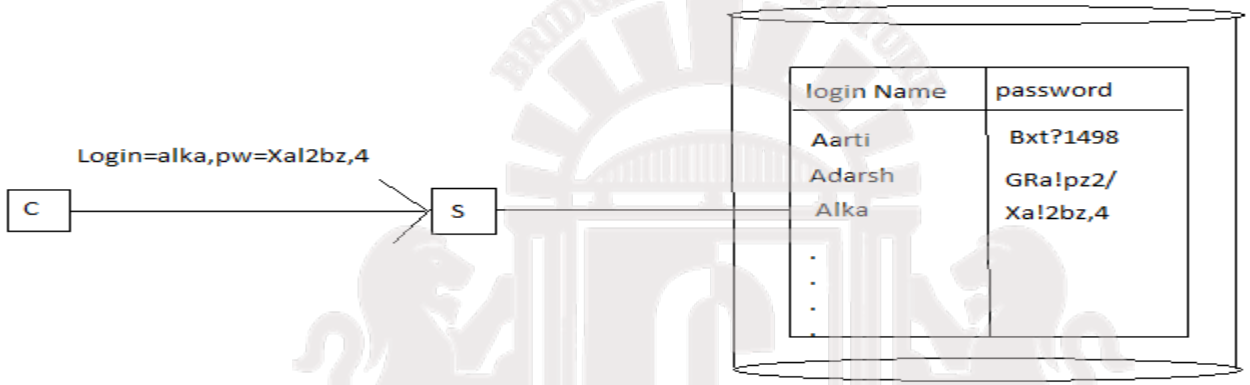
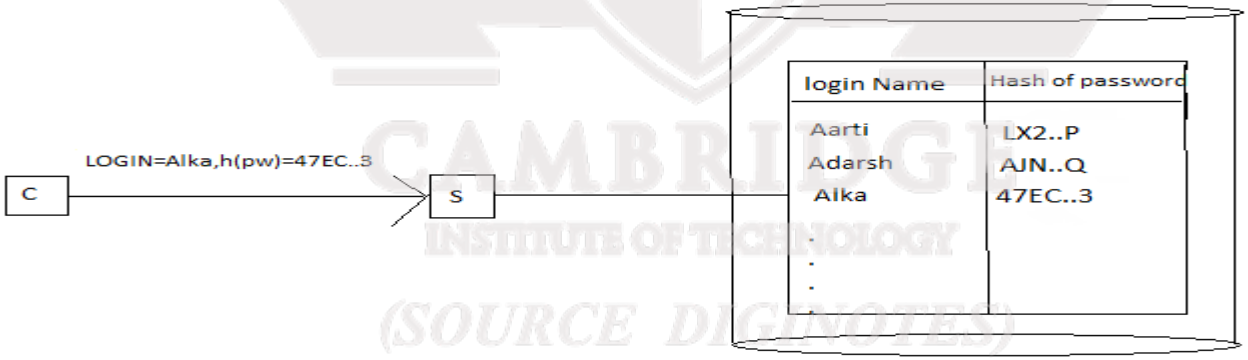
AUTHENTICATION -1

- ONE WAY AUTHENTICATION
- PASSWORD BASED AUHTENTICATION
- CERTIFICATION BASED AUHTENTICATION
- MUTUAL AUHTENTICATION
- DICTIONARY ATTACKS

