CS 161 Final Note Sheet

Kerchoff's Principle

You should not rely on the secrecy of the algorithm/protocol and or keysize, as wall as the possible plain text for security because eventually the adversary will figure them out.

Mono-Alphabetic Ciphers: 1 to 1 mapping of characters to symbols

- Subsitution
  - Shift or Caesar's Cipher $E_k(m) \leftarrow m + k(\texttt{mod}\ N)$ $D_k(c) \leftarrow c - k(\texttt{mod}\ N)$
  - Affine Cipher: $E_k(m) \leftarrow k_! m + k_2(\texttt{mod}\ N)$ $D_k(c) \leftarrow k_!^{-1}(c - k(\texttt{mod}\ N)$
  - Substitution Ciphers have an extreme vulnerability to frequency attacks.

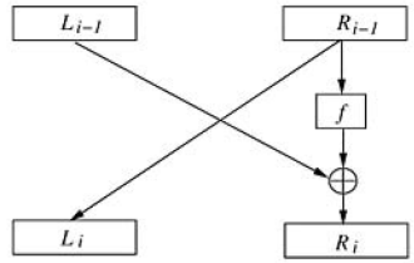
Poly-Alphabetic Ciphers

- Vigenere Cipher: Shift by a repeated key
- Book Cipher (Beale Cipher) key is hidded in a passage of a set book.
- Vernam Cipher
  - Message is m bits and the key is n bits.
  - Bitwise xor the message and the key, if m is greater than n, then use the key multiple times.
- One-Time Pad
  - Same idea as the Vernam Cipher except we use a key that is the same length or greater than the length of the message, then discard it after each use.
- Transposition/Permutation Cipher
  - Break the message into n bit blocks, then on each block perfom the same permutation
  - Despite being polyalphabit, the cipher is still vulnerable to frequency attacks. Because the original patterns are still basically present. You can attack by checking anagrams.

Data Encryption Standard (DES)

DES is a block cipher in which messages are divided into data blocks of a fixed length and each block is treated as one message either in M or in C. The DES encryping and decryption algorithms take as an input a 64-bit plaintext or ciphertext message and a 56-bit key, and output a 64-bit ciphertext or plaintext message. DES is done in 3 steps:

1. Apply a fixed "initial permutation" IP to the input block. $(L_0, R_0) \leftarrow IP(\texttt{Input Block})$ This step has no apparent cryptographic significance.
2. Iterate the following 16 rounds of operations (Feistel Cipher)



- the function is nonlinear and is considered a Substitution Cipher
- the move from $L_i \to R_{i-1}$ is a Transposition cipher
- Vernam cipher is used at the xor
- k is a 48 bit subsection of the 56 bit, "round key"

Single DES

- vulnerable to brute force or exaustive key search attacks

Triple DES

Triple DES uses an encryption-decryption-encryption scheme,
$c \leftarrow E_{k_1}(D_{k_2}(E_{k_1}(m)))$
$m \leftarrow D_{k_1}(E_{k_2}(D_{k_1}(m)))$
This scheme enlarges the keyspace while maintaining backward compatibility with single DES if $k_1 = k_2$

Advanced Encryption Standard (AES)

AES is a block cipher with variable block size and variable keysize. (block size can be 128, 192, 256 bit)
AES has 4 states:

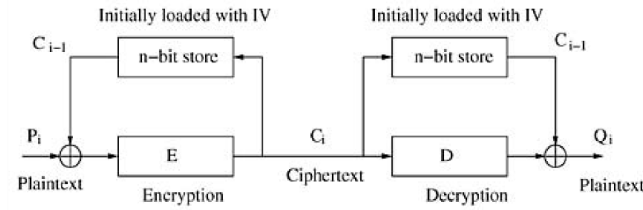
1. Sub Bytes State: nonlinear substitution on each byte
2. Shift Rows State: Transposition rearranges the order of elements in each row
3. Mix Columns State: Polynomial multiplication after converting column to polynomial.
4. Add Round Key State: adds elements of round key to the state, basically bitwise "OR"

Decryption is the inverse of these steps.

