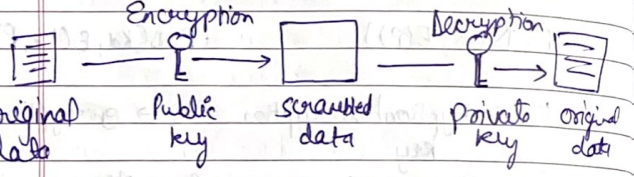
Symmetric key Cryptosystem	Asymmetric key Cryptosystem
1) It only require a single key for both encryption and decryption	It requires two keys, a public and a private key, one to encrypt and the other to decrypt
2) The size of cipher text is same or smaller than the original plain text	The size of cipher text is same or larger than the original Plain text
3) The encryption process is very fast	The encryption process is very slow
4) It is used when large amount of data is required to transfer	It is used when small amount of data is required to transfer
5) It only provides confidentiality	It provides confidentiality, authencity and non repudiation
6) The length of key used is 128 or 256 bits	The length of key used is 2048 or higher
7) Resource utilization is low	Resource utilization is high
8) efficient	less efficient
9) security is less (One key is used)	security is more (two keys are used)
10) Mathematical Representation	Mathematical Representation
$P = D(K, E(P))$	$P = D(Kd, E(Ke, P))$
K → encryption/ decryption key	Ke → encryption key
P → Plain text	P → Plain text
D → Decryption	D → Decryption
E(P) → encryption of plain text	E(Ke,P) → encrypting plain text using encryption key
11) Example DES, 3DES, AES	Example ECC, DSA, RSA

Sunil
DATE
PAGE
"Rough 'n' Fair"

Public key cryptography

Public key cryptography involves a pair of keys as a public key and a private key. Each public key is published and the corresponding private key is kept secret. Data that is encrypted with the public key can be decrypted only with the corresponding private key.



Note: Public key encryption requires many calculations ∴ it is not always appropriate for large amount of data.

Principles of Public key Cryptosystem

There are two basic principles of any cryptosystem i.e Confidentiality and authenticity. We have seen that the symmetric cryptosystem has a problem associated with these two principles.

Sunil
DATE
PAGE
"Rough 'n' Fair"

In symmetric cryptosystem the problem associated with confidentiality is that we all know in symmetric cryptograph a secret key is used to encrypt as well as decrypt the message. So, this key must be shared by both the communicating parties by any means or they must rely on a third party for the distribution of the key i.e key distribution centre. But relying on a third party again risk the secrecy of the secret key.

Symmetric key also had an issue with authentication. To become widespread there was a need for digital signatures as well as that assure all parties that a particular message has been sent from a particular person.

The public key cryptosystem is successful in achieving both these principles.

Confidentiality :- Because the content is encrypted with an individual's public key, it can only be decrypted with the individual's private key, ensuring

only the intented receipient can decrypt and view the contents

Authenticity :- since the individual's unique private keys was used to apply the signature, receipient can be confident that the individual was the one to actually apply the signature.

Nobody is able to modify the message without having the sender's private key. So, public key cryptosystem has achieved authentication in both the terms of data integrity and source

Any public key cryptographic algorithm has six element as follows

Plain text
This is a readable message which is given as input to the algorithm. In a public key algorithm, the plain text is encrypted in blocks.

Encryption Algorithm
The encryption algorithm is implemented on the plain text which performs several transformations on plain text.

Public and Private keys
These are the set of keys among which if one is used for encryption the other would be used for decryption. The transformation of plain text by encryption algorithm depends on the key chosen from the set to encrypt the plain text.

Cipher Text
This is the output of encryption algorithm. The generated cipher text totally depends on the key selected from the set of public and private key. Both of these keys one at a time with plain text would produce different cipher texts.

Decryption Algorithm
This would accept the output of the encryption algorithm i.e cipher text and will apply the related key to produce the original plain text.

Steps in public key cryptography

1) Each user has to generate two keys one of which will be used for encryption and other for decryption of messages

2) Each user has a pair of keys among which one has to be made public by each user. And the other has to be kept secret.

