

COMS3000 Exam Notes

Advanced Persistent Threat (APT)

- Advanced: the attacker is an expert in cyberintrusion methods and is capable of crafting custom exploits and tools
- Persistent: the attacker has a long-term objective and will persistently work to achieve it without detection and without regard for time
- Threat: the attacker is organised, funded, well trained, and highly motivated

Lifecycle of an APT attack:

1. Initial intrusion through system exploitation
2. Malware is installed on compromised system
3. Outbound connection is initiated
4. Attacker spreads laterally
5. Compromised data is extracted
6. Cover tracks

Firewalls:

- Primary purpose
 - Control the traffic from one network to another
 - Use packet header
 - Access control lists – source IP, destination IP/port – block by default
 - Allow or deny connection
- Secondary purpose:
 - VPN
 - Network address translation
 - DMZ

Intrusion Detection/Prevention System:

- Primary purpose:
 - Control the traffic from one network to another
 - Use packet payload
 - Allow or deny each connection
 - Example
 - Firewall lets web traffic to ports 80, 443
 - IPS examines payload looking for web application attack e.g. SQL injection, cross site scripting

Endpoint fraud detection technologies – prevent fraud on the end user's laptop, desktop or mobile

- Preventing malware being installed or removing malware already installed
- Providing patch protection, module loading protection, anti-screenshot, anti-keylogging

Detecting and Profiling Malware

- Execute files in a safe environment (virtual machine) – web interactions, email attachments, scan file systems, create MD5 signatures
- Understand the characteristics – IP address/URLs, cloud based interaction


Classes of technology for detecting APT:

- Event Correlation – logs, flows, IP location, geo location
- Anomaly Detection (activity baselining & anomaly detection) – user activity, application activity, network activity
- Offense Identification – credibility, severity, relevance

Cloud Service Categories:

- Infrastructure-as-a-Service (IaaS) – cloud service category in which the cloud capabilities provided to the cloud service customer is an infrastructure capabilities type
- Platform-as-a-Service – cloud service category in which the cloud capabilities provided to the cloud service customer is a platform capabilities type

- Software-as-a-Service – cloud service category in which the cloud capabilities provided to the cloud service customer is an application capabilities type



Cloud Service Categories & Cloud Capabilities

Cloud Service Categories	Cloud Capabilities Types		
	Infrastructure	Platform	Application
Software as a Service			X
Platform as a Service		X	
Infrastructure as a Service	X		
Network as a Service	X	X	X
Data Storage as a Service	X	X	X
Compute as a Service	X		
Communication as a Service		X	X

Security issues with Cloud Services:

1. The cloud customer assumes the cloud service is secure without understanding the contract
2. Insecure management or administration interfaces
3. No separation of duties, detection of abuse, or escalation of privilege
4. Weak, vague, or one-sided SLAs and contracts

CIA Triad:

- **Confidentiality** – prevention of unauthorised disclosure of information
- **Integrity** – prevention of unauthorised modification of information
- **Availability** – prevention of unauthorised withholding of information or resources

Authenticity – making sure the author/sender of a message is as it is claimed

Non-repudiation – ensures that the purported maker of a statement, or signer of a contract, will not be able to successfully challenge the validity of the statement or contract. E.g. the authenticity of a signature can be challenged or “repudiated”

Risk – the likelihood that a particular threat using a specific attack, will exploit a particular vulnerability of a system that results in an undesirable consequence

Threat – any circumstance or event with the potential to cause harm to an information system in the form of destruction, disclosure, adverse modification of data, and/or the denial of service

Vulnerability – Weakness in an information system, cryptographic system, or other components (e.g. system security procedures, hardware design, internal controls) that could be exploited by a threat

Access Control:

1. Identification – determine identity (person or machine)
2. Authentication – verify identity (proof)
3. Authorisation – once the identity is known, a decision can be made about granting or denying access (role based access control)

Methods of authentication:

- Something you know – password, PIN
- Something you have – physical key, token, smart card

- Something you are (or do) – fingerprints, voice, hand geometry

Criteria for a good biometric:

- **Universality** – each person should have the characteristic
- **Distinctiveness** – any two people should be sufficiently different in terms of this characteristic
- **Permanence** – the characteristic should be sufficiently invariant over a period of time
- **Acceptability** – extent to which people are willing to accept biometric in their daily lives
- **Circumvention** – how easily can the system be tricked

Cryptographic One-Way Hash Functions:

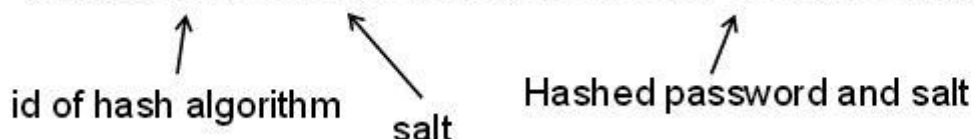
- Used by OS to store the hash of a password rather than the password itself. The OS then checks to see if the user's typed password hashes to the same value
- **Easy to compute**
- **Compression** – $h()$ maps an input x of arbitrary finite bit length to an output $h(x)$ of fixed, typically short, bit length
- **Pre-image Resistance (one-way property)** – for essentially all outputs $h(x)$, it is computationally infeasible to find x
- **Strong Collision Resistance** – it is computationally infeasible to find any two distinct inputs x_1 and x_2 such that $h(x_1) = h(x_2)$. This is analogous to finding the same result show up in any of the previous rolls of a 2^n sided dice. $2^{n/2}$ complexity.
- **Weak Collision Resistance (2nd Pre-image Resistance)** – for a given $h(x_1)$, it is computationally infeasible to find another input x_2 such that $h(x_1) = h(x_2)$. This is analogous to rolling a 2^n sided dice and expecting a particular result. Probability of a collision is 0.5^n or a collision will be found on average after 2^n tries.
- Also known as 'digest functions' or 'digest fingerprints'

Salt – a random bit string that is concatenated to users' passwords

- Ensures different password hashes, even if users share the same password
- Defends against attacks with pre-computed hash tables (rainbow tables)
- Makes dictionary attack harder. Instead of having to hash each password in a dictionary, an attacker needs to try all possible salt values with each dictionary word. Using a 12 bit (two 6-bit character) salt, this increases the work load by a factor of 4096.

Salt in Linux Shadow Password Files

```
arlin:$1$/80DehIe$.11g0DCK3CmY/UtX.mr6c/:14687:::::::
```

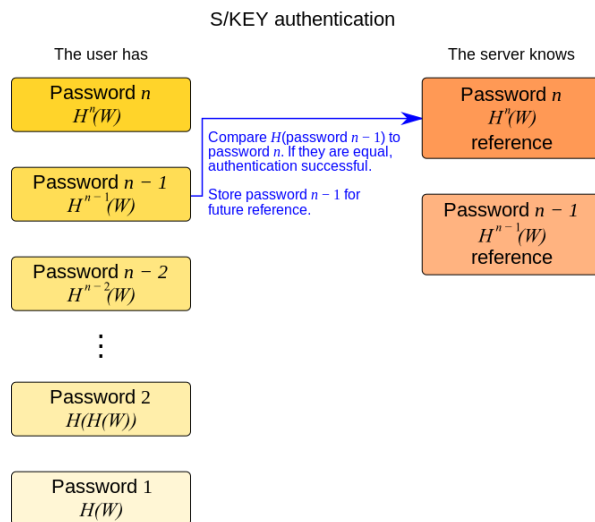


Password-based Authentication Protocols

Over a communication channel, Alice can send a hash $h(p_A)$ of her password to Bob as a means of logging into a system or authentication to a web server. However, if Eve is eavesdropping on this channel, she can obtain that hash value and simply resend that hashed password (pretending to be Alice). This is known as a replay attack. This is similar to how HTTP Basic Authentication works.

To overcome this problem, a challenge-response protocol is used. The server sends a challenge c (also known as nonce – number used once) which the client uses to calculate a response $r = h(c || p)$ or $r = h(c \oplus p)$. $||$ is concatenation. This prevents eavesdroppers from performing replay attacks. It is important that the challenge c is not repeated and that the sequence of challenge values appears random and unpredictable. This is similar to how HTTP Digest Authentication works.

Lamport's Hashed Password Scheme



Discretionary Access Control (DAC):

- Owner decides access rights
- Unix criteria: Owner, Group, Other (RWX for each)
- Set User Id (SUID) used to execute code at owner privileges (S rather than X). Used by passwd.