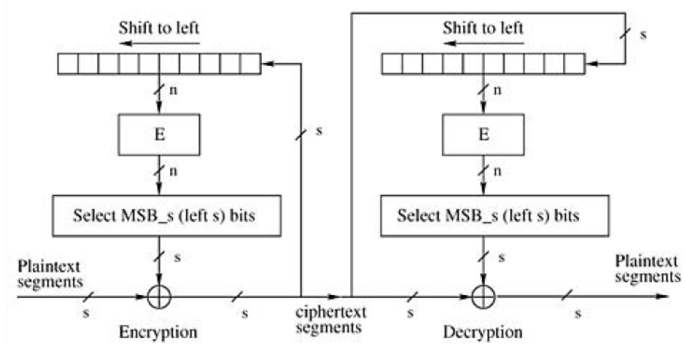
Confidentiality Modes of Operation

Different modes of operation have been devised on top of an underlying block cipher algorithm

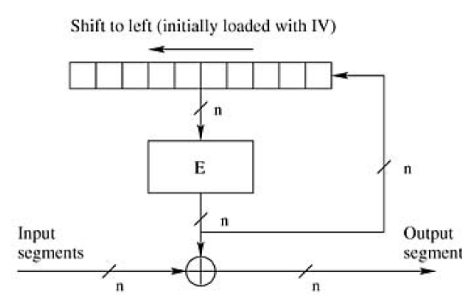
- Electronic Codebook (ECB) Mode This mode encrypts and decrypts every block seperately. It is deterministic and leaves patterns in the cipher text. (for example images.)
- Cipher Block Chaining (CBC) Mode
  - This is the most common mode of operation. In this mode the output is a sequence of n-bit cipher blocks which are chained together so that each cipher block is dependent on all the previous data blocks.
  - Decryption can be done in parallel
  - CBC cannot prived data integrity protection.
  - If the CBC claims data integrity protection, Eve can use (Bomb Oracle Attack) a Decryption Oracle to figure out the padding scheme and eventually the last byte of the cipher text.



- Cipher Feedback (CFB) Mode
  - CFB mode of opration features feeding successive cipher segments which are output from the mode back as input to the underlying block cipher algorithm.
  - CFB requires an IV as the initial n-bit input block



- Output Feedback (OFB) Mode
  - The OFB mode feeds successive output blocks from the underlying block cipher back to it.
  - The feedback blocks form a string of bits which used as the key stream of the Vernam cipher.



- Counter (CTR) Mode
  - The CTR mode features feeding the underlying block cipher algorithm with a counter value which counts up from an initial value. With a counter counting up, the underlying block cipher algorithm outputs successive blocks to form a string of bits. This string of bits is used as the key stream of the vernam cipher, that is, the key stream is XOR-ed with the plaintext blocks. $C_i \leftarrow P_i \oplus E(Ctr_i, i = 1, 2, \ldots, m$ $P_i \leftarrow C_i \oplus E(Ctr_i, i = 1, 2, \ldots, m$

Bomb Oracle Attack

Asymmetric Cryptography

Oneway Trapdoor Function

- Asymmetric crypto system, Public Key Cryptography
- $D \to R$ is oneway, it is easy to evaluate $\forall x \in D$ and difficult to invert for all values in R.

Textbook Encryption Algorithms

- All or Nothing Secrecy: Given Cipher Text the attacker must not be able to get any information about the plain text
- Passive Attacker: The attacker doesn't modify or manipulate ciphertexts they also don't ask for encryption or Decryption services.

Diffie-Hellman Key Exchange Protocol

`Common Input`	$(p, g) : p$ `is a large prime, g is a generator element in` F_p^*
`Output`	`An element in` F_p^* `shared between Alice Bob.`

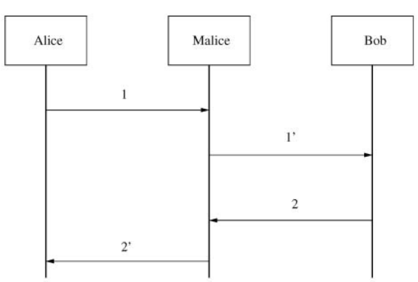
1. Alice picks $a \in U(1, p-1)$; computes $g_a \leftarrow g^a (\texttt{mod}\ p)$; sends g_a to Bob.
2. Bob picks $b \in U(1, p-1)$; computes $g_b \leftarrow g^b (\texttt{mod}\ p)$; sends g_b to Alice.
3. Alice computes $k \leftarrow g_b^a (\texttt{mod}\ p)$
4. Bob computes $k \leftarrow g_a^b (\texttt{mod}\ p)$

Alice and Bob both compute the same key,

$$k = g^{ba} (\texttt{mod}\ p) = g^{ab} (\texttt{mod}\ p)$$

P is a public 2048 bit prime number.