3) If a user has to send a message to a particular receiver then the sender must encrypt the message using the intended receivers public key and then send the encrypted message to the receiver

4) On receiving the message the receiver has to decrypt the message using his private key.

note In public key there is no need for key distribution as in symmetric key cryptography.

Rivest Shamir Adleman

Rivest Shamir Adleman

RSA Algorithm – in 1978

RSA is an asymmetric cryptography algorithm.

The idea of RSA is based on the fact that it is difficult to factorize a large integer.

2 keys are used public and private key concept is used

- Public key → known to all users
- Private key → kept secret, not shareable to all

The RSA scheme is a block cipher in which plain text and cipher text are integers blw 0 and n-1 for some value n.

Key Generation

i) select two large prime no p and q

ii) calculate $n = p * q$

iii) calculate $\phi(n) = \phi(p * q)$ (Euler's totient function)

$= \phi(p) * \phi(q)$

$= (p-1) * (q-1)$

iv) choose the value of e

$1 < e < \phi(n)$ (coprime to $\phi(n)$)

RSA

↳ Block Cipher

↳ 0 to n-1, for given value of n

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

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

vi) public key = $\{e, n\}$

vii) private key = $\{d, n\}$

Eg $p = 61$ $q = 53$

$n = p * q$
$= 61 * 53 = 3233$

$\phi(n) = \phi(p * q)$
$= (p-1) * (q-1)$
$= 60 * 52$
$= 3120$

$e = 17$

public key (17, 3233)

$d = e^{-1} \bmod \phi(n)$
$ed = 1 \bmod \phi(n)$
$17 \times d = 1 \bmod 3120$
$d = 2753$

private key (2753, 3233)

GCD → 1 - coprime

Finding d.

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

$d = \frac{(\phi(n) * i) + 1}{e}$

$= \frac{(3120 * 1) + 1}{17} = 183.58$

$= \frac{(3120 * 2) + 1}{17} = 367.11$

$= \frac{(3120 * 3) + 1}{17} = 550.647$

$= \frac{(3120 * 15) + 1}{17} = 2753$

Encryption $(13, 143)$ (e, n)

→ Plain text

$C = P^e \bmod n \quad P < n$

$C = 13^{13} \bmod 143$

$13 \bmod 143 = 13$
$13^4 \bmod 143 = 104$
$13^8 \bmod 143 = 91$

$C = [(13^8 \bmod 143)(13^4 \bmod 143)(13 \bmod 143)] \bmod 143$
$= (91 \times 104 \times 13) \bmod 143$
$C = 52$

C → ciphu text

Decryption $(37, 143)$ (d, n)

$P = C^d \bmod n$
$= 52^{37} \bmod 143$

$52 \bmod 143 = 52$
$52^4 \bmod 143 = 26$
$52^{32} \bmod 143 = 130$

$P = [(52^{32} \bmod 143)(52^4 \bmod 143)(52 \bmod 143)] \bmod 143$
$= (130 * 26 * 52) \bmod 143$
$P = 13$

based on quadratic congruence.

Rabin Cryptosystem

It is a public key cryptosystem invented by Michael Rabin. It uses a asymmetric key encryption for communicating b/w two parties and encrypting the message.

The rabin cryptosystem can be thought as an RSA cryptosystem in which the value of e & d are fixed

The encryption is $C \equiv P^2 \pmod{n}$
decryption is $P \equiv C^{1/2} \pmod{n}$.

In rabin cryptosystem the public key is n,
private key is (p, q)

Key generation:-

1) Choose two large prime no p and q which satisfy the condition

$$p \neq q \rightarrow p \equiv q \equiv 3 \pmod{4}$$

2) Calculate the value of n
$n = p * q$

3) publish n as public key and save p and q as private key.

Encryption
For encryption only public key is used

1) get public key n

2) convert message to ASCII value.
convert it to binary
change binary value back to decimal M

3) $C = M^2 \bmod n$ M = plain text
C = cipher text

4) send c to receipient

Decryption

1) Accept C from sender

2) Specify a and b with extended Euclidean GCD
$a * p + b * q = 1$

3) $r = C^{(p+1)/4} \bmod p$
$s = C^{(q+1)/4} \bmod q$

4) calculate x and y

$x = (a.p.r + b.q.s) \bmod p$
$y = (a.p.r - b.q.s) \bmod q$

Four roots are then
$M1 = x$ $M3 = y$
$M2 = -x$ $M4 = -y$

Convert them into binary and divide all in half

6) Determine in which the left and right half are same. Keep that binary's one half and convert it to decimal M. Get the ASCII character for the decimal value M. The resultant character gives the correct message sent by sender.