Mandatory Access Control (MAC):

- System-wide policy defines access to objects
- Each file (object) has a classification indicating the sensitivity level
- Each user (subject) has a clearance indicating the sensitivity level the user has access to
- Users only have access to their level of classification within their assigned compartments
- **Bell-LaPadula Model:**
 - **No read up** – a subject can only read an object if its clearance dominates the object's classification
 - **No write down** – a subject at a given security level must not write to any object at a lower security level
- **High water mark** – the security level of subjects can only be increased

Information Theory:

$$I(x) = \log_2 N = -\log_2 p$$

$$\text{Average Information: } H(x) = -\sum_{i=1}^N p_i \log_2 p_i$$

$$\text{Average Message Length} = \text{relative frequency} \times \text{encoding length}$$

$$\text{Redundancy} = \text{average message length} - \text{entropy}$$

Cryptography:

- **Frequency Analysis** – used to decipher mono-alphabetic substitution ciphers
- **Kasiski Test** – in the Vigenère cipher, the key length is likely to be the GCD (Greatest Common Divisor) of the distances between repeating n-grams
- **One Time Pad:**
 - **Encryption** – $C_i = M_i \oplus K_i$
 - **Decryption** – $M_i = C_i \oplus K_i$

Cipher Modes (Block Ciphers)

Prevents:

- Reordering Attack
 - Insertion Attack
 - Replay Attack
 - Dictionary attacks
- **Electronic Book Code (ECB) mode:** parallel both ways
 - **Cipher Block Chaining Mode (CBC):** parallel decryption

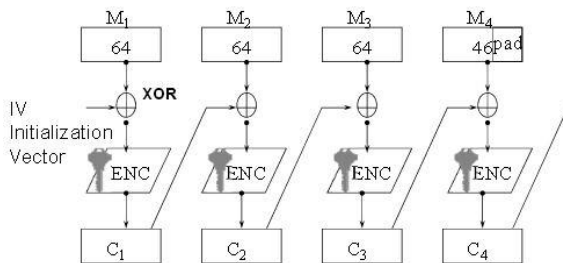


Figure 1: Encryption

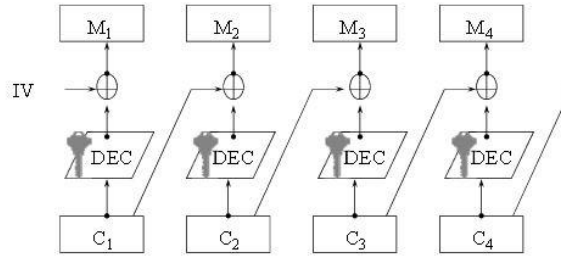


Figure 2: Decryption

- **Cipher Feedback Mode (CFB):** parallel decryption

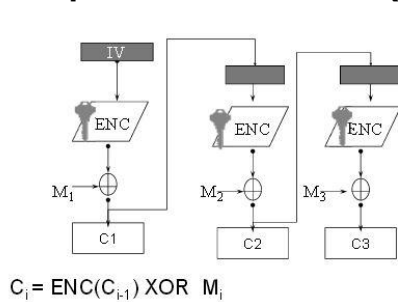


Figure 3: Encryption

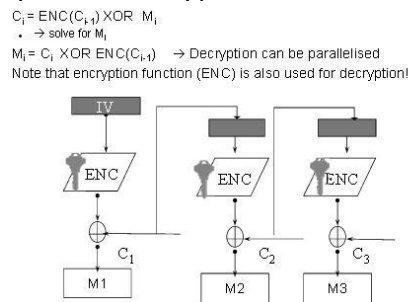


Figure 4: Decryption

- **Output Feedback Mode (OFB):** not parallelizable

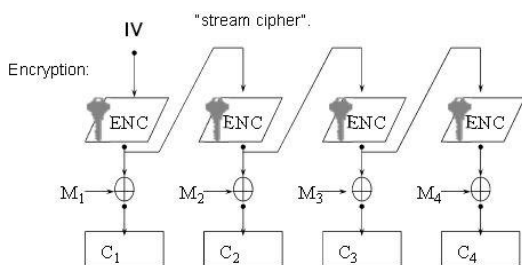


Figure 5: Encryption

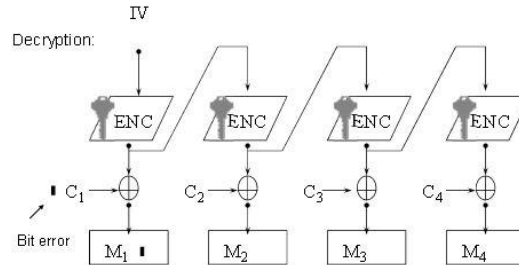
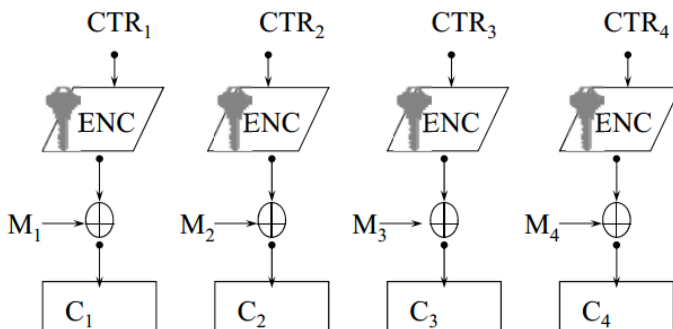


