificate representing a certificate authority

```
Certificate:[16]
    Data:
        Version: 3 (0x2)
        Serial Number:
            04:00:00:00:00:01:15:4b:5a:c3:94
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
        Validity
            Not Before: Sep  1 12:00:00 1998 GMT
            Not After : Jan 28 12:00:00 2028 GMT
        Subject: C=BE, O=GlobalSign nv-sa, OU=Root CA, CN=GlobalSign Root CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:da:0e:e6:99:8d:ce:a3:e3:4f:8a:7e:fb:f1:8b:
                    ...
```

End-entity certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            10:e6:fc:62:b7:41:8a:d5:00:5e:45:b6
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Organizati
        Validity
            Not Before: Nov 21 08:00:00 2016 GMT
            Not After : Nov 22 07:59:59 2017 GMT
        Subject: C=US, ST=California, L=San Francisco, O=Wikimedia
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
            pub:
                    00:c9:22:69:31:8a:d6:6c:ea:da:c3:7f:2c:ac:a5:
                    af:c0:02:ea:81:cb:65:b9:fd:0c:6d:46:5b:c9:1e:
                    9d:3b:ef
```

Public-Key Cryptography Standards

- PKCS = group of public-key cryptography standards
  - https://en.wikipedia.org/wiki/PKCS
- devised and published by RSA Security LLC

RSA PKCS Standards:

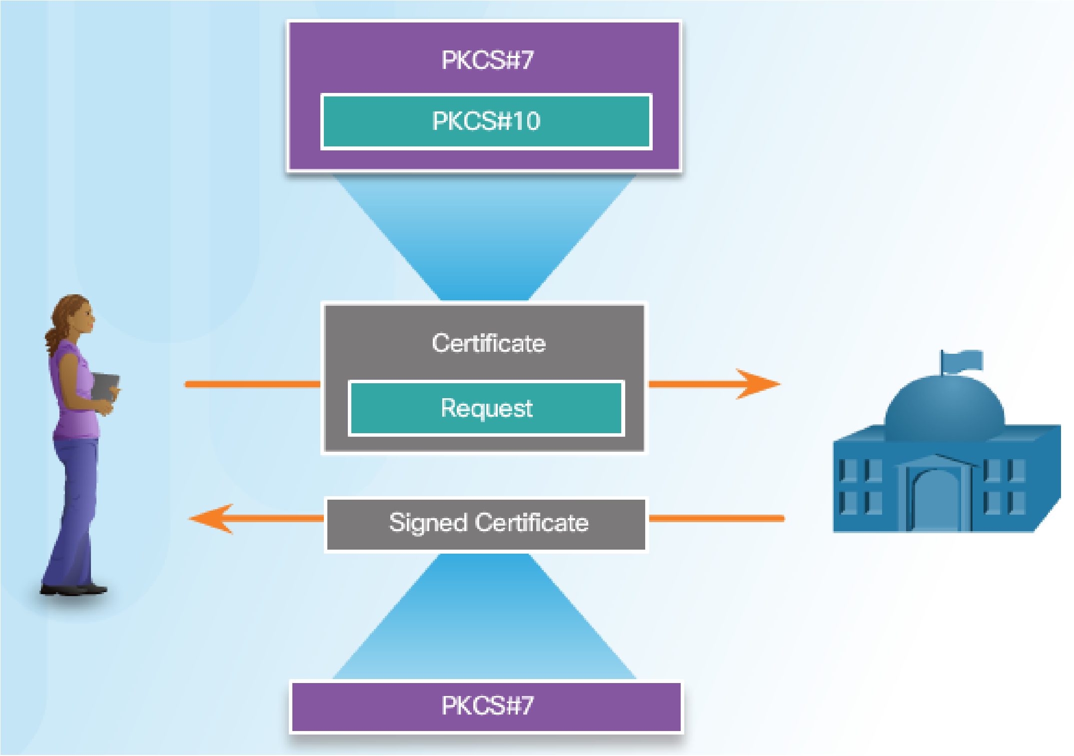
- PKCS #1: RSA Cryptography Standard
- PKCS #3: DH Key Agreement Standard
- PKCS #5: Password-Based Cryptography Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #10: Certification Request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

	Version	Name	
PKCS #1	2.2	RSA Cryptography Standard[1]	See RFC 8017. and encoding/pa
PKCS #2	-	*Withdrawn*	No longer active
PKCS #3	1.4	Diffie–Hellman Key Agreement Standard[2]	A cryptographic communications
PKCS #4	-	*Withdrawn*	No longer active
PKCS #5	2.1	Password-based Encryption Standard[3]	See RFC 8018 a
PKCS #6	1.5	Extended-Certificate Syntax Standard[4]	Defines extensio
PKCS #7	1.5	Cryptographic Message Syntax Standard[5]	See RFC 2315. message). Form used for single s
PKCS #8	1.2	Private-Key Information Syntax Standard[6]	See RFC 5958.
PKCS #9	2.0	Selected Attribute Types[7]	See RFC 2985. information, and
PKCS #10	1.7	Certification Request Standard[8]	See RFC 2986.
PKCS #11	3.0	Cryptographic Token Interface[9]	Also known as " public-key crypto PKCS 11 Techni
PKCS #12	1.1	Personal Information Exchange Syntax Standard[11]	See RFC 7292. symmetric key. F This container fo format for the Ja
PKCS #13	–	Elliptic-curve cryptography Standard	*(Apparently aba*
PKCS #14	–	Pseudo-random Number Generation	*(Apparently aba*
PKCS #15	1.1	Cryptographic Token Information Format Standard[13]	Defines a standa implementation

PKI

Simple Certificate Enrollment Protocol

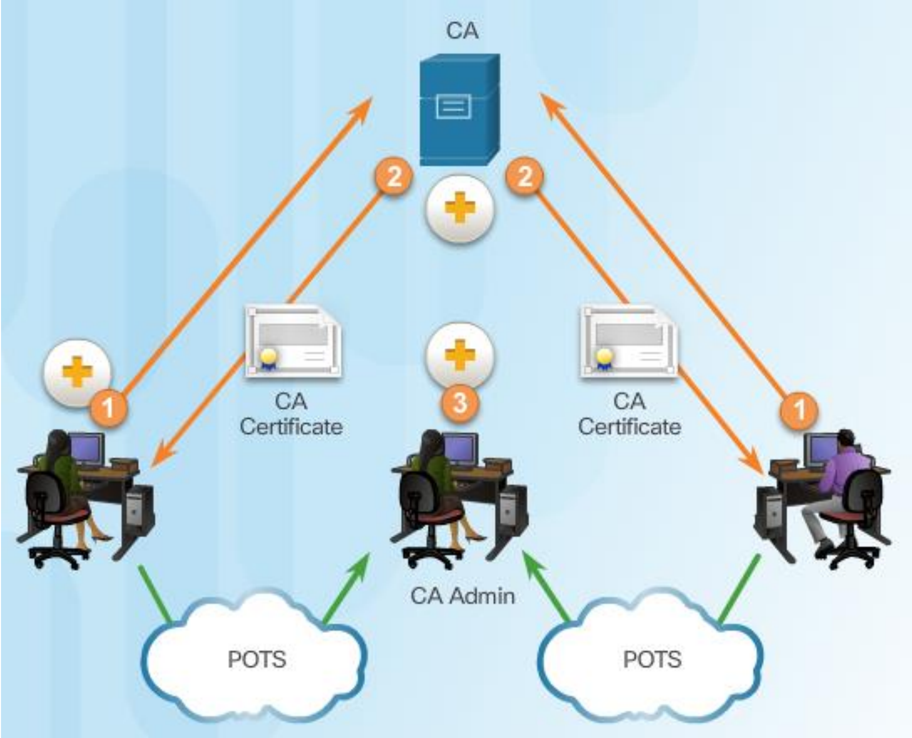
- PKI technology
  - Actually is extensively used
  - Became defacto basis for standardized security
  - Arises requirements on scalable certificates management suitable for PKI clients and CA servers
    - Whom perform enrolment, revocation, lifecycle…
  - Previously almost processed manually by admins
    - Not suitable
      - For large deployments
      - For easy and electronic use
- => IETF is preparing the **Simple Certificate Enrollment Protocol** (SCEP) (only draft)
  - Effort to make issuing and revocation of digital certificates as scalable as possible
  - Allows simplified means of handling certificates for large-scale implementation
