Unit-III

Cryptography hash functions:-

1) message Authentication:-

*Authentication is nothing but, it will verifying the identity of user (from correct person (or) not).

*Authentication can be done by using authenticator.

*Authenticator generated by authentication functions.

*Here we have three Authentication functions..

1) Message encryption
2) Message Authentication code (MAC)
3) Hash functions (H).

1) Message function Encryption:-

*It converts plain text to cipher text by using key is known as message encryption.

*Here the cipher text act as authenticator.

2) Message authentication code:-

$c(M,k) = o/p$ (fixed length code)

c = authentication function

m = message → (plain text)

k = key

o/p = MAC code

*MAC code is act as authenticator.

3) Hash function (H):-

* Here in case of key, we will be using hash function.
* key is replaced with hash function

H(m) = fixed length code (Hash code h)

H = hash function
h = hash code

* This hash code will act as authenticator.

---

2) secure hash algorithm (SHA):-

* It is a modified version of MD5 (Message digest)
* In MD5 the length of the o/p is 128 bits.
* In SHA the length of the o/p is 160 bits.

Working:-

i) padding:-

* In this, we have to add extrabits to the original message.

original message + (padding) → extrabits.

* so, that total length is 64 bit, less than exact multiple of 512.

Example:- original msg = 1000 bits

512 × 1 = 512 bits
512 × 2 = 1024 bits
512 × 3 = 1536 bits

1536
−64
1472

∴ add 472 bits

1000 bits + 472 bits = 1472 bits.

2) Appending :-

* Append the original length before padding

Calculate length mod 64

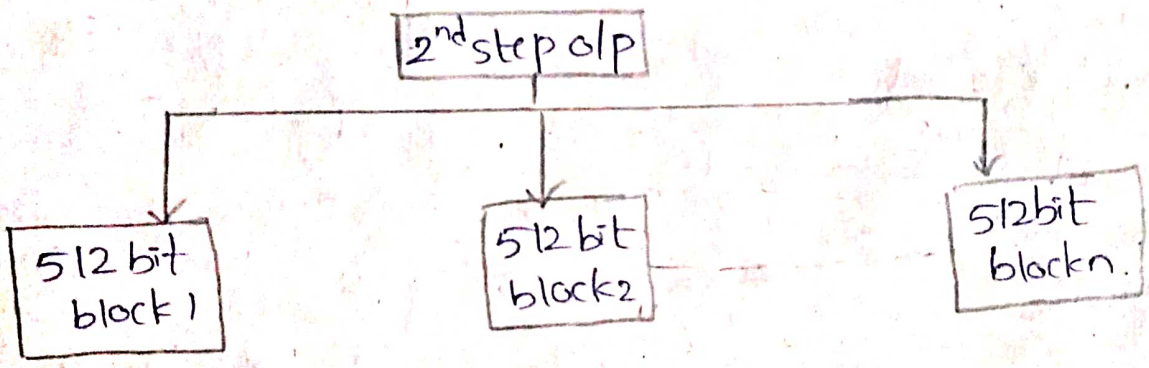
* Most of the cases, 64 bits is obtained as answer

(∴ append 64 bits)

* so, it again becomes multiple of 512.

Means 1472 bits + 64 bits = 1536 bits.

3) Dividing :- (each 512 bits)



4) Initialising :- (5 chaining variables)

each = 32 bit

*Here we have consider 5 chaining variables, the values are predefined.

(A) (B), (C), (D) & (E) → values predefined.