1.PASSWORD BASED AUHTENTICATION

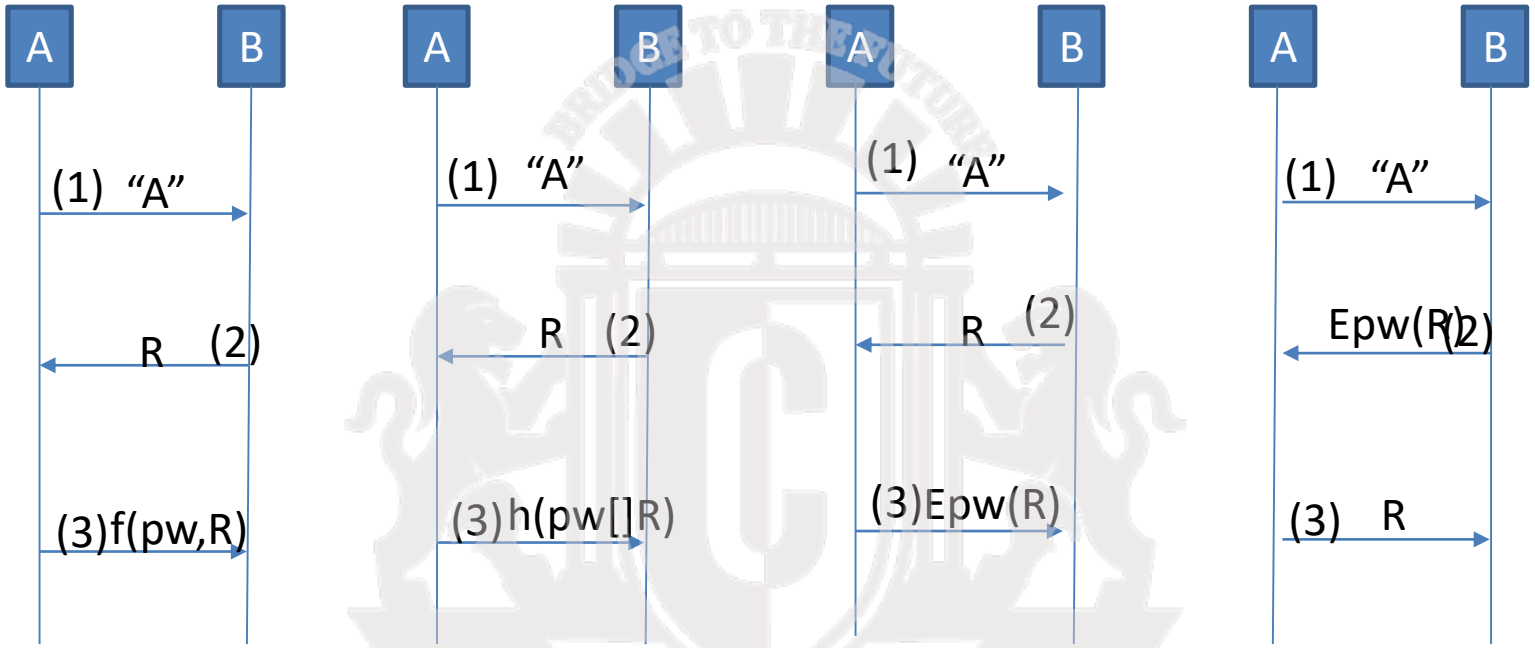


(A) Communicating password

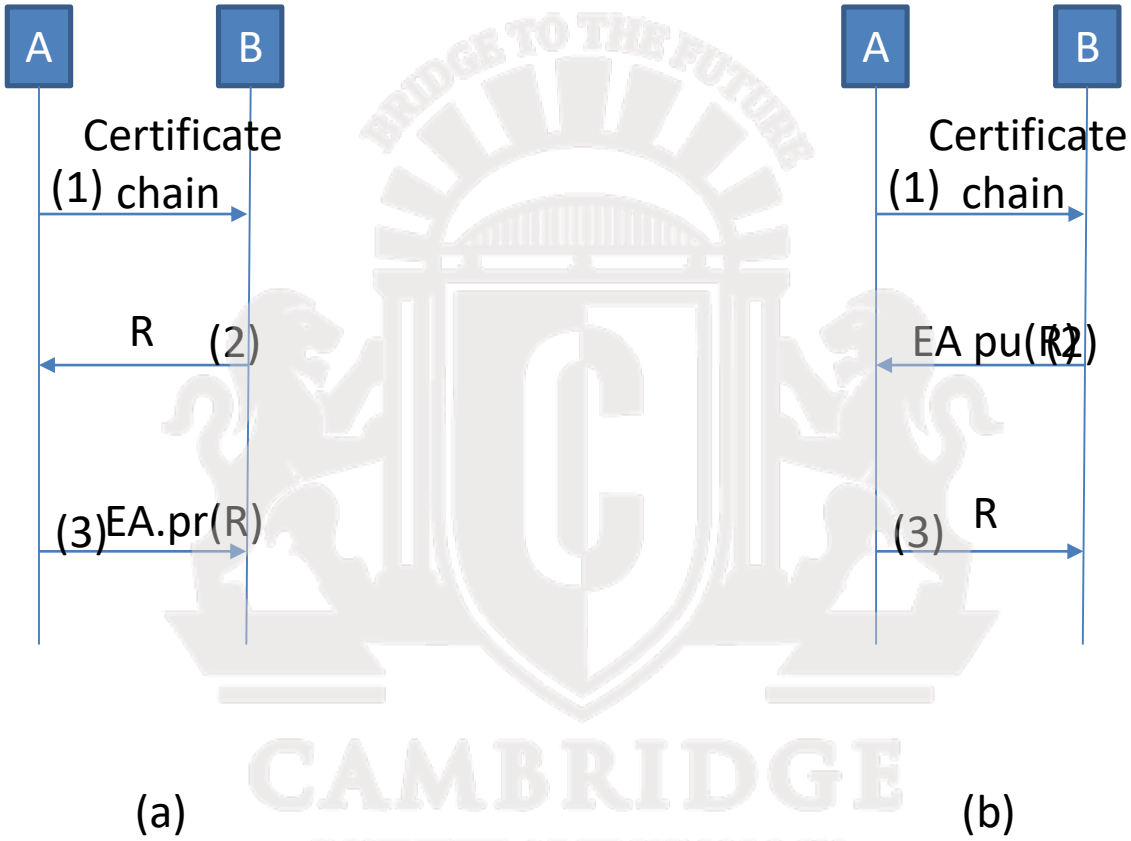


(B) Communicating hash of password

One way authentication using challenge-response protocol

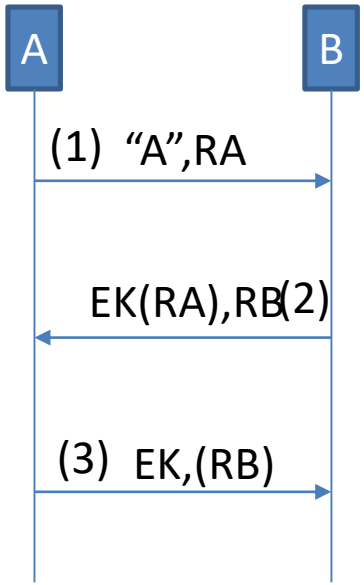


2. Certification –based one way-authentication

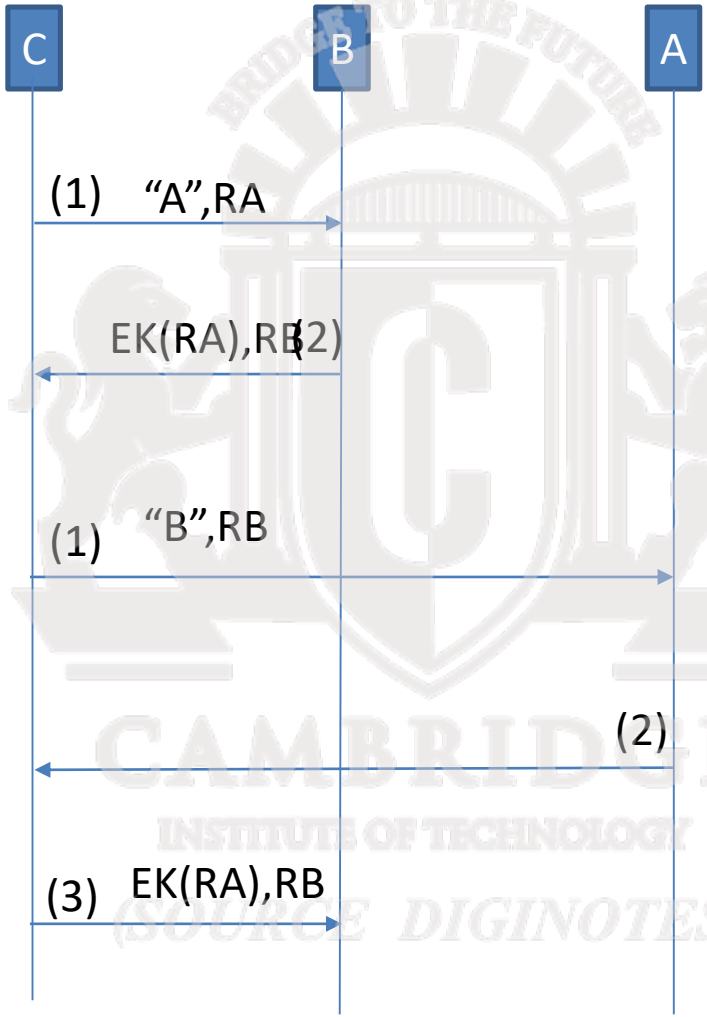


MUTUAL AUTHENTCATION

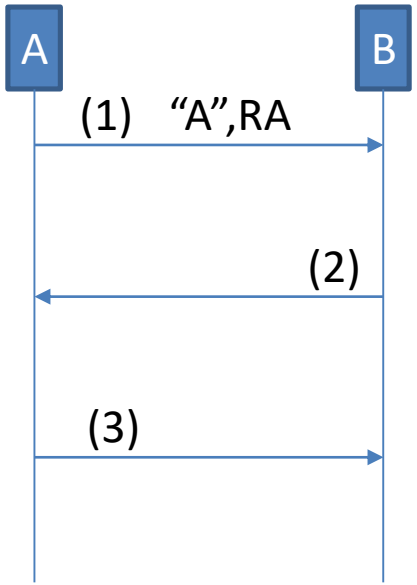
SHARED SECRET-BASED AUTHENTICATION



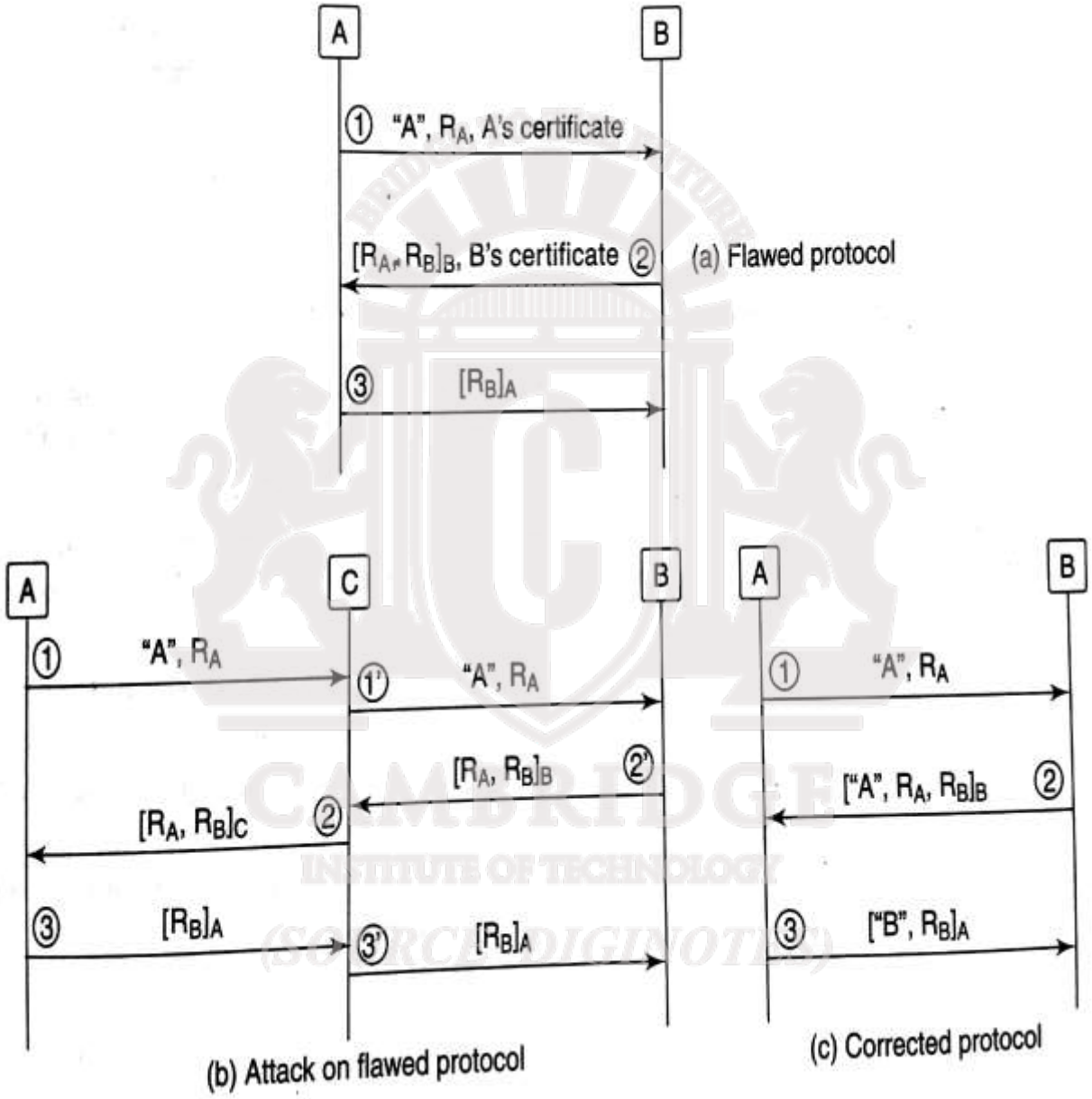
(a) Flawed protocal

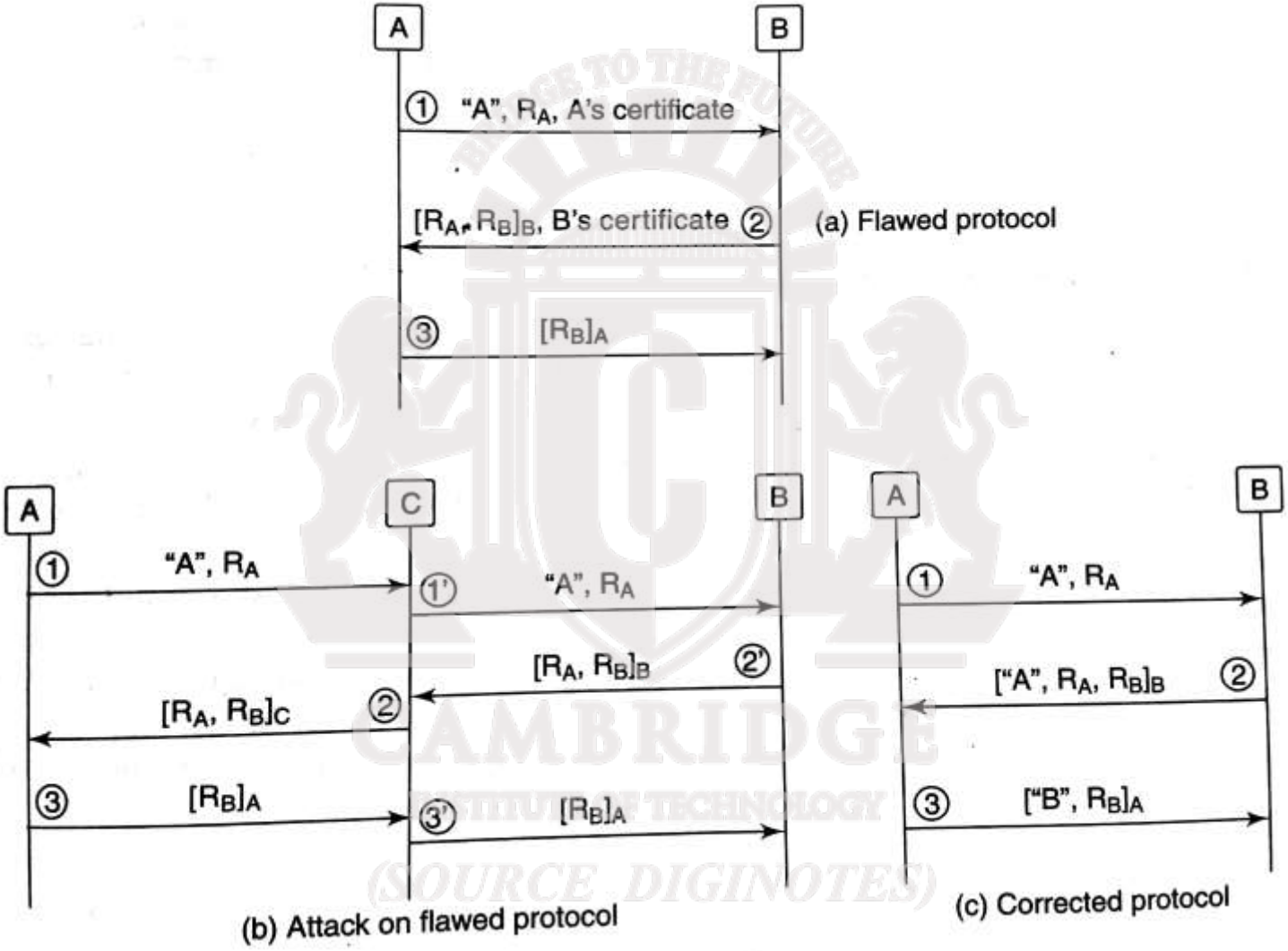


(b)Parallel session attack



(c) Corrected protocal





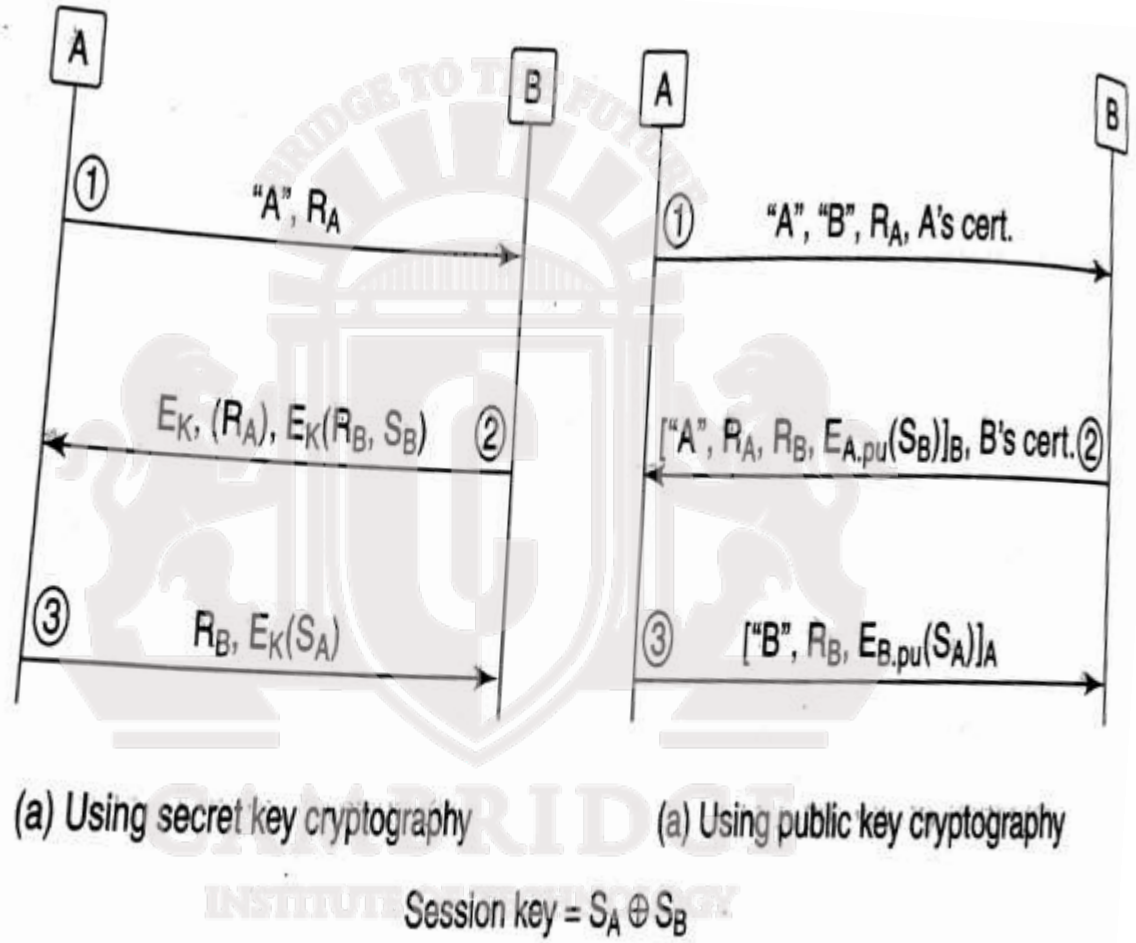


Figure 11.6 *Combined mutual authentication and key exchange*

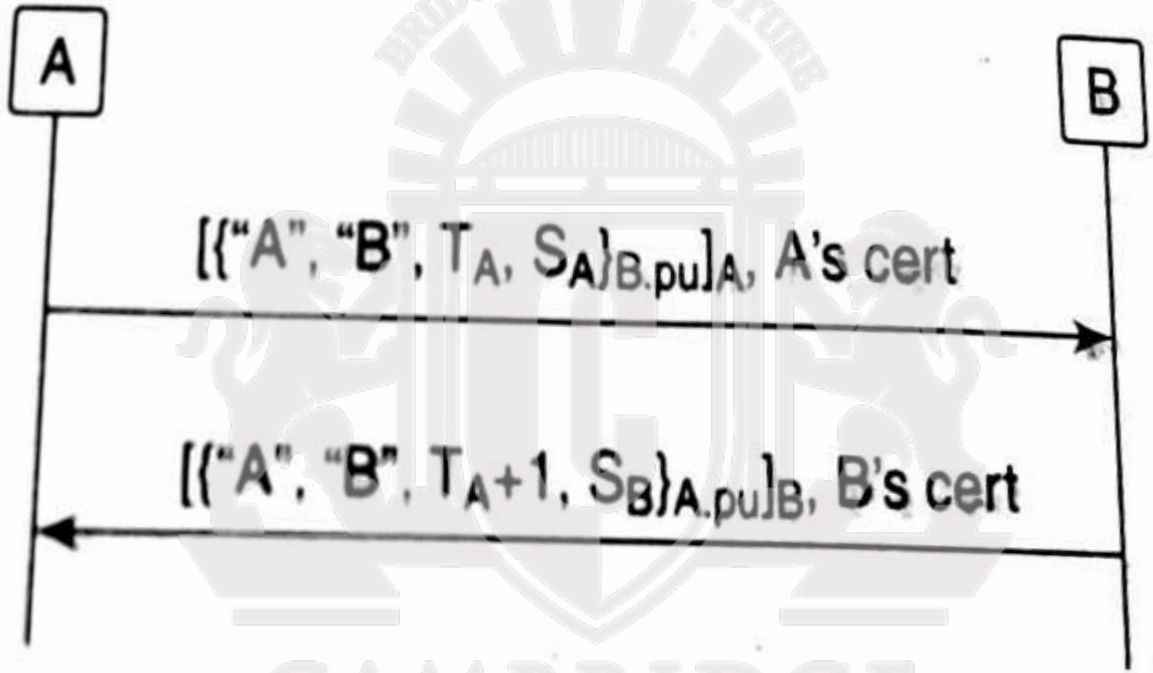


Figure 11.7 *Mutual authentication with timestamps*

Dictionary attacks

1. Attack types

- Two types of dictionary attacks are on-line and off-line.
- In online attacks, an intruder attempts to login to the victim's account by using the victim's login name and a guessed password.
- In online there is a limit on the number of failed login attempts.
- In off-line attack leaves few fingerprints.
- One possibility is the attacker to get a hold of the password file.

Cont…

- Another possibility is for the attacker to eavesdrop on the communication link during client authentication.

```
// let D be an array containing the dictionary
// let F denote f(pw,R) where pw is client's password
// let n be the number of permissible guesses(size of D)
Found=false
i=0
While(~found && i<n)
{
X=f(D[i],R)
If(x==F) {
Print("CORRECT  PASSWORD is D[i]")
Found=true
}
}
```

2. Defeating Dictionary Attacks

- One approach is to increase the cost of performing such an attack.
- The cost is the time to successfully complete the attack.
- The most time consuming operation in each iteration of the dictionary attack program is f(D[i],R).
- Hence to decrease the attacker's chance of success, the function f(D[i],R) could be made more computationally expensive.
- H(……..h(h(D[i],R))….)

A protocol that eliminates off-line dictionary attack is theEncrypted Key Exchange(EKE)

- It is a password-based protocol.
- It combines Diffie-Hellman key exchange with mutual authentication based on a shared secret.
- DHKE is vulnerable to a man-in the middle attack which is due to the unauthenticated exchange of partial secrets g^a mod p and g^b mod p.
- In EKE, each side transmits its partial secret after encrypting it. The encryption key, PW, is the hash of the password.
- Fig shows the 4 messages that are exchanged in EKE.

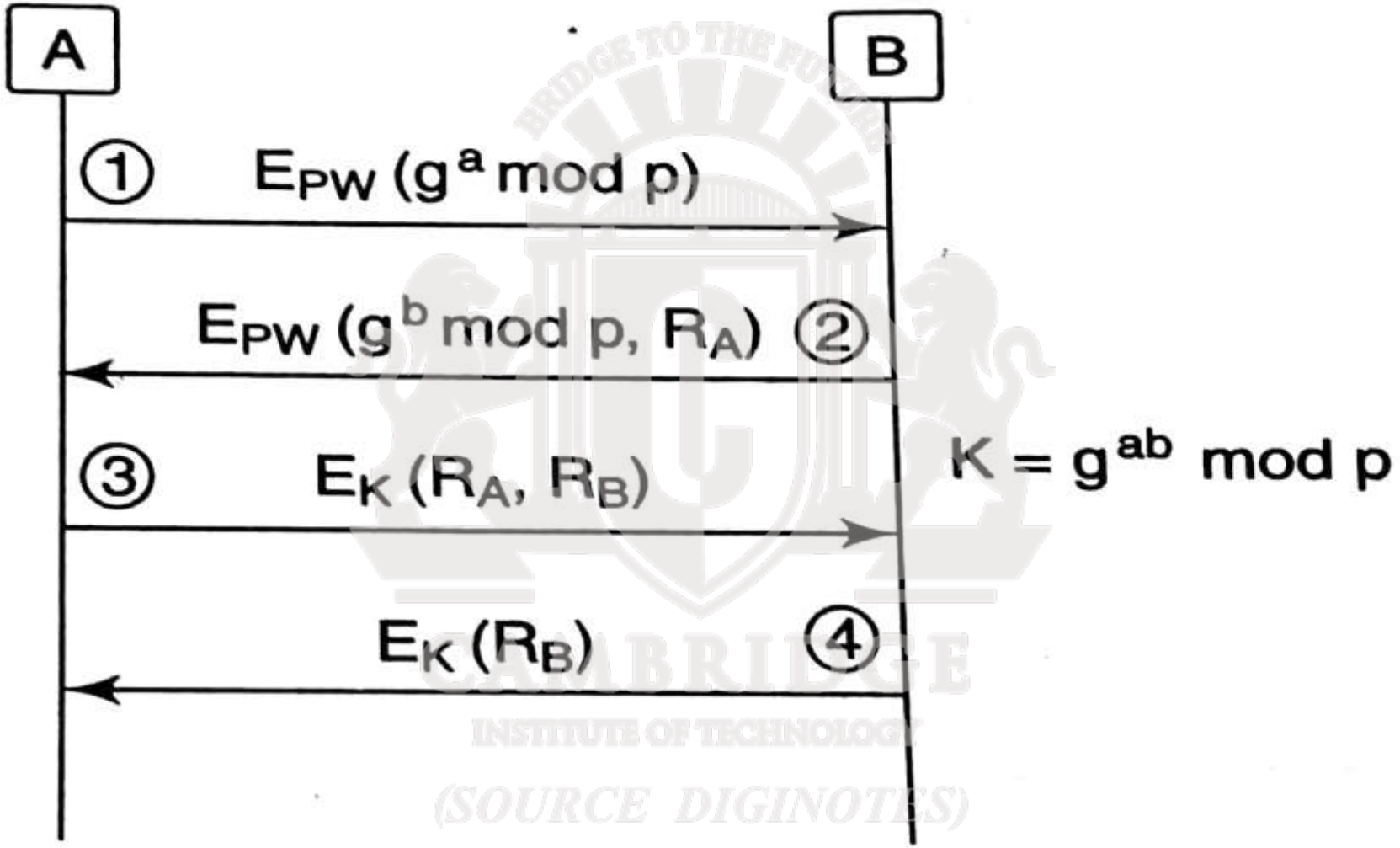


Figure 11.8 *EKE protocol*

AUTHENTICATION –II

Advantages of secret key cryptography over public key cryptography.

- First, DC and PKI are needed in support of public key cryptography. So there is a substantial cost to set up and maintain a PKI.
- Second public/private key operation are relatively slow compared to secret key operations.

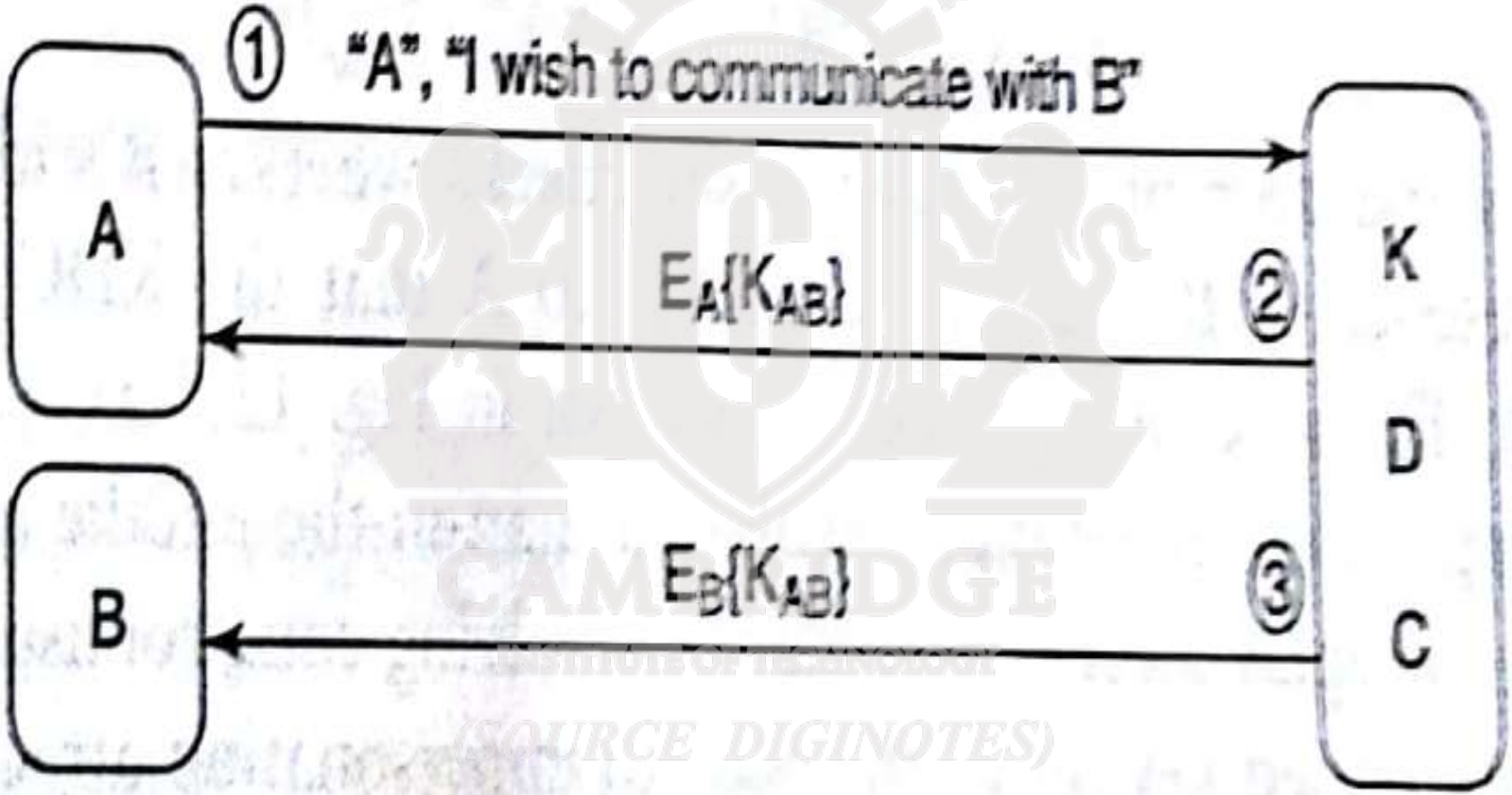
Disadvantages of secret key cryptography

- An entity must share a key with each party it wishes to communicate with.
- Suppose if entity communicates with large number of other entities over time, it must share a secret with each of those parties.
- So managing and securely storing a large number of keys is a non-trival task.

One approach is to use trusted third party

- It function as a key distribution centre(KDC).
- Each user registers with a KDC and chooses a password.
- A long-term secret, which is a function of the password, is to be exclusively shared by that user and the KDC .
- The main function of the KDC is to securely communicate a fresh, common session key to the two parties who wish to communicate with each other.