Man in the Middle Attack on Diffie-Helman



1. Alice picks a $\in_u [1, p-1)$, computes $g_a \leftarrow g^a (\texttt{mod}\ p)$ she sends g_a to Malice("bob");
2. (1') Malice("Alice") computes $g_m \leftarrow g^m (\texttt{mod}\ )$ for some $m \in [1, p-1)$; he sends g_m to Bob;
3. (2) Bob picks $b \in_U [1, p-1)$, computes $g_b \leftarrow g^b (\texttt{mod}\ p)$; he sends g_b Malice("Alice");
4. (2') Malice("Bob") sends to Alice: g_m;
5. (3) Alice computes $k_1 \leftarrow g_m^a (\texttt{mod}\ p)$;
6. (4) Bob computes $k_2 \leftarrow g_m^b (\texttt{mod}\ p)$;

Diffie-Helman and the Discrete Logarithm Problem

- Computational Diffie-Hellman (CDH) Problem

`INPUT`	`desc`(F_q) `: a finite field` F_q
	$g^a, g^b \in F_q^*$ `for some integers` $0 < a, b < q$
`OUTPUT`	g^{ab}

- Discrete Logarithm Problem

`INPUT`	`desc`(F_q) `: a finite field` F_q
	$g \in F_q^*$
	$h \in F_q^*$
`OUTPUT`	`the unique integer` $(a < q)$ `such that` $h = g^a$

If the Discrete Logarithm Problem is solved, then the CDH Problem is also solved. But not necessarily vice versa.

Eulers Theorem

given a,n are coprime,
$a^{\phi(n)} = 1(\texttt{mod}\ n)$

Fermat's Little Theorem

given a coprime to N,
$a^{(N-1)} = 1(\texttt{mod}\ N)$, $a \in \mathbb{Z} < N$

Cryptanalysis Against Public-key Cryptosystems

Chosen-Plain Text Attack (CPA)

- An attack chooses a plaintext messages and gets encryption assistance to obtain the corresponding ciphertext messages. The task for the attacker is to weaken the targeted cryptosystem using the obtained plain-text pairs.
- The attacker has an encryption box.
- All public key encryption systems must resist CPA otherwise it is useless.

Chosen-Ciphertext Attack (CCA)

- An attacker chooses ciphertext messages and gets decryption assistance to obtain the corresponding plaintext messages. The task for the attacker is to weaken the targeted cryptosystem using the obtained plaintext-ciphertext pairs. The attacker is successful if he can retrieve some secret plaintext information from a "target ciphertext" which is given to the attacker after the decryption assistance is stopped. That is, upon the attacker receipt of the target ciphertext, the decryption assistance is no longer available.
- The attacker is entitled to a conditional use of a decryption box. The box turns off before the ciphertext is sent.

Adaptive Chosen-Ciphertext Attack (CCA2)

- This is a CCA where the decryption assistance for the targeted cryptosystem will be available forever, except for the target ciphertext.
- The attacker has the decryption box for as long as he wishes, except they can't decipher the original message.

RSA Cryptosystem

- Key Setup
  1. Choose two random prime numbers p and q (typically done by applyling a Monte-Carlo prime number finding algorithm.
  2. Compute $N = pq$
  3. Compute $\phi(N) = (p-1)(q-1)$
  4. Choose a random integer $e < \phi(N)$ such that $\gcd(e, \phi(n)) = 1$, and compute the integer d, such that,
  $$ed \equiv 1(\texttt{mod}\ \phi(N))$$
  5. Publicize (N,e) as the public key, safely destroy p, q, and $\phi(N)$, and keep d as the private key.
- Encryption
  To send a confidential message $m < N$ to Alice, the sender Bob creates the ciphertext c as follows,
  $$c \leftarrow m^e (\texttt{mod}\ N)$$
- Decryption
  To decrypt the ciphertext c, Alice computes,
  $$m \leftarrow c^d (\texttt{mod}\ N)$$

Proof of RSA

$m = (m^e)^d \texttt{mod}\ N$
$m = m^{e^d} \texttt{mod}\ N$, `because` $ed = 1 \texttt{mod}\ \phi(N)$
$ed = a + k\phi(N)$
$m = m^{1+k\phi(N)} \texttt{mod}\ N$
$m = m(m^{\phi(N)})^k \texttt{mod}\ N$
$m = m(\texttt{mod}\ N)$, `iff` $gcd(m, N) = 1$

