Digital Signatures

Introduction

- Digital signature is an authentication mechanism.
- The signature guarantees the source & integrity of the message.
- The other authentication mechanisms protects the two parties from any third party.
- They doesn’t provide protection of two parties against each other.
- The relationship between the signature and message are one to one.

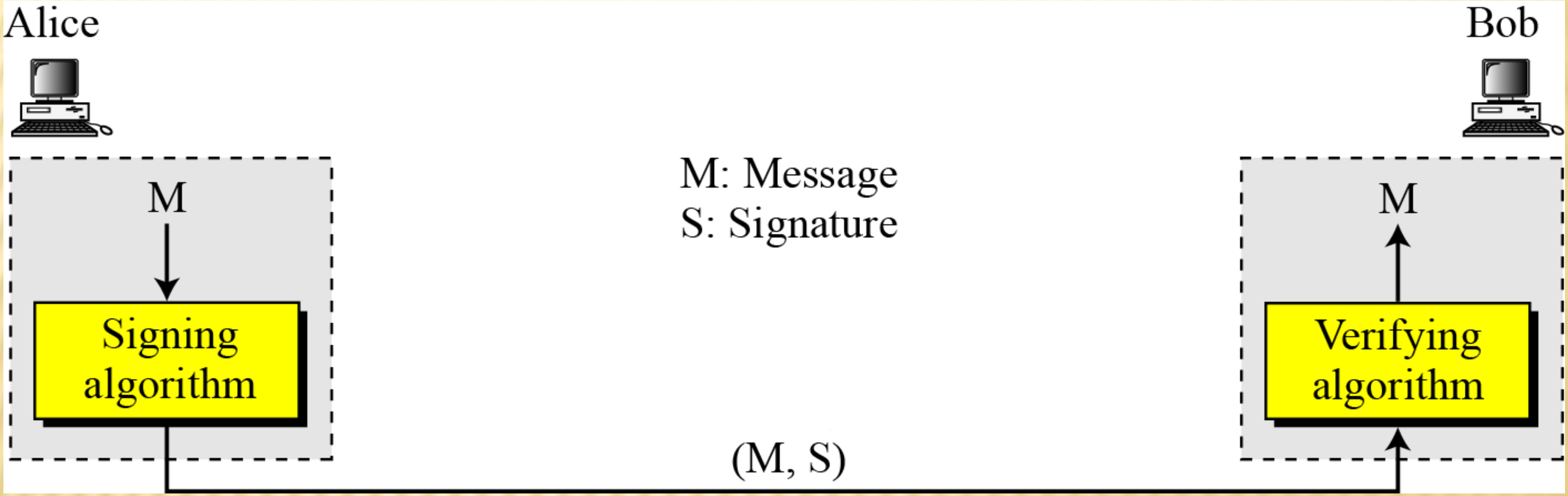
Requirements for a digital signature

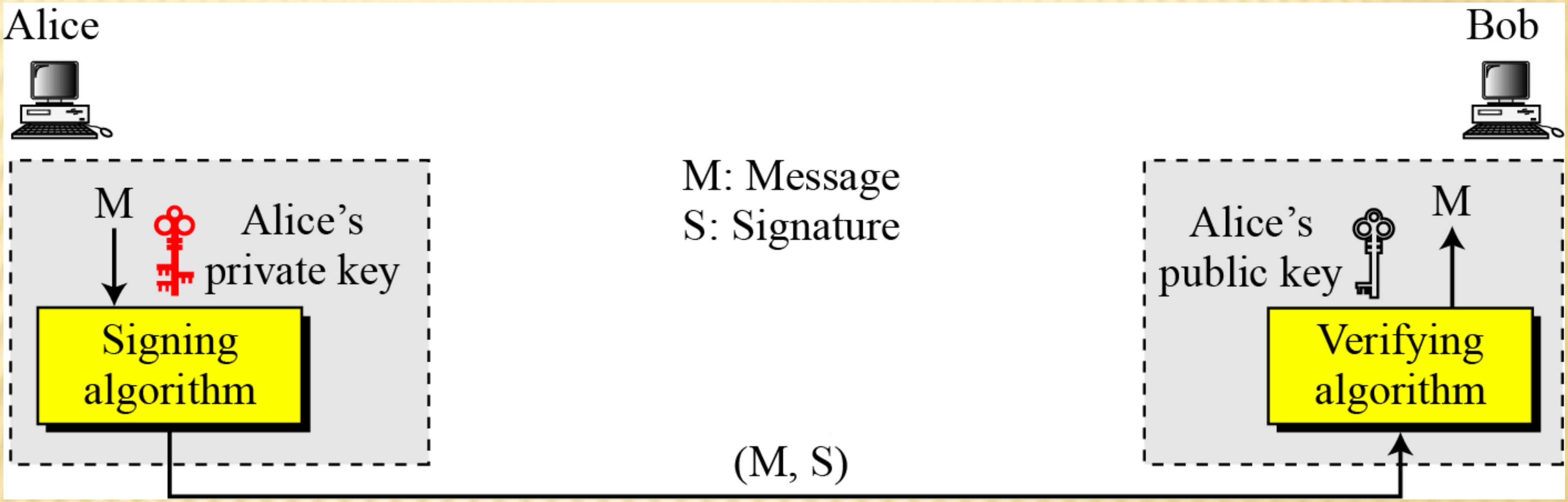
- The signature depends on the message being signed.
- The signature must use some information unique to the sender, to prevent forgery & denial.
- It must be relatively easy to produce the digital signature.

