No.

Date

Diffie-Hellman key exchange

$A = g^a \bmod p$

$B = g^b \bmod p$

$S = A^b \bmod p = B^a \bmod p$

$= g^{ab} \bmod p$

Vulnerable to MITM Attack

El Gamal Encryption

$B = g^b \bmod p$

$C = (R, S) = (g^r \bmod p, m \cdot B^r \bmod p)$

$M = R^{-b} \cdot S \bmod p$

SHA: Secure Hash Algorithm

vulnerable to a length-extension attack

i.e. If Mallory knows len(s), H(s),

He can calculate H(s||M) for arbitrary M

Because all internal state is derivable from H(s) and len(s) when computing H(s||M)

HMAC

o_key_pad = 0x5c5c... ⊕ key

i_key_pad = 0x3636... ⊕ key

hmac = hash(o_key_pad ||

hash(i_key_pad || message))

The second hash can prevent appending data if the underlying hash is vulnerable to length-extension attack

Passwords Protection

a) store with hash and salt. {Alice, Salt, H(P||Salt)}

b) Slower Hashes

RSA Encryption: Secure because big integer factoring is believed to be hard

a) Generate large primes, p, q — prime

Use a probabilistic algorithm to check if it's

b) $n = p * q$

$\phi(n) = (p-1)(q-1)$

c) Chose random $2 < e < \phi(n)$

e should be relative prime to $\phi(n)$

but it can be small

d) $d = e^{-1} \bmod \phi(n)$

n, e are public / d, p, q and $\phi(n)$ are secret

public key / private key

Encryption: $C = m^e \bmod n$

Decryption: $M = c^d \bmod n$

$d = e^{-1} \bmod \phi(n)$

$ed = 1 \bmod \phi(n)$

$ed - 1 = k \cdot \phi(n)$ for some k

$c^d \bmod n = m^{ed} \bmod n$

Euler's Theorem: $a^{\phi(n)} \bmod n = 1$

$= (m^{ed-1}) \cdot m \bmod n$

$= [m^{k\phi(n)}] \cdot m \bmod n$

$= [m^{\phi(n)}]^k \cdot m \bmod n$

$= 1^k \cdot m \bmod n$

$= m$

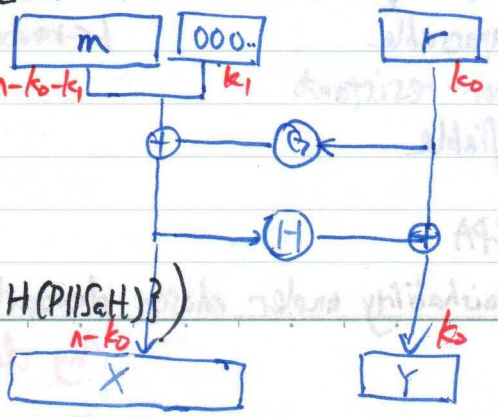
Sign: $s = (H(m))^d \bmod n$

Verify: $v = s^e \bmod n = H(m)$

Note: Use RSA-OAEP to ensure IND-CPA

it can make input random but reversible

Feistel network



Anonymity only works in a crowd

Security is economics

Least privilege

Use fail-safe defaults

Separation of responsibility

Defense in depth

Psychological acceptability

Human factors matter

Ensure complete mediation

Know your threat model

Detect if you can't prevent

Don't rely on security through obscurity

Design security in from the start

Conservative design

Kerckhoffs' principle

A cryptosystem should be secure even if everything about the system, except session key is the key, is public knowledge.

Proactively study attacks

Shannon's Maxim

The enemy knows the system

TOCTTOU Vulnerability

TCB (The Trusted Computing Base)

Design Principles

a) Unbypassable

b) Tamper-resistant

c) Verifiable

IND-CPA

Indistinguishability under chosen plaintext attack

Defenses against buffer overflow

a) memory safe languages

b) canaries

c) Non-Executable Stack Space

d) W^X

e) Address Space Layout Randomization

ROP: Return-Oriented Programming

PRG: Pseudorandom Generator.

DLP: Discrete Log Problem

(an integer k that solves $b^k = a$)

Forward Secrecy

a) The past communication cannot be decrypted if one key is compromised.

b) The past communication cannot be decrypted if private key is compromised. But future communication could be compromised.

AES: Electronic Code Book (ECB)

$C_i = E_k(M_i)$

The most effective attack is just bruteforce.

