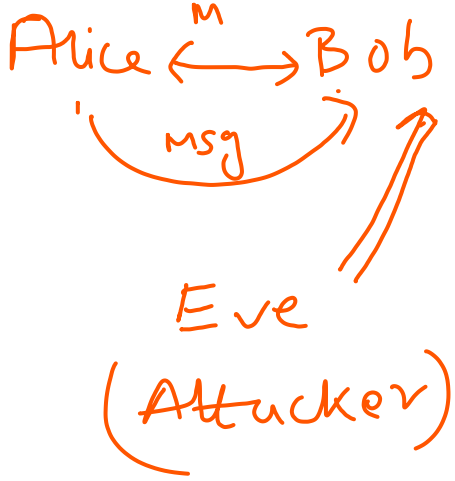
# Module 4

## Authentication Protocols & Digital signature schemes

# Services of digital signature

1. Message Authentication
2. Message Integrity
3. Non-repudiation
4. Confidentiality

# Message Authentication




- Bob can verify that the message is sent by Alice because Alice's public key is used in verification
- How?
  - Assume that Eve is an attacker in the channel between Alice and Bob. Both Alice and Bob are unaware of Eve's presence
    - If Eve is sending some message to Bob,
      - At first, the message digest is created from the message.
      - Secondly, the digest needs to be signed by Eve's private key.
      - Then the message along with the signature is sent to Bob

Msg → Msg digest
↓
Signed by the private key of sender

# Message Authentication

- Bob receives both the message and signature
- He needs to verify the signature
- Since Bob wants to make communication only with Alice, he has the public key of Alice only
- So, assuming that the message which received is coming from Alice, he will try to verify it with the public key of Alice
- But the public key of Alice can't verify the signature signed by Eve's private key.
- Hence, Bob comes to know that the message is not coming from an authenticated source

# Message Integrity

- Integrity of the message is preserved even after signing. This is because we cannot get the same signature if the message is changed.
- How?
  - The digital signature scheme uses a hash function in the signing and verifying algorithms to preserve the integrity of the message

# Nonrepudiation

- If Alice sends a message and then denies it, then Bob can prove that Alice actually signed it.
- How?
  - If Alice sends a message to Bob asking him to transfer Rs. 50,000 to Ariana. Bob transfers the amount. Alice later denies that she has sent this message.
  - Solution:- Bob must keep the signature in a file and use the public key of Alice to create the original message. Then he has to prove that the new message created is same as the original message. Then only Alice will accept.

# Nonrepudiation

1. Msg → Private key
2. Msg → New private key

- Problem:- This solution is infeasible because Alice might have changed her public or private key. She might claim that the file containing the signature is not authentic.

- The proper solution is to use services of the **trusted third party**
- People can create an established trusted party among themselves.