- It must be relatively easy to recognize & verify the digital signature.

- It must be computationally infeasible to forge a digital signature.

Digital signature process

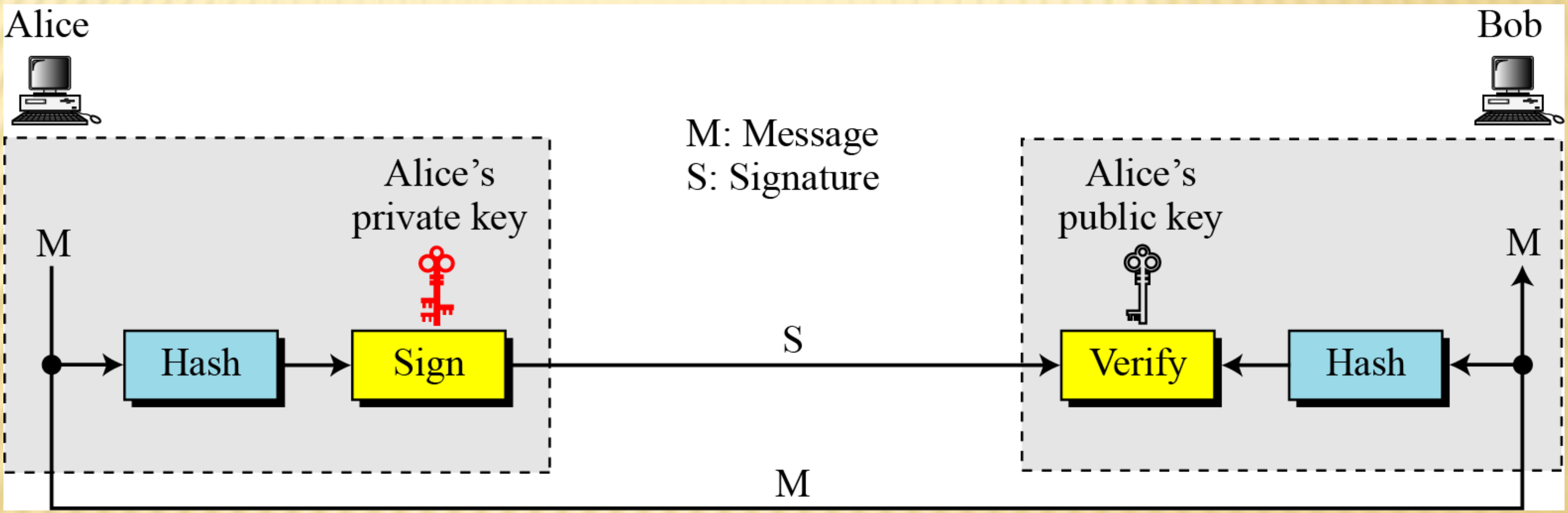



Alice
Bob
M
Alice's
private key
Signing
algorithm
M: Message
S: Signature
(M, S)
Alice's
public key
M
Verifying
algorithm

- A digital signature needs a public-key crptosystem.
- The signer signs with her private key; the verifier verifies with the signer’s public key.
- Symmetric key cryptosystem is not used because the secret key is known only to two entities.