Message confidentiality using a KDC



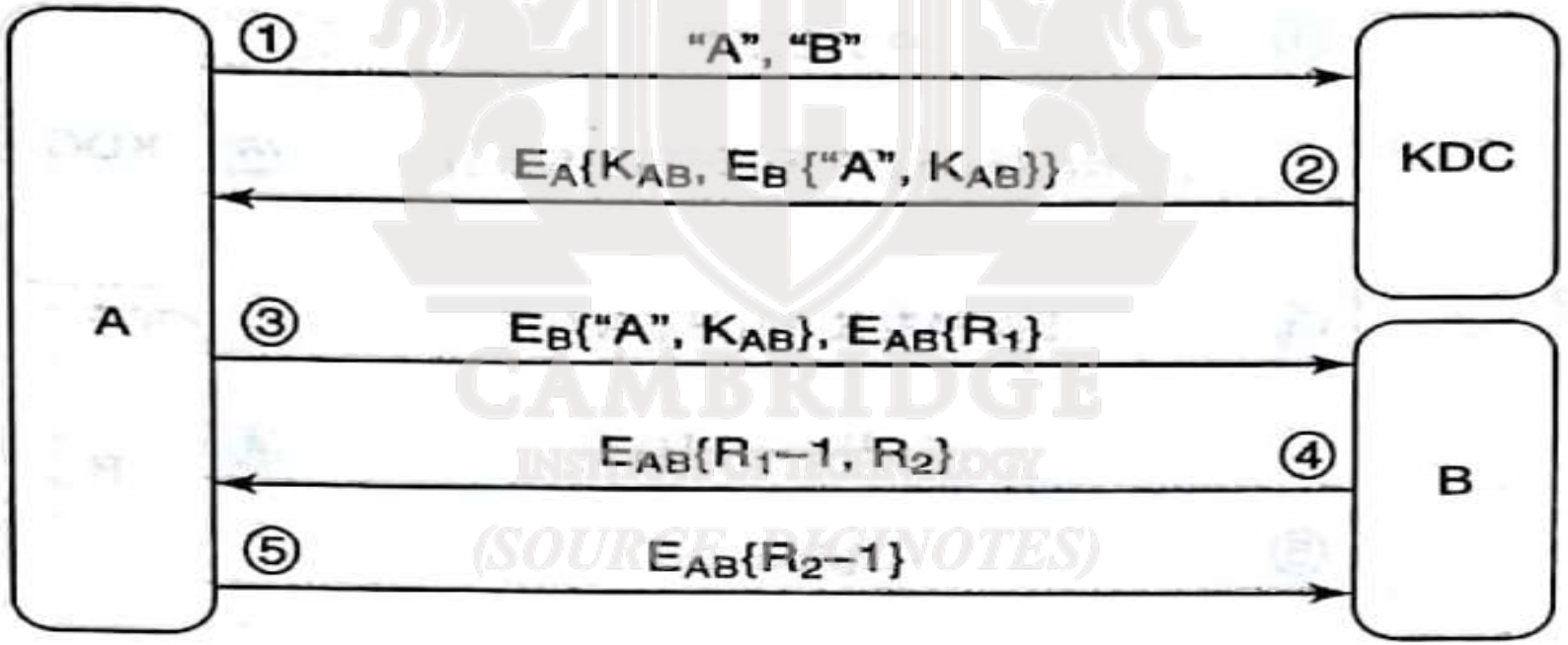
Source diginotes.in

The Needham-Schroeder protocol

- In this protocol, both sides proceed to challenge the other to prove knowledge of the session key.
- The challenge is a nonce.
- The response involves decrementing the nonce and encrypting the nonce with the session key.
- MSG1:A informs the KDC that it intends to communicate with B.
- MSG2:KDC dispatches session key and the ticket to B[Encrypted with long term key shared b/w B & KDC] in its msg to A[Encrypted with long term key shared b/w A & KDC].
- MSG3:A then forwards the ticket together with her challenge to B.
- MSG4:B response involves decrementing the nonce and new challenge to A, both encrypted using a session key.
- MSG5: A response to B by decrementing the nonce encryptrd using a session key.

The Needham-Schroeder protocol

- Provide mutual authentication by including a challenge-response phase.

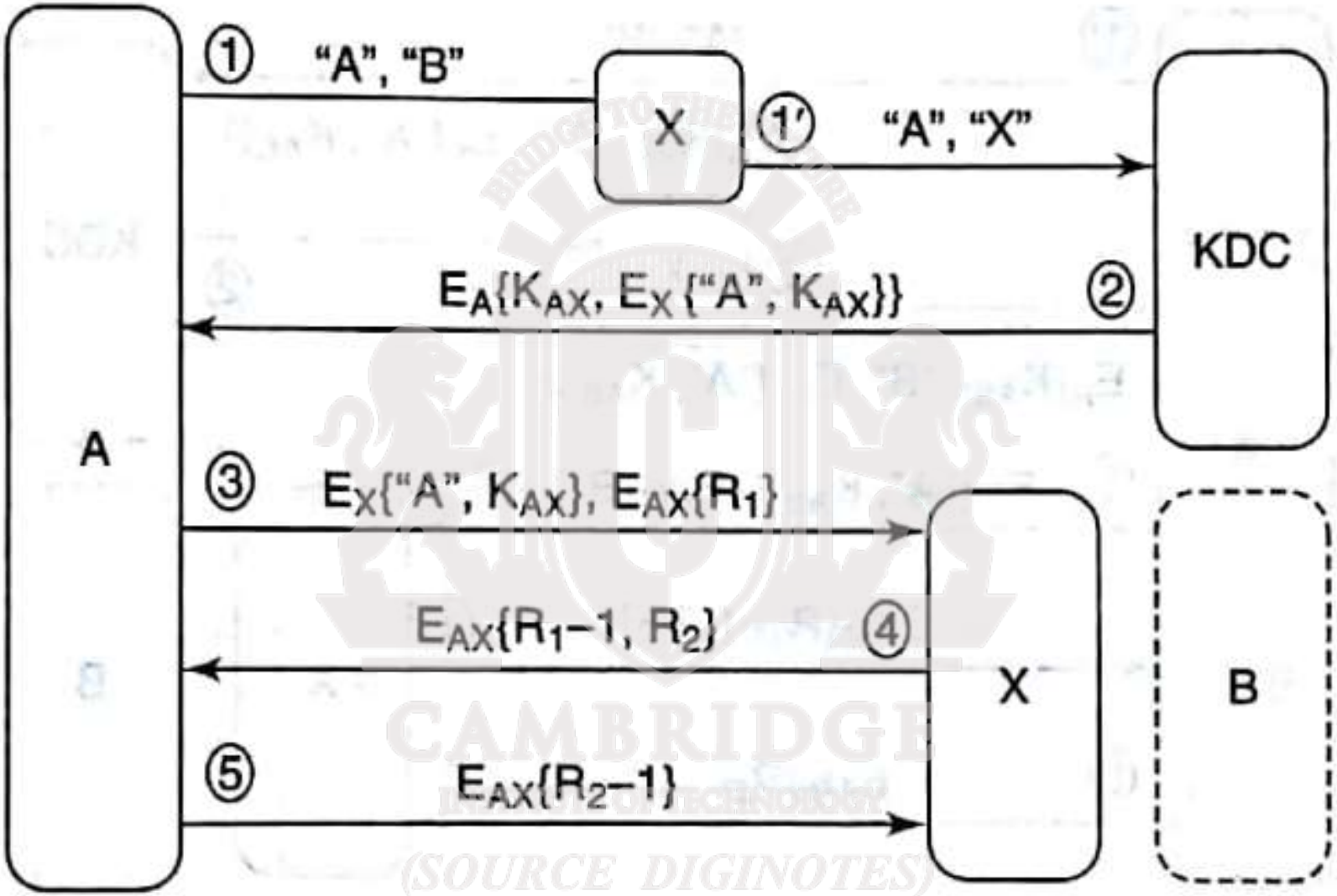


(a) : Preliminary version 1

Source diginotes.in

Man-in-the middle attack on preliminary version1

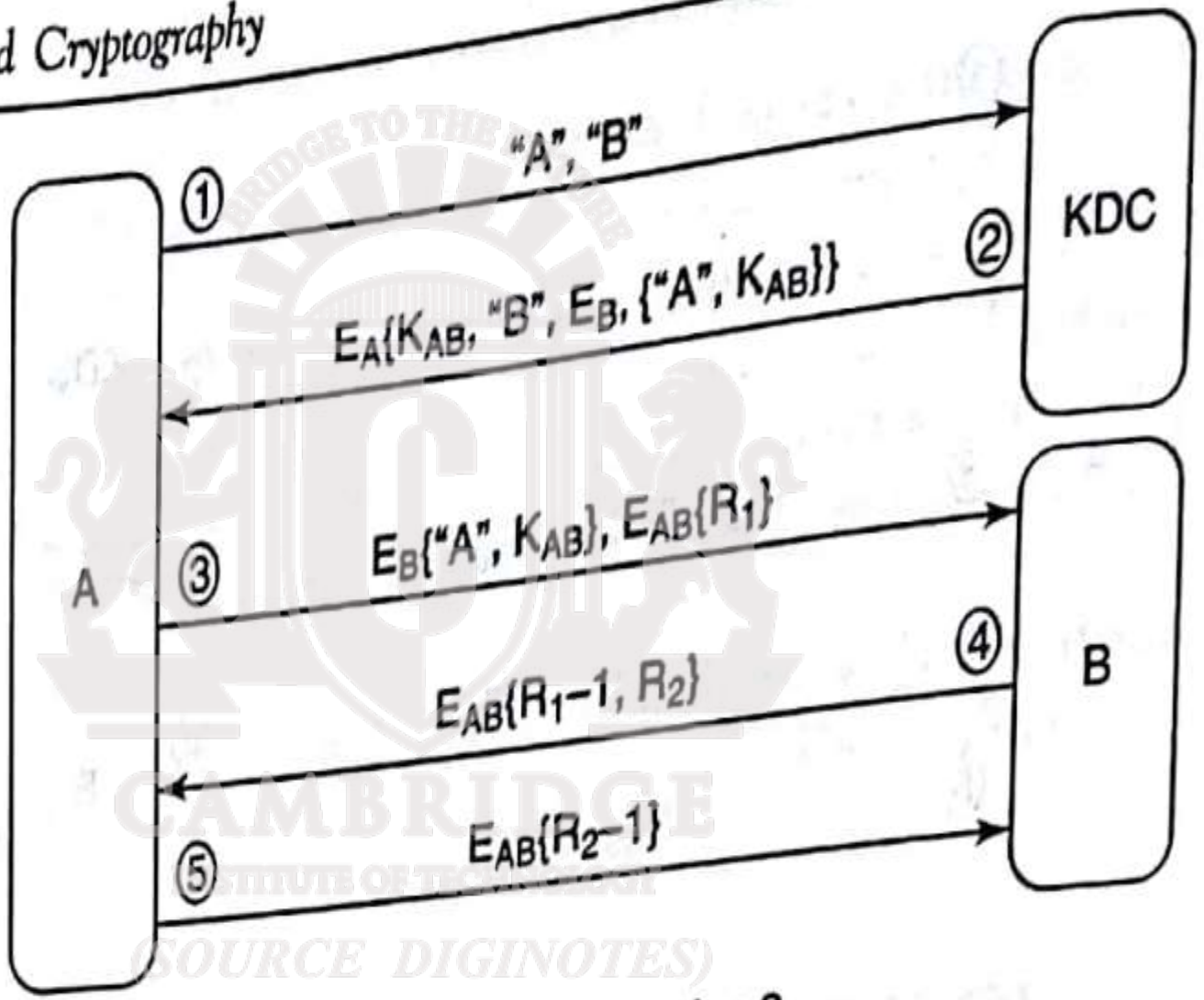
- The attacker, X, is an insider who shares a long-term key with the KDC.
- The attacker , X, intercepts MSG1, substitutes B for X and sends the modified msg to the KDC.
- In response, the KDC creates a ticket encrypted with X's long-term key and send it to A.
- Now X intercepts MSG3.He decrypts the ticket using the long term secret he shares with the KDC. He thus obtains the session key.
- MSG 3 also contains A's challenge R1.X uses the session key to decrypt the part of the msg containing A's challenge. He successfully responds to A's challenge in MSG 4.
- Thus, X successfully impersonates B to A.



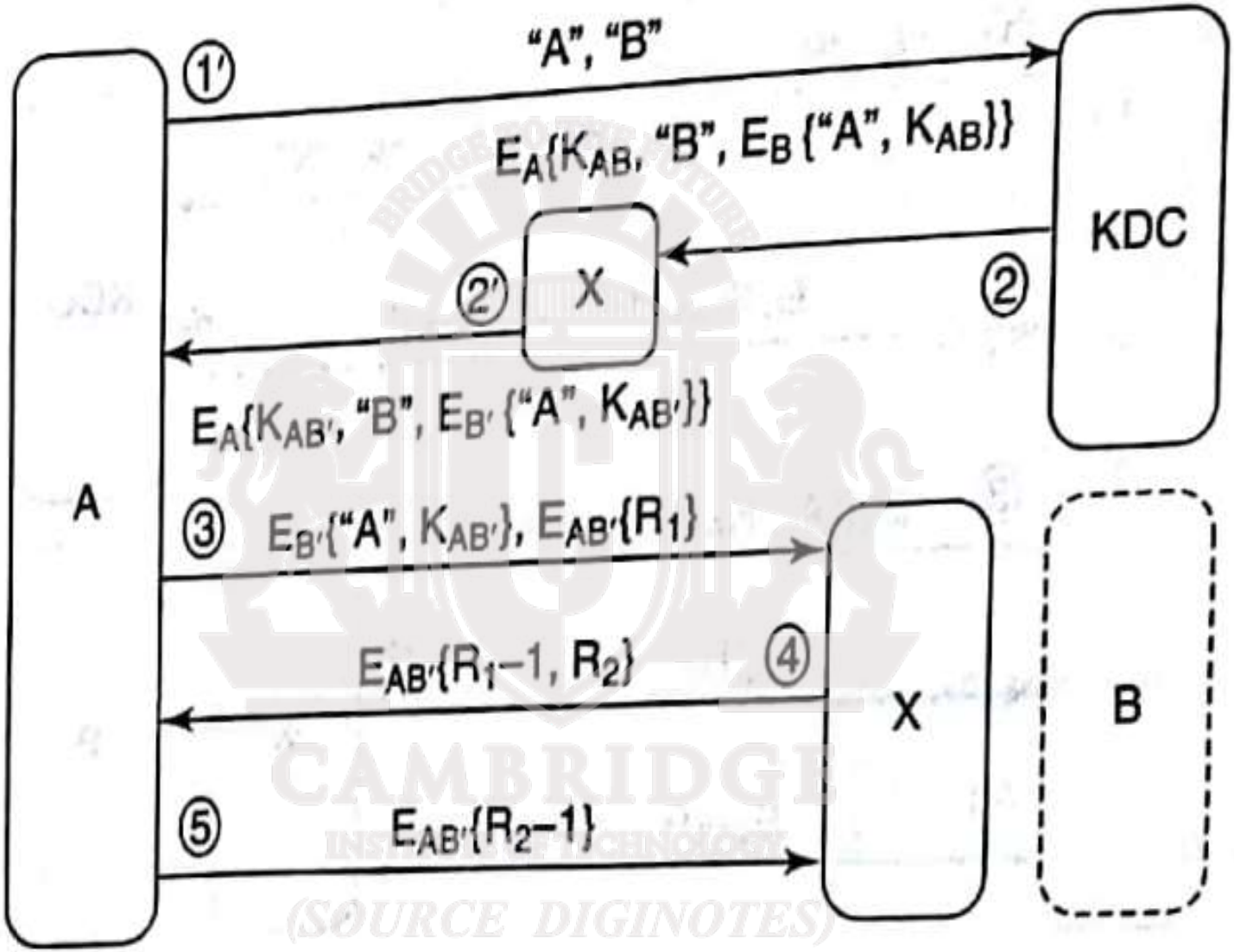
(b) : Man-in-the middle attack on preliminary version 1

Preliminary Version 2

- Solution to previous problem is to include B's identity in the encrypted message from the KDC to A in MSG 2.
- Now, after A receives and decrypts MSG2, she checks whether B's identity is contained inside the msg.
- The presence of B's identity confirms to A that the KDC knows that A wishes to communicate with B.



(a) Preliminary version 2



(b) Man-in-the middle and replay attack on preliminary version 2

A determined attacker X does the following:

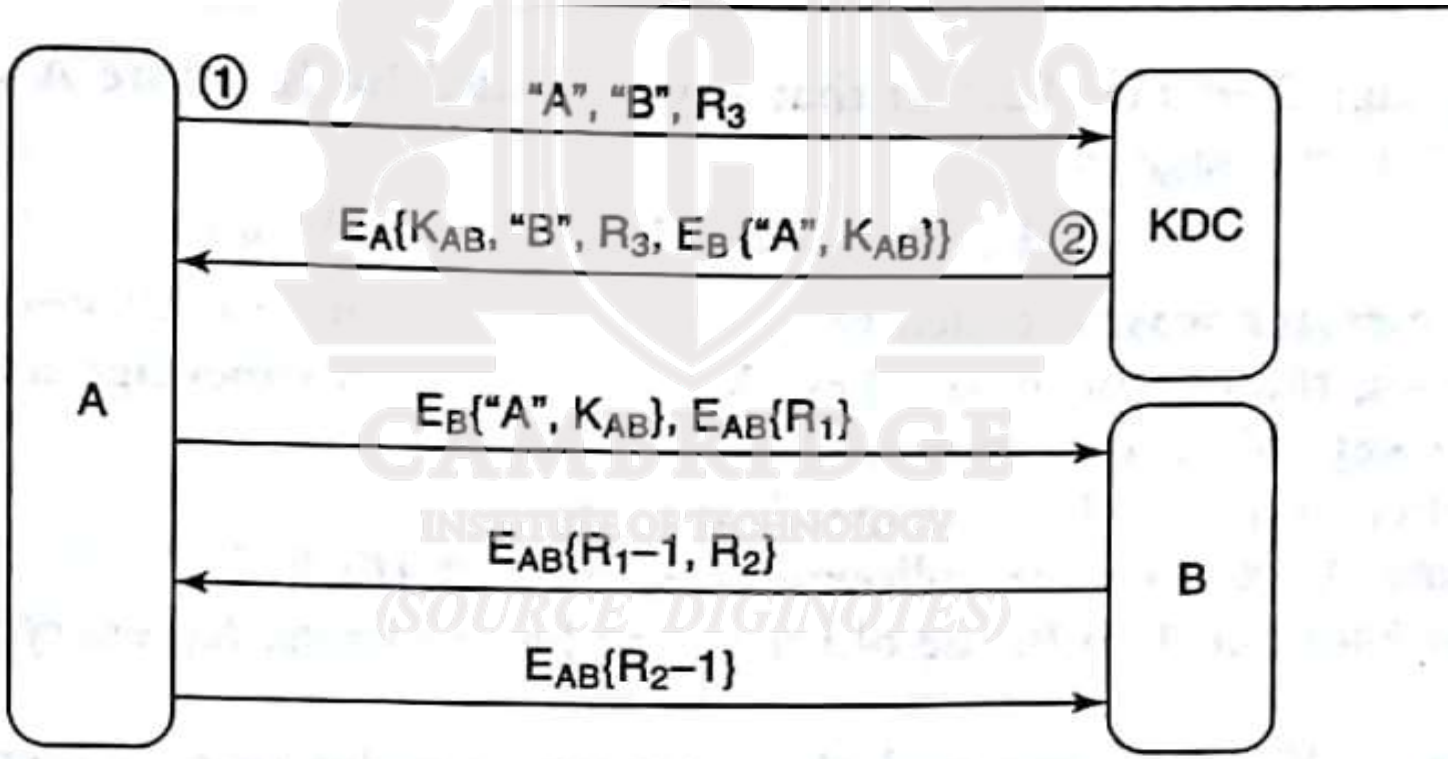
- X eavesdrops upon and meticulously records many of A's sessions with the KDC and with B over a period of time.
- He then steals B's password or long-term key.
- B recognizes that his password has been stolen and immediately reports the incident to the KDC. He obtains a new long-term key which he uses subsequently.