    - secure issuance of certificates to network devices



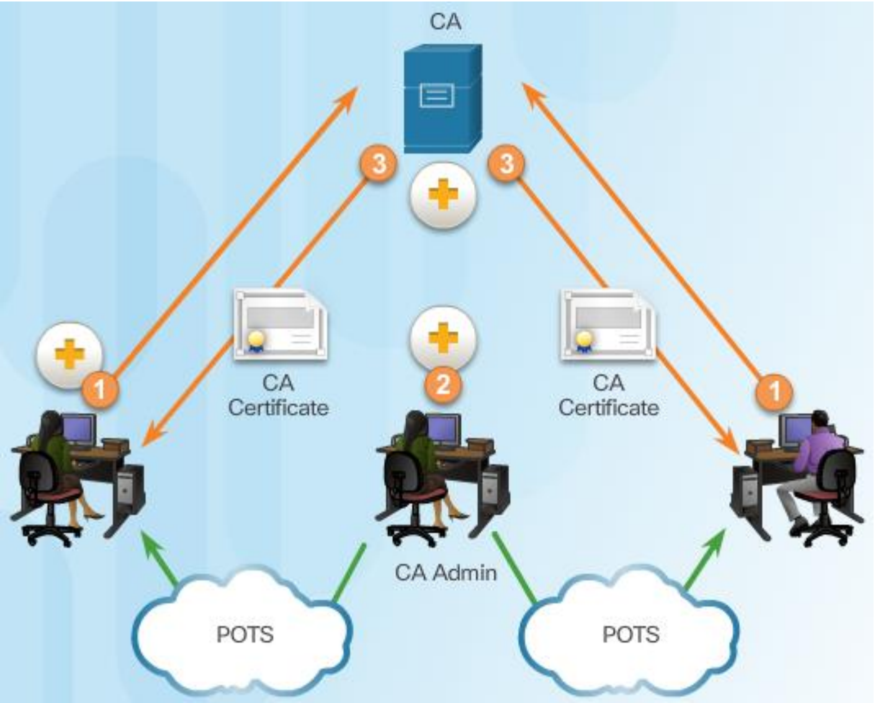
https://datatracker.ietf.org/doc/rfc8894/

Digital Certificates and CAs - processes

- 1-2) User has to securely obtain CA's public key (self-signed certificate)
  - Key verifies all the certificates issued by the CA
    - it is essential for PKI, only root CA issues it
  - Downloaded in-band, stored locally
- 3) User authenticates CA's certificate out-of-band to CA admin (personally, call …)
  - Serial number

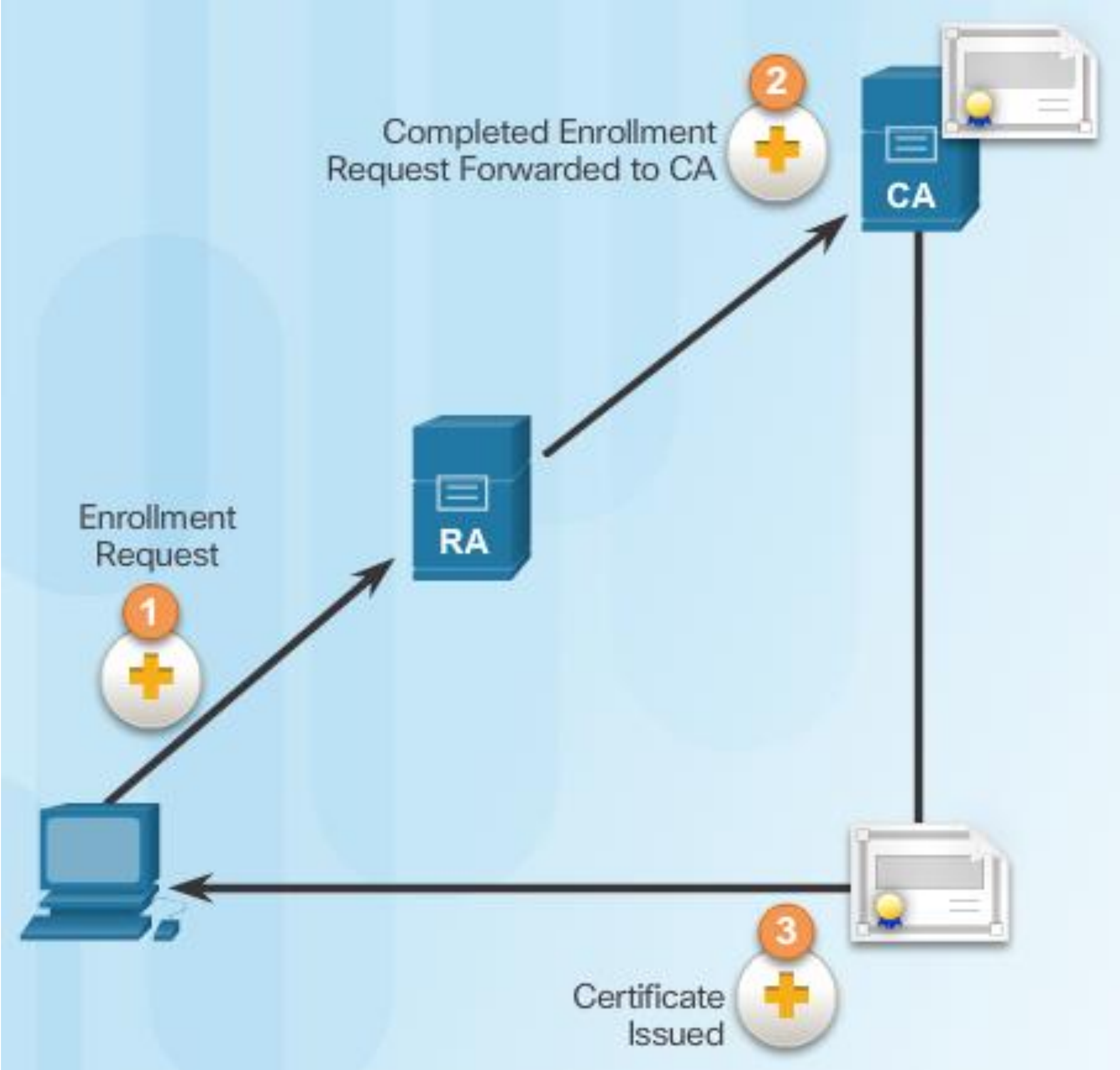


- 1) Users submit his certificate requests
- 2) CA server receive requests
  - CA admin call to users for confirmation
  - Add registration data, digitally sign it, issues certificates
- 3) Users download and install theirs certs
  - manually or using SCEP



Registration Authority - RA

- Reduce the burden of CAs
  - where is high volume of cert transaction
  - RA can accept requests for enrollment in the PKI
    - Identification and authentication of subscribers
    - Accept registration and certificate requests (1)
    - Forwards requests to CA (2)
    - Does not sign or issues certs,
      - only CA may do that (3)
- May handle three tasks
  - Authentication of users when they enroll with the PKI
  - Key generation for users that cannot generate their own keys
  - Distribution of certificates after enrollment

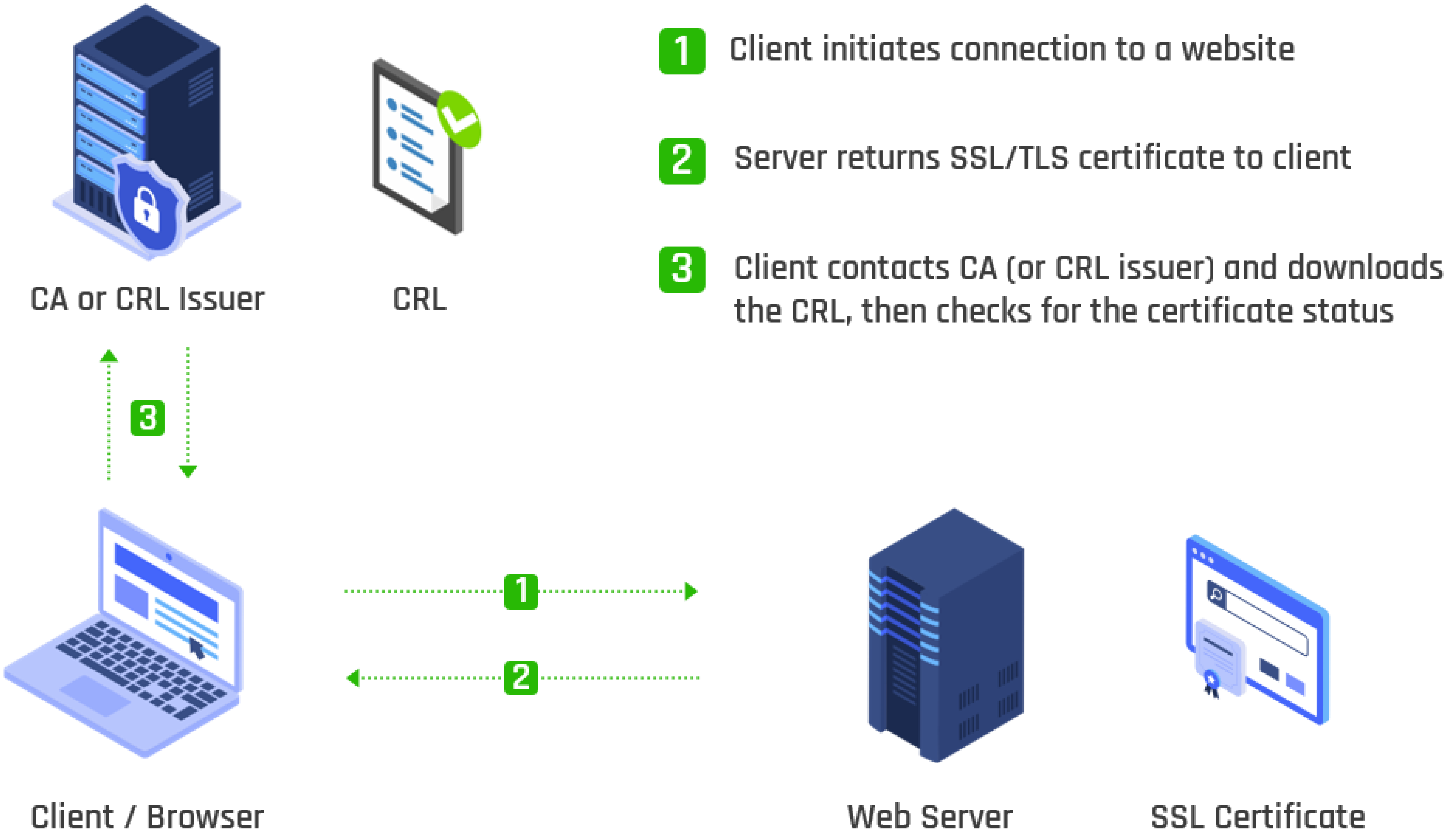


Certificate Enrollment, Authentication, and Revocation

- All systems that leverage the PKI
  - must have the CA's public key =
    = called as the self-signed certificate
  - CA public key verifies all the certificates issued by the CA and is vital for the proper operation of the PKI
  - For many systems such as web browsers, the distribution of CA certificates is handled automatically (certs are preinstalled)
- The certificate enrollment process
  - Used by a host system to enroll with a PKI.
  - Therefore
    - CA certificates are retrieved **in-band** over a <u>network</u>
    - the authentication is done **out-of-band** (OOB) using the <u>telephone</u>
- Further authentication
  - no longer requires the presence of the CA server
  - each user exchanges their certificates containing public keys
- Revocation
  - Certificates must sometimes be revoked
    - if key is compromised or if it is no longer needed
  - Two of the most common methods of revocation
    - Certificate Revocation List (CRL)
      - A list of revoked certificate serial numbers that have been invalidated because they expired.
      - PKI entities regularly poll (query) the CRL repository to receive the current CRL.
    - Online Certificate Status Protocol (OCSP)
      - Protocol used to query an OCSP server for the revocation status of an X.509 digital certificate.

2 methods of revocation: 1. Certificate Revocation List (CRL)

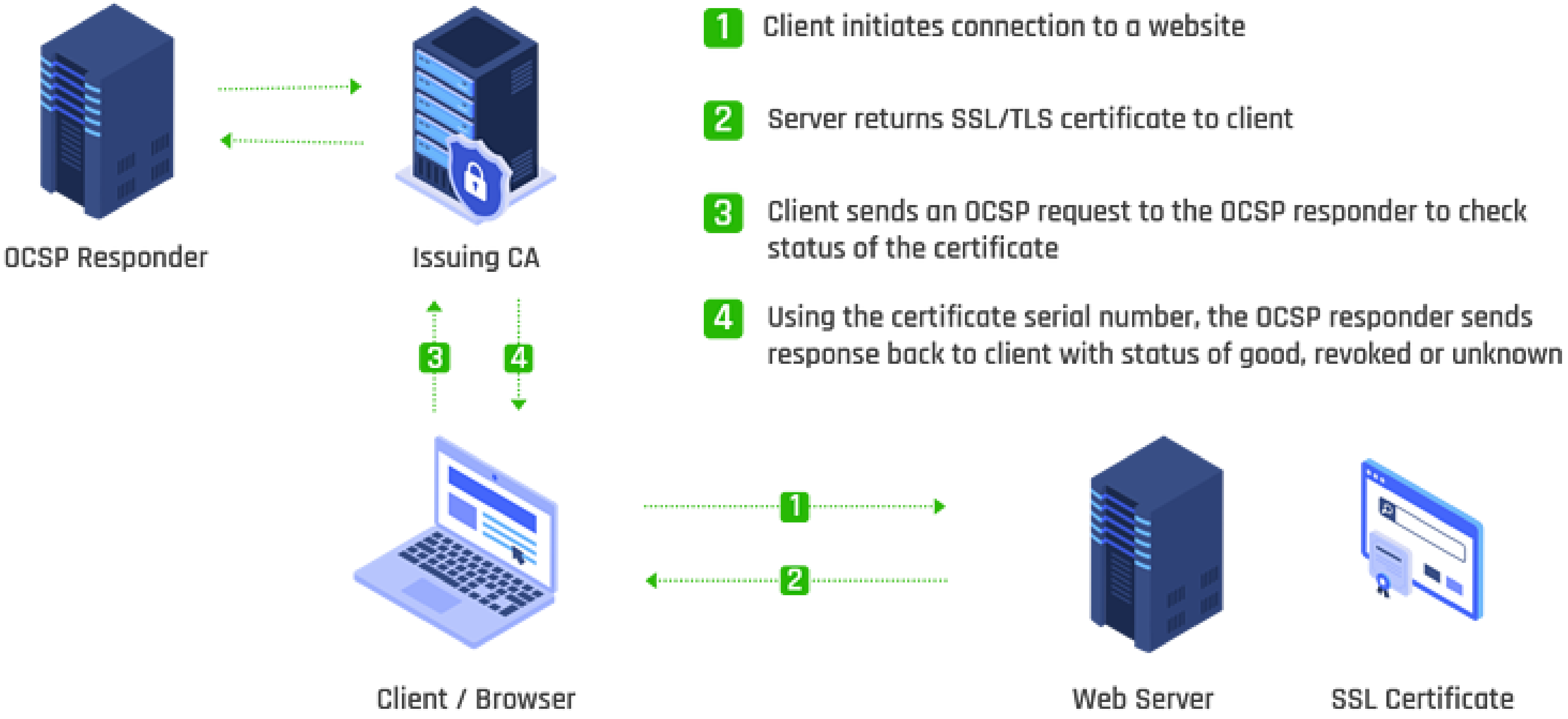
How web browsers and applications check CRLs



2 methods of revocation: 1. Online Certificate Status Protocol (OCSP)

How web browsers and applications check CRLs

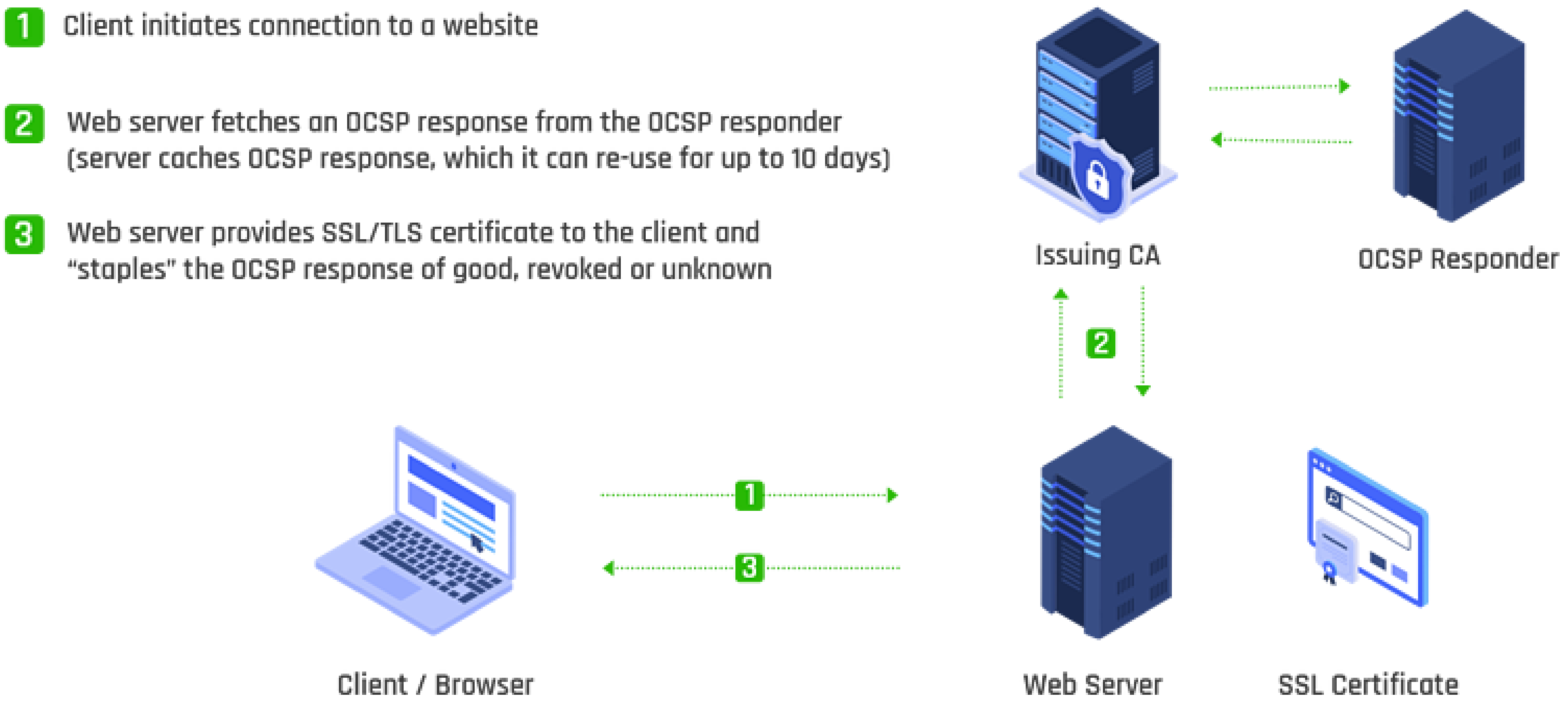
- Another method used to convey information to users about revoked certificates is the Online Certificate Status Protocol (OCSP). Instead of downloading the latest CRL and parsing it to check whether a requested certificate on the list, the browser requests the status for a particular certificate from the issuing CA's revocation server.