RSA Problem and the Integer Factorization Problem

- The RSA Problem

`INPUT`	$N = pq$ `with` p, q `prime numbers`
	e`: and integer such that,`
	$gcd(e, (p-1)(q-1)) = 1$
	$c \in Z_N^*$
`OUTPUT`	`the unique integer` $m \in Z_N^*$ `satisfying,`
	$m^e \equiv c(\texttt{mod}\ N)$

- The Integer Factorization Problem

`INPUT`	N`: odd composite integer with at least two distict prime factors.`
`OUTPUT`	`prime p such that` $p \mid N$

- The difficulty of the RSA problem depends, in turn, on the difficulty of the integer factorization problem.

RSA Euler's Theorem

$m^{p-1)(q-1)} \equiv 1(\texttt{mod}\ pq)$

Insecurity of Textbook RSA Encryption

- The RSA Cryptosystem is "all-or-nothing" secure against CPA if and only if the RSA assumption holds, meaning if the attacker has some prior knowledge of the contents of a message (ex a number or bid), they may be able to successfully bruteforce a solution.

  For a plaintext m (<N), with a non-negligible probability, only $\sqrt{m}$ trials are needed to pinpoint m if $\sqrt{m}$ size of memory is available, exploiting,

  $$(m_1 \cdot m_2)^e \equiv m_1^e \cdot m_2^e (\texttt{mod}\ N)$$

- Let $c = m^e(\texttt{mod N})$ such that Malice knows $m < 2$. With non-negligible probability m is a composite number satisfying,

  $$m = m_1 \cdot m_2 \texttt{ with } m_1, m_2 < 2^{\frac{l}{2}}$$

  and with RSA's multiplicative property, we have,

  $$c = m_1^e \cdot m_2^e (\texttt{mod}\ N)$$

  Malice can build a sorted database

  $$\{1^e, 2^e, \ldots, (2^{\frac{l}{2}})^e\}(\texttt{mod}\ N)$$

  Then they can search through the sorted database trying to find $c/i^e(\texttt{mod}\ N)$ for $i = 1, 2, \ldots, 2^{\frac{l}{2}}$ a finding signaled by,

  $$c/i^e \equiv j^e (\texttt{mod}\ N)$$

  Acheiving a square-root level reduction in time complexity.

- Let Malice be in conditional control of Alice's RSA decryption box. If the ciphertext submitted by Malice is not meaningful, then Alice should return the plaintext to Malice. This is reasonable because of the following:

  1. "A random response for a random challenge" is a standard mode of operation in many cryptographic protocols. (including in the Needham-Schroeder protocol)
  2. This random-looking decryption result should not provide an attacker with any useful information.

  Malice wants to know the plaintext of a ciphertext $c \equiv m^e(\texttt{mod}\ N)$ which he has eavesdropped.

  He picks a random number, $r \in Z_N^*$, and computes $c' = r^e c(\texttt{mod}\ N)$ and sends his chosen cipher text, c', to Alice. Alice will return,

  $$c'^d \equiv rm(\texttt{mod}\ N)$$

  Which will appear completely random to Alice.
  Then because Malice has r, he can obtain m with a division modulo N.

Data Integrity

Data Integrity is the security service against unauthorized modification of messages.

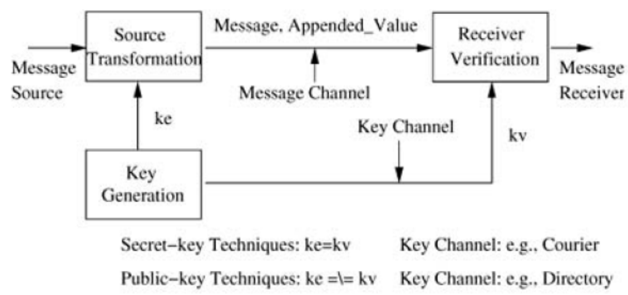
Manipulation Detection Codes (MDC)

- Let K_e denote an encoding key and K_v denote a verification key which matches the encoding key.
- Manipulation Detection Code creation:

  $$\texttt{MDC} \leftarrow f(K_e, \texttt{Data})$$

- Manipulation Detection Code verification:

  $$g(K_v, \texttt{Data}, \texttt{MDC}) = \begin{cases} \texttt{True} & \texttt{if MDC} = f(K_e, \texttt{Data}) \\ \texttt{False} & \texttt{if MDC} \neq f(K_e, \texttt{Data}) \end{cases}$$



Symmetric Techniques of Data Integrity

A MDC is another term for a Message Authentication Code (MAC). A Mac can be created an verified using a keyed hash function technique, or using a block cipher encryption algorithm.