Q) $p = 23$ $q = 7$

Encryption

1) both 23 and 7 are congruent to 3 mod 4

$23 \% 4 = 3$
$7 \% 4 = 3$

2) $n = p \times q$
$= 23 \times 7$
$= 161$

3) public key $(n) = 161$
private key p and q = 27 and 7

4) plain text $P = 24$
n and p are prime

$C = P^2 \bmod n$
$= 24^2 \bmod 161$
$= 93$

Decryption

1) $C = 93$

$a_1 = +(C^{(p+1)/4}) \bmod p$
$= +(93^{24/4}) \bmod 23$
$= 1 \bmod 23$

$a_2 = -(C^{(p+1)/4}) \bmod p$
$= -(93^6) \bmod 23$
$= -1 \bmod 23$
$= 22 \bmod 23$

$a_3 = +(C^{(q+1)/4}) \bmod q$
$= +(93^2) \bmod 7$
$= 4 \bmod 7$

$a_4 = -(c^{(q+1)/4}) \bmod q$

$= -4 \bmod 7$

$= 3 \bmod 7$

The possible answers are

(a_1, b_1) (a_2, b_2) (a_3, b_1) (a_1, b_2)

Use CRT to obtain possible ans

ELGamal Encryption Algorithm

ELGamal is a public-key cryptosystem. It uses asymmetric key encryption for communicating between two parties and encrypting the message.

Key Generation

i) select large prime no (P)
ii) select decryption key (Private key) (D)
iii) select second part of encryption key (Public key) (E_1)
iv) third part of encryption key or public key (E_2)

$E_2 = E_1^D \bmod P$

v) Public key = (E_1, E_2, P), Private key = D

Encryption

i) Select Random no (R)

$C_1 = E_1^R \bmod P$

$C_2 = (PT \times E_2^R) \bmod P$

$CT = C_1, C_2$

Decryption

$PT = [C_2 \times (C_1^D)^{-1}] \bmod P$

Eg $P = 11$ $D = 3$ $E_1 = 2$

$E_2 = E_1^D \bmod P$

$= 2^3 \bmod 11$

$= 8 \bmod 11 = 8$

Public key = (2, 8, 11)
Private key = 3

R = 4

$C_1 = E_1^R \bmod P$
$= 2^4 \bmod 11$
$= 16 \bmod 11$
$= 5$

PT = 7

$C_2 = (PT \times E_2^R) \bmod P$
$= (7 \times 8^4) \bmod 11$
$= 28672 \bmod 11$
$= 6$

Encryption

$CT = (5, 6)$

$PT = [C_2 \times (C_1^D)^{-1}] \bmod P$
$= [6 \times (5^3)^{-1}] \bmod 11$

$(125)^{-1} \bmod 11$
$(125 \times x) \bmod 11 = 1$
$(125 \times 3) \bmod 11 = 1$
$x = 3$

Decryption

$6 \times 3 \bmod 11$
$18 \bmod 11$
$PT = 7$

Diffie – Hellman key exchange

i) It is not an encryption algorithm
ii) It is used to exchange secret keys b/w two users
iii) We will use asymmetric encryption to exchange the secret key between users

This algo is used because when we are sending a key to receiver it can be attached in b/w.

i) consider a prime number 'q'
ii) select α such that it must be the primitive root of q and $\alpha < q$

a is a primitive root of q if

$a \bmod q$
$a^2 \bmod q$
$a^3 \bmod q$ ---- $a^{q-1} \bmod q$

given results $\{1, 2, 3, \ldots, q-1\}$

i.e. the values should not be replated & we should have all values in the O/P set from 1 to q-1

eg Let $q = 7$

$\alpha < q$ i.e it is a primitive root

Let $\alpha = 5$ we can take any of two primitive root 3 or 5

α and q → global public element (known to everyone)