Enhancement tu the standard OCSP protocol (RFC 6066)

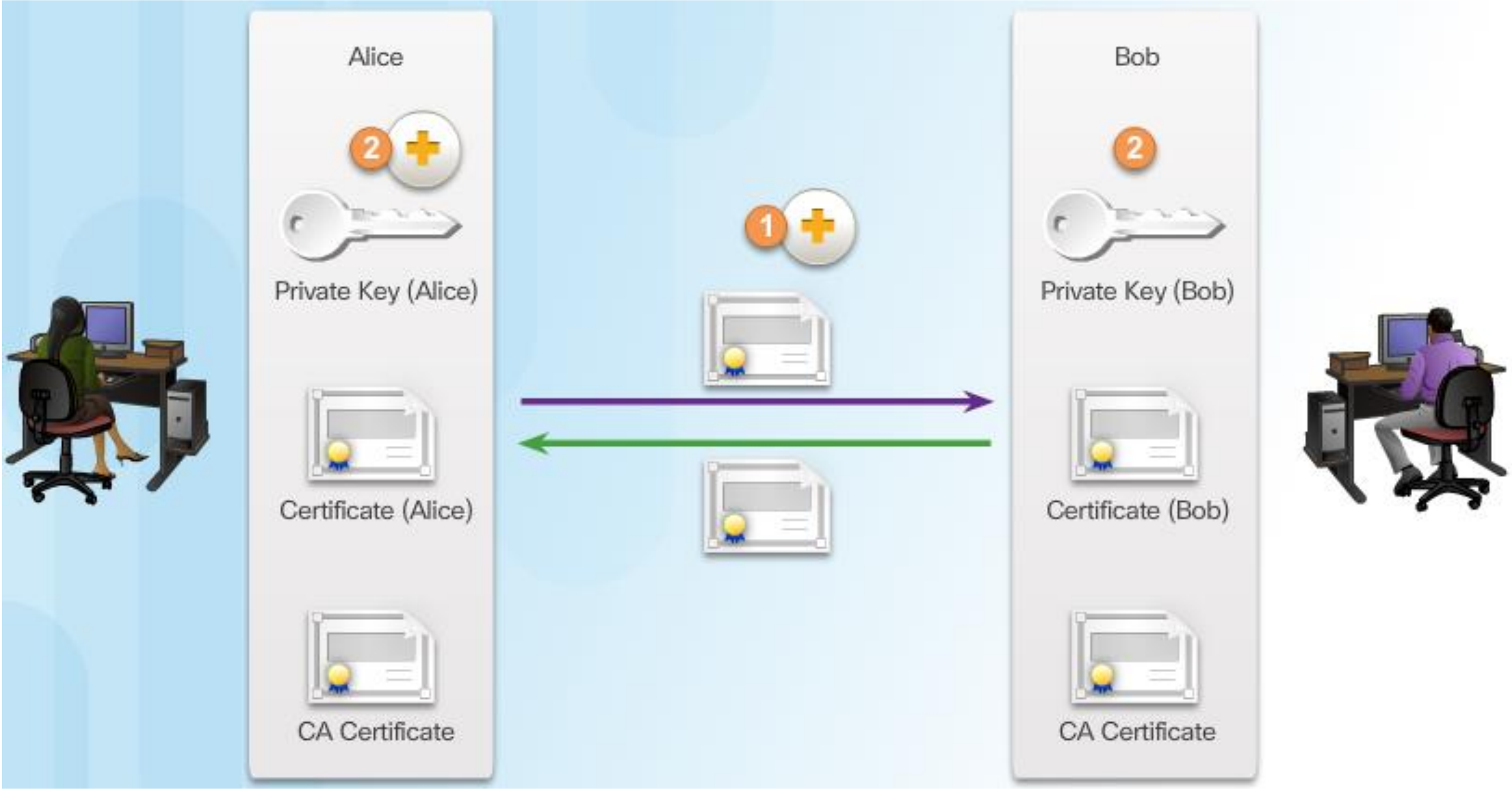
OSCP stapling

- the web server caches the OSCP response from the CA and when a TLS handshake is initiated by the client, the web server “staples” the OSCP response to the certificate it sends to the browser.



Digital Certificates and CAs - processes

- And they may start communicate securely

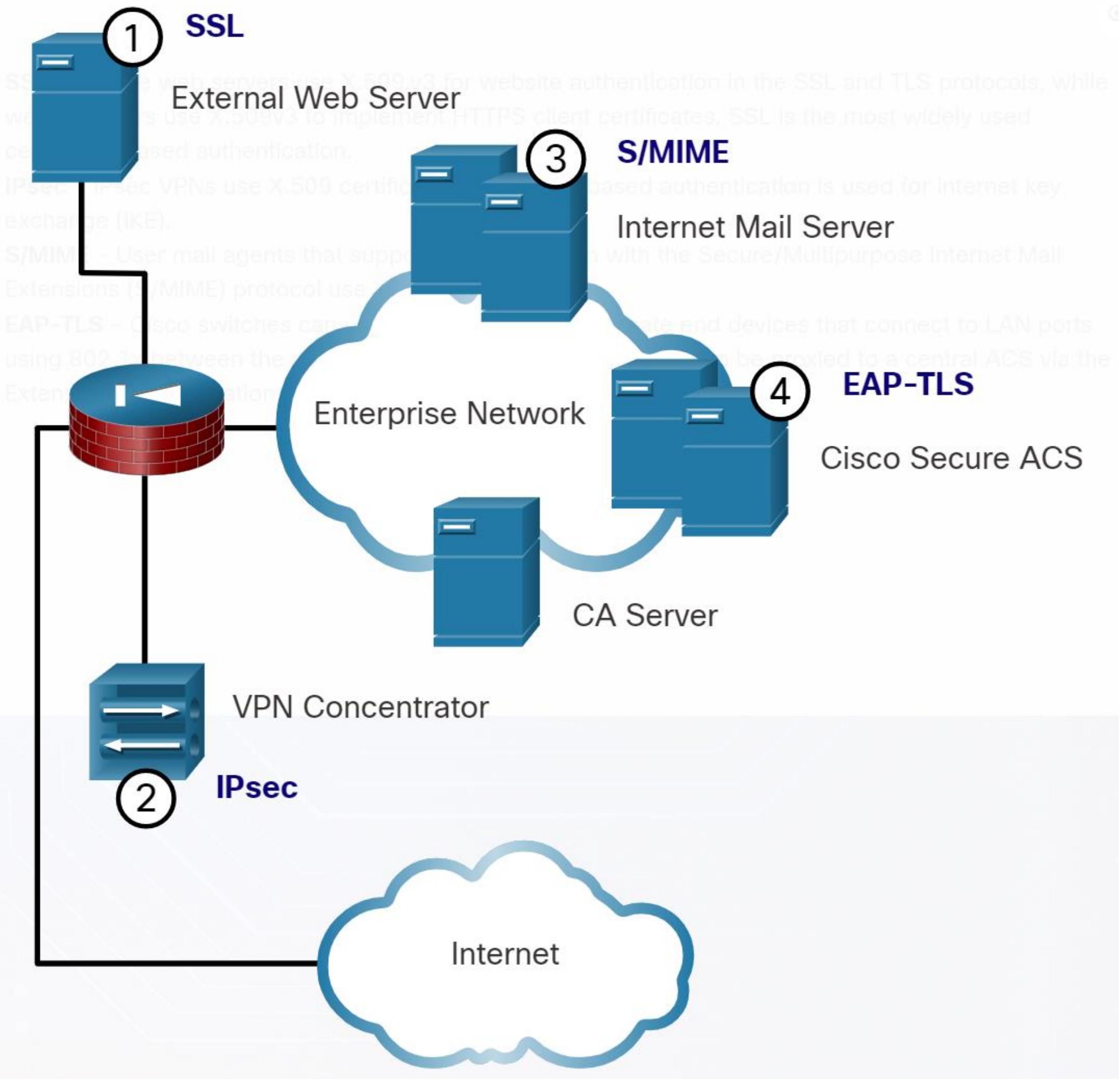


21.5 Applications and Impacts of Cryptography

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

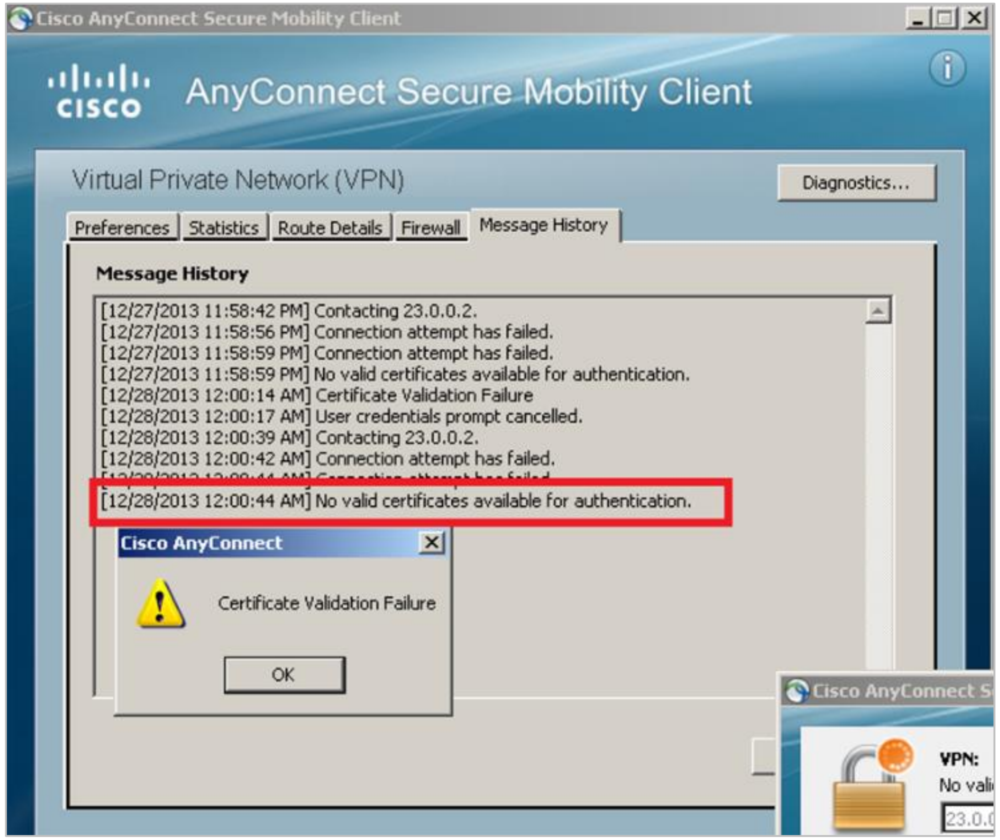
PKI Applications

- Common uses of PKI in an enterprise:
  1. **SSL/TLS** certificate-based peer authentication
  2. Secure network traffic using **IPsec VPNs**
  3. Secure **email** using the **S/MIME** protocol
  4. Control access to the network using **802.1x** authentication + **EAP TLS**
  5. **HTTPS Web** traffic
  6. Secure **instant messaging**
  7. Approve and authorize applications with **Code Signing**
  8. Protect user data with the Encryption File System (**EFS**)
  9. Implement **two-factor authentication** with **smart cards**
  10. Securing **USB storage** devices



Encrypted Network Transactions

- Threat actors can use SSL/TLS to introduce regulatory compliance violations, viruses, malware, data loss, and intrusion attempts in a network.
- Other SSL/TLS-related issues may be associated with validating the certificate of a web server. When this occurs, the web browsers will display a security warning. PKI-related issues associated with security warnings include:
  - **Validity date range** - The X.509v3 certificates specify “not before” and “not after” dates. If the current date is outside the range, the web browser displays a message.



Signature validation error

- If a browser cannot validate the signature on the certificate
- there is no assurance that the public key in the certificate is authentic.

Encryption and Security Monitoring

- Network monitoring becomes **more challenging when packets are encrypted**.
- As HTTPS introduces end-to-end encrypted HTTP traffic (via TLS/SSL), it is not as easy to peek into user traffic.
- Security analysts must know how to circumvent and solve these issues. Here is a list of some of the things that a security analyst could do:
  - Configure rules to distinguish between SSL and non-SSL traffic, HTTPS and non-HTTPS SSL traffic.
  - Enhance security through **server certificate** validation using CRLs and OCSP.
  - Implement antimalware protection and URL filtering of HTTPS content.
  - Deploy a **Cisco SSL Appliance** to **decrypt SSL** traffic and send it to intrusion prevention system (IPS) appliances to identify risks normally hidden by SSL.
- Why CRL and OCSP monitoring is critical
  - If a CA is down, it is not unable to issue new certificates, but if the CRL is expired or unreachable, all certificates become immediately unusable.
  - As discussed, most applications need to check the validity of certificates against a CRL or OCSP server. If they cannot reach the CDP or OCSP responder, or if the CRL itself is expired, users won't be able to access their application

Encryption and Security Monitoring (Contd.)