5) process Blocks :- 5 chaining variables into

i) copy ~~5 chaining~~ corresponding variables

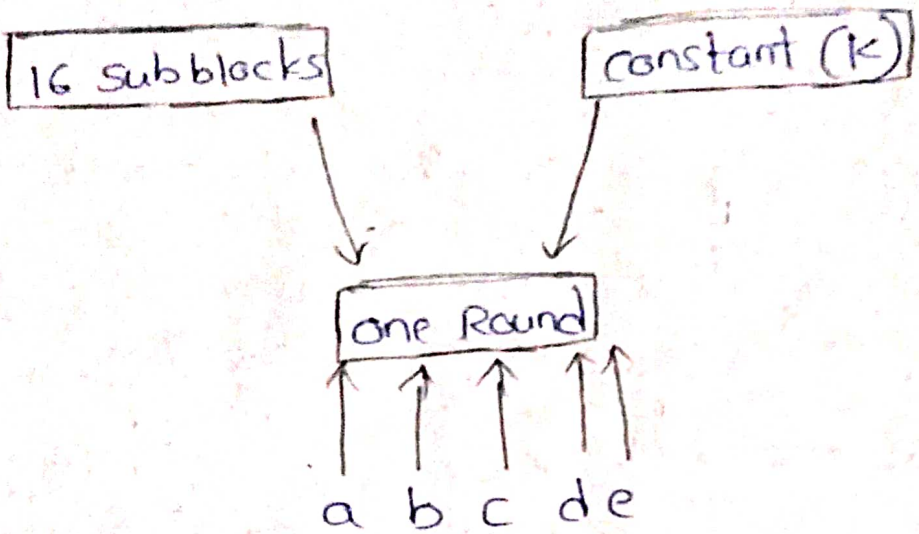
A = a, B = b, C = c, D = d, E = e.

ii) divide into no. of 512 bit blocks

(16 – 32) → each 32 bits.
↓
subblocks

iii) Four Rounds (each round = 20 steps)

16 subblocks and a constant (k)



$$a = b + ((a + process, p(b,c,d,e) + m[i] + T[k]))$$

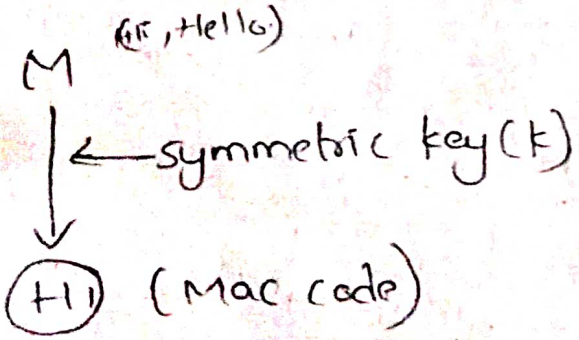
PART-B

Message Authentication code (MAC):-

* It is similar to message digest 5
* In this, symmetric key cryptography is used.

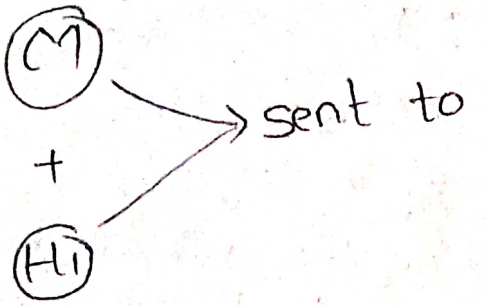
Working:-

* If a sender wants to send a message M by using symmetric key (k), then we get the H1 (MAC code) i.e., cipher text



* Now, the message and Hash code (H1) combined together sent to the Receiver.
* The Receiver calculate his own MAC(H2) by using key (k) i.e., same key that is used at

the sender side.



Now, on Receivers side, H_1 and H_2 are compared.

$H_1 = H_2$ → no change in message

$H_1 \neq H_2$ → message is changed.

Significance of MAC:-

1. Receiver can know if message is changed / not.
2. Receiver has assurance that message is from correct sender. (because same key for S and R)

2) Message Authentication Requirements:-

*There are seven message Authentication Requirements.

1) Disclosure (Release of msg content)
2) Traffic Analysis
3) Masquerade

→ [which is same in attacks of security]

4) content modification
5) sequence modification
6) Timing modification
7) Repudiation

Content modification:- changes to the contents of a message, including insertion, deletion, transposition, or modification.

4) ~~Insertion~~

5) Sequence modification:- Any modification to a sequence of messages blw parties, including insertion, deletion and reordering.

6) Timing modification:- Delay (or) replay of messages. In a connection oriented application, individual messages in the sequence could be delayed (or) replayed.

7) Repudiation:-
Denial of receipt of message by destination (or) Denial of transmission of message by source.

---

3) Hash based MAC (HMAC)

* It is used in secure sochet layer. algorithm.