The following scenario shows X successfully impersonating B to A.

- A wishes to communicate with B and sends MSG1
- X intercepts the KDC's response(MSG2) and instead plays a previous recording of MSG2. X is careful to replay a copy of MSG2, which he recorded before B's key was compromised(contains a ticket encrypted with B's old key.
- X then intercepts MSG3 from A, which contains the old ticket and a fresh challenge to B. X has B's old key, he can decrypt this ticket and recover the session key.
- X knows session key , he can respond to A's challenge in MSG4.
- X's response is exactly what A expected to receive from B. Hence A is convinced that she is talking to B.

Preliminary Version 3

- Previous problem solved by ensuring the freshness of MSG2.
- A sending a nonce in MSG1 and receiving confirmation of its receipt by the KDC.



(a) : Preliminary version 3

- X could still attack the protocol by recording previous messages and selectively replaying them when the right opportunity presents itself.
- He attempts to steal A's password or long-term key and success in it.
- MSG2 was recorded by X before A's key was compromised.

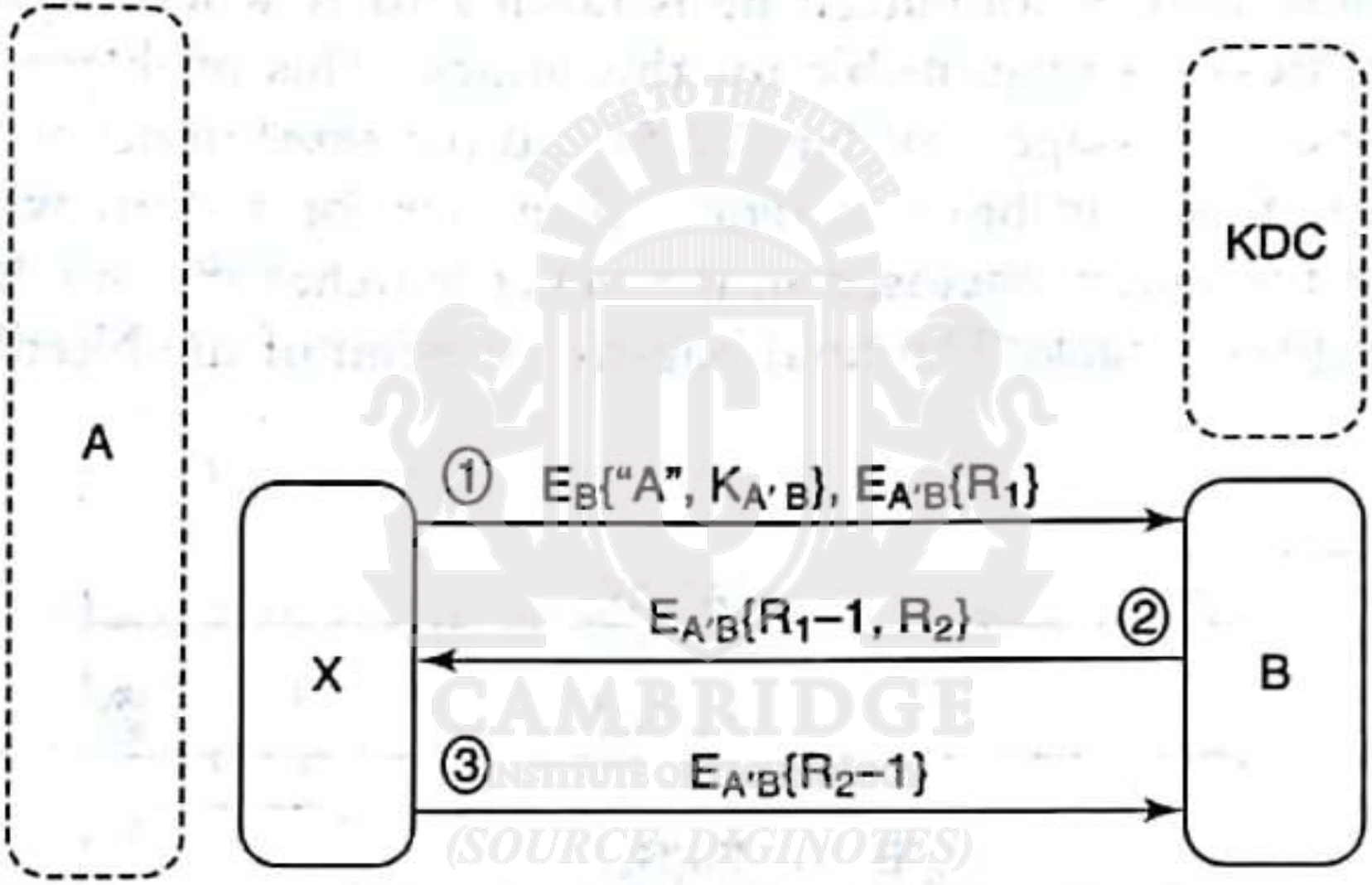
Using the compromised key, X can decrypt this msg and recover the

- Old session key used then and the old ticket dispatched to B.

To impersonate A, X does the following:

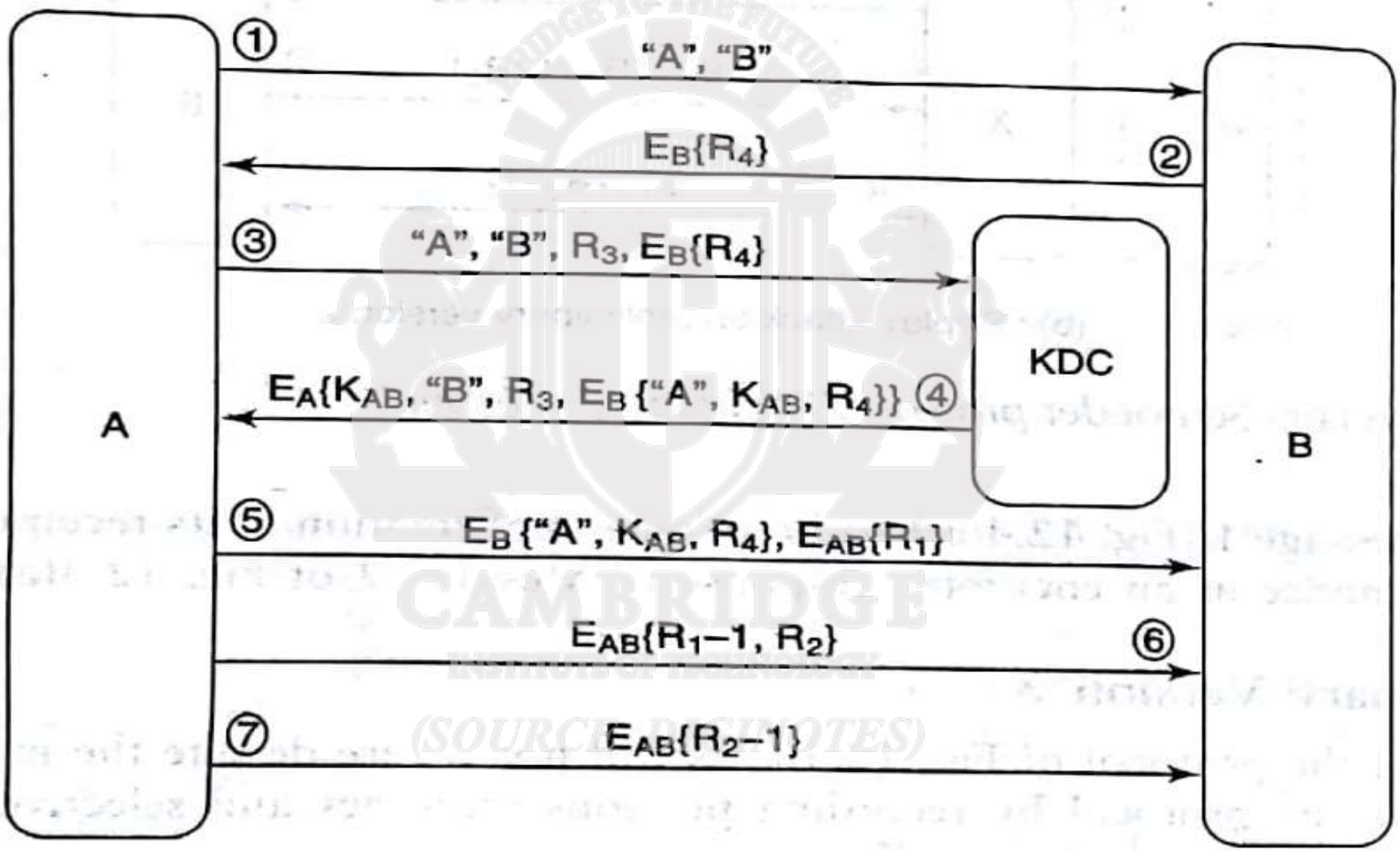
- X sends, in MSG1 to B, the old ticket and a challenge R1, encrypted with the old session key.
- B responds to X's challenge and also communicates his own challenge,R2.
- Because X has the session key, he responds to the challenge by encrypting R2 with the old session key.

B receives the response and is convinced he is talking to A but he is talking to X.



(b) : Replay attack on preliminary version 3

Needham-Schroeder protocol: Final Version



Needham–Schroeder protocol: Final version

KERBEROS

- A scenario with multiple users and multiple servers in an organization.
- A user, once logged in, may then wish to access different resources such as e-mail or a file server in the course of that login session.
- One possibility is for the user to have multiple passwords on each of these servers.
- Humans remember and update multiple passwords is not practical.
- A user could use the same password for all servers but distributing and maintaining a password file across multiple servers is a security risk.

A password-based system should ensure the following:

- The password should not be transmitted in the clear.
- It should not be possible to launch dictionary attacks using the eavesdropped-upon messages containing a function of the password.
- The password itself should not be stored on the authentication server, rather it should be cryptographically transformed before being stored.
- A user enters her password only ONCE during login. Thereafter, she should not have to renter her password to access other servers for the duration of the session. This feature is called single sign-on.
- The password should reside on a machine for only few milliseconds after being entered by the user.

- The KDC is logically split into two entities here- the authentication server(AS) and the Ticket Granting Server(TGS).

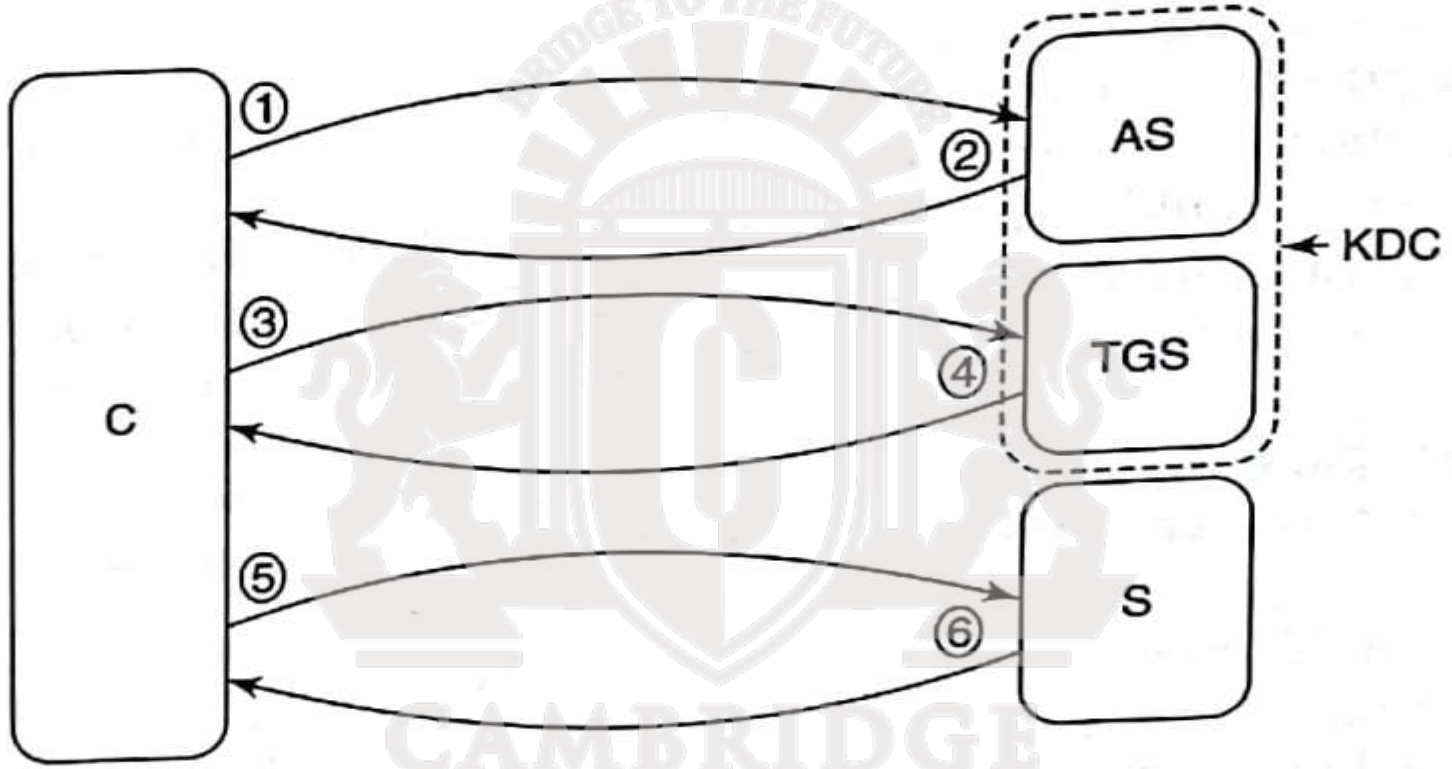
- The Ticket is the mechanism used to safely distribute session keys.

- User A shares a secret Ka with the AS.

- Each server, B shares a secret Kb with the TGS.

- Kerberos also makes use of timestamps.

Kerberos message sequence



① C request Ticket-Granting Ticket

② C receives Ticket-Granting Ticket

③ C request Service-Granting Ticket

④ C receives Service-Granting Ticket and session key

⑤ C authenticates itself to S

⑥ S authenticates itself to C

Kerberos message sequence Source diginotes.in

BIOMETRICS

- A biometric is a biological feature or characteristic of a person that uniquely identifies him/her over his/her lifetime.
- Common forms of biometric identification include face recognition, voice recognition, manual signatures and fingerprints.
- More recently, patterns in the iris of the human eye and DNA have been used.
- Biometric forms were first proposed as an alternative or a complement to passwords.
- Passwords are based on what a user knows and are based on what a person has.
- A biometric, on the other hand, links the identity of a person to his/her physiological or behavioural characteristics.

The two main processes involved in a biometric system are:

- **Enrolment:** A subject's biometric sample is acquired. The essential features of the sample are extracted to create a reference template. Sometimes multiple samples are taken and multiple templates are stored to increase the accuracy of a match in the subsequent recognition phase.

- **Recognition:** A fresh biometric sample of the person is obtained. This is then compared with the reference templates (created during enrolment) to determine the extent of a match.

Biometrics is used in at least two different situations:

Authentication or Identity verification:

- A biometric systems stores login name and biometric sample pairs.
- During a login attempt, a biometric sample (such as a fingerprint scan) of the user is taken.
- The biometric sample is compared with the sample stored on the server.
- The user is authenticated only if a match between the two occurs.

Identification

- Subject's identity is not presumed to be known beforehand.
- It is assumed that a database of biometric samples of several users already exists.
- The subject's biometric sample is compared with the samples in the database to determine if a match exists with any one of them.

- Authentication involves a one-to-one match, identification involves a one-to-many match.

A characteristics of a good biometric include the following:

- Universality: All humans should be able to contribute a sample of the biometric.
- Uniqueness: biological samples taken from two different humans should be sufficiently different that they can be distinguished by machine intelligence.
- Permanence: The biometric should not change over time

KEY MANAGEMENT

* Key management is related to the generation, storage, distribution and backup of keys.
* Public key-private key pairs are used for encryption, decryption, signature generation/verification and for authentication.
* TO encrypt a session key for use in communication between A and B, A needs to know B's public key.
* TO verify B's Signature on a msg, A needs B's public key
* The key issue here is "How does A know B's public key?"

Possibility 1: A may frequently communicate with B in a secure fashion, so she may already have B's public key.

Possibility 2: Every entity's public key is securely maintained in a centralized directory.

Possibility 3: A receives a document signed by a trusted source C, containing B's public key.

DIGITAL CERTIFICATES.

1. Certificate types

* A digital certificate is a signed document used to bind a public key to the identity of a person.
* The entity that issues certificates is a trusted entity called a certification Authority (CA).
* The CA may have to obtain and verify several details of the applicant including his/her employ e-mail address etc. practically speaking, this task would be delegated by the CA to a Registration Authority.

2. X.509 Digital certificate format.

* certificate serial number and version: Each certificate issued by a given CA will have a unique no
* Issuer Information: The distinguished name of an entity includes his/her/its "common name", email address, organization, country etc.
* Subject information: It includes the name of the certificates owner, other information, such as the Subject's country, State & organization may be included.
* Subject public key information: The public key, the public key algorithm and the public key parameters.

* validity period: There are two date fields that specify the start date and end date b/w which the certificate is valid.