# Nonrepudiation

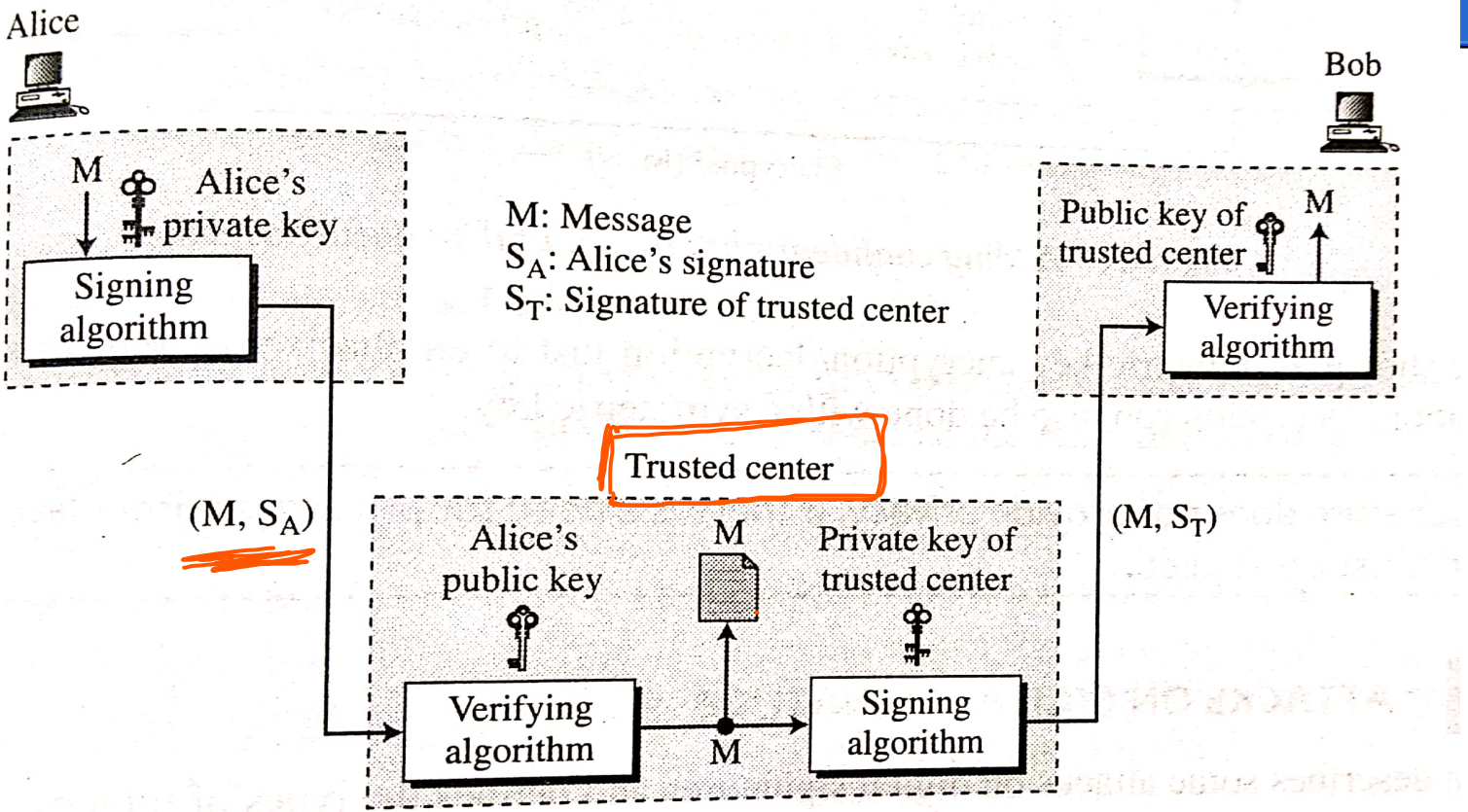



**Fig. 13.4** *Using a trusted center for nonrepudiation*

# Nonrepudiation

- Alice creates a signature from her message ($S_A$) and sends the message, the signature, her identity and Bob's identity to the center
- The center, after checking the validity of Alice's public key, verifies through the same key that the message came from Alice.
- The center saves a copy of message with the sender's identity, recipient's identityand a timestamp in its archive
- The sender uses its private key to create another signature ($S_{R)}$ from the message

→ Trusted center is doing verification

→ Stored with trusted center

# Nonrepudiation

.

- The center sends the following to Bob: message, the new signature, Alice's identity and Bob's identity
- Bob verifies the message using public key of the trusted center
- If in future, Alice denies sending the message then the center can show a copy of the saved message

# Confidentiality

- Digital signature scheme does not provide privacy or confidentiality.
- But if privacy is required then it can be provided
- How?
  - Using another layer of encryption/decryption
  - The message and the signature must be encrypted using either symmetric or asymmetric communication