Signing the digest

- Asymmetric key cryptosystems are very inefficient in dealing with long messages.
- In digital signature the messages we are having are long.
- The solution is to sign the digest of the message which is much shorter than message.



Signing the digest

Services

- A digital signature provides
  - Message authentication.
  - Message Integrity
  - Non repudiation
    - It can be provided only by using trusted third party.

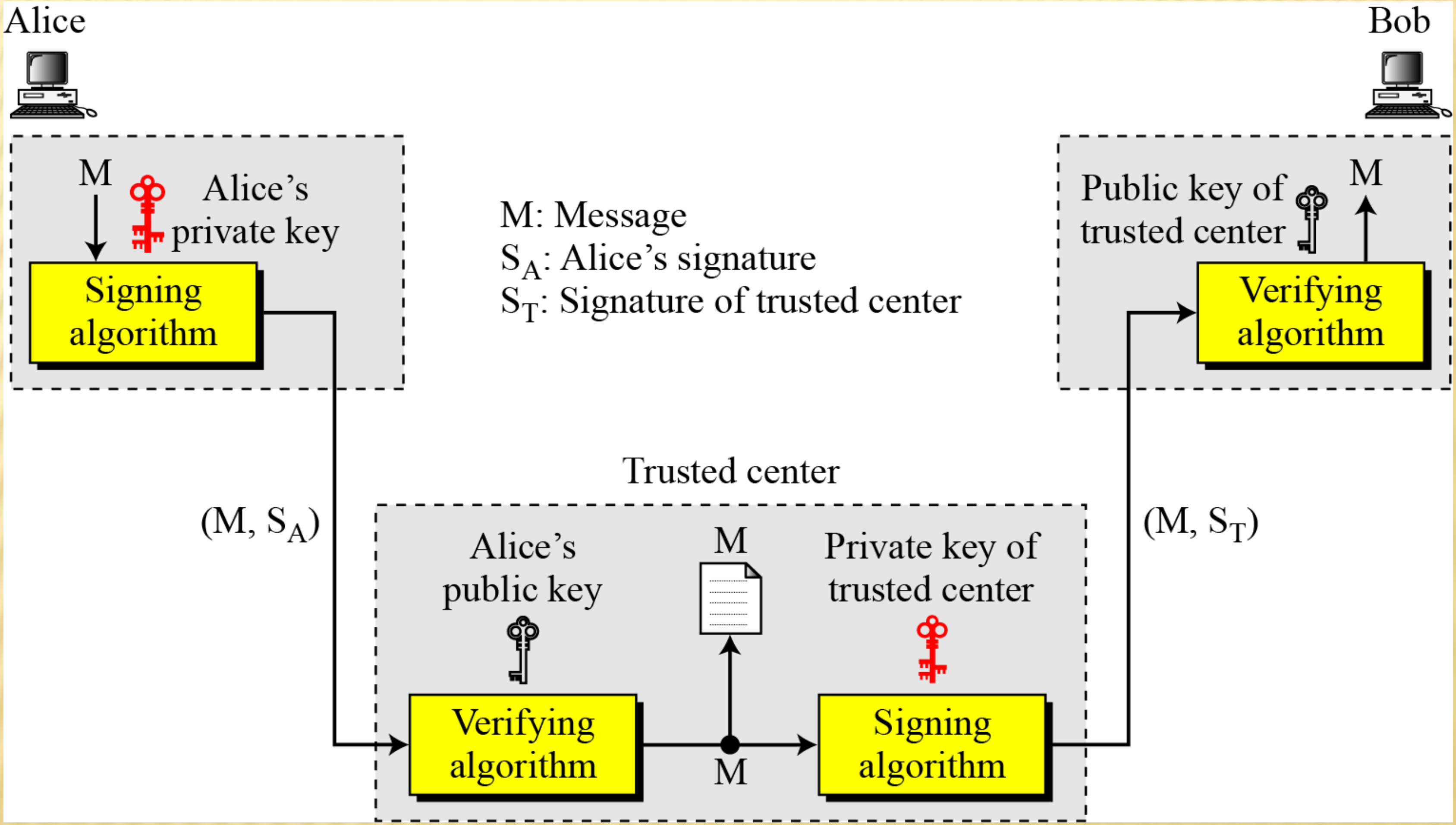
- Alice creates a signature from her message and sends the message, her identity , receiver’s identity, the signature to the trusted center.