- Cryptography is dynamic and always changing.
  - a security analyst must maintain a good understanding of cryptographic algorithms and operations
  - to be able to investigate cryptography-related security incidents.
- There are 2 main ways in which cryptography impacts security investigations.
  1. attacks can be directed to specifically target the encryption algorithms themselves.
     - After the algorithm has been cracked and the attacker has obtained the keys
       - any encrypted data that has been captured can be decrypted by the attacker and read
         - thus exposing private data.
  2. the security investigation is also affected because data can be hidden in plain sight by encrypting it.

PKI
Final notes

- PKI is mature technology, widely supported
- Problems
  - Complex and pricy to built up
  - How to built net of trust
    - Lot of local, regional, national or global PKI providers
  - PKI and certificate issuing is service provided usually for a price
    - Ignoring self-signed certs
    - Price depends
      - Type of cert (email, ssl/tls), validity, warranty: from tens to hundreds dollars per year and cert

Rank	Issuer	Usage	Market share
1	IdenTrust	20.4%	39.7%
2	Comodo	17.9%	34.9%
3	DigiCert	6.3%	12.3%
4	GoDaddy	3.7%	7.2%
5	GlobalSign	1.8%	3.5%

- Trends
  - Open CA – global SSL/TLS certs free of charge
    - Let's Encrypt
    - CAcert.org
  - SSL Free certs
  - BlockChain PKI
    - CertCoin
    - FlyClient
    - BlockQuick
- Criticism
  - Price
  - Security issues
    - CA compromise, attack on key storage, implementation weaknesses,

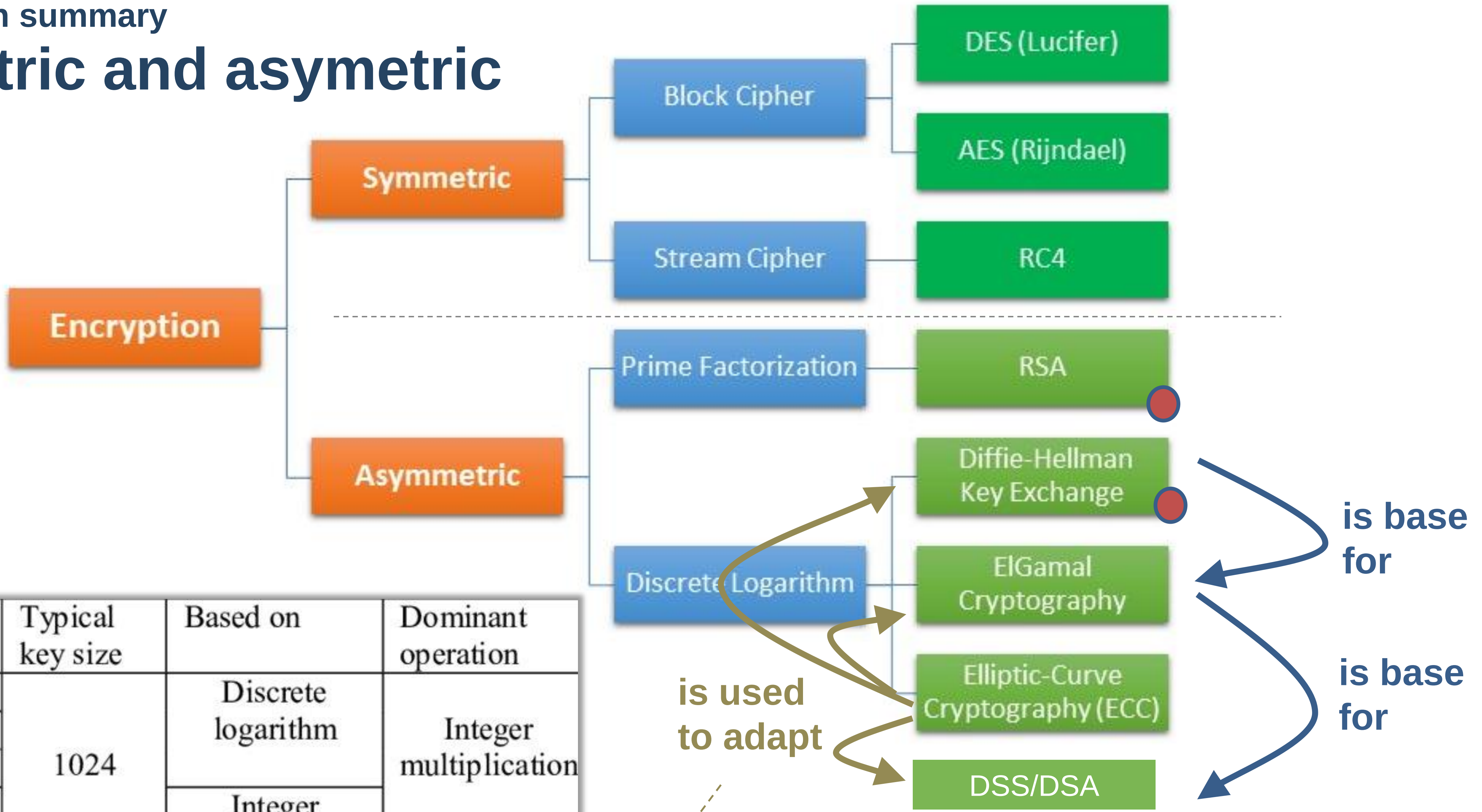
Platené SSL certifikáty – distribútor pre SR: https://www.verisign.sk/?gclid=Cj0KCQjwzozsBRCNARIsAEM9kBOU_CXDk39Ke46vy5mEB-yUoU6Vt0zpTyw0kPk3V5_J2G11MvkWoooaAiMBEALw_wcB

21.6 Public Key Cryptography Summary

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Encryption summary

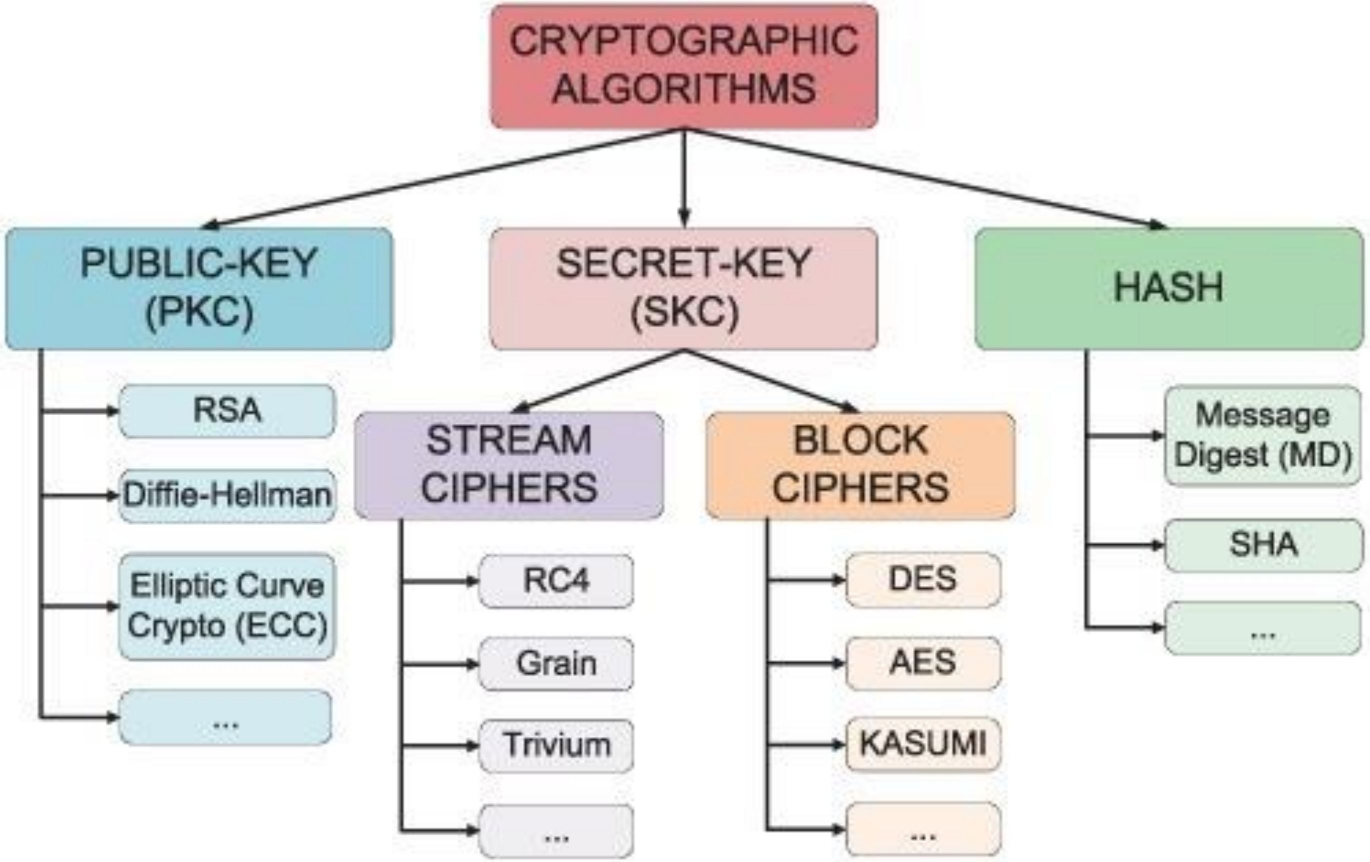
Symetric and asymetric



Name	Typical key size	Based on	Dominant operation
Diffie-Hellman	1024	Discrete logarithm	Integer multiplication
El-Gamal			
DSA			
RSA		Integer factoring	
RSA Signature			
eDH	163	Elliptic-Curve Discrete logarithm	Polynomial multiplication
eEl-Gamal			
eDSA			

Secret key = symetric, public key = asymetric, hash

PARAMETERS	RSA	DIFFIE-HELLMAN
Ephemeral Keys	Generating ephemeral keys for RSA is extremely difficult.	Generating ephemeral keys for Diffie-Hellman is extremely easy.
Security	Relies on the difficulty of integer factorization.	Relies on the difficulty of discrete logarithm.
Encryption Cost	Encryption is cheaper.	Encryption is expensive.
Public Key Encoding	Public key is smaller to encode.	Public key is bigger to encode.
Strength	RSA 1024 bits is less robust than Diffie-Hellman.	Diffie-Hellman 1024 bits is much more robust.
Authentication	Authenticates only the sender.	Authenticates both the sender and the receiver.
Attacks	Susceptible to low exponent, common modules and cycle attack.	Susceptible to man-in-the-middle attack.



What Did I Learn in this Module?

- The four elements of secure communications: data integrity, origin authentication, data confidentiality, and data non-repudiation.
- A hash function takes a variable block of binary data, called the message, and produces a fixed-length, condensed representation, called the hash.
- There are two classes of encryption that are used to provide data confidentiality: asymmetric and symmetric.
- Symmetric encryption algorithms, such as DES, 3 DES, and AES are based on the premise that each communicating party knows the pre-shared key.
- Asymmetric algorithms (public key algorithms) are designed so that the key that is used for encryption is different from the key used for encryption.
- Data confidentiality can also be ensured using asymmetric algorithms, including Rivest, Shamir, and Aldeman (RSA) and PKI. The process is summarized using this formula: Public key (Encrypt) + Private Key (Decrypt) = Confidentiality.

Public Key Cryptography Summary

What Did I Learn in this Module? (Contd.)

- The authentication objective of an asymmetric algorithm is initiated when the encryption process is started with the private key. The process can be summarized with this formula: Private Key (Encrypt) + Public Key (Decrypt) = Authentication.
- Diffie-Hellman (DH) is an asymmetric mathematical equation algorithm that allows two computers to generate an identical shared secret key without having communicate before.
- Digital signatures are a mathematical technique used to provide three basic security services: authenticity, integrity, and non-repudiation. Digital signatures are commonly used in code signing and digital certificates.
- The Public Key Infrastructure (PKI) consists of specifications, systems, and tools that are used to create, manage, distribute, use, store, and revoke digital certificates.
- There are many common uses of PKIs including a few listed here: SSL/TLS certificate-based peer authentication, HTTPS Web traffic, secure instant message, and securing USB storage devices.
- A security analyst must be able to recognize and solve potential problems related to permitting PHI-related solutions on the enterprise network.