# Confidentiality

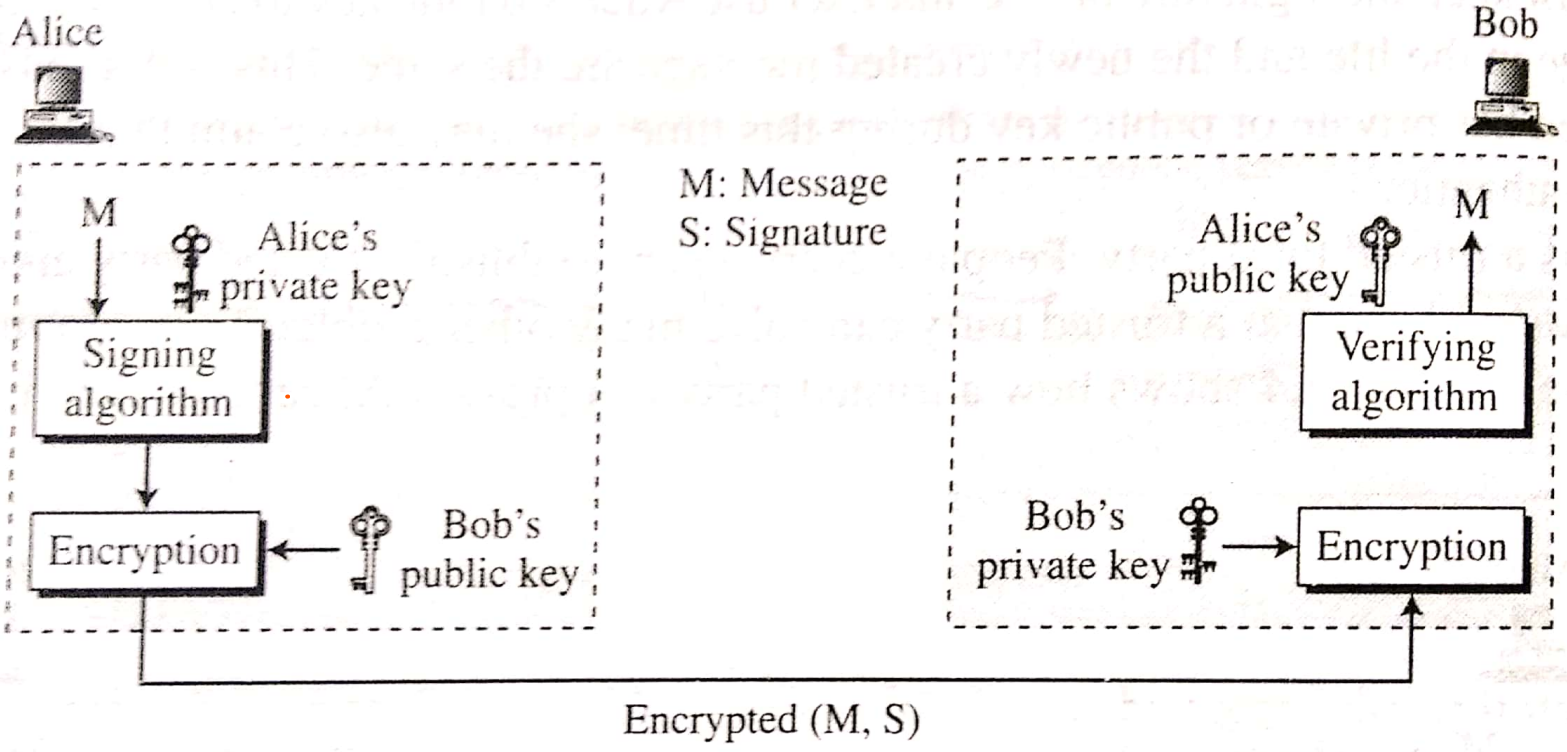



**Fig. 13.5** *Adding confidentiality to a digital signature scheme*

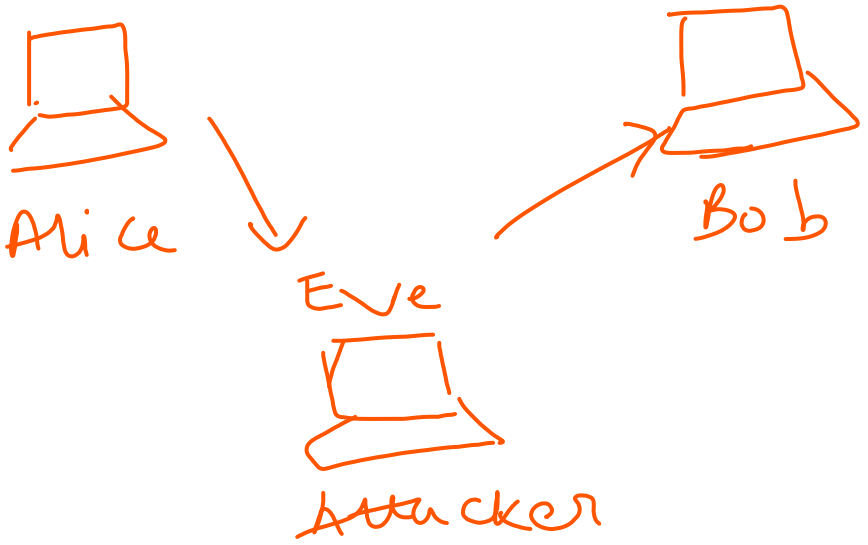
# Attacks on digital signature

1. Key only attack
2. Known Msg attack
3. Chosen Msg attack

# Key-Only attack

- Eve has access to the public information released by Alice
- To forge a message, Eve needs to create signature of Alice to convince Bob that the message is coming from Alice