- The center verifies that the message came from Alice through Alice’s public key.

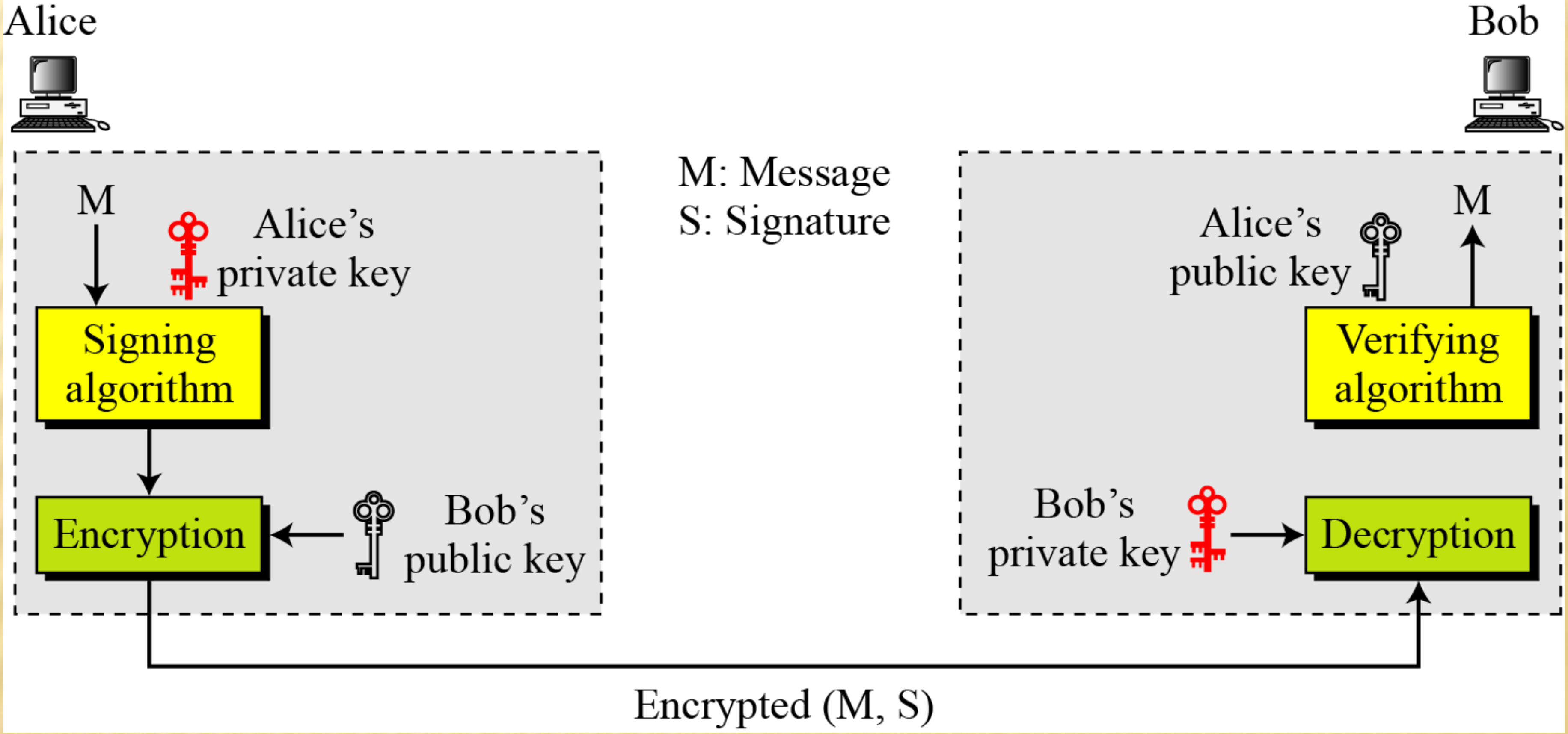
- The center then saves a copy of the message with senders identity , receivers identity and a timestamp .

- The center uses its private key to create another signature from the message.