Module 22: **Endpoint Protection**

Module Objective: Explain approaches to network security defense

Topic	Topic Objective
Antimalware Protection	Explain methods of mitigating malware
Host-based Intrusion Prevention	Explain host-based IPS/IDS log entries
Application Security	Explain how a sandbox is used to analyze malware

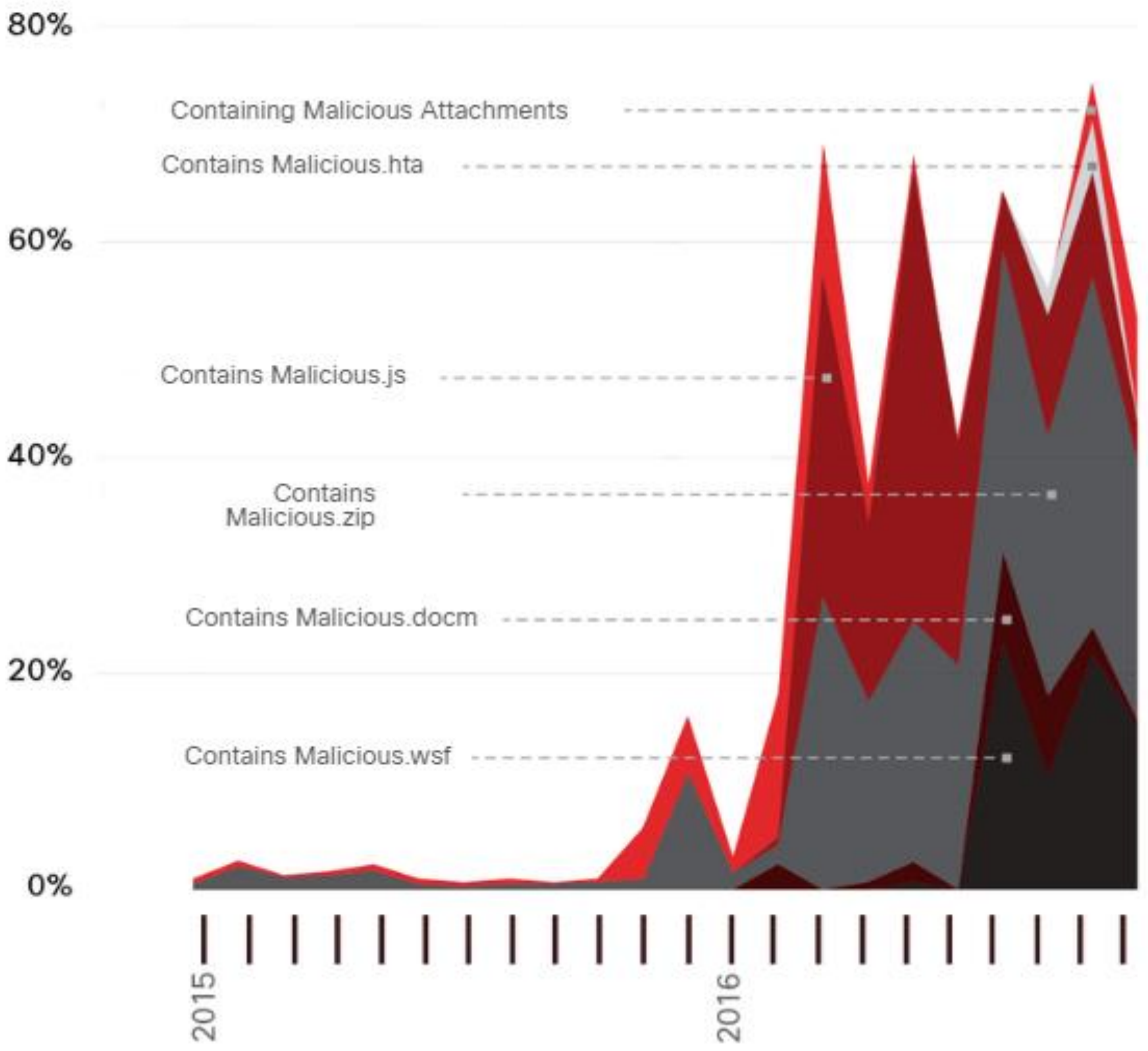
22.1 Antimalware Protection

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Endpoint (E) Threats

- E = host on the network
  - that can access the network
  - or be accessed by other hosts
  - or remotely access network through VPN
- E is potentially a way for malicious SW
  - to gain access to a network
  - to inject malware into the VPN net from public net
- CAMOUFLAGE by malware
  - several common types of malware have been found to significantly change features in less than 24 hours in order to evade *(vyhnúť sa)* detection

Malicious Spam Percentage



Source: Cisco Security Research

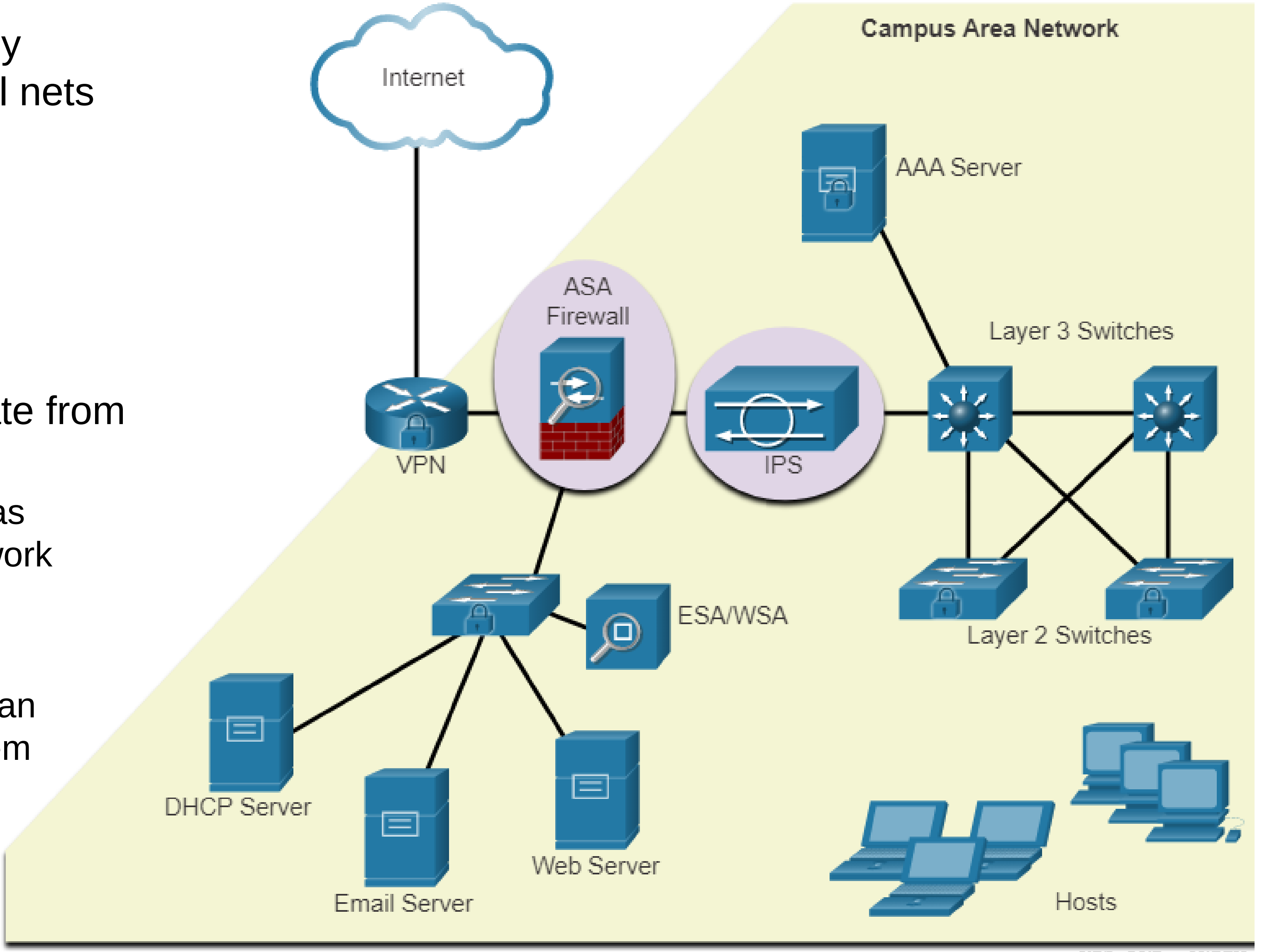
Endpoint Protection

Endpoint Security

There are two internal LAN elements to secure:

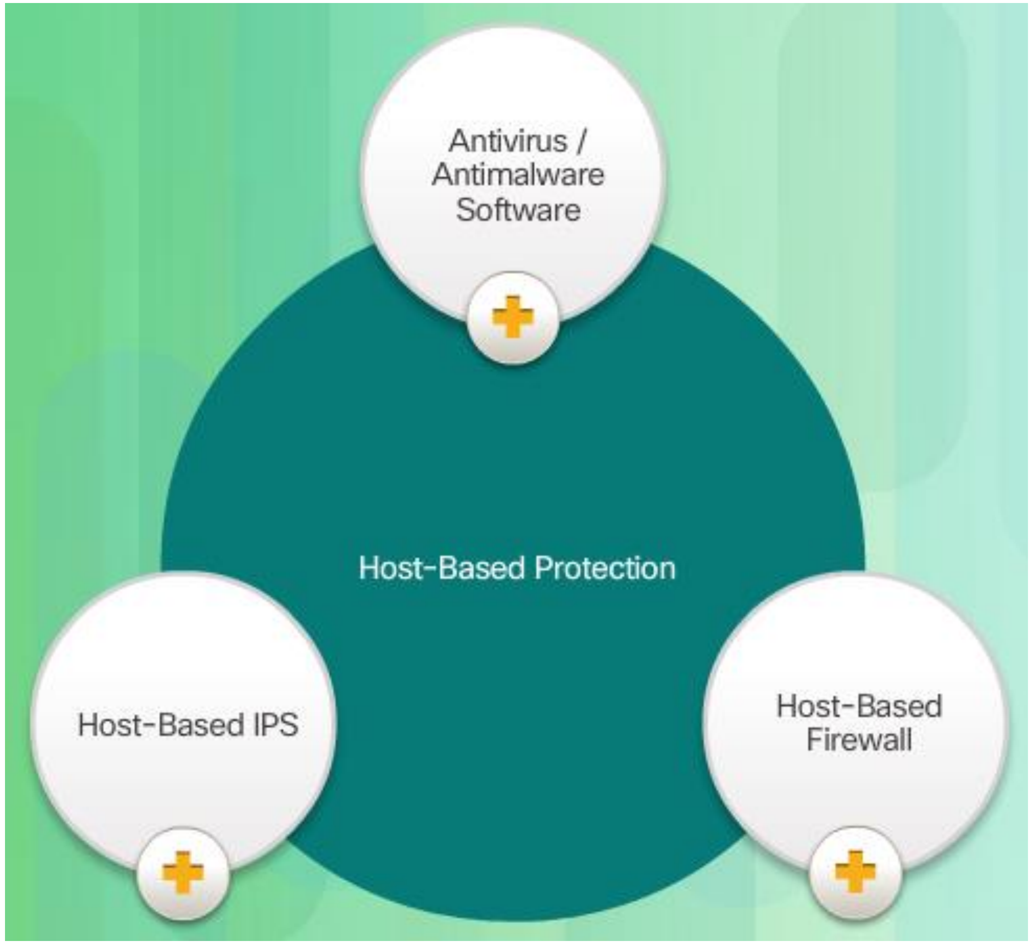
1. Endpoints - susceptible to malware-related attacks.
2. Network infrastructure – LAN devices

- People under a network attack usually imagine attacks from outside external nets
  - DoS/DDoS
  - Breach of organization's servers
    - Web, data, mail …
- Before: focus of perimeter security
  - Hardened ISR, ASA, IPS, AAA
- However today: many attacks originate from **inside** the network
  - => securing an internal LAN is nearly as important as securing the outside network perimeter
- After an **internal** host is **infiltrated**
  - => it can become a **starting point** for an attacker to gain access to critical system devices
    - such as servers and sensitive information



Traditional Endpoint Security (before...)

- Endpoints
  - (Before) usually employee company-issued computers
    - With nicely defined security border – LAN access perimeter
  - (Before) protected a traditional way
    - OS security and updates
    - Application updates
    - Host based firewall
    - Host based IPS
    - Antivirus / antimalware



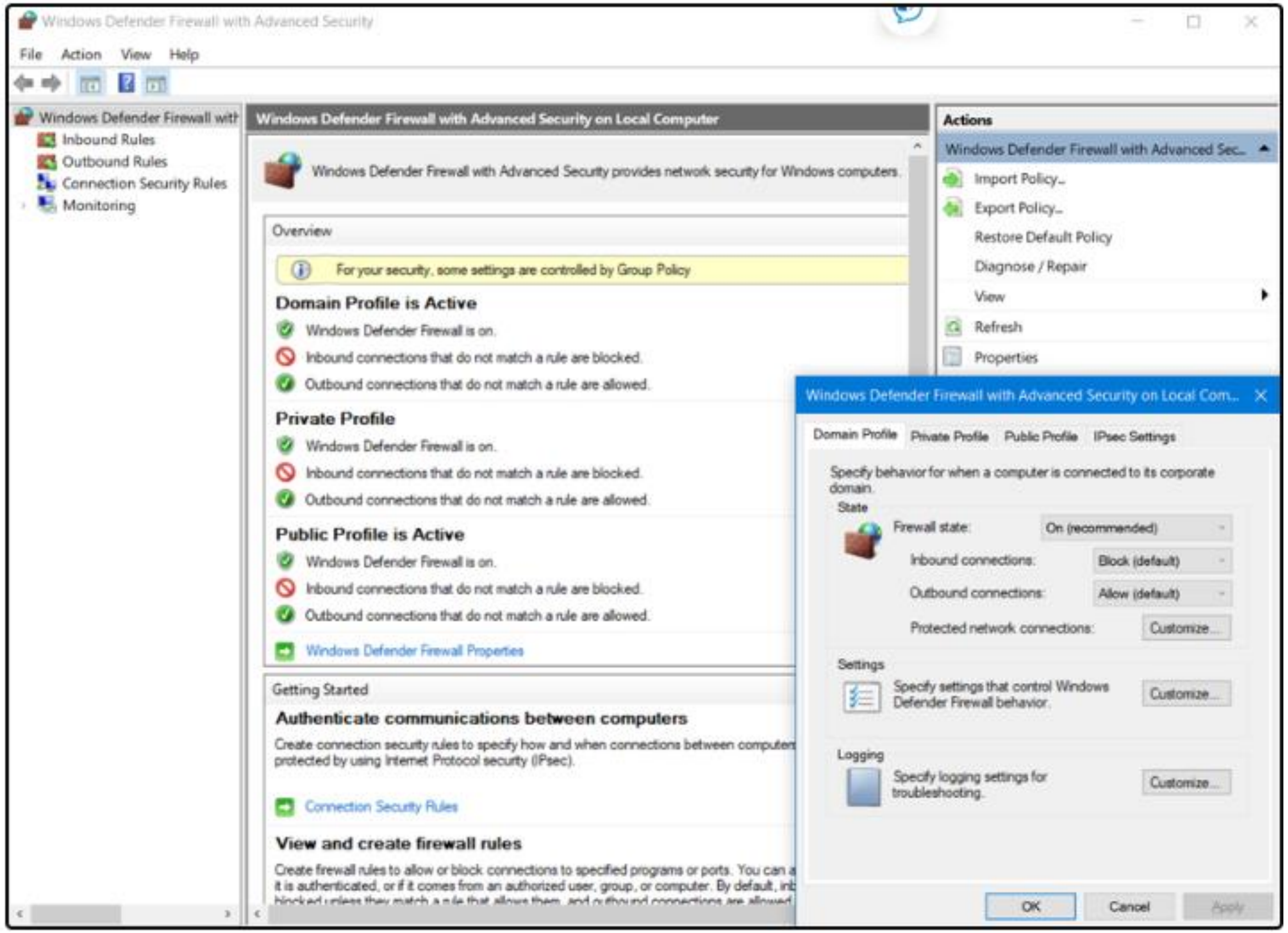
Endpoint Protection

Antivirus/Antimalware Software

- It is a software that is installed on a host to detect and mitigate viruses and malware
  - Windows Defender Virus & Threat Protection
  - Cisco AMP for Endpoints
  - Norton Security
  - McAfee
  - Trend Micro
  - and others.