Working of HMAC:-

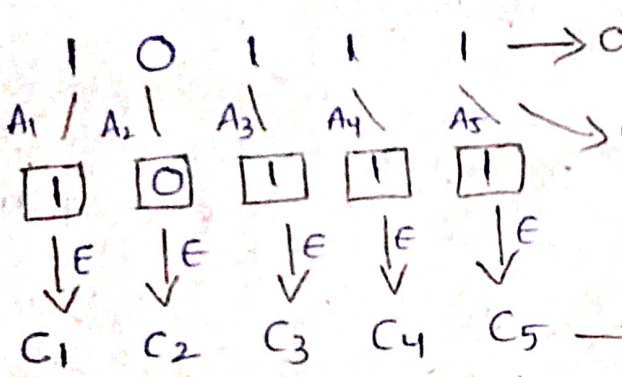
original msg(m) → message digest is generated
↓ MD5/SHA
↓ ← key(K)
encryption
↓
MAC (CT)

* By using MD5/SHA, the original message (M) is generated the message digest.
* The message digest is encrypted by using the key (K). Then MAC is obtained which is known as cipher text.
* In MAC - direct MAC is generated
* In HMAC - MAC is generated with the help of msg digest

1) Cipher Based MAC (CMAC):-

* It has message size limit.
* It is based on block cipher algorithm.
* The given message is divided into equal number of blocks and each block is encrypted separately.

Ex:-



* Here the last cipher text (C5) act as a MAC. Because we have 5 cipher texts from that we have choose last one.

* Theoretically:-

$$C_1 = E(K, A_1)$$

$$C_2 = E(K, (A_2 \oplus C_1))$$

$$C_3 = E(K, (A_3 \oplus C_2))$$

$$C_4 = E(K, (A_4 \oplus C_3))$$

$$\vdots$$

$$C_n = E(K, (A_n \oplus C_{n-1}))$$

C_n ↓

Act as MAC.

5) Digital Signature:-

* It is Asymmetric key cryptography.
* It uses both private and public key.
* For encryption, private key is used.
* For decryption, public key is used.
* It is used for both authentication and Non-Repudiation.

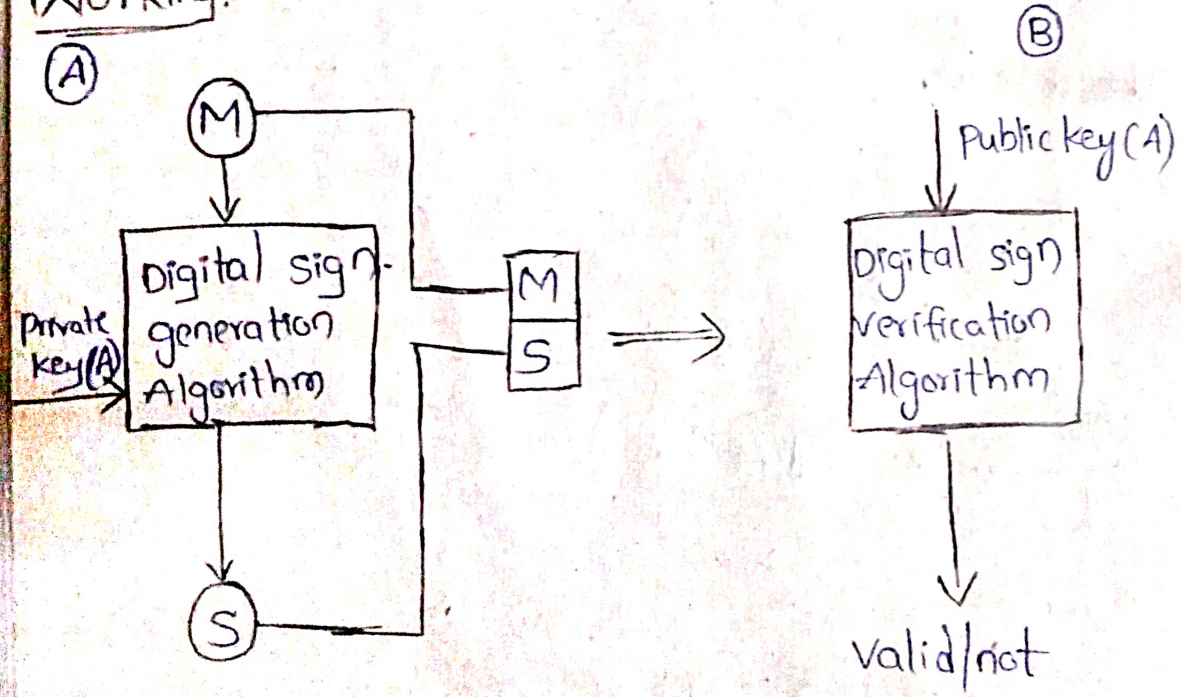
* **authentication:-** sending the message to the correct person.