- The center then sends the message , the new signature , Alice’s identity, Bob’s identity to Bob.
- Bob verifies the message using the public key of the trusted center.
- If in future Alice denies that she sent a message, the center can show the copy of the saved message.


Alice
Bob
M
Alice's private key
Signing algorithm
M: Message
S_A: Alice's signature
S_T: Signature of trusted center
Public key of trusted center
M
Verifying algorithm
Trusted center
(M, S_A)
(M, S_T)
Alice's public key
M
Private key of trusted center
Verifying algorithm
M
Signing algorithm

- Digital signature does not provide confidentiality.
- It can be provided only by using the encryption schemes.
- Confidentiality can be provided by either symmetric or asymmetric encryption schemes.



Adding confidentiality to digital signature scheme

ATTACKS ON DIGITAL SIGNATURE

- Key only attack
  - Eve has the access to the public information of sender.

- Known message attack
  - Eve has access to one or more message signature pairs.
  - Eve tries to create another message and forge Alice’s signature on it.

- Chosen Message Attack
  - Eve makes Alice sign one or more messages for him.
  - Eve later creates another message with the contents he want and forge’s Alice’s signature on it.

Forgery types

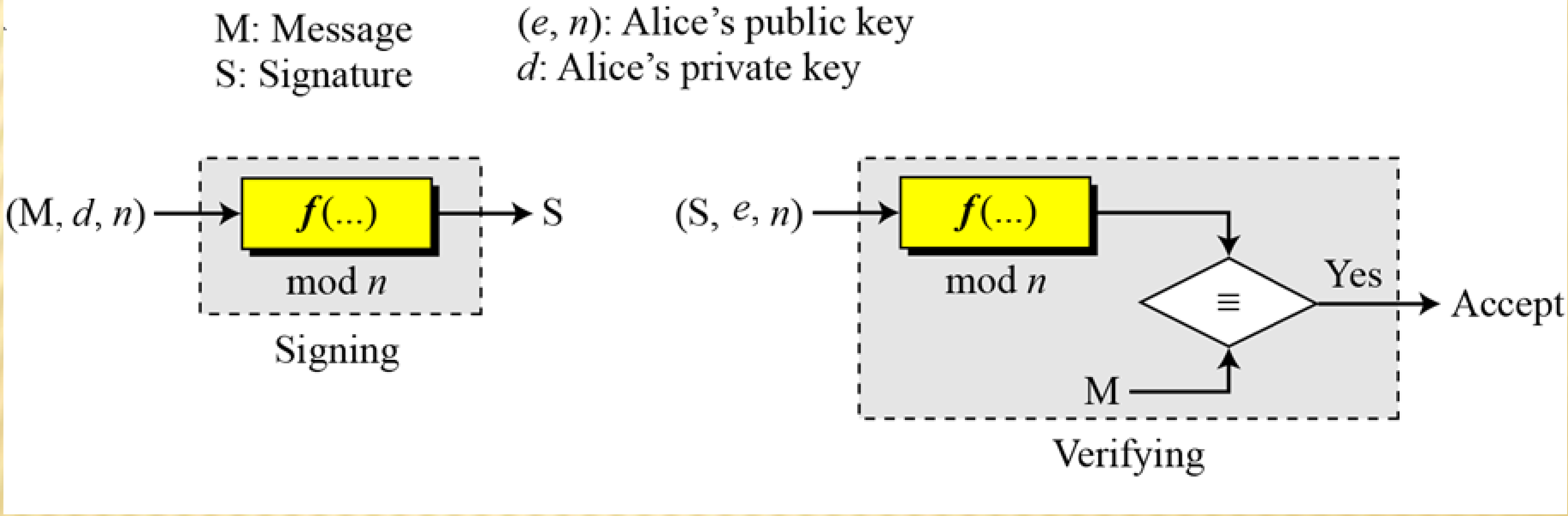
- If the attack is successful the result is a forgery.

- Existential forgery
  - Eve may be able to create a valid message signature pair but she cannot use it.

- Selective forgery
  - Eve may be able to forge Alice's signature on a message with the content selectively chosen.
  - The probability of such a forgery is less.

RSA Digital Signature Scheme

- RSA algorithm can be used for signing and verifying a message.
- The sender uses its own private key to sign the document.
- The receiver uses senders public key to verify it.