Cryptographic Hash Functions

A Hash function is a deterministic function which maps a bit string of an arbitrary length to a hashed value which is a vit string of a fixed length.

Properties of a Hash Function

- Mixing-transformation
  On any input x, the output hashed value h(x) should be computationally indistinguishable from a uniform binary string.
- Collision Resistance
  It should be computationally infeasible to find two inputs x,y with $x \neq y$ such that $h(x) = h(y)$
- Pre-image resistance
  Given a hashed value h, it should be computationally infeasible to find an input string such that $h = h(x)$
- Practical efficiency
  Given input string x, the computation of $h(x)$ can be done in time bounded by a small-degree polynomial (ideally linear)

Hash Function Applications

- In digital signatures, hash functions are generally used for generating "message digests" or "message fingerprints."
- In public-key cryptosystems, has functions are widely used for realizing a cipher text correctness verification mechanism. This is necessary to protect an encryption scheme from active attackers.
- Hash Functions are used in wide range of applications, pseudo-randomness is required.

Random Oracle

A Random Oracle is a very powerful and imaginary function that occurs if on any input, the distribution of the output hashed value is uniform.

Birthday Attack

Assuming a hash function behaves as a random oracle, the square-root attack (birthday attack) suggests that,

$$2^{|h|/2} = \sqrt{2^{|h|}}$$

random evaluations of the hash function will suffice an attacker to obtain a collision with a non-negligible probablility. To mount the attack, the attacker should generate random message-hash pairs.

$$(m_1, h(m_1)), (m_2, h(m_2)), \ldots$$

until he ends up with finding two messages m and m' satisfying,

$$m \neq m', \ \ h(m) = h(m')$$

This is only really usefull if the hashed message contains a meaningful sub-message.

MAC based Keyed Hash Function

In a shared-key scenario a hash function takes a key as part of its input.

$$\texttt{MAC} = h(k \| M)$$

Where k is a secret key shared between the transmitter and receiver. ($\| = \texttt{concatenation}$)
The receiver should then recalculate the MAC from the message. If they match, then the message is believed to have come from the transmitter. In order for the receiver to verify itself to the sender, we must compute the HMAC,

$$\texttt{HMAC} = h(k \| M \| k)$$

MAC based on Block Cipher Encryption

Let $E_k(m)$ denote a block cipher encryption algorithm keyed with the key, k, on the input message, m. In order to authenticate a message, M, we need to divide M.

$$M = m_1 m_2 \ldots m_l$$

Where each sub-message block $m_i, i = 1, 2, \ldots, l$ has the size of the input of the block cipher algorithm. Let $C_0 = IV$ be a random initializing vector. Here the transmitter applies the CBC encryption:

$$C_i \leftarrow E_k(m_i \oplus C_{i-1}), i = 1, 2, \ldots, l$$

Then pair,

$$(IV, C_l)$$

Digital Signatures

the RSA Signature Scheme

- Key setup
  The key setup is identical to that of standard RSA cryptosystems.
- Signature Generation
  To create a signature of message, m, Alice create,

  $$s = \texttt{Sign}_e(m) \leftarrow m^e(\texttt{mod}\ N)$$

- Signature Verification
  Let Bob be a verifier who knows that the public-key material (N,e) belongs to Alice. Given a message-signature pair (m,s), Bob's verification procedture is,

$$\texttt{Verify}_{(N,d)}(m,s) = \texttt{True if } m \equiv s^d (\texttt{mod } N)$$

Non-repudiation

Non-repudiation is basically the idea that if there is a single entity responsible for creating a digital signature, then it is easy to settle disputes over who has created the signature. No denial of connection with a message.

Entity Authentication

Entity authentication is a communication process by which a principal establishes alively correspondence with a second principal whose cliamed identity should meet what is sought by the first.