Cipher Block Chaining (CBC)

$C_0 = IV$

$C_i = E_k(C_{i-1} \oplus M_i)$

NOT parallelizable for Enc

parallelizable for Dec

Counter Mode (CTR)

$Z_i = E_k(IV+i)$

$C_i = Z_i \oplus M_i$

parallelizable for Enc and Dec

Output Feedback Mode (OFB)

$Z_0 = IV$

$Z_i = E_k(Z_{i-1})$

$C_i = Z_i \oplus M_i$

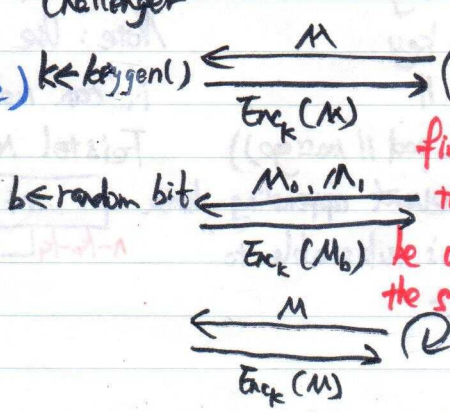
Stream Cipher

Enc(K, M):

IV ← random value

C ← PRG(K|IV) ⊕ M

return (IV, C)



Rollback-resistant

Even if the attacker finds out the state at time T, they should not be able to determine what the state was at time T-1.

An encryption scheme is IND-CPA iff $Pr[b=b'] \leq \frac{1}{2} + negl$.

Any deterministic scheme is not IND-CPA

No.

Date

Cookie Scope

SSL: Secure Sockets Layer
TLS: Transport Layer Security
PS: Premaster Secret (a random number)

Browser (RSA) Server

prevent replay attack

R_B →
← R_S
← Cert
← $\{PS\}_{PK_{server}}$
→

Derive C_B, C_S, I_B, I_S from PS, R_B, R_S

MAC_{I_B}(dialog) →
← MAC_{I_S}(dialog)

} detect MITM

(Ephemeral key Exchange via Diffie-Hellman)

$\{g, P, g^a \bmod p\}_{PK_{server}}$ →
or
$g^b \bmod p$ →
$(PS = g^{ab} \bmod p)$

RSA key exchange lacks forward secrecy

a) What scope a server is allowed to set on a cookie

domain: Any domain-suffix of URL-hostname except TLD (Top-level Domains)

path: can be set to anything

b) when browser sends cookie Browser sends all cookies in URL scope

i) cookie-domain is domain-suffix of URL-domain
ii) cookie-path is prefix of URL-path
iii) protocal=HTTPS if secure flag is set

Flags a) Secure: sent over HTTPS only
b) HttpOnly: cookie cannot be accessed by Javascript, but only sent by browser

Expires: the expiration date
Set it to date in past can delete cookie

Same-origin Policy
origin = (protocal, hostname, port)

Cross-origin communication
A narrow API: postMessage

XSS: Cross-site scripting { Stored XSS / Reflected XSS
CSRF: Cross-site Request Forgery

SQL Injection Prevention
a) Sanitizate user input
b) Use Parameterized/Prepared SQL

How to store session tokens
a) cookies: vulnerable to CSRF
b) Embed in URL links: token leaks via HTTP Referer header / users may share URLs
c) In a hidden form field: short session only

CSRF Defenses
a) CSRF token embeded into forms
b) Referer Validation

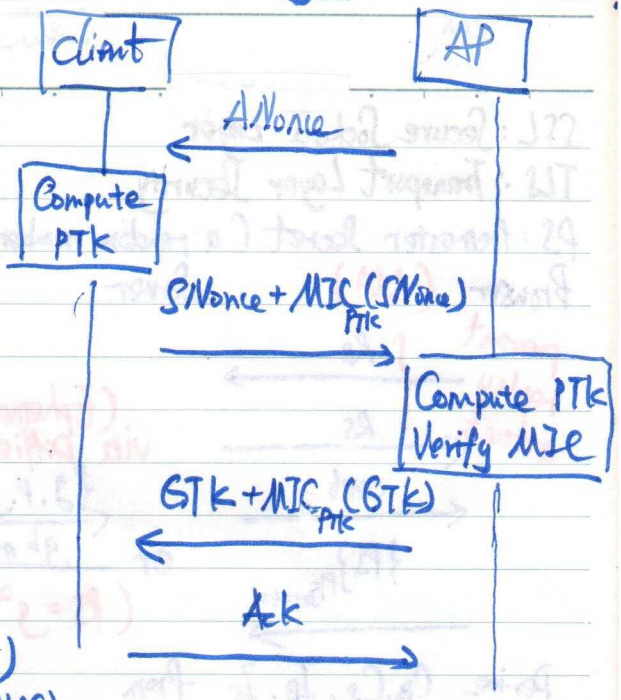
XSS Prevention
a) Input validation: use whitelisting, avoid blacklisting
b) Output escaping: escape dynamic data before inserting it into HTML
c) Content-security policy (CSP)

MAC: Media Access control (Address)
AP: Access Point
(Wi-Fi Protected Access 2)
WPA2

PBKDF2(PRF, P, S, c, len)
PBKDF2: Password Based Key Derivation Function 2
PRF: Pseudo Random Function (e.g. SHA-256)
P: Password/Passphrase
S: Salt
c: Iteration count (make it slower)
len: Number of bits/bytes requested

PSK: Pre-Shared Key
PTK: Pairwise Transient key
PSK = PBKDF2(SHA1, passphrase, ssid, 4096, 256)
PTK = F(PSK, ANonce, SNonce, AP MAC, Client MAC)
GTK: Group Transient Key