General idea of RSA Digital Signature Scheme

- Key Generation

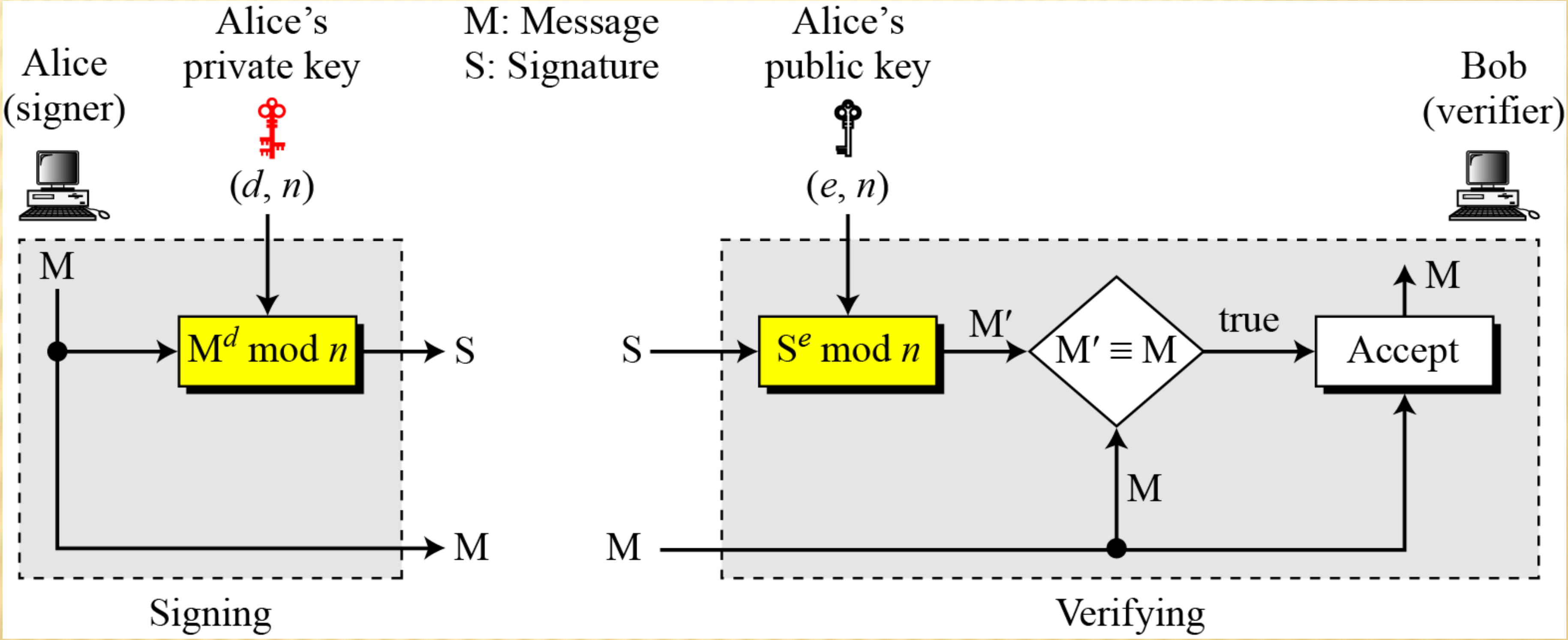
- Sender chooses two prime numbers p and q.
- Calculates $n = p * q$
- $\phi(n) = (p-1) * (q-1)$
- Then an integer e is chosen and its public.
- d is calculated such that $e * d \equiv 1 \bmod \phi(n)$ and d is private.

Signing and Verifying

- Signing
- Alice creates a signature from the message using her private key.
- $S = M^d \bmod n$
- Then sends the message and signature to bob.