* Certificate signature and associated signing algorithm information: It is necessary to verify the authenticity of the certificate. For this purpose, it is signed by the issuer. So the certificate should include the issuer's digital signature and also the algorithm used for signing the certificate.

3. Digital certificate in Action

* Assume that A needs to securely transmit a session key to B. So she encrypts it with B's public key. A will need to retrieve the public key from B's certificate.

* A may already have B's certificate or she may send a msg to B requesting it.

* There are no of checks that A ll have to perform on B's certificate prior to using B's public key.

1. Is this indeed B's certificate?

2. A should check if the certificate is still valid.

3. The certificate must be signed by a CA or RA.

PUBLIC KEY INFRASTRUCTURE.

1. Functions of PKI.

* public key Infrastructure includes
  a. certificate creation, issuance, storage.
  b. key Generation (if necessary)
  c. certificate/key updation (if necessary)
  d. certificate revocation.

2. PKI Architecture.

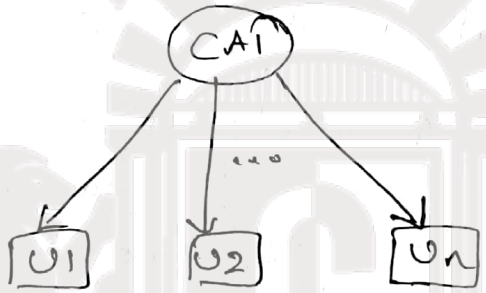
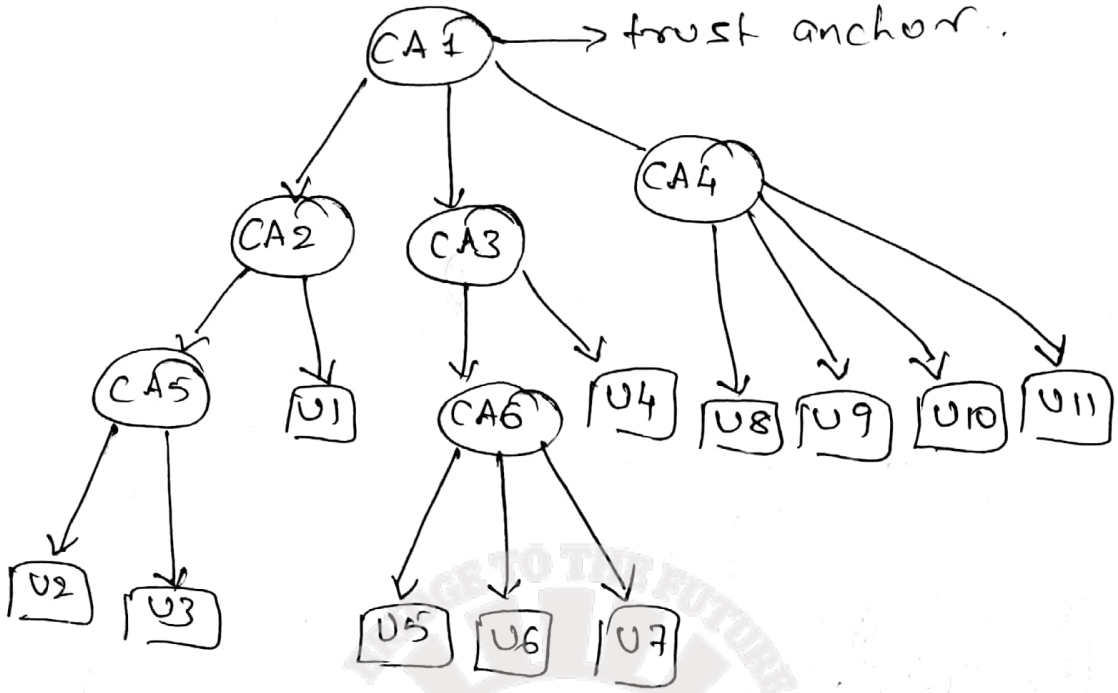


fig: PKI with single CA

* CA1 could issue certificates to multiple users U1, U2, etc. enabling any pair of these users to communicate securely using certificates exchanged b/w them.
* This architecture, however is not scalable.
* Suppose if there are tens of millions of users who may need certificates. It is not practical for CA1 to issue certificate to them all

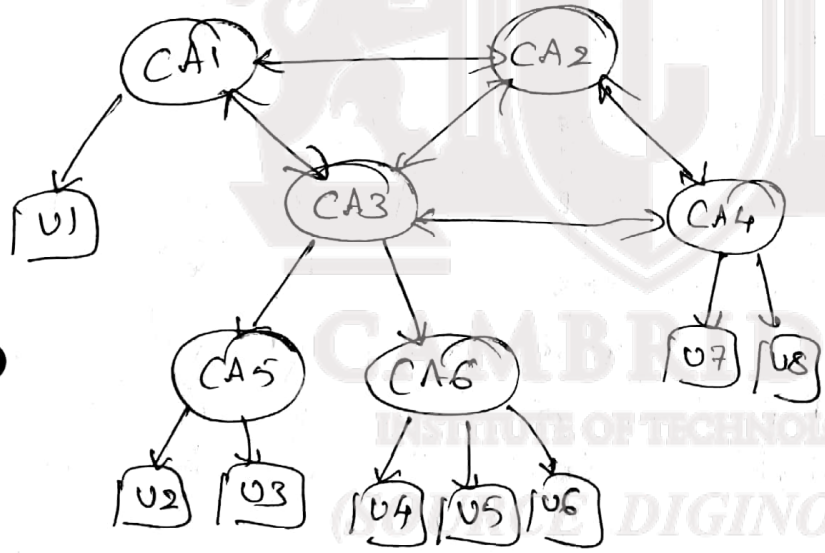
---

* A practical solution to the problem of scalability is to have CA1 certify other CAs who in turn certify other CAs & so on.
* This creates a tree of CAs known as a hierarchical PKI architecture.



- Fig:- Hierarchical (tree-based) PKI architecture.

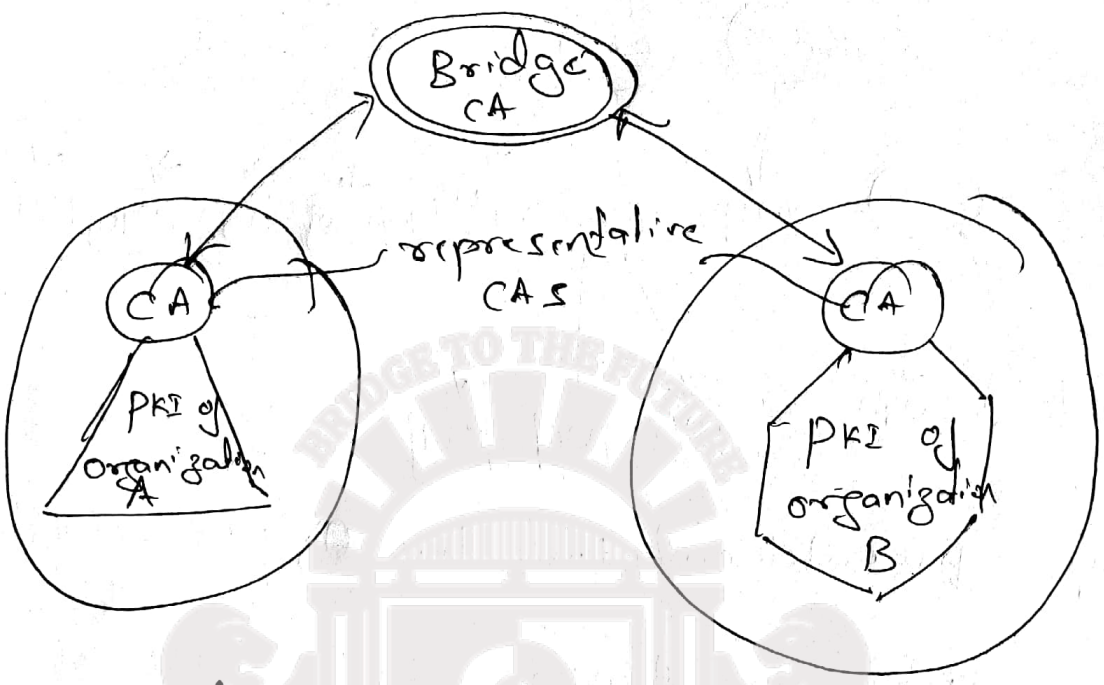
Fig:- Mesh-based PKI



- * This include mutually trusting CAs - CA1 trusting CA2 & CA2 trusting CA1 depicted by a bidirectional arc b/w CA1 & CA2.
- * There may be multiple trust paths b/w 2 users
- * one trust path b/w user U1 & U7 passes through CA1, CA3 & CA4
- * Another trust path involves CA1, CA2 & CA4.
- * Multiple paths provide greater resilience in the

event of one or more CAs being compromised.

Fig:- Bridge-based PKI



* Motivated by the need for secure communication b/w organization in a business partnership.
* Suppose that the partnering organization already have their own PKIs, A bridge CA is introduced that establishes a trust relationship with a representative CA from each organization. This is accomplished by the bridge CA & the organizational representatives issuing certificates to each other.

3. Certificate Revocation

a. Revocation Scenarios.

Scenario 1: The certificate's subject, prashant was issued a certificate valid b/w Jan 01, 2010, & Dec 31, 2010. However, he left the organization

on April 1, 2010. Assume that prashant's certificate is to be used for key exchange and that he has made a copy of it.

* note that the public key in a key exchange certificate is used by another party to encrypt a random session key. The session key itself is then used to encrypt all msges in both directions for the duation of the ensuingsession.

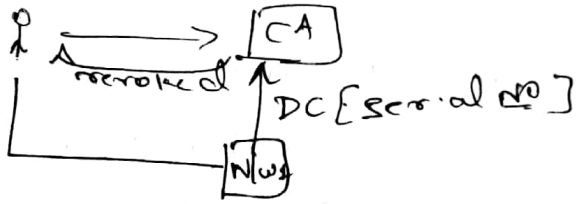
* It is not legal for prashant to act on behalf of his company beyond the date of his resignation.

Scenario 2:

* suppose that the private key of CA3 were compromised. An attacker with access to the compromised private key could then do the following.

->Generate a public key, private key pair (x, y)
->Create a certificate containing the public key
  * with subject name = U,
->Sign the above certificate using the compromised private key of CA3

Handling Revocation

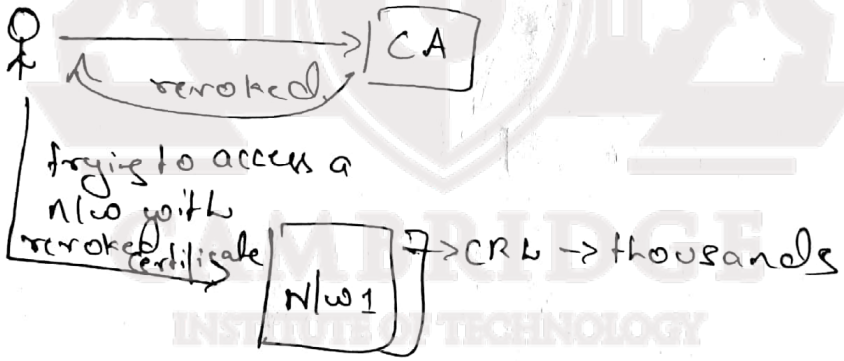


Solution 1:

* It is to use an on-line facility that provides information on the current status of digital certificates
* for this purpose, a protocol called on-line certificate status protocol is employed
* Browser sends D.c to CA for status update.

Solution 2:

* certificate Revocation lists (CRL)
* If CRLs are distributed too frequently, they could consume considerable bandwidth.
* CRL contains lists of all revoked certificates.



Authentication - I.

* Authentication is a process in which a principal proves that he/she/it is the entity it claims to be.
* The principal is referred to as the prover, while the party to whom proof is submitted for identity verification is called the verifier.

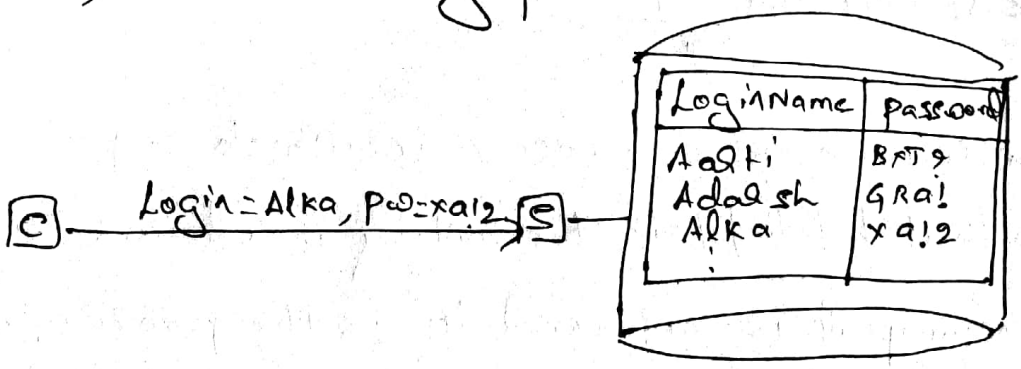
ONE - WAY AUTHENTICATION

- In client - server communication, the client authenticates itself to the server. The server may or may not be authenticated to the client.

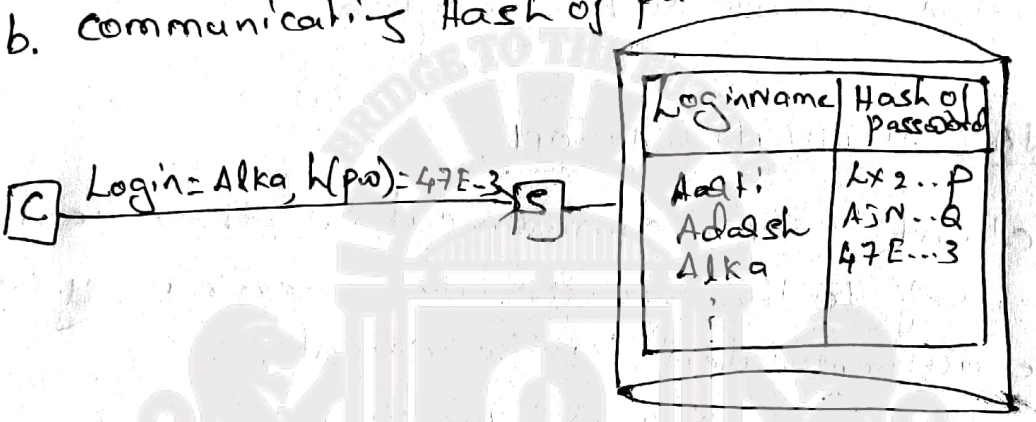
1. Password - based Authentication

* The common mechanisms to implement authentication is the password.
* To login to a server, a user enters his/her
- login name and password.
* The password is the secret i.e known only to the user and server
* The login name identifies a user, while the user's knowledge of the corresponding password constitutes proof that he/she is the person with the given login name.

Fig: a) communicating password.



b. communicating Hash of password.



* Two danger associated with such an implementation
* First, the password is sent in the clear, so an attacker can eavesdrop on the msg containing the password and later impersonate the realuser
* Second, the passwords are stored in unencrypted form in a file on the server. If an internal attacker obtains access to that file, all passwords stored on that server could get compromised.

* Solution is the cryptographic hash of the password rather than the password itself is stored on the server.
* The one-way property of the cryptographic hash helps prevent an attacker from deducing user passwords from information in the password file or from communications on the transmission line.

* However, an attacker could snoop on the communications b/w Alka & the Server and obtain the hash of the password. He can, at a later point in time, replay it to the Server thus impersonating Alka. Such an attack in which one play back all or a part of one or more previous msgs with the intent of impersonating a legitimate user, is referred to as a replay attack.

The solution to replay attack is for the verifier to offer a fresh challenge to the prover. In response, the client does not communicate its password but rather proves that it knows the password. The Server is thus able to verify whether the client is genuine or not. Such an authentication protocol is commonly referred to as a challenge-Response protocol.

* Fig shows a three-msg one-way authentication protocol.
* In the first msg, A conveys its identity. The second msg contains the challenge from the Server. The challenge is a random number called a nonce. The third msg is the client's response- a cleverly chosen function of the challenge & the password.
* The function, $f(pw, R)$ has the following properties:

* Given x & y, it should be easy to compute $f(x,y)$.
* f is one-way; so knowing $f(pw, R)$ & R, it should be infeasible to compute pw.
* Given an R, it should be infeasible to compute $f(pw, R)$ even if one knows
  - $f(pw, R_1), f(pw, R_2), f(pw, R_3) \ldots$
  - the corresponding $R_1, R_2, R_3 \ldots$

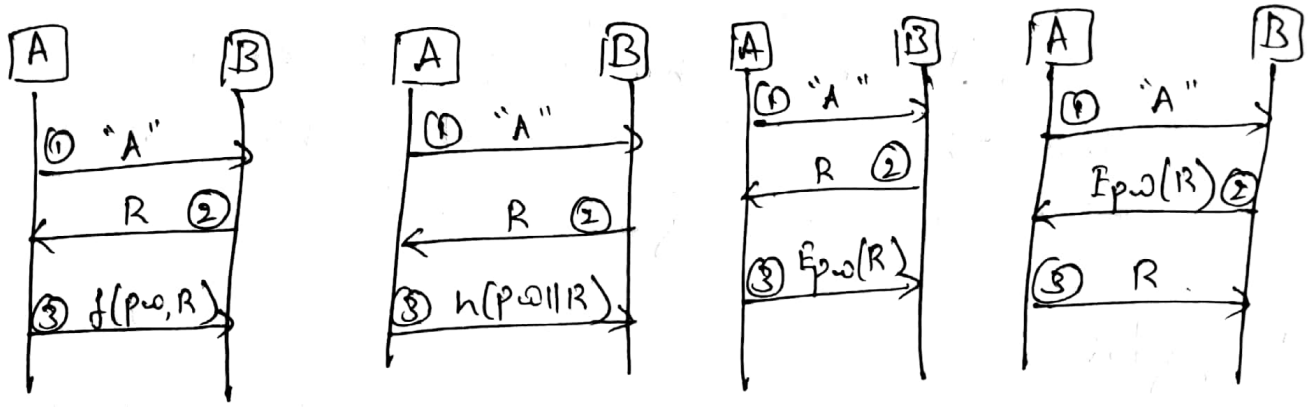
* Fig b: Another choice for f is the cryptographic hash, which is applied over the concatenation of the password and the nonce.