4-way handshake

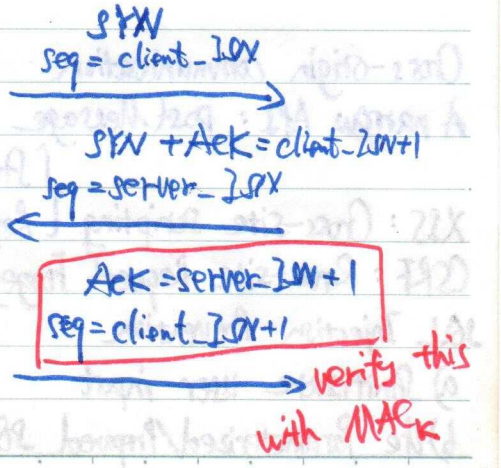


Vulnerable to brute-force attack: Attacker can brute-force password and verify it by MIC_{PTK}(SNonce)
Reason: people don't chose good password.
WPA-Enterprise: The 4-way handshake is secured with a unique PSK

DHCP: Dynamic Host Control Protocol
ARP: Address Resolution Protocal
(provide address of gateway/router)

Kaminski Attack
a) try to poison a.google.com
b) state that "www.google.com" is an authority
Reason to work: NX Domains are not likely to be cached, so that you can always try again.
Defense: Randomize the UDP source port (Adds 16 bits of entropy)

SYN-cookies:
server_ISN = $HMAC_k$(SIP|DIP|SPORT|PPORT|client_ISN)



UDP Blind Spoofing: Identification number is random, so that $\frac{1}{2^{16}}$ chance of succeed.

Visual Integrity
Target & Pointer are visible } Context Integrity
Temporal Integrity
↳ Target clicked = Target checked
↳ Pointer clicked = Pointer checked.
Two-Factor authentication
Authentication using two of:
a) Something you know (account details / password)
b) Something you have (tokens / phones)
c) Something you are (biometrics)
(fingerprint, iris scan, facial recognition)

Phishing attack
Attackers creates fake website that appears similar to a real one

URL obfuscation attack
Attacker can choose similarly looking URL
Homeograph attack
Unicode characters from international alphabets may be used in URLs.

Clickjacking attacks
Exploitation where a user's mouse click is used in a way that was not intended by the user.

Defenses
a) User confirmation: pops dialogue box to let user re-confirm the action (Degrades user experience)
b) UI randomization (difficult & unreliable)
c) Framebusting (can be defeated) Enforcing temporal integrity
Website includes code that prevents other pages from framing it
d) remove cursor customization } Ensure visual integrity of pointer
e) Freeze screen outside of the target display area
f) UI delay: after visual changes on target or pointer, invalidate clicks for X ms
g) Pointer re-entry: after visual changes, invalidate clicks until pointer re-enter target
h) X-Frame-Options (Good, but poor adoption by sites)

Monitoring
a) Network-based Intrusion Detection
b) Host-based Intrusion Detection
c) Log Analysis: can only detect after the fact happened.
d) System Call monitoring

Styles of Detection
a) Signature-Based (Blacklisting)
b) Anomaly-Based (Whitelisting)
c) Specification-Based
d) Behavioral-Based
(Look for evidence of compromise)
e) Honeypots
A sacrifical system that has no operational purpose
Any access is by definition, not authorized

Merkle Trees: one can prove that data item is in the tree in a logarithmic number of steps and using a logarithmic number of nodes in the tree
Audit proof for a node: siblings of nodes on the path from the node to root

No.

Date

DNSSEC

KSK: Key Signing key is used to sign the DNSKEY RRSET

ZSK: Zone Signing key is used to sign everything else

DS: Delegated Signer

The parent signs DS records for the child's KSK

RRSIG: Resource Record Signature

RRSET: Resource Record Set

NSEC: vulnerable to domain enumeration

NSEC3: Attest that there're no domains whose HASH H falls between H(N) and H(M)

Cryptographic hash functions

a) One-way: Given x, it's easy to compute H(x). However, given y, it's infeasible to find any input x such that y = H(x) (preimage resistant)

b) Second preimage resistant: Given x, it's infeasible to find another message x' such that $x' \neq x$ but $H(x) = H(x')$

c) Collision resistant: It's infeasible to find any pair of messages x, x' such that $x \neq x'$ but $H(x) = H(x')$

Collision resistant implies Second preimage resistant

If we modify the compiler to have a backdoor which inserts a backdoor when compiling the login program and also when compiling the compiler, we'd have an awful time "trusting trust"...

Polymorphic Code

Everytime virus propagates, it inserts a newly encrypted copy of itself.

Metamorphic Code

Everytime virus propagates, generat semantically different version of it

Certificate Transparency is used to monitoring and auditing digital certificates

Tor: anonymity of content against local adversaries

vulnerable to global adversaries (traffic/timing analysis)