- Verifying

- Bob receives M and S.
- Bob applies Alice's public key to the signature to retrieve the message and the retrieved message is denoted as M'.
- $M' = S^e \bmod n$
- Bob then compares the value of M and M'.
- If the two values are congruent Bob accepts the message.
- $M' \equiv M \bmod n$



RSA Digital Signature Scheme

- Example :Suppose that Alice chooses $p = 823$ and $q = 953$, and calculates $n = 784319$. The value of $\phi(n)$ is 782544. Now she chooses $e = 313$ and calculates $d = 160009$. At this point key generation is complete. Now imagine that Alice wants to send a message with the value of M = 19070 to Bob. She uses her private exponent, 160009, to sign the message:

$$\text{M: } 19070 \quad \rightarrow \quad S = (19070^{160009}) \bmod 784319 = 210625 \bmod 784319$$

- Alice sends the message and the signature to Bob. Bob receives the message and the signature. He calculates.

$$M' = 210625^{313} \bmod 784319 = 19070 \bmod 784319 \quad \rightarrow \quad M \equiv M' \bmod n$$

- Bob accepts the message because he has verified Alice's signature.

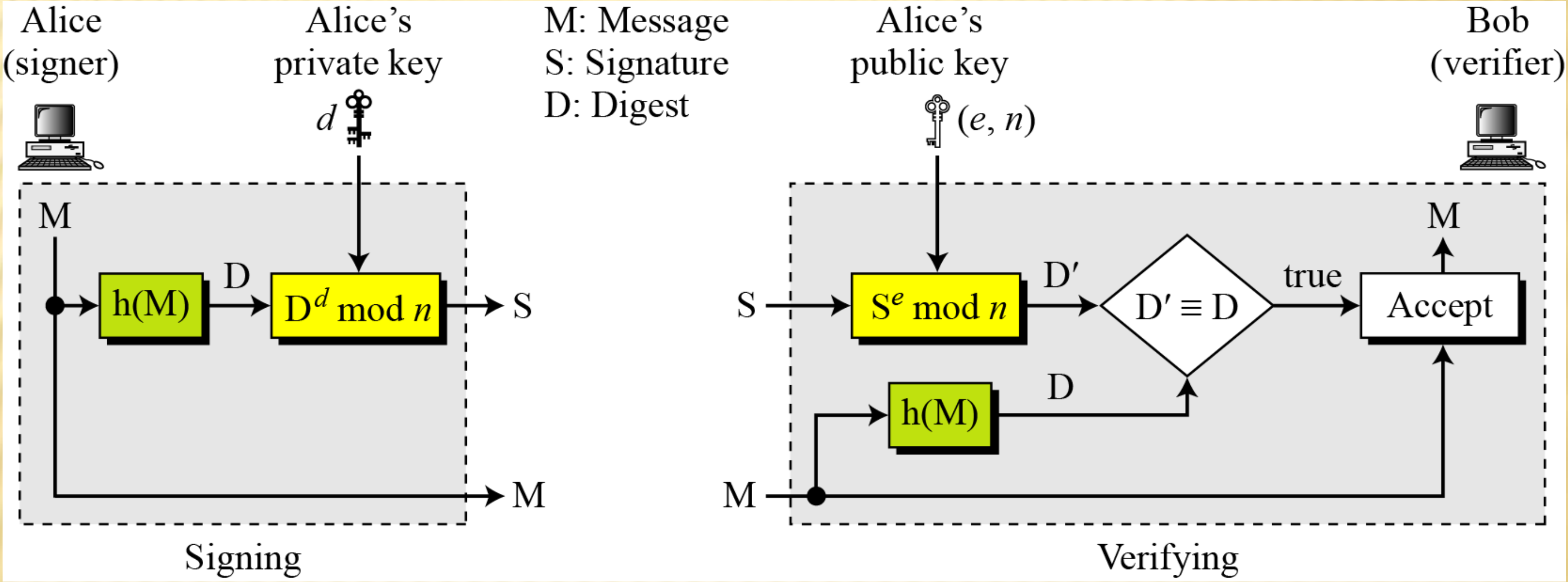
Known message attack

- Assume that attacker has intercepted two message signature pairs (M1, S1) and (M2,S2).
- The two pairs have been created using the same private key.
- If M = (M1 * M2) mod n , then S = (S1 * S2) mod n.
- The attacker can create M = (M1 * M2) mod n and can also create S =(S1 * S2) mod n.
- Thus the attacker fool Bob that S is signature of Alice on the message M.
- It’s an existential forgery.

- Chosen message attack

- Eve makes Alice sign two legitimate messages M_1 and M_2 .
- Eve then creates a message $M = M_1 \times M_2$
- Eve later claims that Alice has signed on M.
- It's a selective forgery.