→ 1) Public key of Alice
2) Msg
3) Signature



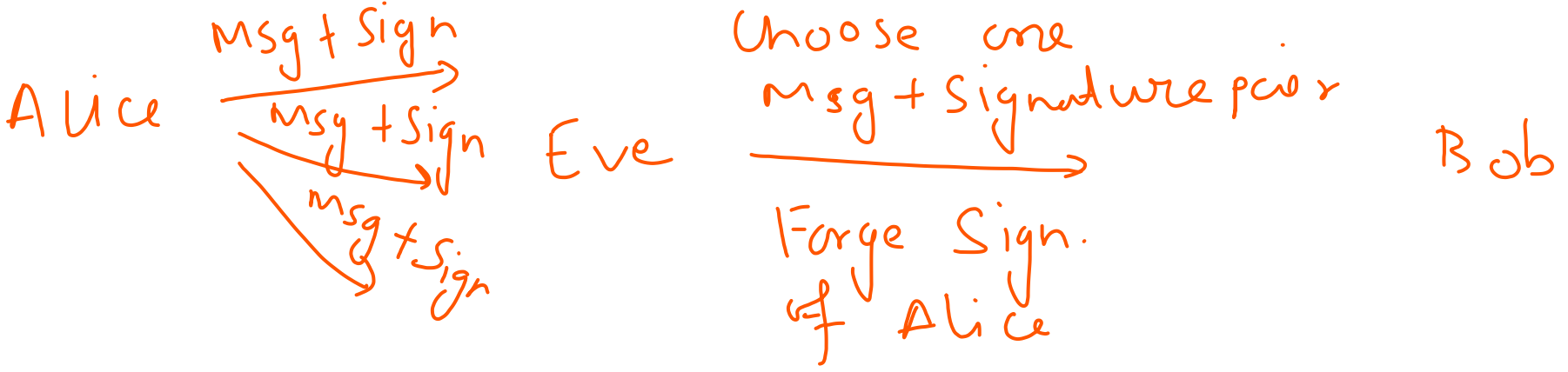

Eve forges a msg and convinces Bob that the msg is coming from Alice

# Known Message attack

- Eve has access to documents previously signed by Alice
- Eve tries to create another message and forge signature of Alice on it

# Chosen Message attack

- Eve somehow makes Alice sign on more than one message for her
- Eve now has a chosen message-signature pair
- Eve later creates another message with the contents she wants and forges signature of Alice on it



# Forgery Types

1. Existential Forgery

2. Selective Forgery

# Existential Forgery

- Eve may be able to create a valid message-signature pair but she can not really use it

contents of Msg are different

# Selective Forgery

- Eve may be able to forge signature of Alice on a message with the content selectively chosen by Eve

# RSA digital signature scheme

- RSA can be used for signing and verifying a message
- The private and public keys of the sender is used
- The sender uses its own private key to sign the document and the receiver uses the sender's public key to verify the document
- RSA digital signature scheme changes the role of private and public key
- Private key plays the role of the sender's own signature
- Sender's public key plays the role of copy of the signature that is available to the public

private & public keys

Alice is Signing
↳ Private of Alice

Bob is Verifying
↳ Public key of Alice

# RSA digital signature scheme

- The signing and verifying uses the same function but with different parameters
- The verifier compares the message and the output of the function for congruence.
- If result is true, message is accepted

# RSA digital signature scheme

- Key Generation:
  - Same as RSA. Public key is(e,n) and Private Key is *d*

# RSA digital signature scheme

Private key is d

- Signing and Verifying
  - **Signing:-**
    - Alice creates a signature of the message using her private exponent, $S=M^d \bmod n$
    - Alice sends this message and signature to Bob

$n = p \cdot q$

M → msg

$$S = M^d \bmod n$$

Alice $\xrightarrow{S + M}$ Bob

# RSA digital signature scheme

- **Verifying:-**
  - Bob receives M and S
  - He applies public exponent of Alice to the signature to create a copy of the message M`=$S^e$ mod n
  - Bob compares the value of M` with value of M
  - If the two values are congruent, then Bob accepts the message
  - The verification criteria used is:

    $M' \equiv M \pmod n \rightarrow S^e \equiv M \pmod n \rightarrow M^{dxe} \equiv M \pmod n$

Public key of Alice is (e, n)

$M' = S^e \bmod n$

M' → copy of msg

Verification Criteria is Congruence of M and M'

# RSA digital signature scheme

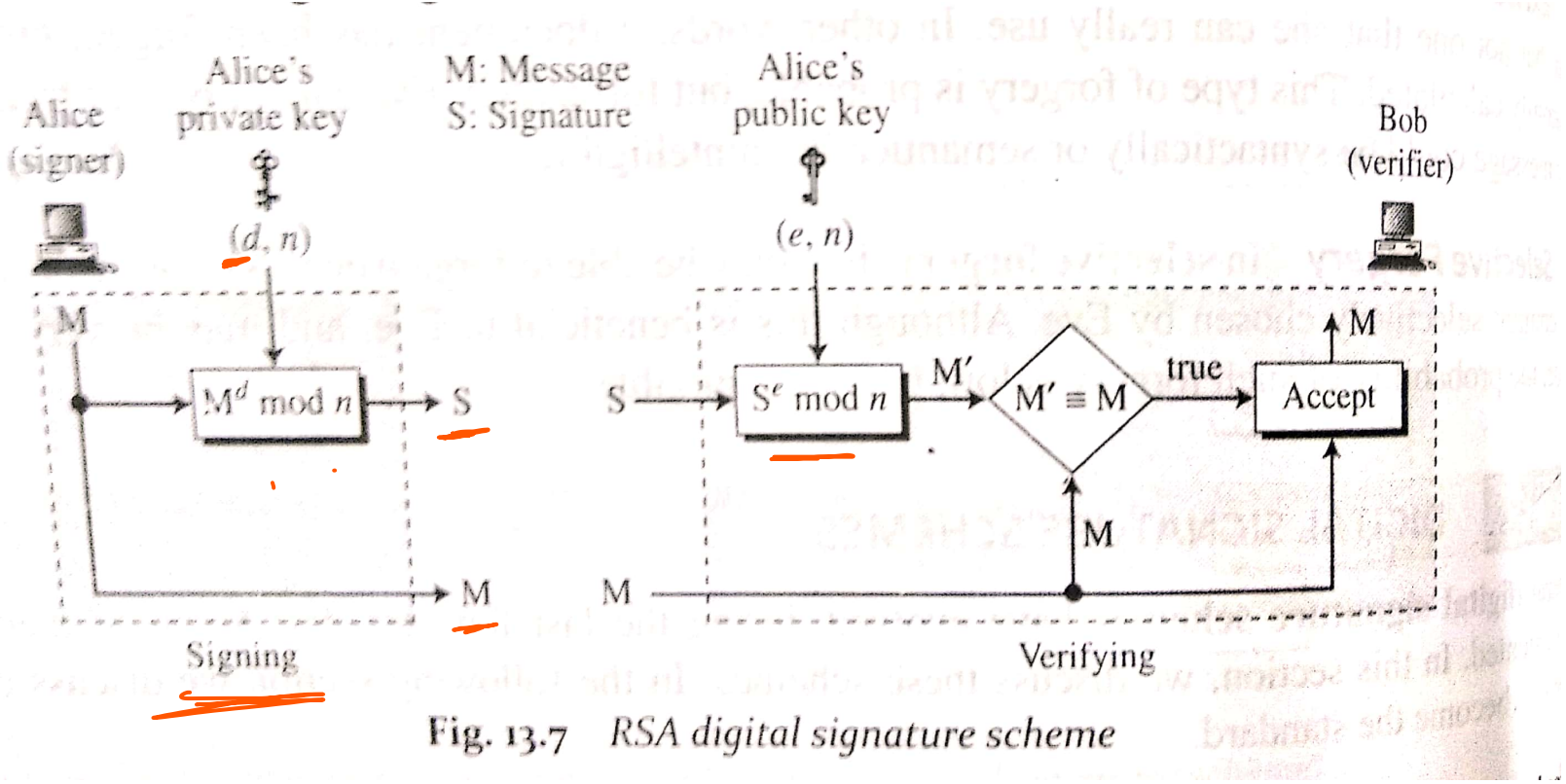



Fig. 13.7 *RSA digital signature scheme*

M' → copy of Msg