* **Non-Repudiation:-** The person who receives the msg, he cannot deny (like phone, gpay)

Signature:-

It is a proof of identity. That is it from correct sender / not.

Working:-



* The message and private key (A) are as the inputs of digital signature generation Algorithm.
* And It generates the signature.
* Again the message and generated signature combined together and sent to the Receiver
* The public key (A) of Receiver side and the combination of message and signature are as the inputs for digital signature verification Algorithm.
* Then it checks wheather it is valid (or) not.
* If message matches then it is valid.
* If message is not matching then it is not valid.

6] Elgamal digital signature:-

* It is one of the digital signature scheme.
* For encryption we are using public key
* For decryption we are using private key

Working:-

1) select a prime number (q)
2) select a primitive root (α) of q
3) generate a random integer (X_A)

$$1 < X_A < q-1$$

4) compute $Y_A = (\alpha)^{X_A} \bmod q$
5) Generate keys for user (A)

private key $\Longrightarrow X_A$

public key $\Longrightarrow \{q, \alpha, Y_A\}$

6) generate hashcode (m) for the plain Text (M)

$$m = H(M) \quad 0 \leq m \leq q-1$$

7) Generate a random Integer K
$1 \le K \le q-1$ and gcd $(K, q-1) = 1$

8) Now calculate S_1 and S_2 ; $S_1 = \alpha^K \bmod q$

$S_2 = K^{-1}(m - X_A S_1) \bmod q-1$

9) Now we got the signature pair (S_1, S_2)

Note, at user B's side

calculate V_1 and V_2

$V_1 = \alpha^m \bmod q$

$V_2 = (Y_A)^{S_1} \cdot (S_1)^{S_2} \bmod q$

if $V_1 = V_2$
signature is valid

if $V_1 \neq V_2$
signature is not valid.

Example:-

Let $q = 19$ and $\alpha = 10$

Now, Random integer X_A $(1 < X_A < q-1)$

$\Rightarrow 1 < X_A < 18$

$\Rightarrow \boxed{X_A = 16}$

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

$= (10)^{16} \bmod 19$

$\boxed{Y_A = 4}$

(A) keys:- private key $X_A = 16$

public key $\{q, \alpha, Y_A\} = (19, 10, 4)$

Now, generate Hash code (m)

$m = H(M) \quad 0 \le m \le q-1$

$0 \le m \le 18$

$\therefore \boxed{m = 14}$

generate (k), $0 \le k \le q-1$ and gcd $(k, q-1) = 1$

$0 \le k \le 18$ and gcd $(k, 18) = 1$

$\therefore \boxed{k = 5}$

calculate $S_1 = \alpha^k \bmod q$

$= (10)^5 \bmod 19$

$\boxed{S_1 = 3}$

$S_2 = k^{-1}(m - X_A S_1) \bmod q-1$

$k^{-1} \Longrightarrow k^{-1} \bmod q-1$

$5^1 \bmod q-1$

$5 \times ? = 1 \pmod{18}$

$\boxed{k^{-1} = 11}$

$\left[\begin{array}{l} \frac{5 \times ?}{18} = 1 \\ \frac{5 \times 11}{18} = 1 \\ \therefore \frac{55}{18} = 1 \end{array}\right]$

$S_2 = 11(14 - 16 \times 3) \bmod 18$

$= 374 \bmod 18$

$\boxed{S_2 = 4}$

At B's end:-

$V_1 = \alpha^m \bmod q$

$= 10^{14} \bmod 19$

$\boxed{V_1 = 16}$

$V_2 = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$

$= 4^3 \times 3^4 \bmod 19$

$= 5184 \bmod 19; \boxed{V_2 = 16}$

Now, $V_1 = V_2$

∴ signature is valid.

PART-C

Key Management and distribution:-

key management:-

* The main aim of key management is to generate a secret key b/w two parties and store it to prove the authenticity b/w communicating users.

* key management is the techniques which support key generation, storage and maintainance of the key b/w authorized users.