- may detect viruses using three different approaches:
  - **Signature-based:** Recognizes various characteristics of known malware files
  - **Heuristics-based:** Recognizes general features shared by various types of malware
  - **Behavior-based:** Employs analysis of suspicious behavior
- Host-based antivirus protection, also known as agent-based, runs on every protected machine.
- Host-based antimalware/antivirus software and host-based firewalls are used to protect mobile devices using VPN

Host-based Firewall (FW) (… also for malware protection)

- restricts incoming and outgoing connections to connections initiated by that host only
- Some FW software can prevent a host from becoming infected and stop infected hosts from spreading malware to other hosts
  - This function is included in some OSs
    - For example, Windows includes Windows Defender Firewall with Advanced Security.



Endpoint Protection

Host-based Security Suites

- It is recommended to install
  - to provide a layered defense
  - that will protect against most common threats.
- These include
  - antivirus
  - anti-phishing
  - safe browsing
  - host-based intrusion prevention system
  - firewall capabilities
  - also telemetry function
  - Also includes robust logging functionality
    - that is essential to cyber security operations.
- Reviews of host-based protections
  - The independent testing laboratory AV-TEST provides high-quality reviews
    - https://www.av-test.org/en/

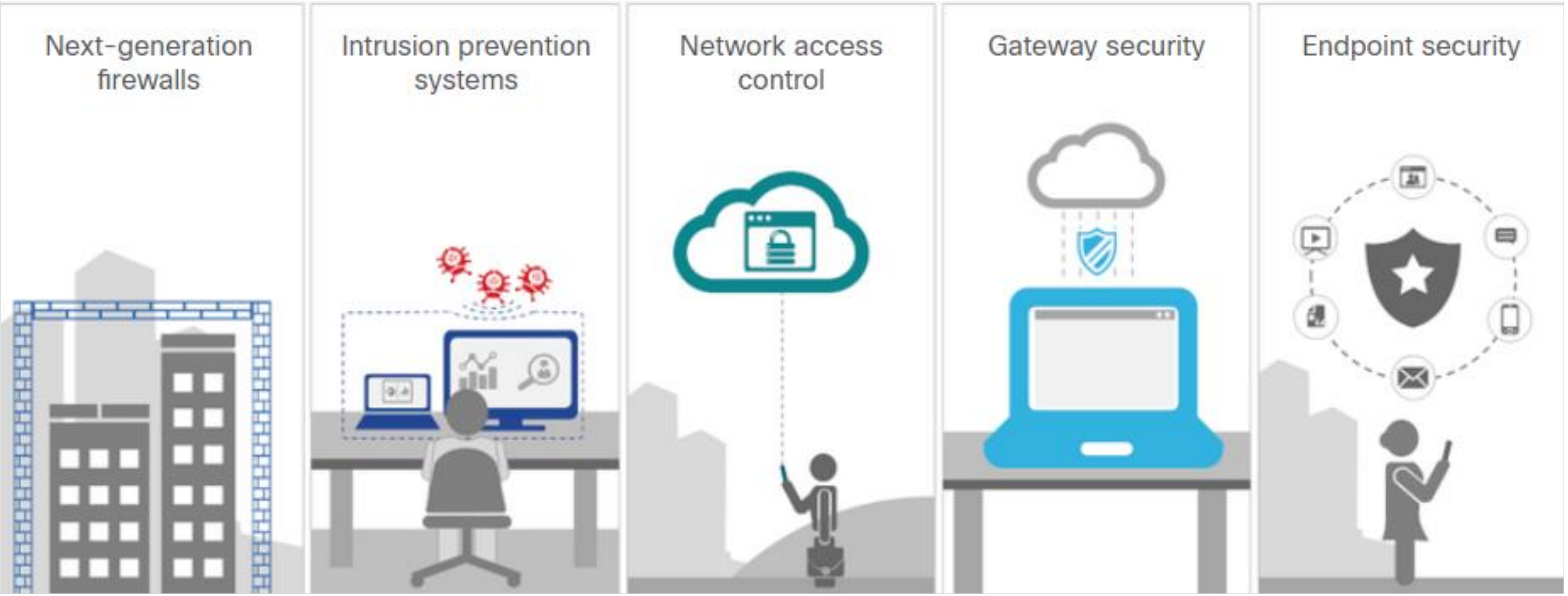
Windows 11: August 2022

Producer		Certified	Protection	Performance	Usability
AhnLab	V3 Internet Security 9.0	TOP PRODUCT	6	6	6
Avast	Free AntiVirus 22.6 & 22.7	TOP PRODUCT	6	6	6
Avast	One Essential 22.6 & 22.7	TOP PRODUCT	6	6	6
AVG	Internet Security 22.6 & 22.7	TOP PRODUCT	6	6	6
Avira	Internet Security for Windows 1.1	TOP PRODUCT	6	6	6
Bitdefender	Internet Security 26.0	TOP PRODUCT	6	6	6
F-Secure	SAFE 18	TOP PRODUCT	6	5.5	6
G DATA	Total Security 25.5	TOP PRODUCT	6	6	6
K7 SECURITY	Total Security 16.0	TOP PRODUCT	6	6	6
kaspersky	Internet Security 21.3	TOP PRODUCT	6	6	6
Malwarebytes	Premium 4.5	TOP PRODUCT	6	6	6
McAfee	Total Protection 26.0	TOP PRODUCT	6	5.5	6
Microsoft	Defender 4.18		5.5	4.5	6
eScan Enterprise Security	eScan Internet Security Suite 22.0		5.5	5.5	6
norton	Norton 360 22.22	TOP PRODUCT	6	6	6

Network-Based Malware Protection

- Network-based malware prevention devices are capable of **sharing information** among themselves to make better informed decisions.
- Protecting endpoints in a borderless network can be accomplished using **network-based**, as well as **host-based** techniques.

NGFW IPS NAC GW sec EPS

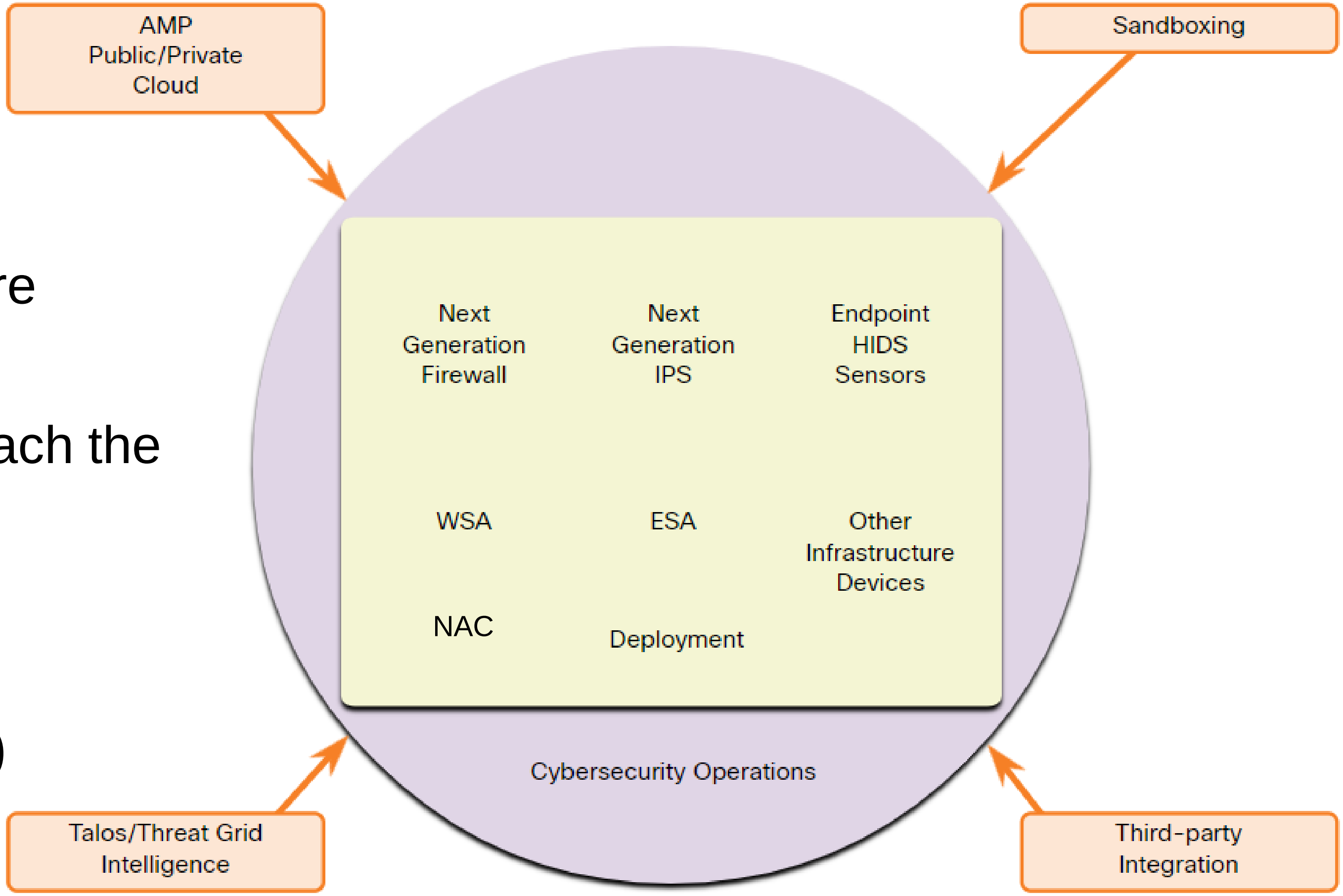


Advanced Malware Protection Everywhere

Network-Based Malware Protection (Contd.)