Figure 6: Decryption

- **Counter Mode:** parallel both ways. Decryption same as bellow but feed in cipher to XOR. Similar to a stream cipher.



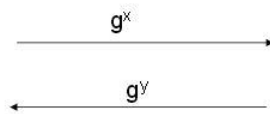
Diffie-Hellman Key Agreement Protocol



- Alice and Bob initially agree on a large prime number p and a generator g
 - \rightarrow discrete logarithms exist
- All computations are done modulo p



- Alice chooses random number x
- Alice computes $X=g^x$ and sends it to Bob



- Bob chooses random number y
- Bob computes $Y=g^y$ and sends it to Alice

- Alice computes $K_{AB} = Y^x = (g^y)^x \bmod p = g^{xy} \bmod p$



- Bob computes $K_{AB} = X^y = (g^x)^y \bmod p = g^{xy} \bmod p$

- Eve is listening on the insecure channel and sees g^x and g^y , but since she cannot calculate discrete logarithms, she does not know x or y
- \rightarrow She cannot compute the secret key K_{AB}

RSA

- Can be used for confidentiality – encryption
- Can be used for authenticity/integrity – digital signatures

RSA (Toy) Example

- Bob chooses $p=3$ and $q=11$
 - (far too small to be secure)
- $n = p * q = 11 * 3 = 33$
- $z = (p-1) * (q-1) = (3-1)*(11-1) = 20$
- $e=?$
 - (e cannot have a common factor with $z=20$)
 - $e = 3$, or 7 , 9 , 11 , ...
- $d=?$ ($e*d \bmod z = 1$)
 - For small numbers, we can do this via trial and error.
 - For large numbers \rightarrow Extended Euclid's Algorithm
 - $d=1: 3*1 \bmod 20 = 3$
 - $d=2: 3*2 \bmod 20 = 2$
 - \vdots
 - $d=7: 3*7 \bmod 20 = 1$

RSA

- 1) choose 2 primes p and q , $n=p*q$
- 2) $z = (p-1)(q-1)$
- 3) Choose e so that e and z are relatively prime
- 4) Compute d so that $e*d \bmod z = 1$

Public Key: (n, e)
Private Key: (n, d)

Encryption: $c=m^e \bmod n$
Decryption: $m=c^d \bmod n$

- Bob's Public Key: $(n, e) = (33, 3)$
- Bob's Private Key: $(n, d) = (33, 7)$

- How can Alice use this RSA system to send Bob a secret message?
 - "Bob, I love you"
- Letters are encoded: 'A'=1, 'B'=2, ... 'Z'=26
- Encrypted individually
- $E('B') = E(2)$?
 - $E('B') = 2^3 \bmod 33 = 8$
- How does Bob decrypt? $D(8)=?$
 - $D(8) = 8^7 \bmod 33 = 2097152 \bmod 33 = 2$

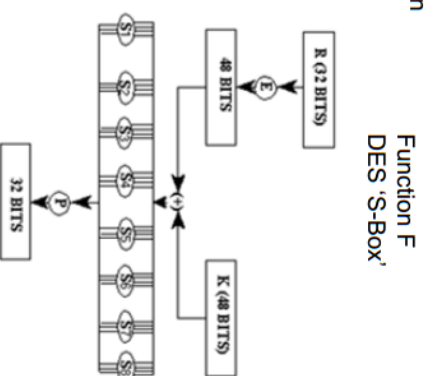
RSA

Public Key: $(n, e) = (33, 3)$
Private Key: $(n, d) = (33, 7)$

Encryption: $c=m^e \bmod n$
Decryption: $m=c^d \bmod n$

Data Encryption Standard (DES)