* key management plays an important role for securing cryptographic goals like confidentiality, authentication, data integrity and digital signatures.

* Basic purpose of key management is key generation, key distribution, controlling the uses of keys, updating destruction of keys and key backup/recovery.

The points to be executed in KM:-

1) user registration
2) user initialization
3) key generation
4) key installation
5) key registration
6) Normal use
7) key backup
8) key update
9) key Recovery
10) key de-registration and revocation.

2] Key distribution in symmetric key:-

* There are four ways:-

1) physical delivery
2) key distribution center (KDC)
3) using previous keys.
4) using third party

1) physical delivery:-

* The sender and receiver will meet physically and exchanging the key.
* It is most secure way to exchange key
* The disadvantage is, it takes more time.

2) key distribution center:-

* It will generate the key and it will distribute the key for both sender and Receiver
* It takes less time.
* It is authentic, but you have to relay on (or) depend. third party (KDC).

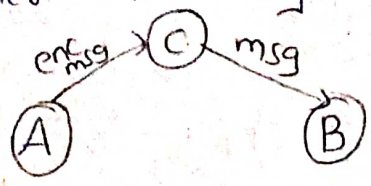
3) using previous key:-

* By encrypting the previous key, we generate the new key.
* We didn't generate the new key directly, by using some Hint of previous key, we generate new key.

4) using third party:-

* The sender send the msg to third party, that third party will send the msg to Receiver. Here we have trusted third party.

* The sender and Receiver will communicate with each other indirectly.



3] key distribution [in Asymmetric key]

* There are four ways:-
  1) public Announcement
  2) public key directory
  3) public key Authority
  4) certificate Authority.

Public Announcement:-

* The particular user will announce the key to all other users in that network.
* So, they can do encryption (or) decryption. They will broadcast.

2) public key directory:-

* It is like telephone directory.
* All users will put (or) store their keys in public key directory.
* User can come and search for it's required key and takes the key
* changes should reflect in the directory also. like updation, adding new keys etc--

3) public key Authority:-

* It is similar to the directory but, improves a security by tightening control over the distribution of keys from the directory.

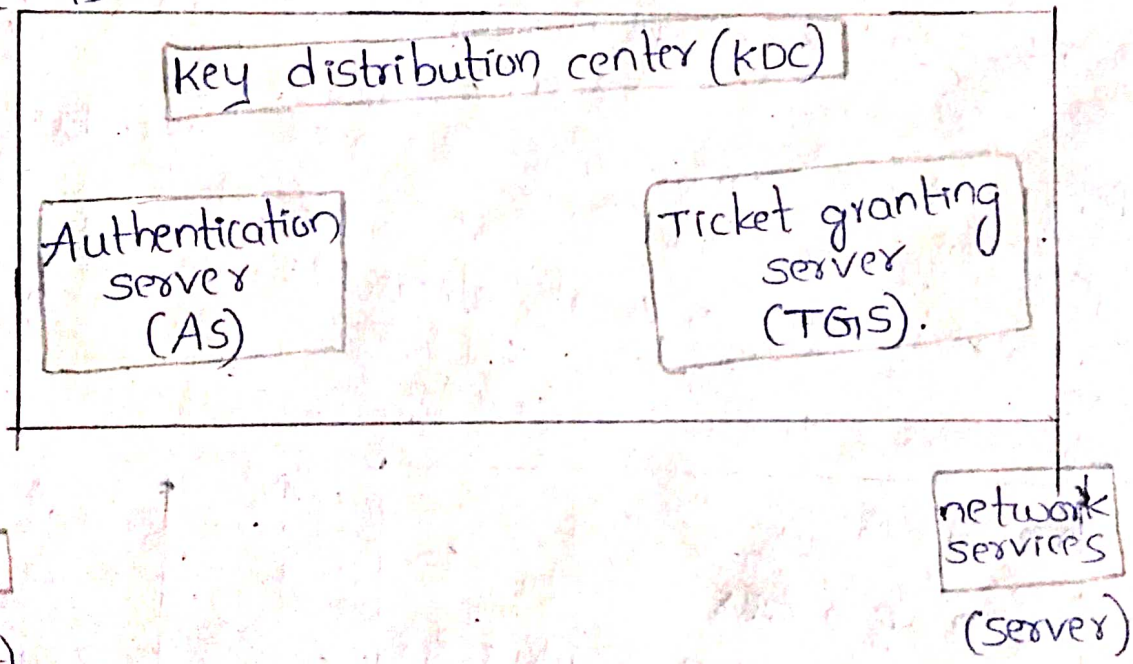
4) Certificate Authority:-

* A trusted third party organization that issues public key certificates is known as a ceritificate Authority (CA).
* The CA can be likened to a notary public.