- Host-host type: for example upon a reboot the system must identify to a trusted server neccessary information (like trusted copy of OS, trusted clock setting, or trusted environment settings) Client-Server setting where one host requests certain services from the other server.
- User-Host type: Computer login via telnet/ssh (Authenticated through some password protocol) In serious cases mutual authentication is required.
- Process-host type: used more for distributed computing. Ex) used for "mobile code" or a "browser based" java applet.
- Member-club type: ex a member of a club showing a membership card for access. Zero-knowledge identification protocols and undeniable signature schemes

Manipulation Detection Code (MDC)

Lively correspondence

- Freshness
  - Freshness verifies that a message was sent sufficiently recently
  - Data origin doesn't guarantee freshness.
  - This prevents attacks that involve doing large computations on a message or replay attacks.
- nonces
  - A random number, used for verifying a challenge response.
  - Simple example: bob sends a nonce to alice. Alice responds with the encrypted nonce, which was encrypted with a shared private key. If bob recieves the encrypted nonce (in a sufficiently short amount of time) and establishes it was encrypted correctly. This does not provide a proper data-integrity service.
- timestamps

Data Origin Authentication

- Consists of transmitting a message from a purported source (the transmitter) to a receiver who will validate the message upon reception.
- The message validation conducted by the receiver aims to establish the identity of the message transmitter.
- The validation also aims to establish the data integrity of the message subsequent to its departure from the transmitter.
- The validation further aims to establish liveness of the the message transmitter.

Authentication vs Key exchange

- Key exchange or key agreement is usually used during entity authentication or as a subtask

Challenge-response

- Standard vs non-standard (encryption-then-decryption) challenge-response mechanisms
- Standard (as set by ISO, International Organization for Standardization and the IEC, International Electrotechnical Commission) of three challenge-response system.
- Non Standard with Symmetric Cryptography
  1. `Bob` $\rightarrow$ `Alice:` N_B
  2. `Alice` $\rightarrow$ `Bob:` $M, \texttt{MDC}(K_{AB}, M, N_B)$
  3. `Bob reconstructs` $\texttt{MDC}(K_{AB}, M < N_B)$ `and`
     $\begin{cases} \texttt{accepts} & \texttt{if two MDCs are identical} \\ \texttt{rejects} & \texttt{otherwise} \end{cases}$
- Non Standard with Asymmetric Cryptography
  1. `Bob` $\rightarrow$ `Alice:` N_B
  2. `Alice` $\rightarrow$ `Bob:` $\texttt{sig}_A(M, N_B)$
  3. Bob verifies the signature using his nonce and
     $\begin{cases} \texttt{accepts} & \texttt{if signature verification passes} \\ \texttt{rejects} & \texttt{otherwise} \end{cases}$

Unilateral Authentication

- Only one of the two participants is authenticated.

Mutual authentication vs trusted third-party authentication

- In mutual authentication, both communicating entities are authenticated to each other.
- Originally simply unilateral authentication done twice, once in each direction, until wiener's attack (the Canadian Attack)
- Trusted Third Party (TTP) Authentication requires the principals to use a centralized open system from a trusted third party to authenticate.
- One of the ISO and IEC standards for mutual authentication is the "ISO public Key Three-Pass Mutual Authentication Protocol"

  `PREMISE:` `A has a public key certificate` $\texttt{Cert}_A$
  `B has a public key certificate` $\texttt{Cert}_B$
  `GOAL` `They achieve mutual authentication`

  1. $B \rightarrow A : R_B$
  2. $A \rightarrow B :$ $\texttt{Cert}_A$, `Token` AB
  3. $B \rightarrow A :$ $\texttt{Cert}_B$, `Token` BA

  Here,

$$\texttt{Token } AB = R_A||R_B||B||\texttt{sig}_A(R_A||R_B||B)$$

$$\texttt{Token } BA = R_B||R_A||A||\texttt{sig}_B(R_B||R_A||A)$$

Needham-Schroeder password protocol

`Premise:` `User, U and Host, H have setup U's password entry` $(ID_u, f(P * u))$`where f is a one-way function; U memorizes password` P_u
`Goal:` `U logs in H using her/his password.`

1. U $\rightarrow$ H : ID_u;
2. H $\rightarrow$ U : "Input password:";
3. U $\rightarrow$ H : P_u;
4. H applies f on P_u, finds entry $(ID_u, f(P_u))$ from its archive; Access is granted if the computed $f(P_u P)$ matches the archived. Where f is a trapdoor function. In the unix password system, f is a series of 25 rounds of DES. Salt (12 bit random number) is used to randomize the flips.