- DES was widely used in the past, but now mostly replaced with AES (Advanced Encryption Standard)
- Based on IBM's (Horst Feistel's) Lucifer cipher
 - NSA made some changes → DES
 - Key length, "S-boxes"
- NIST Standard since 1977
 - National Institute of Standards and Technology
- DES is a Feistel Cipher
 - 16 rounds
 - 64-bit blocks
 - 56-bit key (64-bit key less parity)
- Round function $F(R,K)$:
 - Expand 32 to 48 bits ('expansion permutation')
 - S-Box: 6-bit input → 4 bit output
 - Implemented via lookup tables
 - Permutation (P-box)
- No major weakness has been found
 - Brute-force attack is best option
 - Problem:
 - 56-bit key is too short

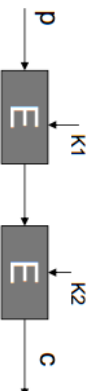


http://en.wikipedia.org/wiki/Data_Encryption_Standard

15

Extending the lifetime of DES

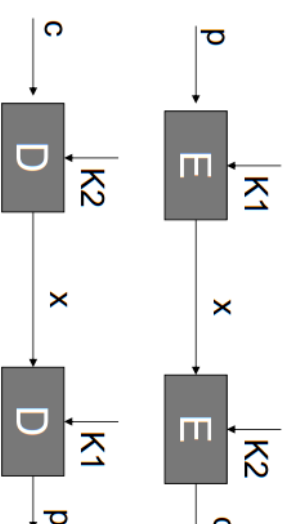
- Q: Given that DES is secure and the only problem is its short key length, how can its lifetime be extended?
- A: Use DES multiple times with different keys
- For example: 2-DES ('Double DES')
 - $c = E_{K2}(E_{K1}(p))$
 - $p = D_{K1}(D_{K2}(c))$
- Is the effective key length = 2×56 bits = 112 bit?
 - 2-DES is vulnerable to the so-called "meet-in-the middle" attack, which makes it not much more secure than single DES.



17

Meet-in-the-middle Attack

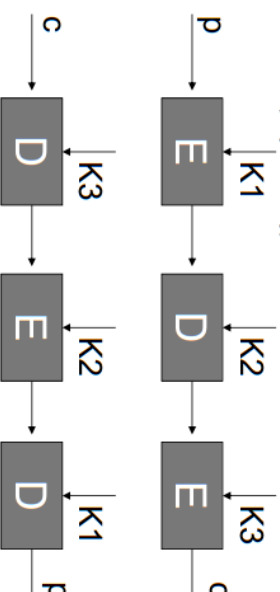
- $x = E_{K1}(p) = D_{K2}(c)$
- For a given plaintext-ciphertext pair (p,c) → known plaintext attack
 - Calculate $E_{K1}(p)$ for all 2^{56} values of $K1$ (and store them in a table)
 - Calculate $D_{K2}(c)$ for all 2^{56} values of $K2$
 - If $E_{K1}(p) = D_{K2}(c)$ we have a match and have $K1$ and $K2$ (with high probability)
 - Computational cost (worst case)
 - Only 2^{57} instead of 2^{112}
 - Trades-off computation with storage cost
 - 'Time-space trade-off'



18

3-DES (Triple DES)

- Keying options:
 - 1) 3 different keys (vulnerable to meet-in-the middle (or two thirds) attack)
 - effective key length 112 bits
 - 2) $K1 = K3$
 - vulnerable to some type of attacks, considered to have an effective key length of 80 bits
 - 3) $K1 = K2 = K3$ (only 56 bit key)
- Why EDE and not EEE?
 - If $K1=K2=K3$, this is equivalent to Single-DES
 - Good for backwards compatibility



19

Advanced Encryption Standard (AES)

- Symmetric block cipher
- Not a Feistel Cipher.
- Data blocks of 128 bits
- Cipher keys with lengths of 128, 192, and 256 bits
- Only known attacks are side channel attacks. Eg. timing information, power consumption

AES – How it works in a Nutshell

➤ Substitution-Permutation Network (Transposition) Network

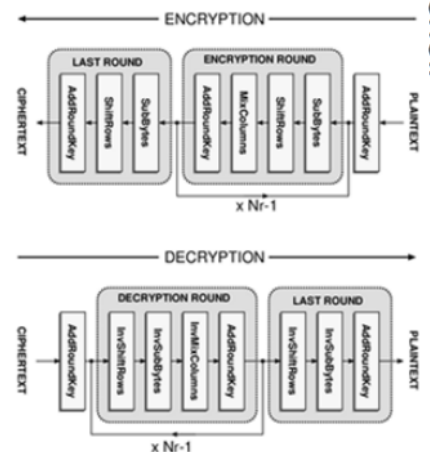
➤ Number of Rounds

- 128-bit key: 10 rounds
- fastest
- 192-bit key: 12 rounds
- 256-bit key: 14 rounds
- slowest