3) assume X_A (private key) and $X_A < q$ of A

calculate $Y_A = \alpha^{X_A} \bmod q$

public key of A

key generation of person 1

assume $X_A = 3$

$Y_A = 5^3 \bmod 7$

$Y_A = 6$

4) assume X_B (private of X_B)

$X_B = 4$

$Y_B = \alpha^{X_B} \bmod q$

key generation of person 2

$Y_B = 5^4 \bmod 7$

$= 2$

→ Now we will calculate secret key

To calculate the secret key both the sender & receiver will use public key

$K_1 = (Y_B)^{X_A} \bmod q$ $K_2 = (Y_A)^{X_B} \bmod q$

↳ public keys known to all ←

$K_A = (Y_B)^{X_A} \bmod q$

$= 2^3 \bmod 7$

$= 1$

$K_B = (Y_A)^{X_B} \bmod q$

$= 6^4 \bmod 7$

$= 1$

$K_1 = K_2$ Thus, the keys are exchanged

Sunil
DATE
PAGE
"Rough 'n' Fair"

Key Management

In cryptography it is very tedious task to distribute the public and private keys between sender and receiver.

Cryptographic key management generally refers to key management. It is basically defined as management of cryptographic keys that is used to deliver different purposes in a cryptographic network.

The basic cryptographic key managment deals with the generation, exchange, storage, use, replacement and destruction of keys.

Key management is essential to maintain the security of cryptography It is one of the most different states of cryptography and involves aspects such as system policy, user training, organizational and departmental interactions

Process of cryptographic Key Managment

DATE
PAGE
"Rough 'n' Fair"

1. Key exchange
Sometimes some cryptographic functions required cryptographic key exchange. Public keys can be openly exchanged symmetric keys requiring a secured connection.

Traditional approach involves secure channels such as diplomatic bags and protocols like diffie Hellman key exchange. While modern methods employ modern systems such as asymmetric key algorithms.

2. Key storage
Key storage is basically the allocation of keys. Distributed keys should be stored securely. This is so done to ensure communication security. There are various methods to ensure perfect storing of the cryptographic keys.

The most common technique employed for the purpose is an encryption application. The application manages the key and its usage depends on an access password to control the use of the key.

3. Key Use
As the duration of the usage of keys increases, risk factors also shoot high. As the risk of a hacker is directly proportional to its duration of use. That's why the keys should be frequently changed. This limits the loss of vulnerable information.

Challenges involved in Cryptographic key management.

1) Unintended use of the keys
These keys should be used for the specific function for which they are designed for. It should not be used for other purposes. As when keys are used for intended purpose, they cause a risk to the protection.

2) Weak keys
Keys are nothing but the combination of random numbers and characters. Easier the combination, the hacker can easily crack these keys. Key should be strong enough to protect the data and its vulnerabilities. An ideally qualified random number generator should be used to create the keys.

3) Reuse
Sometimes improper reuse of the already used keys is also a threat to the protection of cryptographic keys.

4) Non-rotation
Using the same older symmetric algorithms again and again increases the threat issue for the keys. To avoid this key should be rotated and updated regularly.

5) Non destruction
If the expired keys are not destroyed immediately, it can lead to the accidental compromise of future data. Thus, keys should be securely deleted leaving no trace, once they are expired.

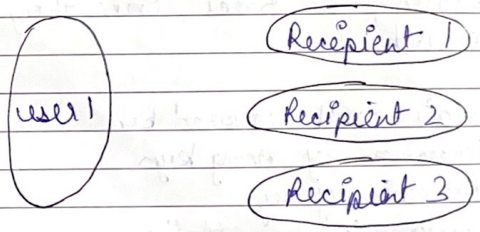
Threats can be covered by
1) Generation of strong keys
2) Protection of keys
3) Automatic key rotation
4) Destroy used keys
5) Full lifecycle management of cryptographic keys

Two aspects of Key Management

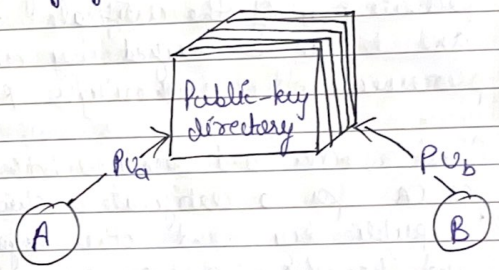
1) Distribution of public keys
2) Use of public-key encryption to distribute secrets

Public Key Distribution
Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:

Public Announcement
Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered an masquerade as claimed user.