* Fig c: Another choice is a secret key encryption function with the key being the password or a function of the password

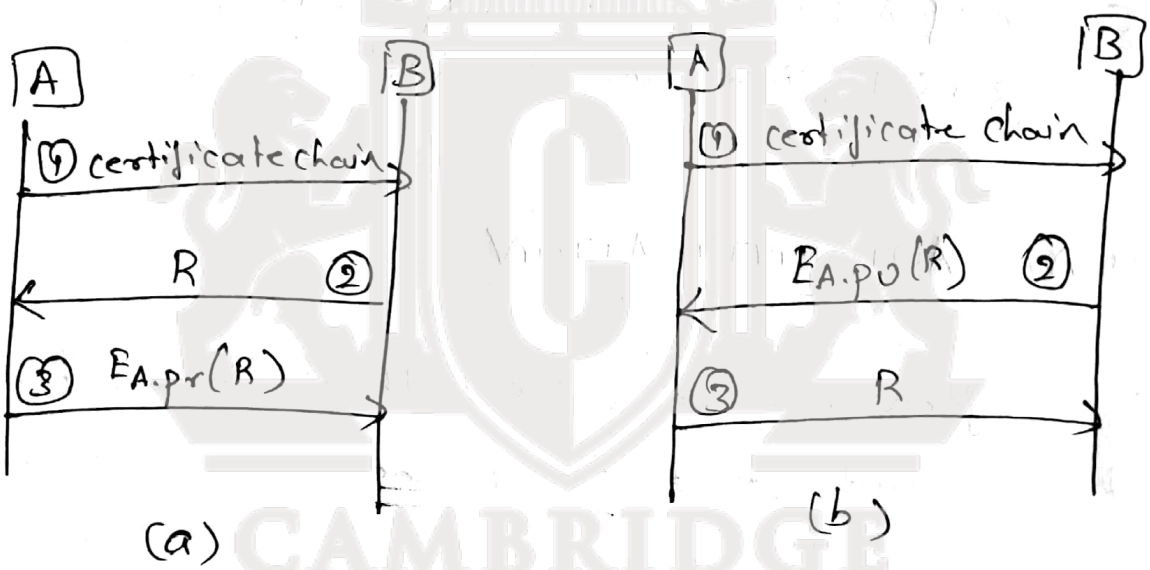
* Fig d: the challenge sent by the server is an encrypted nonce. so the function f is the decryption function the client would need to decrypt the challenge to obtain the nonce and return it to the sender to prove knowledge of his/her password

NONCE : * Nonces are random and nonrecalling.
* Nonce means used only once.
* The size of a nonce is usually large. This provides a large space from which a nonce may be selected
* The large space of nonces means that the probability of choosing the same nonce twice is infinitesimally small.

Fig a: Fig b: Fig c: fig d:



2. CERTIFICATE - BASED AUTHENTICATION.



* Fig a:
  * A sends her certificate in Msg 1.
  * B performs certain checks such as on the validity period & name of principal. He also verifies the signature of the CA on the certificate. He then sends his challenge - a nonce R.
  * A responds by encrypting the challenge with her private key. When B receives $E_{A.pr}(R)$ he decrypts it with A's public key & compares it with the nonce he transmitted in Msg 2.

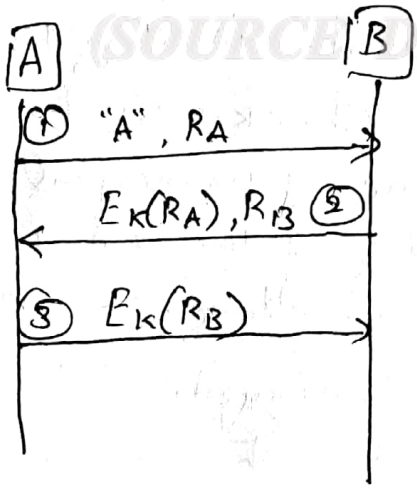
If they match, he concludes that A has used the private key corresponding to the public key in her certificate. Assuming that A's private key is safely protected, she must be the entity who created the correct response in Msg 3.

* Fig b:
  * Here B chooses a nonce, R and encrypts it with A's public key to create the challenge. A decrypts the challenge and sends it to B. Authentication of A to B succeeds if what B receives in Msg 3 is R, the nonce he just chose.

MUTUAL AUTHENTICATION

* It is often necessary for both communicating parties to authenticate themselves to each other.

1. Shared secret-based authentication.

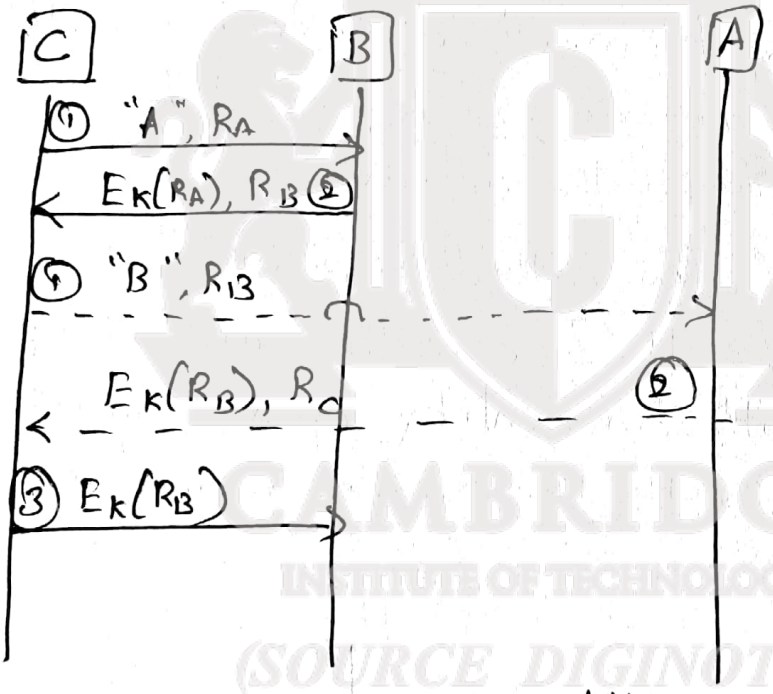


a) Flawed protocol.

*Fig a: In Msg 1, A communicates its identity and its challenge in the form of a nonce R_A.

* In msg 2, B responds to the challenge by encrypting R_A with the common secret, K that A & B share.

* B also sends its own challenge, R_B to A. A's response to B's challenge in the third message appeals to complete the protocol for mutual authentication.



(b) paralled session Attack.

*fig b : Attack scenario is as follows:

* Msg 1 : An attacker, C, sends a msg to B containing a nonce R_A and claiming to be A.

* Msg 2: B responds to the challenge with $E_K(R_A)$ and its own challenge R_B as required by the above protocol.

* Msg 1: Now "C" attempts to connect to A claiming it is B with a challenge R_B. Note that this is the same challenge offered to it by B in Msg 2.

* Msg 2: A responds to the challenge with $E_K(R_{13})$ and a nonce of its own.

* Msg 3: C uses A's response $E_K(R_{13})$ to complete the 3 msg authentication protocol with B.

C has successfully impersonated A to B.

* This attack is termed a Reflection Attack since a part of the msg received by an attacker is reflected back to the victim.

* This attack is also called a parallel session Attack since the attacker, in the midst of a protocol run with one entity, opens another protocol run or session with the same or another entity.

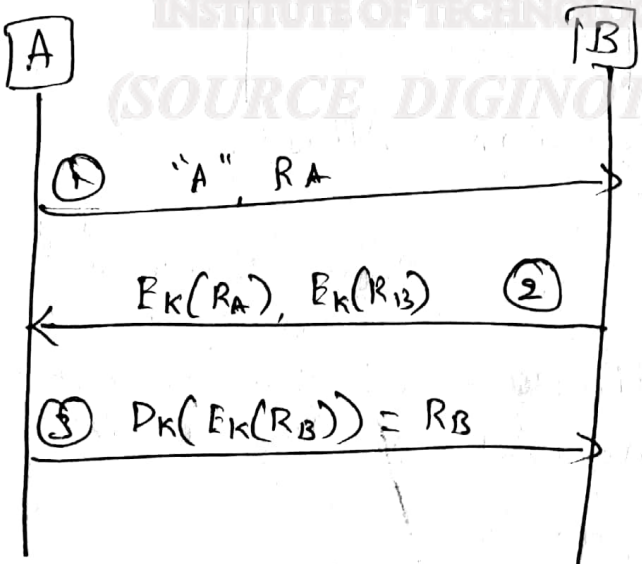
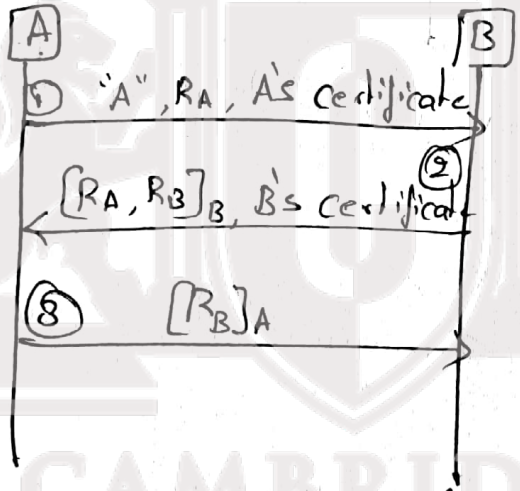


Fig: Corrected protocol.

* Fig c: possibility is to have the initiator and responder handle challenges differently.
for example, the protocol might require the responder to encrypt his challenge, while the initiator would be required to decrypt her challenge.

2. Asymmetric key-based authentication.

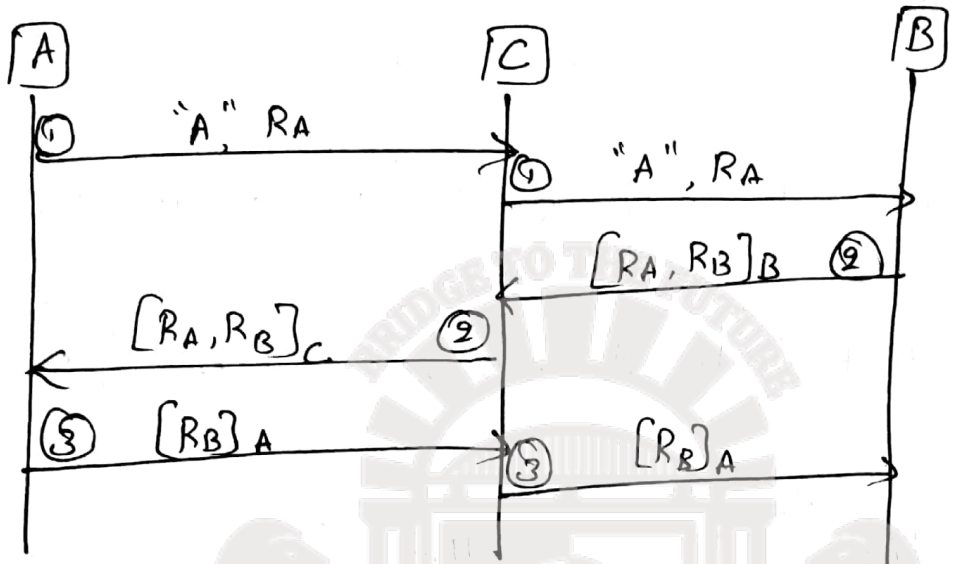
* Assume that both A & B have public/private key pairs.



a) flawed protocol.

* Fig a: Each party transmits its own nonce & challenges the other to sign it.
* Notation $[m]_A$ => m, sent in the clear together with A's signature on m.
* Msg 2: The string obtained by concatenating nonces R_A & R_B is signed by B. Both the nonces and the signature are sent.

* Msg 3: Nonce R_A is the challenge provided by A. R_B is the challenge provided by B and signed by A in response.



b) Attack on flawed protocol.

Fig b:

Msg 1: A initiates communication with C, sending her challenge R_A.

Msg 1: C initiates communication with B using the same nonce R_A supplied by A.

Msg 2: B responds to "A"s challenge & includes a challenge of his own R_B.

Msg 2: C responds to A's challenge and uses B's nonce, R_B as his challenge to A.

Msg 3: A responds to C's challenge (which was actually generated by B). A thus completes the mutual authentication protocol with C.

Msg 4: C forwards A's response to B.

* Analyze the above protocol (fig b)

* A does intend to communicate with C (otherwise A would not have responded in msg 3 to C's challenge that was transmitted in msg 2).

* B wishes to communicate with A. Otherwise B would not have responded in msg 2 to the nonce presented in msg 1.

Note: Msg 1 is sent by C but it includes A's identity, who is C?

C is probably known to A. After all, A intends to talk to C. But C is also the attacker here, when A initiates communication with C, the latter seizes the opportunity & attempts to convince B that A intends to talk to him. B responds to what appears to be A's intention to communicate with him. Note that, in the current scenario, A may not wish to communicate with B & is not aware that C is attempting to do so on her behalf. Yet after B receives msg 3, he feels A intends to communicate with him since msg 3 contains her signature on a nonce chosen by him.

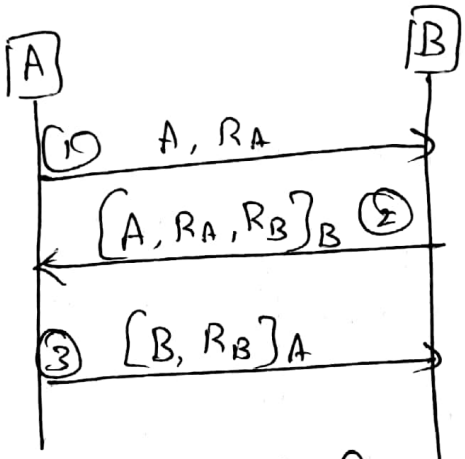


Fig c: corrected protocol.

* Fig c: soln is for the sender to include the identity of the recipient in all meges signed by him. note that with this modification, msg3 would be $[C, R_B]_A$ in fig b. If C tries to forward this msg to B, the latter ll smell a rat since it is C's identity that is included in the msg. So B ll realize that the msg was intended for C, not for him.

3. Authentication and Key Agreement.

* Shows protocols providing both mutual authentication and key agreement.

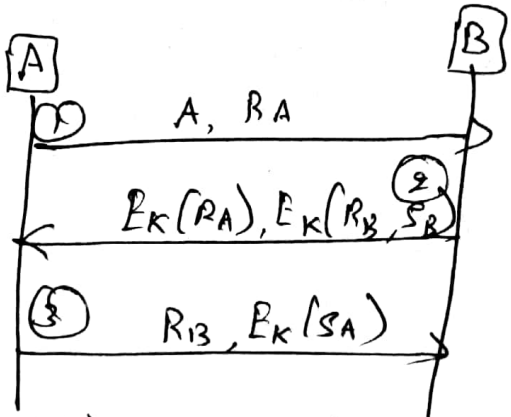
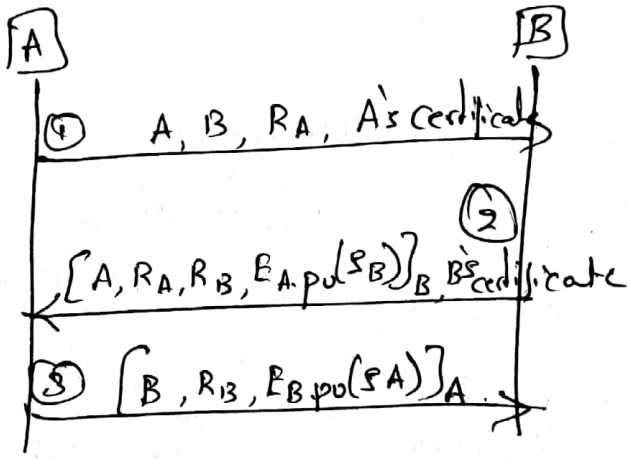


Fig a: using secret key cryptography



b. using public key cryptography.

Session key = $S_A \oplus S_B$.

* Fig a: uses secret key cryptography
  Fig b: uses public key cryptography.
* In both the figures, S_A & S_B are the contributions to the secret key by A & B respectively.
* They are freshly chosen random numbers that are encrypted & sent so that they cannot be eavesdropped upon.
* In fig a, they are encrypted in msg 2 & 3 by the shared secret k.
* In fig b, they are encrypted in msg 2 & 3 using the recipient's public key.
* The key finally chosen could be a simple function of S_A & S_B for example $S_A \oplus S_B$.

4. Use of Timestamps.

* The use of nonces was introduced as a means to prevent replay attacks.
* An alternative to nonces are timestamps.
* Timestamps : Stamping a msg with the current time, we convince the receiving party of its freshness.
* Figure shows the use of timestamps in conjunction with public key cryptography for authentication.

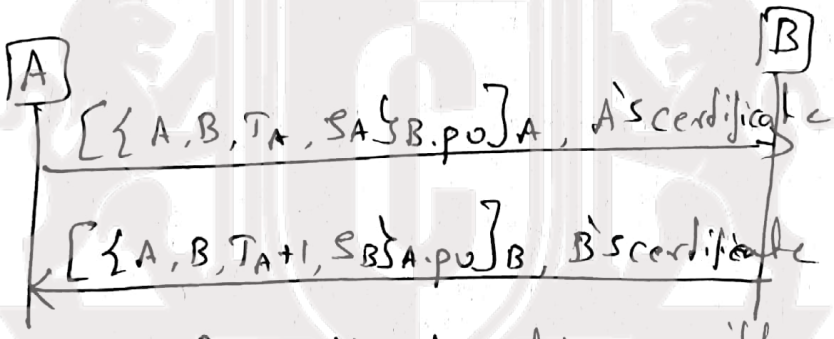


Fig:- Mutual authentication with timestamps.

* Msg 1: A inserts a timestamp, T_A, in her msg & signs it.
* B, on receiving the msg, checks whether the timestamp is sufficiently recent and then verifies the signature. He increments the received timestamp inserts it into his response msg to A & signs the msg.

Notation $\{m\}_{x.pu}$ — m encrypted using the public key of x.

IPsec- Security at the Network Layer

- Security at different layers: Pros and Cons.
- IPsec in Action.
- Internet Key Exchange (IKE) protocol.
- Security Policy and IPsec.
- Virtual Private Networks.

IPsec in action

- IP-level security encompasses three functional areas:

1. **Authentication:** Assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header.
2. **Confidentiality:** Assures that the packet has not been altered in transit.
3. **Key management:** It is concerned with the secure exchange of keys.

Applications of IPsec

- IPsec provides the capability to secure communication across a LAN, across private and public WANs, and across the Internet.

Examples:

- **Secure branch office connectivity over the Internet:** A company can build a secure VPN over the internet.
- **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an ISP and gain access to a company network.
- **Establishing extranet and intranet connectivity with partners:** It can be used to secure communication with other organizations, ensuring authentication, confidentiality & providing key exchange.
- **Enhancing electronic commerce security:** Even though some web and electronic commerce applications have built in security protocols the use of IPsec enhances that security.