➤ Operations are on a matrix of 4x4 bytes (=128 bits), called the 'state'

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

24



AES – High Level Steps

➤ Initial Round

- AddRoundKey
 - Every byte of 'state' combined with round key, using bit-wise XOR
 - Round key is derived from Key Schedule' (Not covered here)

➤ Normal Rounds

- SubBytes—a non-linear substitution step
 - Each byte of the state is replaced with another, based on a lookup table
- ShiftRows—a transposition step
 - Each row of the state is shifted cyclically a certain number of steps
- MixColumns—a linear transformation on columns
- AddRoundKey—combine with the round key
 - Same as in initial round

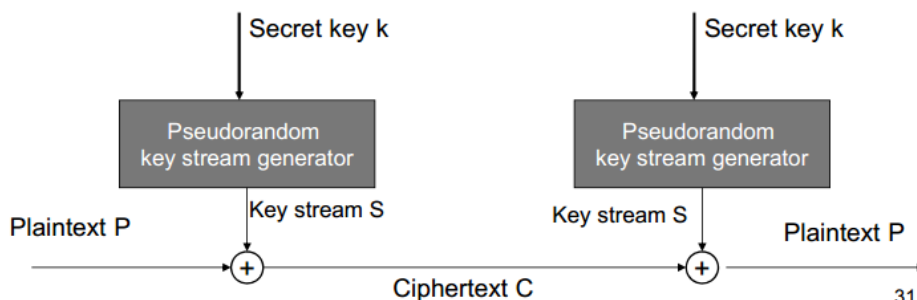
➤ Final Round

- Same as normal round, but no MixColumns step

25

Stream Ciphers

- Work like one-time pad
- Generate pseudorandom key stream S
- S is a function of a secret key k where k is seed to a pseudo random number generator
- Expand k into an arbitrarily long key stream
- Encryption: $C = P \text{ XOR } S$ (bit-wise XOR)
- Decryption: $P = C \text{ XOR } S$
- Not perfect security since key stream S is not truly random and will eventually be repeated.
- Attacker can potentially detect pattern and predict key.
- Security relies on the unpredictability of S
- RC4 is a commonly used stream cipher. "Ron's Code", Ron Rivest.



31

Public Key Certificates:

- Links together identity and public key
- Signed by a trusted third party, called a Certification Authority (CA)
- CAs can have hierarchical authority
- At the top of a certificate chain is the root certificate. This is self-signed and serves as a 'trust anchor'
- Trust is provided via pre-installation in web browser or certificate store
- If a private key is compromised or leaked, the public key should be put on Certificate Revocation List (CRL), published by the CA
- Uses RSA to produce a trusted third party's digital signature

Structure of a X.509 v3 Certificate

- Certificate (This section is hashed then encrypted with the subject's private key)
 - Version
 - Serial Number
 - Algorithm ID
 - Issuer
 - Validity
 - Not Before
 - Not After
 - Subject
 - Subject Public Key Info
 - Public Key Algorithm
 - Subject Public Key
 - Issuer Unique Identifier (optional)
 - Subject Unique Identifier (optional)
 - Extensions (optional)
- Certificate Signature Algorithm
- Certificate Signature

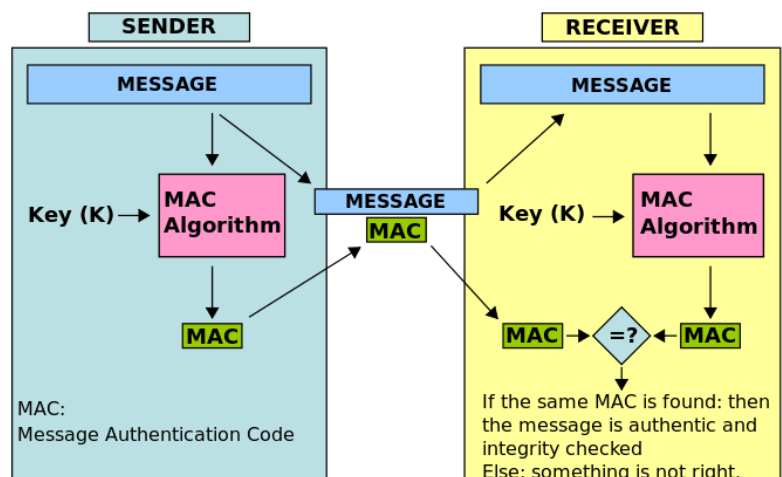
To validate this certificate a second certificate that matches the Issuer (Thawte Server CA) of the first certificate is needed. First verify that the second certificate is of a CA kind; that it can be used to issue other certificates by inspecting a value of the CA attribute in the extension section. Then the RSA public key from the CA certificate is used to decode the signature on the first certificate to obtain a MD5 hash, which must match an actual MD5 hash computed over the rest of the certificate.