Publicly Available Directory
In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participation Registration, access and allow to modify values at any time, contains entries like E name, public key $\}$. Directories can be accessed electronically still vulnerable to forgery or tampering



Public Key Authority
It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

DATE
PAGE
"Rough 'n' Fair"

Public Certification

This time authority provides a certificate to allow key exchange without real-time access to the public authority each time. The certificate is accompanied by some other info such as period of validity, rights of use, etc.
All of this content is signed by the private key of the certificate authority and it can be verified by anyone processing the authority's public key.

First sender and receiver both request for CA for a certificate which contains a public key and other information and then they can exchange these certificates and can start communicating

DATE
PAGE
"Rough 'n' Fair"

Massage Authentication & Hash Functions

Authentication Requirements

Revelation
It means releasing the content of the message to someone who does not have an appropriate cryptographic key.

Analysis of Traffic
Determination of the pattern of traffic through the duration of connection and frequency of connections between different parties

Deception
Adding out of context messages from a fraudulent source into a communication network. This will lead to mistrust between the parties communicating and may also cause loss of critical data.

Modification in the Content
Changing the content of a messages between parties. This includes insertion, deletion and reordering of messages.

Modification in the timings
This includes replay and delay of messages

sent between different parties. This way session tracking is also disrupted.

Repudiation
- Source Refusal
  When the source denies being the originator of a message
- Destination Refusal
  When the receiver of the message denies the reception

(*) Authentication Functions
Message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Message authentication may also verify sequencing and timeliness.
A digital signature is an authentication technique that also includes measures to counter repudiation by either source or destination.

All message authentication and digital signature mechanisms are based on two functionality levels

Lower Level: At this level, there is a need for a function that produces an authenticator, which is the value that will further help in the authentication of a message.

Higher Level: The lower level function is used here in order to help receivers verify the authenticity of messages.

These messages authentication functions are divided into three classes

1) Message Encryption
While sending data over the internet there is always a risk of a Man in the Middle (MITM) attack. A possible solution for this is to use message encryption. In message encryption data is first converted to a ciphertext and then sent any further. Message Encryption can be done in two ways
Symmetric Encryption
Public Key Encryption

2) Message Authentication Code (MAC)
A message authentication code is a security code that the user of a computer has to type in order to access any account or portal. These codes are recognized by

the system so that it can grant access to the right users. These codes help in maintaining information integrity. It also confirms the authenticity of the message.

3) Hash function

A hash function is nothing but a mathematical function that can convert a numeric value into another numeric value that is compressed. The input to this hash function can be of any length but the output is always of fixed length. The values that a hash function returns are called the Message digest or hash values.

4) Message Authentication Codes

MAC stands for Message Authentication or Here in MAC sender and receiver share same key where sender generates a fixed size output called Cryptographic checksum or Message Authentication code and appends it to the original message.

On receiver's side, receiver also generates the code and compare it with what he/she received thus ensuring the originality of the message.

These are components

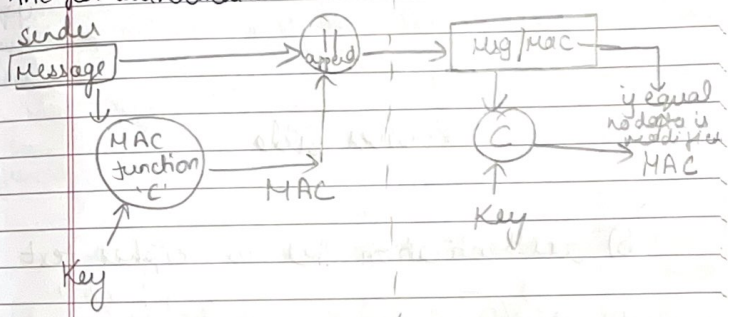
Message
Key
MAC algorithm
MAC value

When A sends a msg to B, it calculates the MAC as a f^n of the message and the key

$MAC = C(K,M)$

M = input message
C = MAC function
K = shared secret key.