Benefits of IPsec

- IPsec is implemented in a firewall or router, it provides strong security that can be applied to all traffic crossing the perimeter.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP and the firewall is the only means of entrance from the Internet into the organization.
- There is no need to change software on a user or server system when IPsec is implemented in the firewall or router.
- IPsec can be transparent to end users. There is no need to train users on security mechanisms, issue keying materials or revoke keying material when users leave the organization.
- IPsec can provide security for individual users if needed.

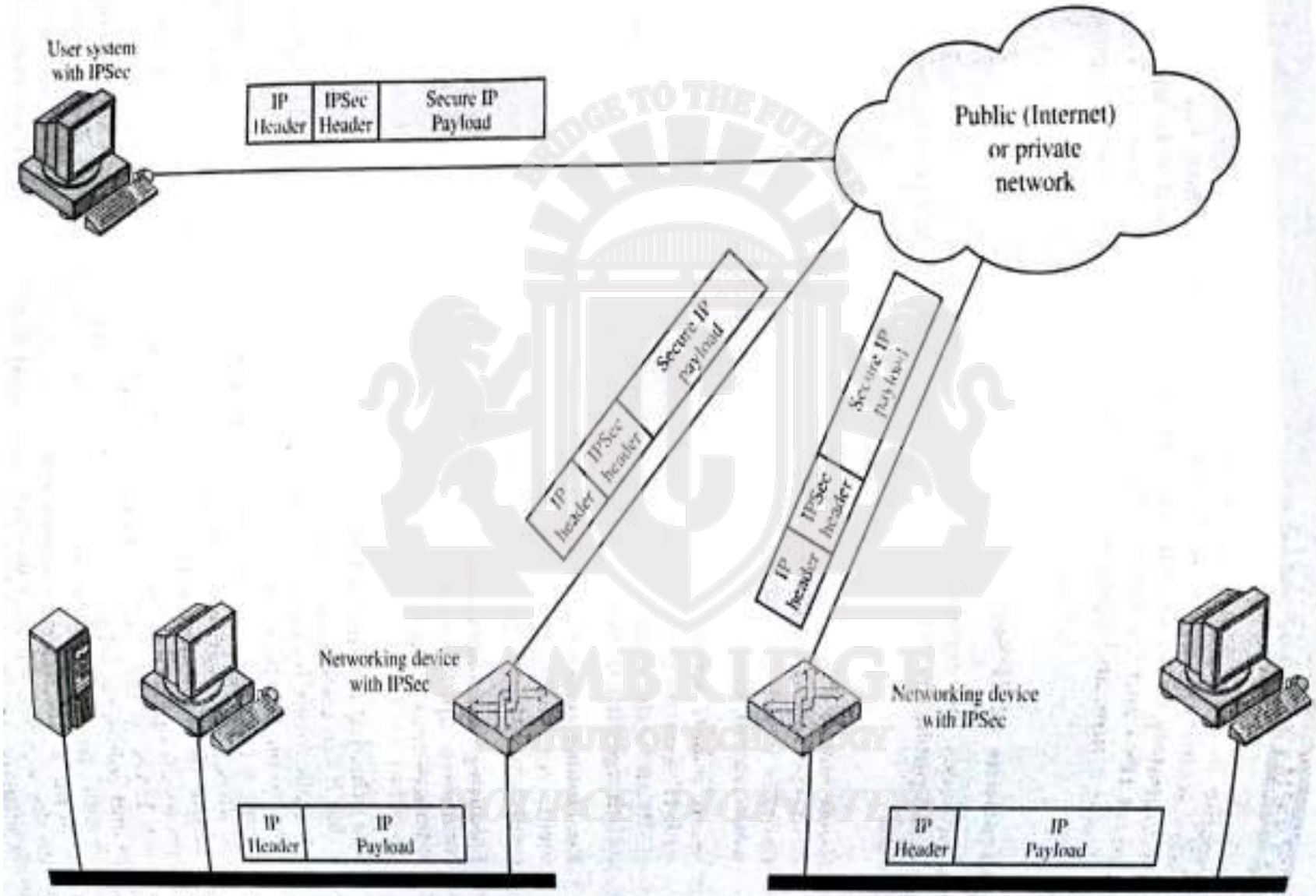


Figure 6.1 An IP Security Scenario

IPsec Services

- The services are

1. Access control.
2. Connectionless integrity.
3. Data origin authentication.
4. Rejection of replayed packets.
5. Confidentiality.
6. Limited traffic flow confidentiality.

Security Associations

- An association is a one-way relationship between a sender and a receiver that affords security services to the traffic carried on it.
- If a peer relationship is needed, for two-way secure exchange, then two security associations are required.
- Each node has a database of SAs for all connection originating from or terminating at it. This database is referred as SA database.

- A SA is uniquely identified by three parameters:

1. **Security Parameters Index(SPI):** SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed.
2. **IP Destination Address:** This is the address of the destination endpoint of the SA, which may be an end user system or a network system such as a firewall or router.
3. **Security Protocol Identifier:** This indicates whether the association is an AH or ESP security association.

SA Parameters

- Sequence number counter.
- Sequence counter overflow.
- Anti-replay window.
- AH Information.
- ESP Information.
- Lifetime of this security association.
- IPsec protocol mode.

Transport mode

- Transport mode is used for end-to-end communication between two hosts.
- When host runs AH or ESP over IPv4, payload is the data that normally follow the IP header.
- For IPv6, the payload is the data that normally follow both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection.
- ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header.
- AH in transport mode authenticates the IP payload and selected portions of the IP header.

Tunnel mode

- Tunnel mode provides protection to the entire IP packet.
- To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new outer IP packet with a new outer IP header.
- The entire original or inner packet travels through a tunnel from one point of an IP network to another: no router along the way are able to examine the inner IP header because the original packet is encapsulated, the new larger packet may have totally different source and destination addresses adding to the security.
- Tunnel mode is used when one or both ends of an SA are a security gateway, such as a firewall or router that implements IPsec.

IPsec protocols: AH and ESP

- The authentication header provides support for data integrity and authentication of IP packets.

- AH consists of the following fields

1. **Next header:** Identifies the type of header immediately following this header.
2. **Payload length:** Length of AH in 32-bit words, minus 2.
3. **Reserved:** For future use.
4. **Security parameters index:** Identifies a SA.
5. **Sequence number:** A monotonically increasing counter value.
6. **Authentication data:** A variable length field that contains the integrity check value or MAC for this packet.

ESP format

- Encapsulating security payload provides confidentiality services, including confidentiality of message contents and limited traffic flow confidentiality.

- ESP contains the following fields:

1. **Security parameters Index:** Identifies a security association.
2. **Sequence number:** A montonically increasing counter value, this provides an anti-replay function.
3. **Payload data:** This is a transport level segment or IP packet that is protected by encryption.
4. **Padding:** If an encryption algorithm require the plaintext to be a multiple of some number of bytes, the padding field is used to expand the plaintext to the required length.

5. Pad length: Indicates the number of pad bytes immediately preceding this field.

6. Next header: Identifies the type of data contained in the payload data field by identifying the first header in that payload .

7. **Authentication data:** A variable length field that contains the integrity check value computed over the ESP packet minus the authentication data field.

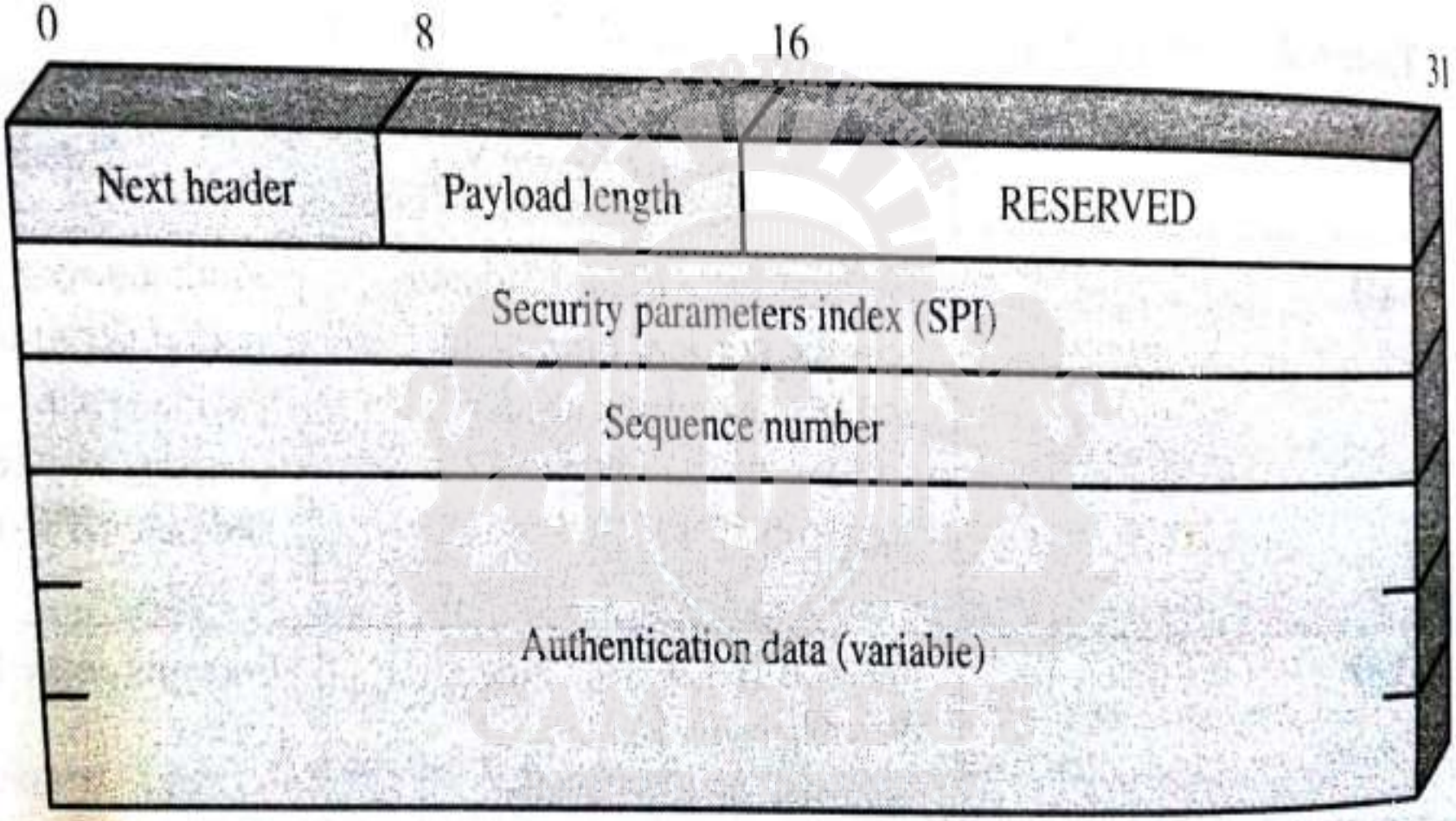


Figure 6.3 IPSec Authentication Header

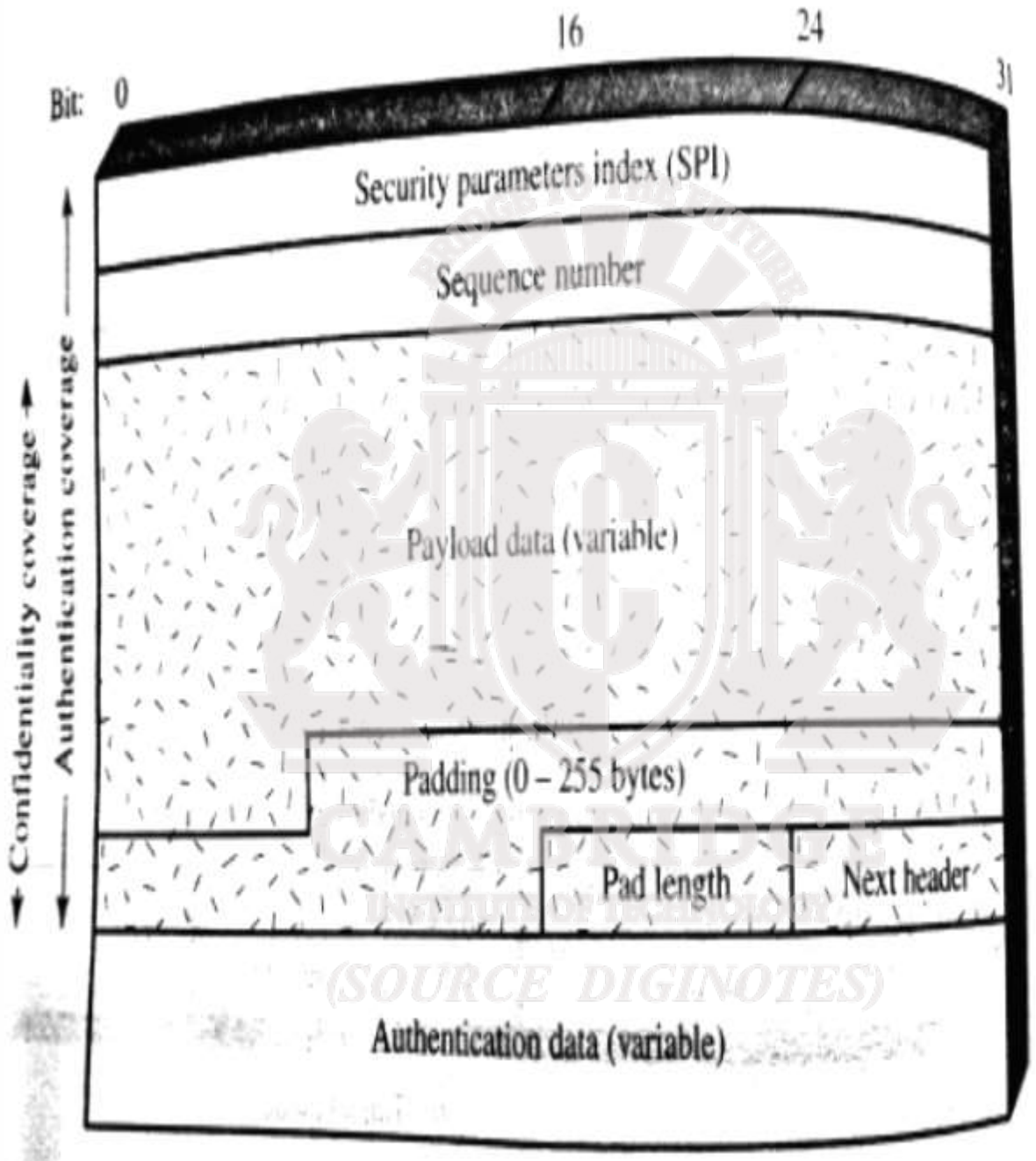
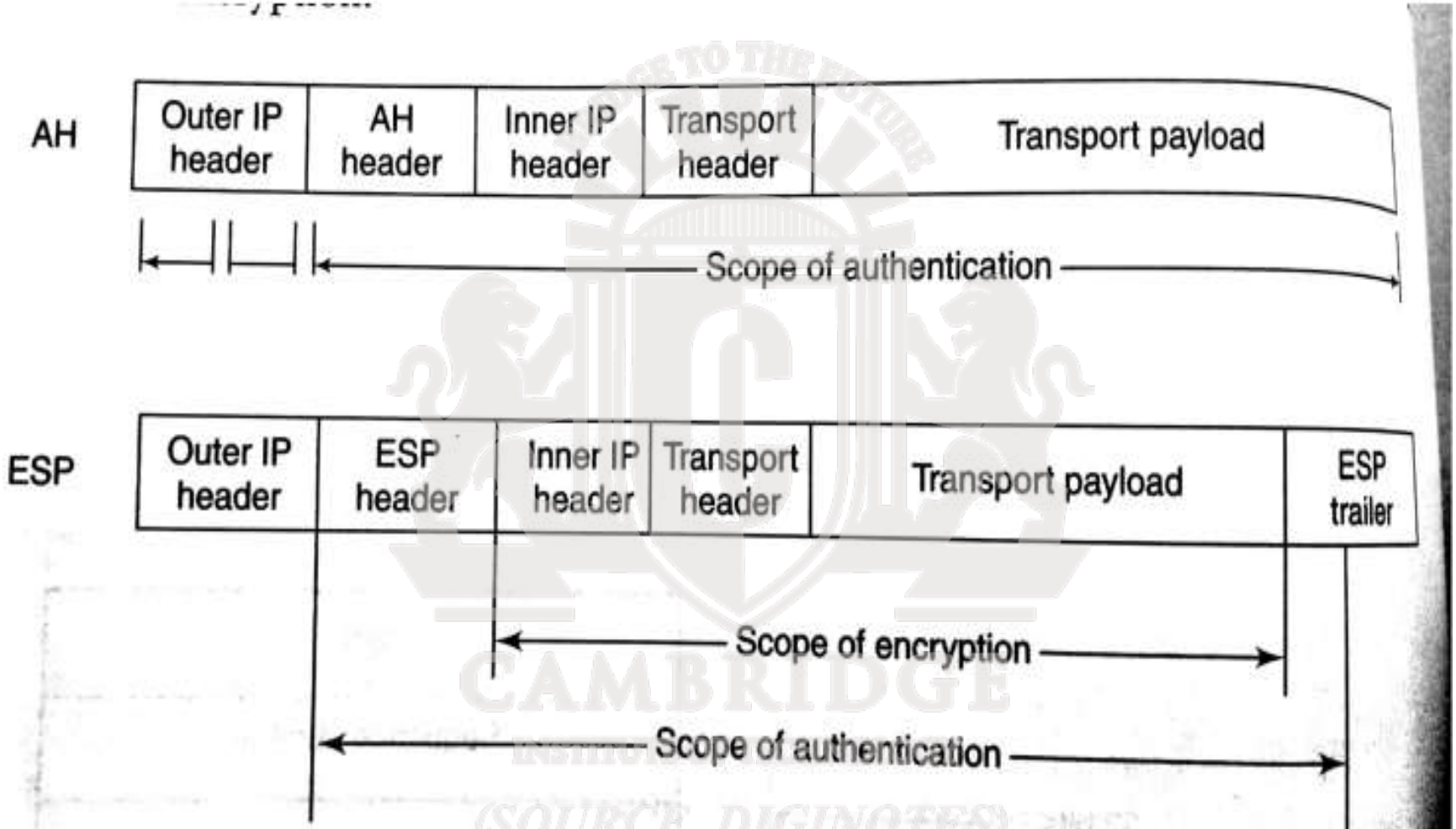


Figure 6.7 IPSec ESP format



3.2 *AH and ESP in tunnel mode*

Internet key exchange protocol

- The main goal of IKE is to establish an SA between two parties that wish to communicate securely using IPsec.
- IKE is an application layer protocol using the connectionless UDP protocol.
- IKE borrows heavily from two major sources- the Internet security association and key management protocol (ISAKMP) and oakley.
- ISAKMP defines formats of various entities such as the digital signature and the digital certificate.
- It also specifies the rules for stringing payloads together to form a valid msg.
- Oakley specifies the kind of information to be exchanged in each message that is part of IKE.