Some examples of devices and techniques that implement host protections at the network level:

- Antimalware Protection (AMP)
  - Protection from viruses and malware
- Email Security Appliance (ESA)
  - SPAM mails filtering before they reach the endpoint
- Web Security Appliances (WSA)
  - Website filtering and blacklisting
- Network Admission Control (NAC)
  - Perform network access decisions
  - Only authorized and compliant systems may connect



22.2 Host-Based Intrusion Protection

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Host-Based Firewalls

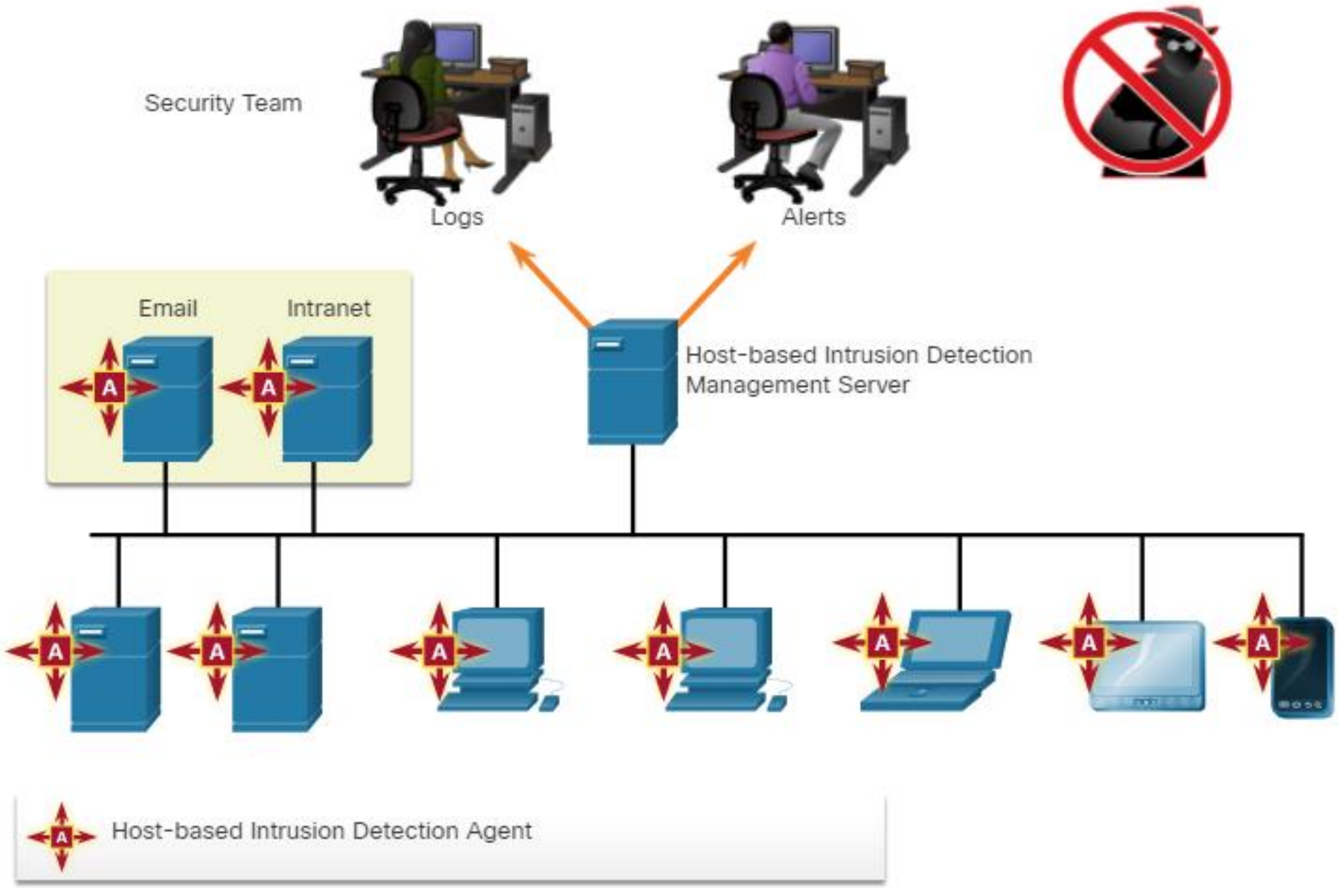
- standalone software programs that control traffic entering or leaving a computer
- can also be configured to issue alerts to users if suspicious behavior is detected
- Some examples of host-based firewalls:
  - **Windows Defender Firewall**
    - First included with Windows XP, Windows Firewall (now Windows Defender Firewall) uses a profile-based approach to firewall functionality
  - **iptables**
    - This is an application that allows Linux system administrators to configure network access rules that are part of the Linux kernel Netfilter modules.
  - **nftables**
    - The successor to iptables, nftables is a Linux firewall application that uses a simple virtual machine in the Linux kernel.
  - **TCP Wrappers**
    - This is a rule-based access control and logging system for Linux.

Host-Based Intrusion Detection (HIDS)

- Is designed to protect hosts against known and unknown malware
- can perform
  - detailed **monitoring**
  - and **reporting**

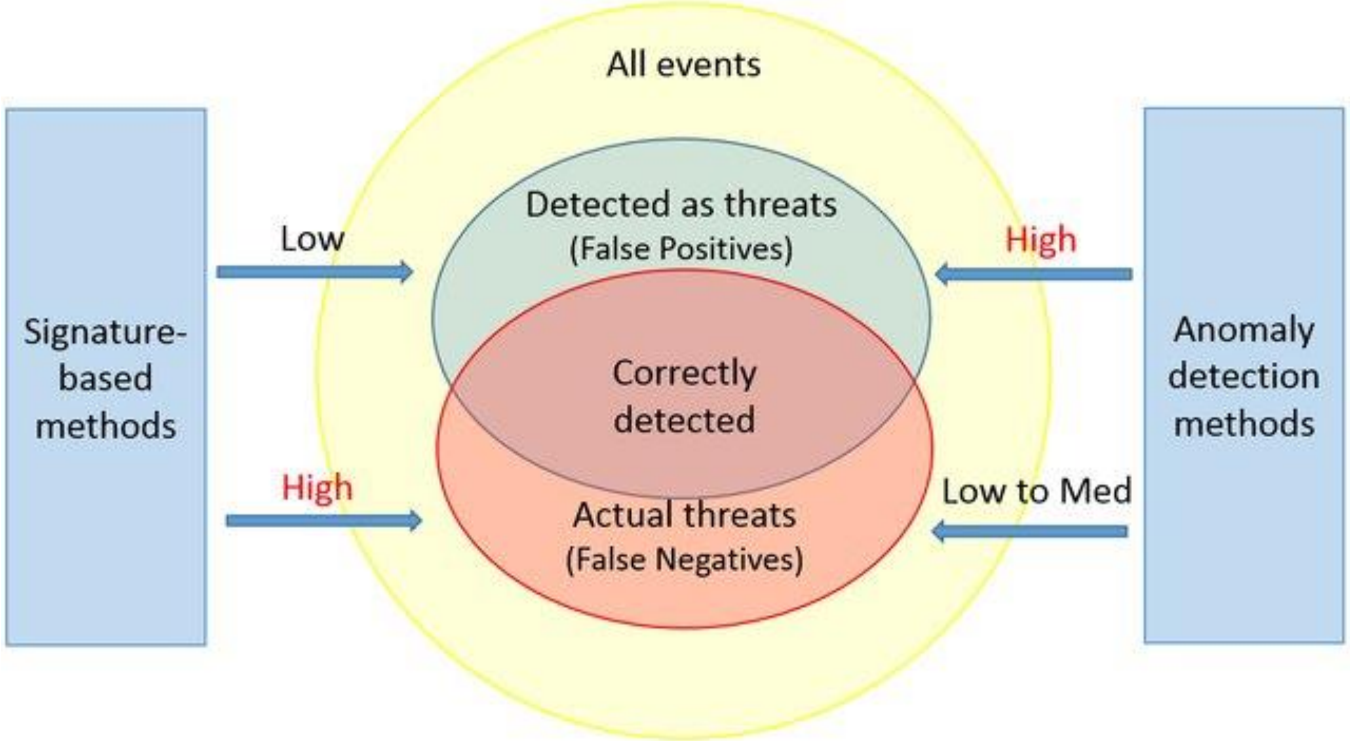
  on the system configuration and application activity
- is a comprehensive security application that **combines** the functionalities:
  - of antimalware applications
  - with firewall functionality
- must run directly on the host
  - => it is considered as an **agent-based system**

Host-based Intrusion Detection **Architecture**



Host-Based Intrusion Protection

HIPS Operation



- Note: Netacad materials don’t distinguish between HIDS and HIPS, in terminology, but we do
- HIPS can prevent intrusion
  - it uses signatures
    - to **detect** known malware
      - log details of the intrusion
      - send alerts to security management systems
      - and take action to prevent the attack
    - and **prevent** it from infecting a system
- Some malware families exhibit polymorphism = constantly changes its identifiable features in order to evade detection
  - signature detection (pattern-matching detection) is not enough => 2 other methods are used:
- HIPS:
  - **Signature based**
  - **Policy based**
    - Normal system behavior is described by rules, or the violation of rules, that are predefined
    - Violation of these policies => action by the HIDS, such as shut down of software processes, …
  - **Anomaly based**
    - Host system behavior is compared to a learned baseline model of normal behavior

HIPS Operation

- Most of the HIDS
  - utilize software on the host (agent-based)
  - and some sort of centralized security management functionality
    - which allows integration with network security monitoring services and threat intelligence
- Some examples are
  - Cisco **Cisco AMP**
  - AT&T **AlienVault USM** (Unified Security Management)
  - FORTRA **Tripwire**
  - OSSEC Open Source HIDS SECurity

Top 10 HIDS open-source tools

1. OSSEC
2. Zeek
3. Snort
4. Splunk
5. Open DLP
6. Sagan
7. Wazuh
8. Samhain
9. Papertrail
10. AgentSmith-HIDS

Formerly: Cisco AMP

Cisco secure endpoint

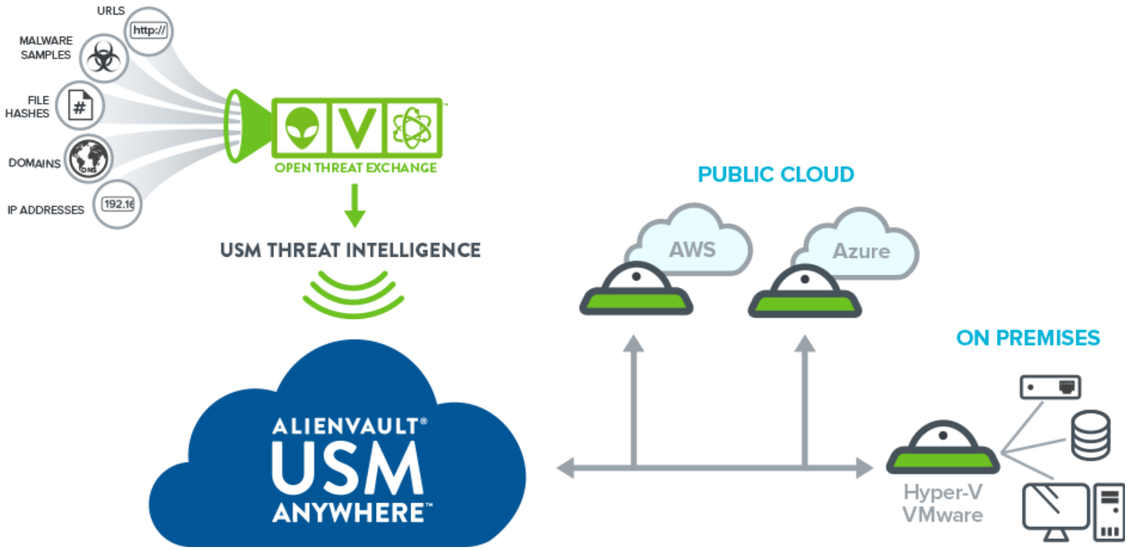
- Formerly Cisco AMP
- Contains EPP, EDR, XDR, plus:
- MDR - Managed Detection and Response
  - Incident Investigation: MDR Security service providers
    - will investigate an alert
    - and determine whether it is a true incident or a false positive
    - accomplished through a combination of
      - data analytics
      - machine learning
      - and human investigation
    - with Cisco dedicated global Security Operations Centers (SOCs)
      - 24x7x365 protection

	Secure Endpoint Essentials	Secure Endpoint Advantage	Secure Endpoint Premier
New! Secure MDR for Endpoint		Available	Available
Next-Gen Antivirus Protection	✓	✓	✓
Continuous Behavioral Monitoring	✓	✓	✓
Dynamic File Analysis	✓	✓	✓
Vulnerability Identification	✓	✓	✓
Endpoint Isolation	✓	✓	✓
Orbital Advanced Search		✓	✓
Secure Malware Analytics		✓	✓
SecureX Threat Hunting			✓
SecureX	Built-In	Built-In	Built-In
Support for Secure Endpoint Private Cloud	✓		

https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/advanced-malware-protection/at-a-glance-c45-731874.html

AlienVault USM platorm

- provides:
  - SIEM
  - vulnerability assessment
  - asset discovery
  - network and host intrusion detection
  - endpoint detection and response (EDR)
  - flow and packet capture
  - file integrity monitoring (FIM)
  - centralized configuration and management
  - AWS-native version is also available



- **Threats blocked:** USM Anywhere detects a broad range of threats, such as:
  - Data breaches
  - Ransomware
  - Advanced malware
  - Advanced persistent threats (APT)
  - Remote access trojans (RAT)
  - Cryptomining
  - Insider threats
  - Phishing attacks
  - DDoS and other threats
  - USM Anywhere also detects indicators of a threat/attack, such as:
- Unusual privilege escalation within an AWS or Azure account
  - Suspicious user downloads from Office 365 or G Suite
  - Bitcoin miners running on endpoints
  - Changes to critical server files or registry
  - Stolen user credentials trafficked on the dark web
  - Signs of lateral movement within a network
  - Communications with a ransomware C&C server

https://www.tripwire.com/solutions

Tripwire

FORTRA

Tripwire Integrity Management

- *„trip wire“ is a thin line of wire that is intended to cause an intruder to trip over it, thus triggering a mechanism elsewhere that will activate an alarm*

TripWIRE by Fortra:

- Supports File Integrity Monitoring (FIM) and Security Configuration Management (SCM)
- to provide real-time change intelligence and threat detection
  - Complete device and asset discovery
  - maintain a secure baseline configuration
  - monitor assets for deviations
  - Automation and guidance for rapid repair of misconfiguration
  - Create and enforce customized compliance policies
  - proactive system hardening and automated compliance enforcement

- **Continuous compliance with standards such as PCI-DSS**
  - PCI DSS is a requirement for any organization or sole trader that stores, processes, and/or transmits credit or debit cardholder data

Open Source HIDS SECurity

OSSEC

- Open-source platform to monitor and control systems
- Features:
  - HIDS (host-based intrusion detection)
  - log monitoring
  - Security Information and Event Management (SIEM)
- uses a central manager server and agents that are installed on individual hosts.
  - OSSEC monitors system logs on hosts and also conducts file integrity checking.
  - The OSSEC server, or Manager, can also receive and analyze alerts from a variety of network devices and firewalls over syslog.