MAC for authentication

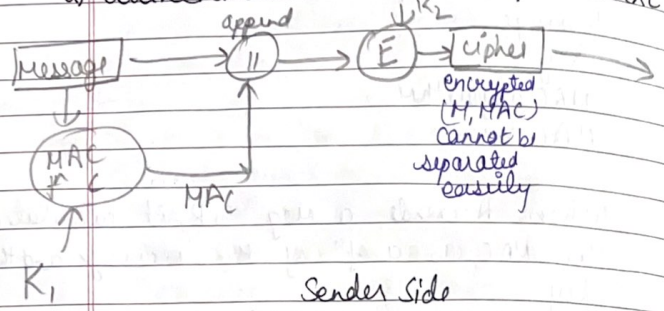


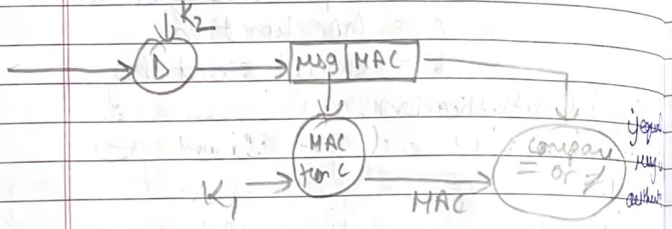
In this only authentication is achieved using MAC function

No confidentiality because if 3rd party come in blw then he can get the message ∴ no security

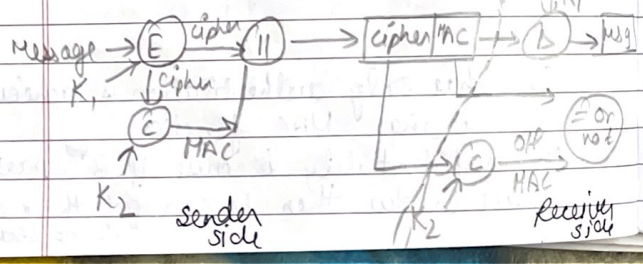
MAC for authentication & confidentiality

a) authentication tied to plain text





b) authentication tied to ciphertext



(X) Hash Functions

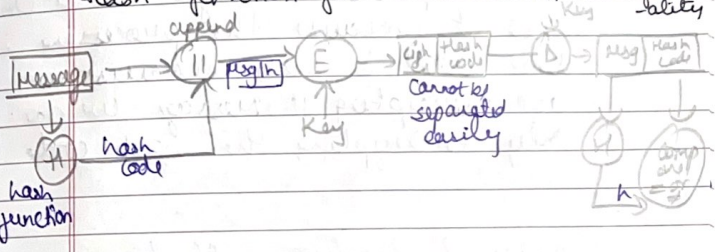
Hash function maps a big number or string to a small integer that can be used as the index in the hash table.

$$h = H(M)$$

M is a variable-length message
H(M) is the fixed length hash value (also referred to as a message digest or hash code)

There are different methods to provide authentication in different situations

Hash function for authentication + confidentiality

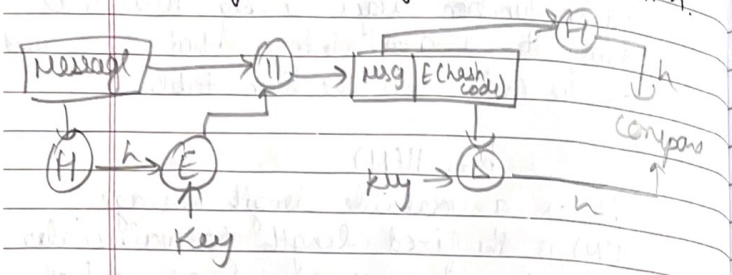


~~Hash function for authentication~~

authentication :- if both hash codes equals in the end.