- Data integrity of the password storage file becomes important and the protocol is still vulnerable to online password eavesdropping attack.
- One-time password scheme. In this flawed modification the number of times the hashing function is iterated one more time.

Passwords and salt

S/Key Protocol

`PREMISE:` `User U and Host H have setup U's initial password entry` $(ID_U, f^n(P_U), n)$`where f is a cryptographic hash function; U memorizes password` P_U
`GOAL:` `U authenticates to H without transmitting` P_U `in cleartext`

1. $U \rightarrow H : ID_U$
2. $H \rightarrow U : c$, "Input Password"
3. $U \rightarrow H : Q = f^{c-1}(P_U)$
4. H finds entry $(ID_U, f^c(P_U), c)$ from its archive
   Access is granted if $f(Q) = f^c(P_U)$, and U's password entry is updated to
   $(ID_U, Q, c-1)$

Encryted key exchange (EKE)

- The EKE protocol protects the password against online eavesdropping and offline dictionary attacks.
- Premise: User, U and Host, H share a password P_u; The system has agreed on a symmetric encryption algorithm, K() denotes symmetric encryption keyed by K; U and H also agreed on an asymmetric encryption scheme, E_u denotes asymmetric encryption under U's key.
- Goal: U and H achieve mutual entity authentication, they also agree on a shared secret key.

1. U generates a random "public" key E_u, and sends to H: U, $P_u(E_u)$
2. H decrypts the cipher chunk using P_u and retrieces E_u; H generates random symmetric key, K, and sends to U: $P_u(E_u(K))$
3. U decrypts the doubly encrypted cipher chunk and obtains K; U generates a nonce N_u, and sends to H: $K(N_e)$
4. H decrypts the cipher chunk using K, generate a nonce, N_H, and sends to U: $K(N_u, N_h)$
5. U decrypts the cipher chunk using K, and return to H: $K(N_h)$
6. If the challenge-response in 3,4,5 is successful, logging-in is granted and the parties proceed further secure communication

Attacks on authentication protocols

Message Replay Attack

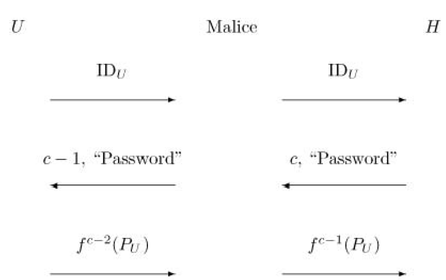
- Malice has previously recorded an old message from a previous run of a protocol and now replays the recorded message in a new run of the protocol
- blocked by freshness

Man-in-the-Middle Attack

Malice intercepts messages between Bob and Alice

Parallel Session Attack

The parallel session attack consists of two or more runs of a protocol executed concurrently under Malice's protection
An Attack on the S/KEY Protocol:



Result: Malice has $f^{c-2}(P_U)$ which he can use for logging-ing in the name of U

Reflection Attack

A reflection attack is when an honest principal sends a message to an intended communication partner. Malice intercepts the message and reflects it back at the host

Interleaving Attack

In an interleaving attack, two or more runs of a protocol are executed in an overlapping fashion under Malic's orchestration. Malic may compose a message and send it out to a principal in one run from which he expects to receive and answer.

Attack Due to Type Flaw

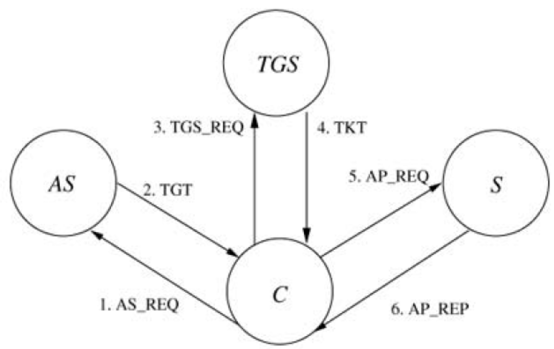
Malice uses a flaw, including a principal being tricked to misinterpret a nonce, a timestamp or an indentification into key

Attack Due to Name Omission

Name omission is a serious problem that could allow exploits

Kerberos

- single signon
  - Each user memorizes a password, this is the single-signon credential for using the Kerberos system.