Public Key Infrastructure (PKI):

- A public-key infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.
- Consists of:
 - Certificates – binds identity to public key, typically X.509
 - Certification Authorities (CAs) – issue certificates, with digital signature
 - Certificate Revocation Lists (CRLs) – list of certificates (serial numbers) that should no longer be trusted

Message Authentication Codes (MAC):

- Used to provide authentication and integrity for packets in a secure network protocol
- Both sender and receiver have a shared secret key K
- $MAC = h(K || m)$
- This provides integrity since an attacker needs to know K to compute a valid MAC
- This provides authenticity since only someone that knows K can compute a valid MAC
- This is susceptible to the 'length extension attack'

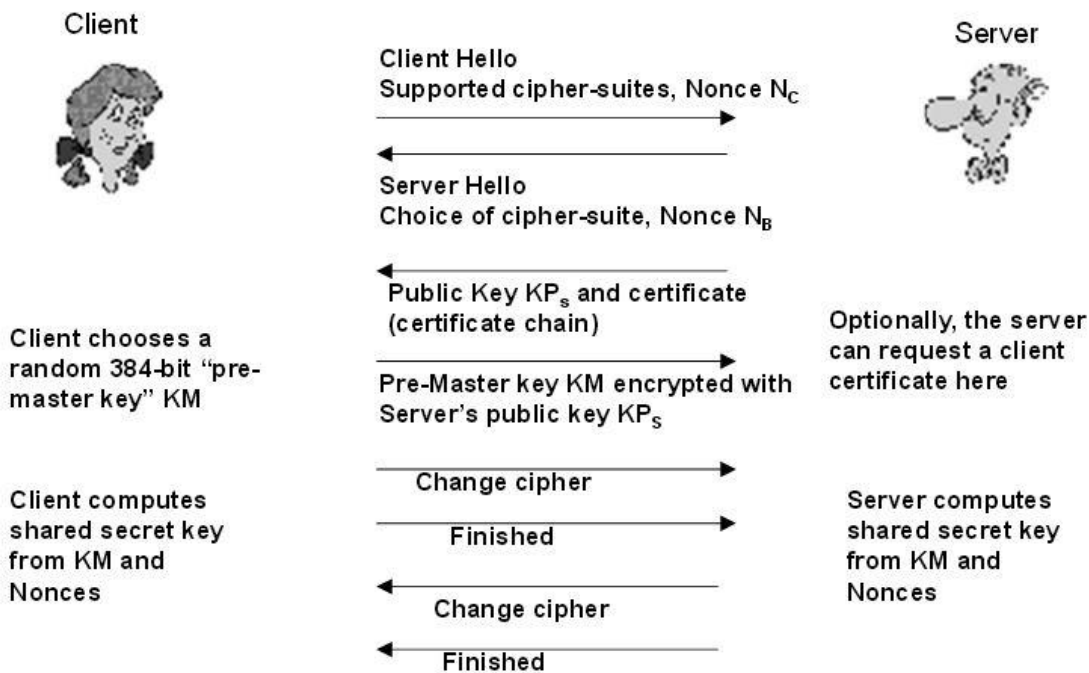


Transport Layer Security (TLS):

- **Handshake Protocol** – establishes shared secret key, negotiates cipher suite
- **Cipher Change Protocol** – enables cipher change
- **Alert Protocol** – reports errors
- **Record Protocol** – main part, provides secure transport
- Provides key establishment, authentication, confidentiality and integrity
- A "cipher suite" is a combination of specific algorithms to be used in a TLS session
 - TLS_RSA_WITH_AES_128_CBC_SHA (this is mandatory)
 - TLS_DH_anon_WITH_RC4_128_MD5
 - TLS_RSA_WITH_DES_CBC_SHA

TLS Handshake:

- Negotiation of cipher suite to be used
- Nonces are used to prevent replay attacks from occurring by attackers. The 3rd and 4th message use nonces.