confidentiality :- maintained because msg was encrypted before sending

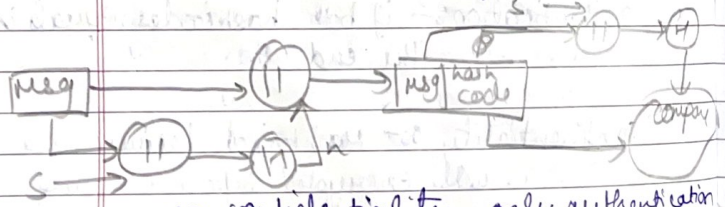
Hash function for authentication.



only hash code is encrypted using symmetric encryption

If you need only authentication & no confidentiality and your msg is not private messages so we can use it because the processing time will be less because we are encrypting the Message. We are only encrypting the hash code

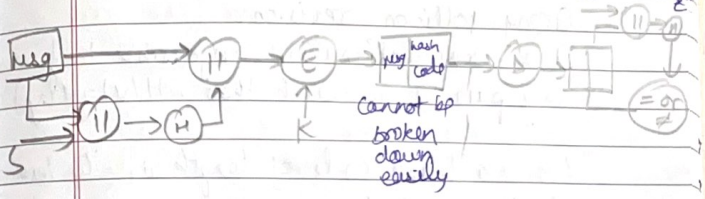
sender & Receiver will have a secret code 's'
and it must be kept secret



no confidentiality, only authentication

confidentiality can be added in prev approach by encrypting the entire msg

Take the msg & append with the secret code 's' then apply hash ~~code~~ function. It gives 'h' now append 'h' & msg. Now encrypt using key 'K' we get encrypted (msg + h)
Now it will be sent to receiver side



Security of Hash Functions & MACs

Two categories: Brute-force & cryptanalysis

Brute-force

1) Hash Functions
The strength of a hash function against brute-force attacks depends on the length of the hash code produced by the algorithm
three desirable properties

Sunil
DATE
PAGE
"Rough 'n' Fair"

One Way
For any given code h, it is computationally infeasible to find x such that $H(x) = h$.

Weak collision resistance
For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$

Strong collision resistance
It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$

For a hash code of length n, the level of effort required, as we have seen is proportional

one way	2^n
weak collision	2^n
strong collision	$2^{n/2}$

2) Message Authentication Codes
A brute force attack on a MAC is a more difficult undertaking because it requires known message - MAC pairs. To attack a hash code, we can proceed in the following way.

Sunil
DATE
PAGE
"Rough 'n' Fair"

Given a fixed message x with n-bit hash code $h = H(x)$, a brute-force method of finding a collision is to pick a random bit string y and check if $H(y) = H(x)$. The attacker can do this repeatedly offline. To proceed, we need to state the desired security property of a MAC algorithm, which can be expressed as follows:

Computation resistance
Given one or more text-MAC pairs $(x_i, C_K(x_i))$ it is computationally infeasible to compute any text-MAC pair $(x, C_K(x))$ for any new input $x \neq x_i$.

In another word, the attacker would like to come up with the valid MAC code for a given message x. There are two lines of attack possible: Attack that key space and attack the MAC value. We examine each of these in turn.

level of effort for brute-force attack on MAC $\rightarrow$ min $(2^k, 2^n)$

key length and MAC length to satisfy
$\min(k, n) \geq N$
where N is perhaps in the range of 128 bits

Sunil
DATE
PAGE
"Rough 'n' Fair"

Cryptanalysis

As with encryption algorithms, cryptanalytic algorithms for hash functions and MAC algorithms seek to exploit some property of the algorithm to perform some attack other than an exhaustive search. The way to measure the resistance of a hash code or MAC algorithm to cryptanalysis is to compare its strength to the effort required for a brute force attack. That is, an ideal hash or MAC algorithm will require a cryptanalytic effort greater than or equal to the brute-force effort.

Sunil
DATE
PAGE
"Rough 'n' Fair"

Extended Euclidean Algorithm

Given two integers a and b we often need to find 2 integers s and t such that

$$s * a + t * b = gcd(a,b)$$

So extended euclidean algo can calculate gcd(a,b) and the value of s and t

$s = s_1 - q s_2$ $\quad t = t_1 - q t_2$

q	a = r_1	b = r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

GCA = 7
s = -1
t = 6

$s * a + t * b$
$-1 * 161 + 6(28) = 7$ gcd.