Internet key exchange

- Purpose

1. Mutual authentication.
2. Shared secret establishment.
3. Crypto algorithms negotiation.
4. Security association establishment.

IKE is composed of two phases.

- In the first phase, an IKE SA is established. This creates a secure channel upon which the communicating parties can then established multiple IPsec SA instances over time.
- It is good security practice to periodically change cryptographic keys used by two communicating parties.
- In phase 1, long term keys are derived.
- In phase 2, shorter term keys are derived for use between two parties. This key is a function of the long term keys computed in phase 1 together with nonce exchanged in phase 2.
- Key agreement use DHKEP, unauthentication key exchange is vulnerable to man-in middle attacks and session hijacking.
- Attacker could induce its victim to compute useless modular exponentiation leading to a DOS attack.
- It is designed to withstand these attacks while at the same time offering a menu of different cryptographic algorithms and authentication methods.

IPsec Cookies

- To thwart DOS attack, IKE makes extensive use of cookies.
- One cookie is created by the initiator A and another by the responder B.
- Phase 1 of IKE uses DHKE, an attacker creates many spurious messages each one being a request to set up an IKE SA with B.
- A spoofed IP source address is used in each of these messages.
- The responder would have no ways of knowing that the message are spoofed.
- To frustrate such attacks, IKE mandates that B should compute a 64-bit integer called a cookie.
- **Cookie:** It is a hash function of many variables including the IP address of A, an secret know only to B and possibly the time.

- A required to send this cookie to B in all subsequent messages.
- In general this cookie will be different for different IP address.
- On receipt of a message from A, B will check to see whether the cookie corresponds to A's IP address.
- If the check fails, B will abort session establishment and hence avoid performing the modular exponentiation.
- The attacker will have no way to spoof the cookie created in response to a request from A.
- The pair (Ca, Cb) plays the role in IKE.

IKE phase 1

- The following are accomplished in IKE phase 1:

1. The authentication method, encryption and hash algorithms together with the diffie- hellman group to be used are negotiated.
2. Both parties authenticate themselves to each other.
3. Keys, key(a) and key(e) are computed. These keys are used for message integrity protection and encryption respectively in both phase 1 and phase2.
4. Cookies are created at the start of phase 1 and serve the purpose of an IKE connection identifier.

Phase 1 use one of two modes

- Main mode: 6 messages, mutual authentication, session key establishment, hiding endpoint identity, negotiating cryptographic algorithms.
- Aggressive mode: 3 messages, mutual authentication, session key establishment.

- The motivation for introducing main mode is to hide the identities of sender and receiver from eavesdroppers.
- The main mode of IKE seeks to protect the confidentiality of these alternative forms of identification through encryption.
- To perform mutual authentication, IKE assumes that either A and B share a secret or A and B each have a public key private key pair.
- There are two ways in which A and B might prove knowledge of their private keys by signing a message(signature private key) or by decrypting a challenge(decryption private key).

Main mode

1. Option 1: A and B share a secret key(s).
2. Option 2: A and B each have private signing keys.
3. Option 3: A and B each have private decryption keys.

Option 1:

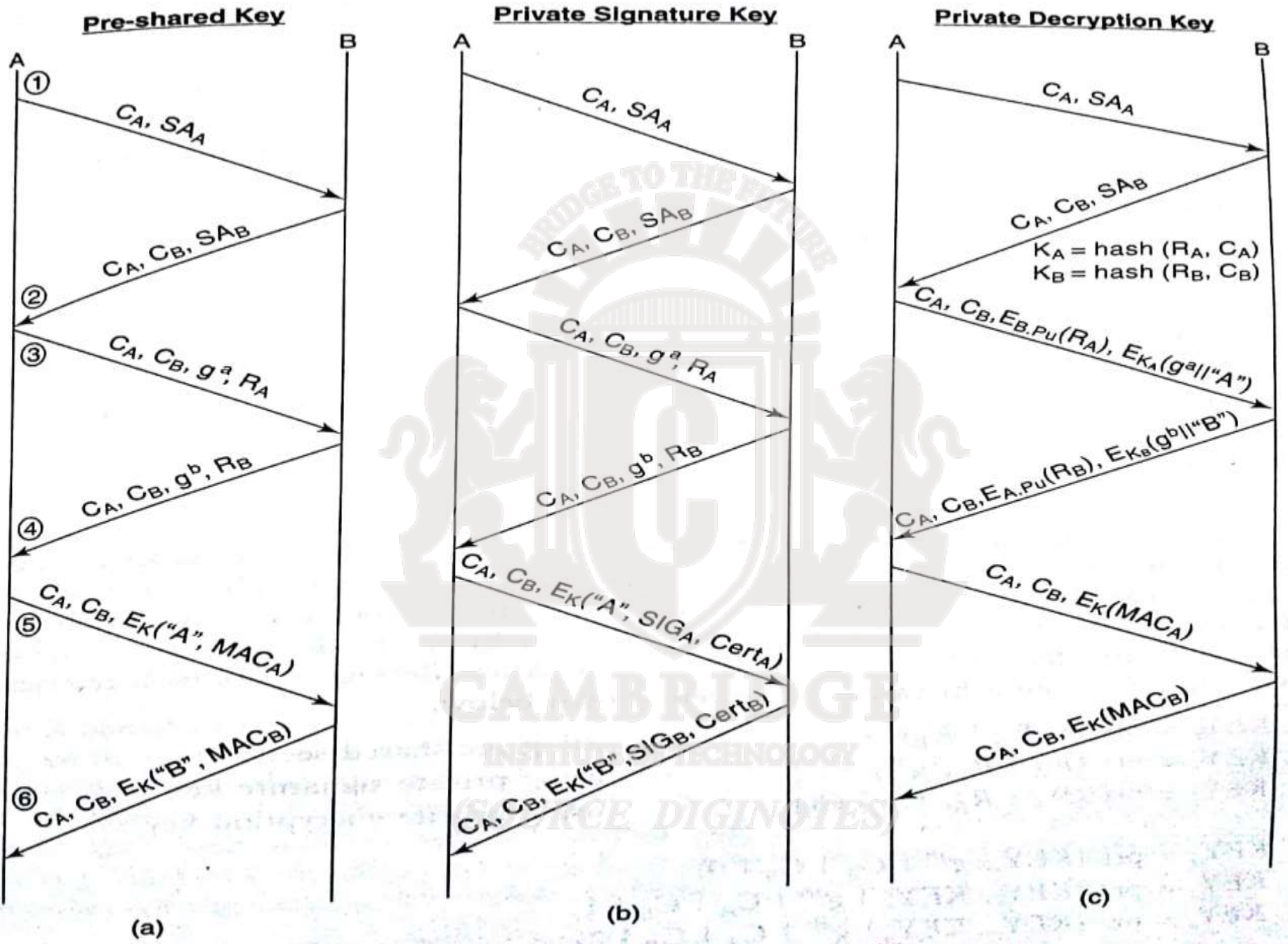
- The sequence of messages exchanged between A and B under the assumption that A and B share a secret keys.
- MSG1: Contains the cryptographic algorithms proposed by A for use in the IKE SA in addition to the cookie Ca, denoted by Sa.
- MSG2: Cryptographic algorithms accepted by B.
- MSG 3 & 4: Both side exchange nonce and the diffie- hellman partial keys.
- MSG 5 & 6: A and B independently compute a hierarchy of secrets.
- Both A and B use a MAC for message authentication and integrity.
- MSG 5 & 6, both sides reveal their identities to one another.
- Messages are encrypted with Key(e) .
- Major drawback is with shared secret.
- Alternatively B, could keep track of all entities that it expects to communicate with from each IP address.

Option 2:

- The main difference is that authentication and integrity protection of messages is by digital signature on MAC(a) and MAC(b) using their private keys.
- A and B dispatch their signing key certificate in MSG 5 and MSG 6 so that other party can perform signature verification.

Option 3:

- Both sides exchange their identities earlier in message 3 &4.
- Each side generate a nonce and encrypts it with the other side's public key.
- Each side encrypts its identity together with its DH partial key with temporary keys K(a) and k(b).
- MSG 5 & 6, each side transmits a MAC.
- An incorrect MAC would be detected by the other party and would result in the IKE exchange being aborted.



13.4 *IKE phase 1 (main mode)*

Aggressive mode

- Identities of the communicating parties are no longer hidden from passive eavesdroppers.
- Diffie – hellman group used and the group parameters are decides by A.
- A chooses a group, computes its partial key and sends it to B in MSG 1.
- B has no choice but to accept the group chosen by A.

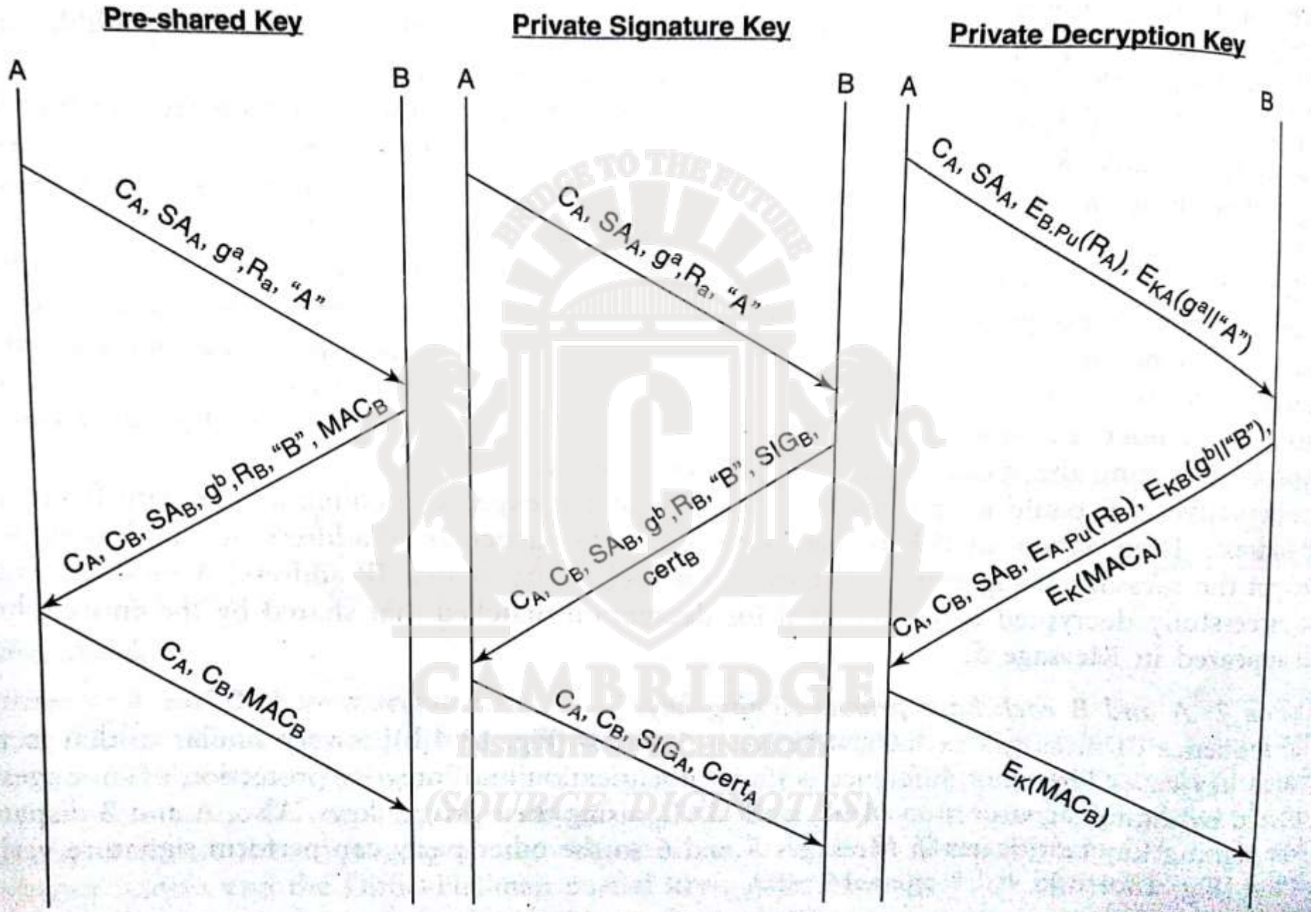


Figure 13.5 *IKE phase 1 (aggressive mode)*

IKE phase 2

- With existing IKE SA, two parties participate in an IKE phase 2exchange in order to establish a new IPsec SA.
- Fig shows the 3 messages exchanged in quick mode.
- All messages are encrypted using the secret key(e) computed in the previous phase.
- Message integrity and data source authentication is provided by using an HMAC. The key for the HMAC is key(a) also computed in phase 1.
- A 32-bit message ID (MID) together with the two cookies Ca and Cb are dispatched as part of each of the three messages.
- Both sides send their proposals of cryptographic algorithms to be used in the IPsec SA. These are denoted SA(a) and SA(b).
- To guarantee freshness both sides also generate and transmit nonces, Na and Nb.
- Is to agree on the secrets to be used for authentication and encryption as part of the IPsec SA. These secrets are computed simultaneously by both sides and are a function of KEY(d) computed in phase 1 and the nonces.

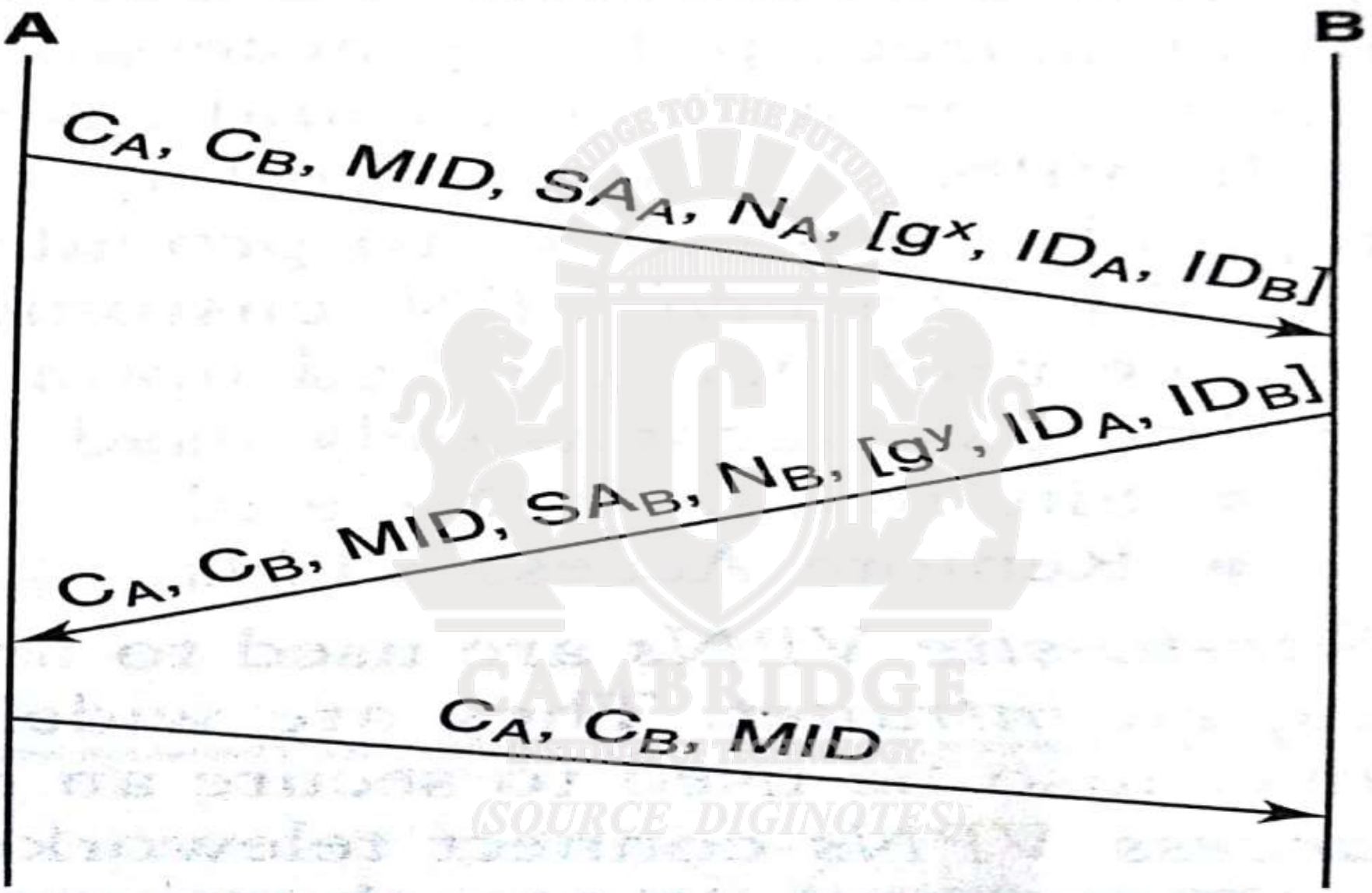


Figure 13.6 *IKE phase 2*

Security policy and IPsec

- Security policy database(SPD) is used to determine whether a packet sent or received should pass through, bypass it, or simply be dropped.
- Decision is made based on fields in the IP and transport headers.
- These fields called selectors include the destination IP address, the type of transport layer protocol and the type of application.
- Selectors are used to index into the SPD.
- The output indicates whether security should be applied.
- If the packet is part of the IP traffic that already has an existing SA, then the SPD returns a pointer to that SA.
- If an SA does not exist or has expired, the IKE protocol is used to establish an SA between the sender and receiver.

Virtual private networks

- VPN enables organizations to communicate securely over a public, shared network such as the internet.
- One possibility is to use dedicated point-to-point lines such as T1 leased lines to keep communications confidential.
- IPsec is just the protocol that helps secure IP traffic over such open and insecure networks.
- A secure VPN uses cryptographic techniques to provide not just confidentiality but also authentication and message integrity.
- In trusted VPN, customer traffic is not usually encrypted. Instead the infrastructure of the service provider is relied upon to guarantee confidentiality of the traffic.

- The two most widely used VPNs are

1. Site-to-site VPNs
2. Remote access VPNs.

- Site-to-site VPNs are used to link multiple offices of an organization in, commonly referred to as intranet.

- It is also used to secure an extranet- a network connecting multiple business partners.

- Remote access VPNs connect teleworkers(mobile users or users from home) to their offices.