- helps meet specific compliance requirements such as **PCI** and **HIPAA**
  - HIPAA - Health Insurance Portability and Accountability Act of 1996 is a federal law of US (protect sensitive patient health information from being disclosed without the patient's consent or knowledge)

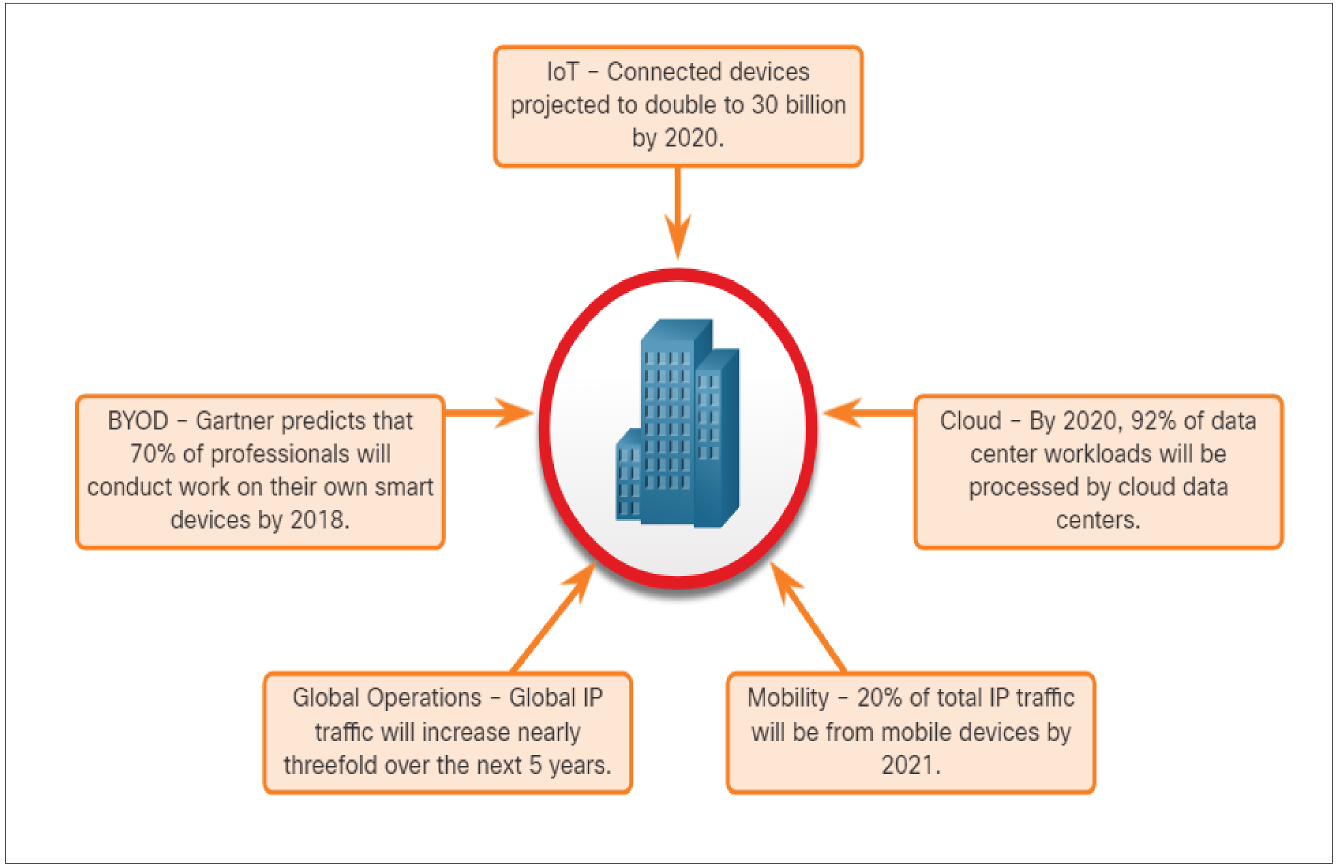
- Wazuh
  - uses anomaly and signature detection methods to:
    - detect rootkits
    - perform log analysis
    - integrity checking
    - Windows registry monitoring
    - plus active response
  - It differs from OSSEC in its **ability**
    - **to be integrated with ELK**
    - its **improved ruleset**
    - and the ability to use **restful API**

22.3 Application Security

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

Attack Surface

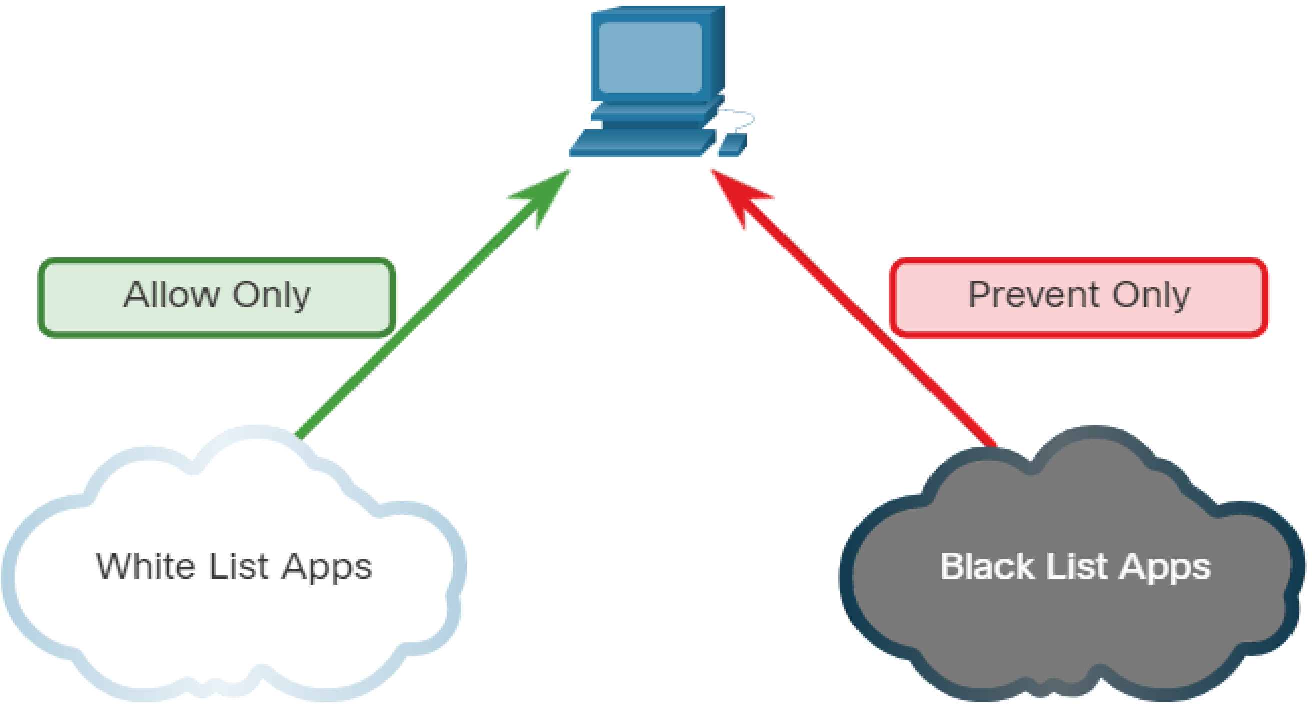
- An attack surface is the total sum of the vulnerabilities in a given system that is accessible to an attacker.
- It can consist of open ports on servers or hosts, software running on internet-facing servers, wireless network protocols, and users.
- Components of the Attack Surface:
  - **Network Attack Surface:** Exploits vulnerabilities in networks.
  - **Software Attack Surface:** Delivered through exploitation of vulnerabilities in web, cloud, or host-based software applications.
  - **Human Attack Surface:** Exploits weaknesses in user behavior.



An Expanding Attack Surface

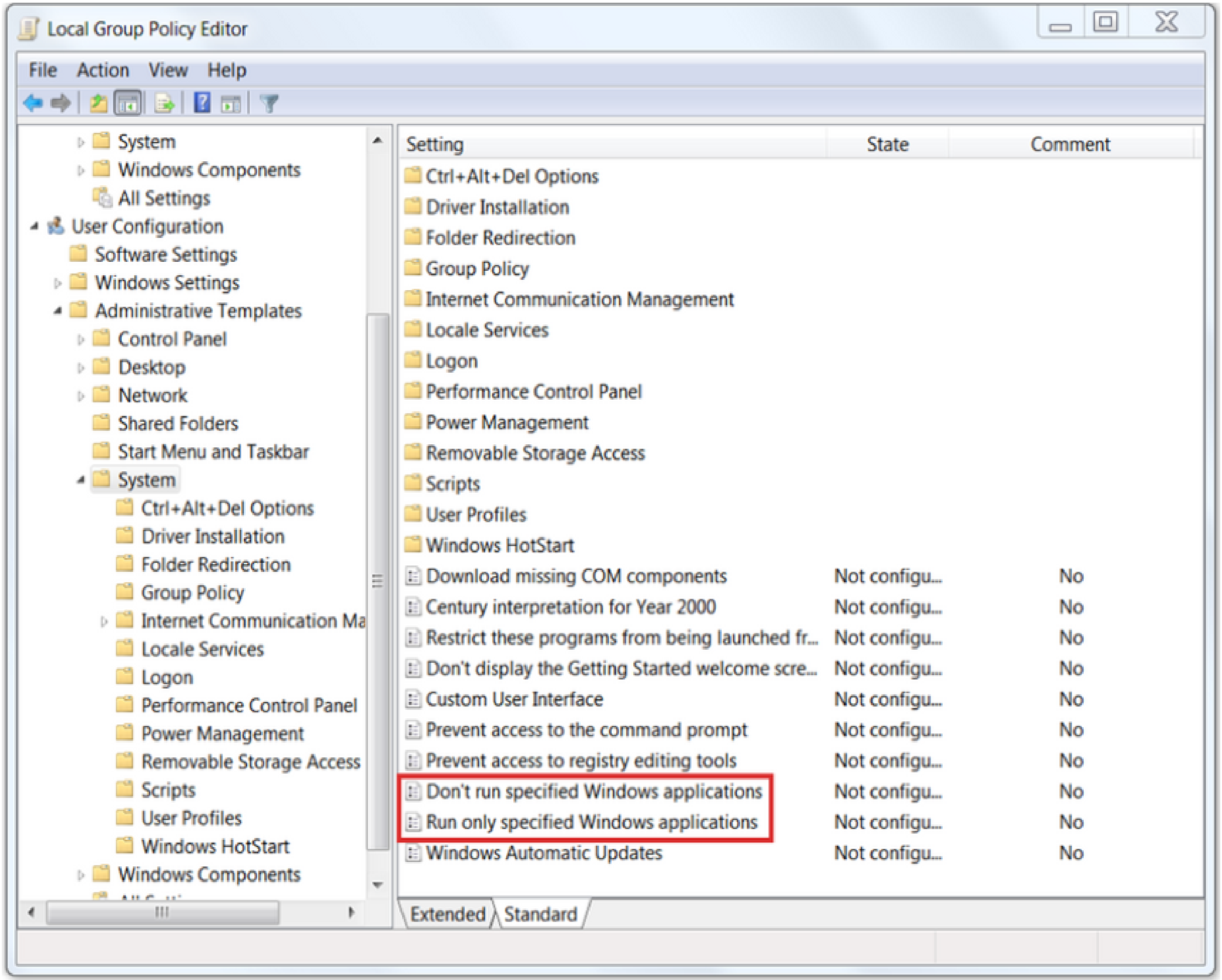
Application Blacklisting and Whitelisting

- Limiting access to potential threats by creating lists of prohibited applications is known as blacklisting.
- Application blacklists can dictate which user applications are not permitted to run on a computer.
- Whitelists specify which programs are allowed to run.
- In this way, known vulnerable applications can be prevented from creating vulnerabilities on network hosts.



Application Blacklisting and Whitelisting (Contd.)

- Websites can also be whitelisted and blacklisted.
- These blacklists can be manually created, or they can be obtained from various security services.
- Blacklists can be continuously updated by security services and distributed to firewalls and other security systems that use them.
- Cisco's Firepower security management system is an example of a system that can access the Cisco Talos security intelligence service to obtain blacklists.



System-Based Sandboxing

- Sandboxing is a technique that allows suspicious files to be executed and analyzed in a safe environment.
- **Cuckoo Sandbox** is a popular free malware analysis system sandbox. It can be run locally and have malware samples submitted to it for analysis.
- **ANY.RUN** is an online tool that offers the ability to upload a malware sample for analysis like any online sandbox.

22.4 Endpoint Protection Summary

© 2020 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

What Did I Learn in this Module?

- Endpoints are defined as hosts on the network that can access or be accessed by other hosts on the network.
- There are two internal LAN elements to secure: Endpoints and Network Infrastructure.
- Antivirus/Antimalware Software is installed on a host to detect and mitigate viruses and malware.
- Host-based firewalls may use a set of predefined policies, or profiles, to control packets entering and leaving a computer.
- Some examples of host-based firewalls include Windows Defender Firewall, iptables, nftables, and TCP Wrappers.
- HIDS protects hosts against known and unknown malware.
- An attack surface is the total sum of the vulnerabilities in a given system that is accessible to an attacker.
- Application blacklists dictate which user applications are not permitted to run on a computer and whitelists specify which programs are allowed to run.

UNIVERSITY OF ŽILINA
Faculty of Management Science
and Informatics

Ďakujem za pozornosť

Obsahom boli moduly:

Chapter 21 Cryptography
Chapter 22 Endpoint Protection

Vyjadrite spätnú väzbu na prednášku a/alebo cvičenie v anonymnej ankete cez google form: [link]

Vytvorené v rámci projektu KEGA 026TUKE-4/2021.