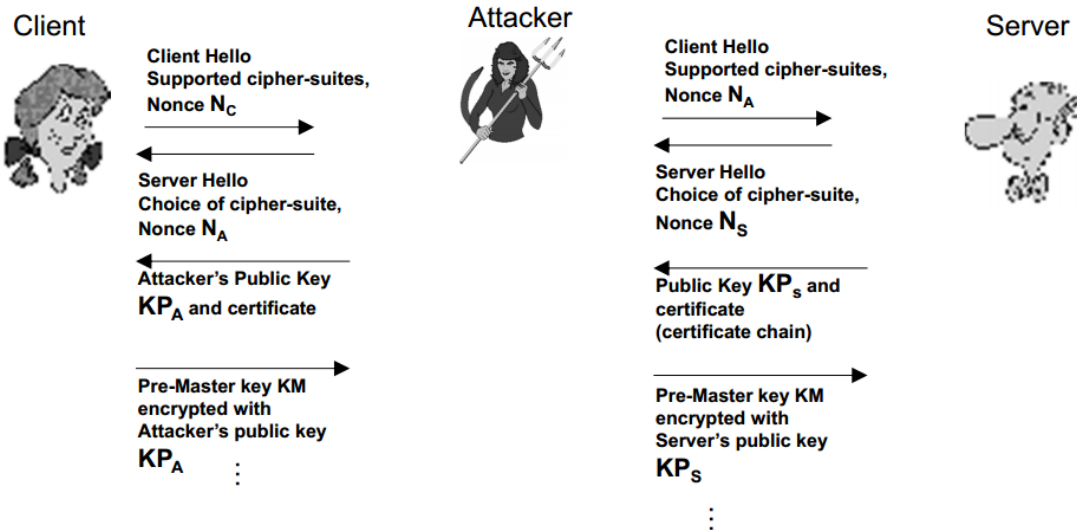
SSL MITM Attack:

Assume an attacker can redirect a web browser's request to a proxy that he/she controls

- DNS poisoning
- ARP spoofing

Ways to launch a Man-in-the-middle attack against SSL/TLS

- If DH_anon (plain Diffie-Hellman) is used for key establishment
- What if RSA Public Key certificates are used in the handshake:



➤ When is this attack successful?

- Attacker needs a way to redirect client's request
 - DNS poisoning, ARP spoofing, ...
- Attack is successful if client accepts attacker's Server certificate
 - Issued by trusted CA?
 - Does name on certificate match the name of the company we want to communicate with?

Glossary:

Acronym	Explanation
AES	Advanced Encryption Standard
ALE	Annualized Loss Expectancy)
AP	Access Point
ARO	Annualized Rate of Occurrence
ASV	Approved Scanning Vendor
CBC	Cipher Block Chaining Mode
CCM	Counter with CBC-MAC
CDE	Cardholder Data Environment
CHAP, MS-CHAP	Challenge Response Authentication Protocol
CHD	Cardholder Data
CP	Certificate Policy
CPS	Certification Practice Statement
CVV, CVV2, CVC2	Card Verification Value
DAC	Discretionary Access Control
DES	Data Encryption Standard
DSS	Data Security Standard
EAL	Evaluation Assurance Level
EAP	Extensible Authentication Protocol
FAR	False Acceptance Rate
FMR	False Match Rate
FNMR	False Non-Match Rate
FRR	False Rejection Rate
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
IV	Initialisation Vector
MAC	Message Authentication Code
OFB	Mandatory Access Control
OTP	Output Feedback Mode
PA	One-Time Password
PAN	Payment Application (off the shelf)
PAP	Primary Account Number = Card number
PCI	Password Authentication Protocol
PED	Payment Card Industry
PGP	PIN Entry Device
PKI	Pretty Good Privacy
PMK	public key infrastructure
PSK	pairwise master key
PTK	Pre-Shared Key
QSA	pairwise transient key
RBAC	Qualified Security Assessor
RFC1760, RFC2289	Role-based access control
RSN	One-Time Password system based on hashing
RSNA	Robust Security Network
SAML	Robust Security Network associations
SASL	Security Assertion Markup Language
SLE	Simple Authentication and Security Layer
SSL	Single Loss Expectancy
TKIP	Secure Sockets Layer
TLS	Temporal Key Integrity Protocol
	Transport Layer Security