4) Distribution of public keys:-

5] Kerberos:-

* It is a network authentical protocol.
* It follows a client server Architecture
* It follows a symmetric key algorithm.
* It Requires a third party for key
* KDC is a database of all secret keys.



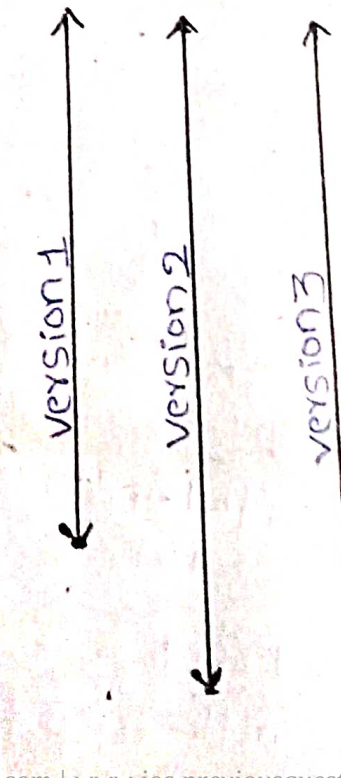
* user sends a request to key distribution center for keys.
* Then Authentication server will respond, and sends a ticket to the user.
* The ticket is in the encrypted form. Then user will decrypt the ticket to get the Hash code.

* The hash code again sent to the authentication server.
* Then authentication server checks that authenticity i.e., if the user will able to decrypt the ticket correctly then it declare that he is a cerifited user. (or) authorized user.
* Then Authentication server gives service ticket to the ticket granting server (TGS)
* The TGS will gives the service ticket (secret key) to the user.
* By using this service ticket, the user will communicate with network services.

6] X-509 Authentication Service:-

* It is a digital cerificate which is accepted by internationally.
* It does not generate any keys. but it provides a way to access public keys.
* There are several elements in x509 certificate.
* It has three versions.

Versions
serial number
signature Algorithm Identifier
Issuer Name
validity period
subject Name
public key Information
Issue unique ID
subject unique ID
Extentions.



Versions:-

* we have 3 versions:-
* version 1 is from version to public key information.
* version 2 is from version to subject unique ID.
* version 3 is from version to extentions.

Serial number:-

* It is a serial number of certificates.

Signature Algorithm identifier:-

* In order to sign on the cerificate, which algorithm the user used. The algorithm may be a RSA, IDEA etc...

Issuer name:- (organization name):
The name of the person who issued the certificate.

Validity period:-

* Validity period means from which date to date and time to time it will have. validity of certificate.

Subject name:-

* It is name of the person to whom you are giving the certificates.

Public key information:-
In order to encrypt (or) decrypt the msg, the user (subject) will be using the public key information.

Issue unique ID & subject unique id:-
Every issuer have a unique id and every subject have unique id.

Extensions:

* If you want to add any descriptions, These extensions are optional
* This may (or) may not be included.

7] public key infrastructure:-

* It is standard which is followed for managing, storing and revoking the digital certificate
* It follows asymmetric key cryptography.
* It includes the:-
  - message digests (Integrity)
  - Digital signature (Authentication, Non-Repudiation)
  - encryption services (confidentiality).

Architecture of PKI:-

* There are four parts.

1) Certificate Repository
2) Entity
3) Registration Authority (RA)
4) certificate Authority (CA).

1) certificate Repository:-

* storing the certificates and information of certificates.
* certificates ID, the name, owner all the information stored in certificate Repository.

2) entity:-

It is the user of PKI, it can be a single person, organization, Router etc....

3) Registration Authority:-

* It is used for Registration and verification
* It is a function for certificate enrollment used in public key infrastructures.

4) certificate Authority:-

* A trusted organization that issues public key certificates is known as certificate Authority.
* It can be likened to a notary public.