Sunil
DATE
PAGE
"Rough 'n' Fair"

Euclidean / Euclid's Algorithm.

used for determining GCA of two very int (generally for large no)

$gcd(a,b) = gcd(b, a \bmod b)$
$gcd(a,0) = a$

Q9 gcd (1025, 35)
gcd (b, a % b)
gcd (35, 10)
gcd (10, 5)
gcd (5, 0)
5 Answer

gcd (600, 45)
gcd (45, 15)
gcd (15, 0)
15 Answer

Sunil
DATE
PAGE
"Rough 'n' Fair"

Fermat's Theorem

Also known as Fermat's little theorem and Fermat's primality test, in no theory, the statement first given in 1640 by French Mathematician Pierre de Fermat, that for any prime number n and any integer x such that n does not divide x, n divides exactly into $x^n - x$.

Eulers totient function

$$x^{n-1} \equiv 1 \bmod n \qquad \phi(n) = n-1$$

x & n are coprime

n = prime no
x = is not divisible by n
$(x \neq 0 \bmod n)$

$x = 3 \quad n = 5$
$3^{5-1} = 3^4 = 81$
$81 \equiv 1 \bmod 5$

Special case of Eulers — should be prime in fermat

$x^{\phi(n)} = 1 \bmod n$ — n prime in fermat
$\Rightarrow x^{n-1} = 1 \bmod n$

variant of fermat

$x^{\phi(n)+1} = x \bmod n$
$x \times x^{\phi(n)} = x \cdot 1 \bmod n$
$x^{(n-1)+1} = x \bmod n$
$x^n = x \bmod n$

x and n should be coprime — $\gcd(x,n) = 1$

Sunil
DATE
PAGE
"Rough 'n' Fair"

Euler's Theorem

It is a generalization of Fermat's little theorem handling with powers of integers modulo positive integers. It increase the applications of elementary no theory such as theoretical supporting structure for the RSA cryptosystem

This theorem states that every x and n that are ϕ relatively prime

$$x^{\phi(n)} = 1 \bmod n$$

where $\phi(n)$ is Euler's totient function which counts the no of the integers less than n that are relatively prime to n

eg $x = 4 \qquad n = 165$
$\gcd(4, 165) = 1$ — coprime then we can write as
$(4)^{\phi(165)} = 1 \bmod 165$

note: $x^{\phi(n)a} = 1 \bmod n$

$\phi(165) = \phi(15 * 11)$
$= \phi(15) * \phi(11)$
$= \phi(3*5) * \phi(11)$
$= \phi(3) * \phi(5) * \phi(11)$
$= 2 * 4 * 10$
$= 80$
$(4)^{80} = 1 \bmod 165$

Sunil
DATE
PAGE
"Rough 'n' Fair"

Multiplicative inverse

$x \div n$ x should not be $\neq 0$

$x \not\equiv 0 \bmod n$ (n is a prime no)

$x \cdot y \equiv 1 \bmod n$ (y is multiplicative inverse of x)

$y = x^{-1} \bmod n$

eg $3 \not\equiv 0 \bmod 5$

$3 \times 2 \equiv 1 \bmod 5$

$2 = 3^{-1} \bmod 5$

2 is MI of 3

or

2 is MI of 3 mod 5

if n is not a prime no then x and n should be coprime

$5 \not\equiv 0 \bmod 9$

$5 \times 2 \equiv 1 \bmod 9$

Sunil
DATE
PAGE
"Rough 'n' Fair"

Euler's Totient function

$\phi(n)$ for $[n \geq 1]$ is defined as the number of +ve integers less than n that are coprime to n

$\phi(5) = \{1, 2, 3, 4\} = (4)$

$\phi(6) = \{1, 5\} = (2)$

when n is prime no

$\phi(n) = n-1$ $\phi(23) = 22$

{a & b are coprime} $\phi(a*b) = \phi(a) * \phi(b)$

$\phi(35) = \phi(7*5)$

$= \phi(7) * \phi(5)$

$= 6 * 4 = 24$

Euler's Toient function is the mathematical multiplicative function which count the positive integers up to the given integer generally known as n that are a prime to n and the function can be used to understand the no of prime no. that exist up to the given integer n.