- exchanges (authentication service exchange, ticket-granting service exchange, client-server authentication application exchange)
  - Authentication Service Exchange (AS Exchange): runs between a client, C and an "authentication server", AS
  - Ticket-Granting Service Exchange (TGS Exchange): runs between C and a "ticket granting server" TGS after the AS Exchange
    * Checks to see if the difference in time between the client time and host time are within a reasonable range for freshness.
  - Client/Server Authentication Application Exchange (AP Exchange): runs between C and an application server, S after the TGS Exchange
    * The AP Exchange a client, C that uses the newly obtained application session key, to obtain application services fromt he Application Servers
- key distribution center (authentication server, ticket granting server)

SSL/TLS

- Process: *=optional

1. $C \to S$: ClientHello
2. $S \to C$: Server Hello, ServerCertificate*, ServerKeyExchange*, CertificateRequest*, ServerHelloDone
3. $C \to S$: ClientCertificate*, ClientKeyExchange, CertificateVerify*, ClientFinished
4. $S \to C$: ServerFinished

- Hello Message Exchange: Server and Host let each other know what protocols they are capable of running
- handshake with key exchange
- crypto-suite selection, certificates
- use of nonces and random secrets

Access Control

Access Control Matrix

	Asset 1	Asset 2	file	device
Role 1	read,write,execute,own	execute	read	write
Role 2	read	read,execute		

Simple Access Control List

User	Accounting Data
Sam	rw
Alice	rw
Bob	r

Middleware

Doing access control at the level of files and programs has grown dramatically in complexity over the years. Middleware is an attempt to provide authentication inside programs or databases.

Object Request Brokers (ORBs)

an ORB is a software component that mediates communications between objects (object as in OOP).
Languages have been created to express security policies, this was specifically driven by the defence and rights management industies.

Sandboxing

Starting with Java, another means to implement access control is through a software sandbox. Code is stuck in and executed in an interpreter, the Java Virtual Machine (JVM)

Virtualization

Virtualization refers to systems that enable a single machine to emulate a number of machines independently. At the client end, virtualization allows people to run a host operating system on top of a guest (for example Linux on top of OSX).

Trusted Computing

'Trusted Computing' was an initiative launched by Microsoft and a number of other tech companies to provide a more secure computer. The aim was to provide software and hardware add-ons to the PC architecture that would enable people to be sure that a given program was running on a machin with a given specification. Originially created to support Digital Rights Management.

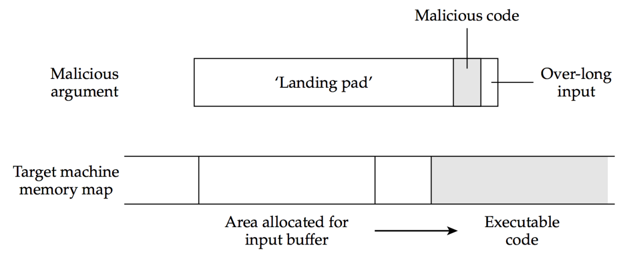
Hardware Protection

Is the idea of preventing one process from interfering with another, commonly called the protection problem. The confinement problem attepts to keep programs from communicating outward other than through their authorized channels The confinement problem attepts to keep programs from communicating outward other than through their authorized channels.

Common Attacks

Stack Smashing

Half of all technical attacks on operating systems that are reported come from memory overwritting attacks or smashing the stack. The basic idea behind it is that programmers are often careless about checking the size of arguments, so an attacker who passes a long argument to a program may find that some of it gets treated as code rather than data.



Format String Vulnerability

This arises when a machine accepts input data as a formatting instruction (ex %n in the C command printf()). This can allow the string's author to write to the stack.

SQL Insertion Attacks

These commonly arise when a careless webdeveloper passes user input to a back-end database without checking to see whether it contains SQL code.

Integer Manipulation Attack

If you don't check for data sizes before writing, the integer manipulation attack uses an overflow, underflow, wrap-arround or truncation that can result from writing an inappropriate number of bytes to the stack.

Race Conditions

Race conditions occur when a transaction is carried out in two or more stages, and it is possible for someone to alter it after the stage which involves verifying access rights.

User Interface Failures

One of the earliest attacks was the Trojan Horse, a program that the administrator is invited to run and which will do some harm if he does so. Ex) a game that would check whether the player was the administrator, and if so create another administrator account with a known password.