RSA Signature on the message digest



- Key only attack

a) Eve intercepts the pair (S,M) and tries to find another message M' that creates the same digest

$h (M) = h(M')$.

The attack is difficult if the hash algorithm is second pre image resistant.

b) Eve finds two messages M and M ‘ such that h(M) = h (M’).

- If eve makes Alice to sign h (M) to get S, then eve has a pair (M ‘ , S).
- This pair can pass the verifying test.
- The attack is difficult if the hash algorithm is collision resistant.

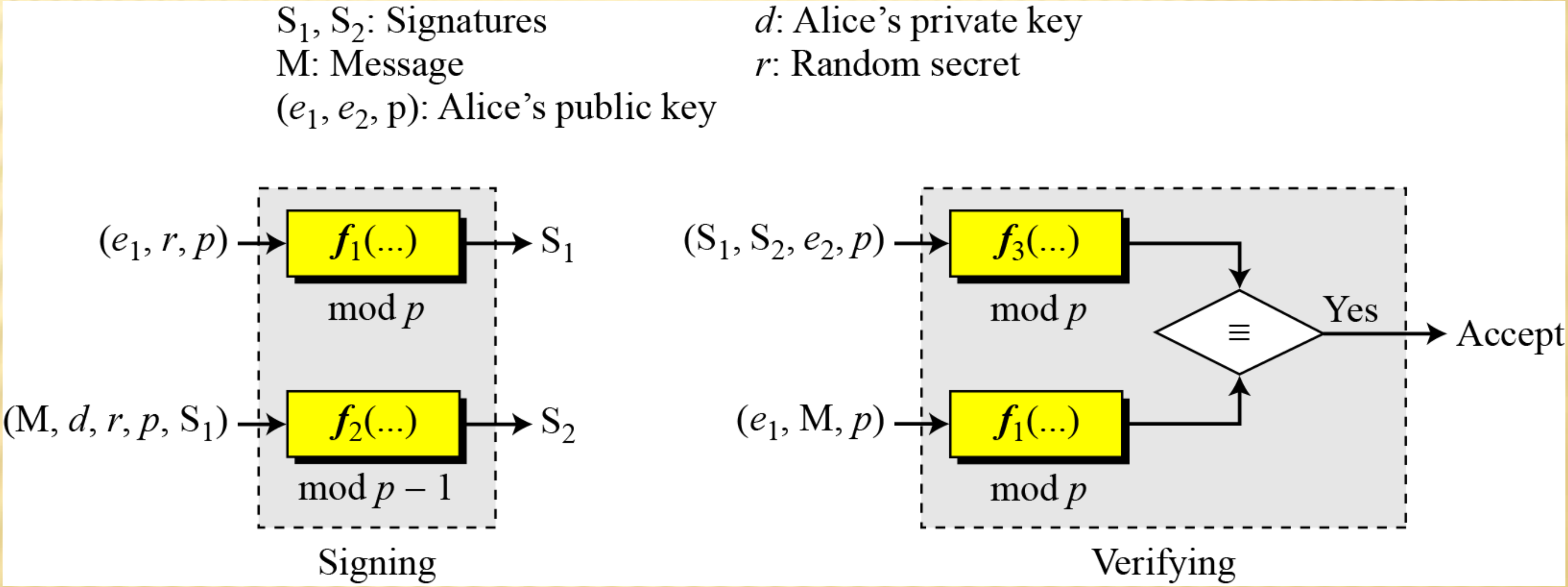
- Known message attack

- Assume that the attacker is having two message signature pairs (M1,S1) and(M2 , S2).
- The attacker calculates S = S1 * S2
- If the attacker can find a message M such that h (M) = h (M1) * h (M2) the attacker has a forged new message.

Chosen message attack

- Alice signs two messages M_1 and M_2 for eve.
- Eve then creates a new signature $S = S_1 \times S_2$.
- Eve can calculate $h(M) = h(M_1) \times h(M_2)$
- Given h (M) if eve can find a message M, the new message is forgery.

ELGAMAL DIGITAL SIGNATURE SCHEME



- Key Generation

- Let P be a large prime number .
- Let e_1 be the primitive element in Z_p^*
- Alice chooses her private key as d which is less than p-1.
- $e_2 = e_1^d \bmod p$
- The public key is (e_1, e_2, p).

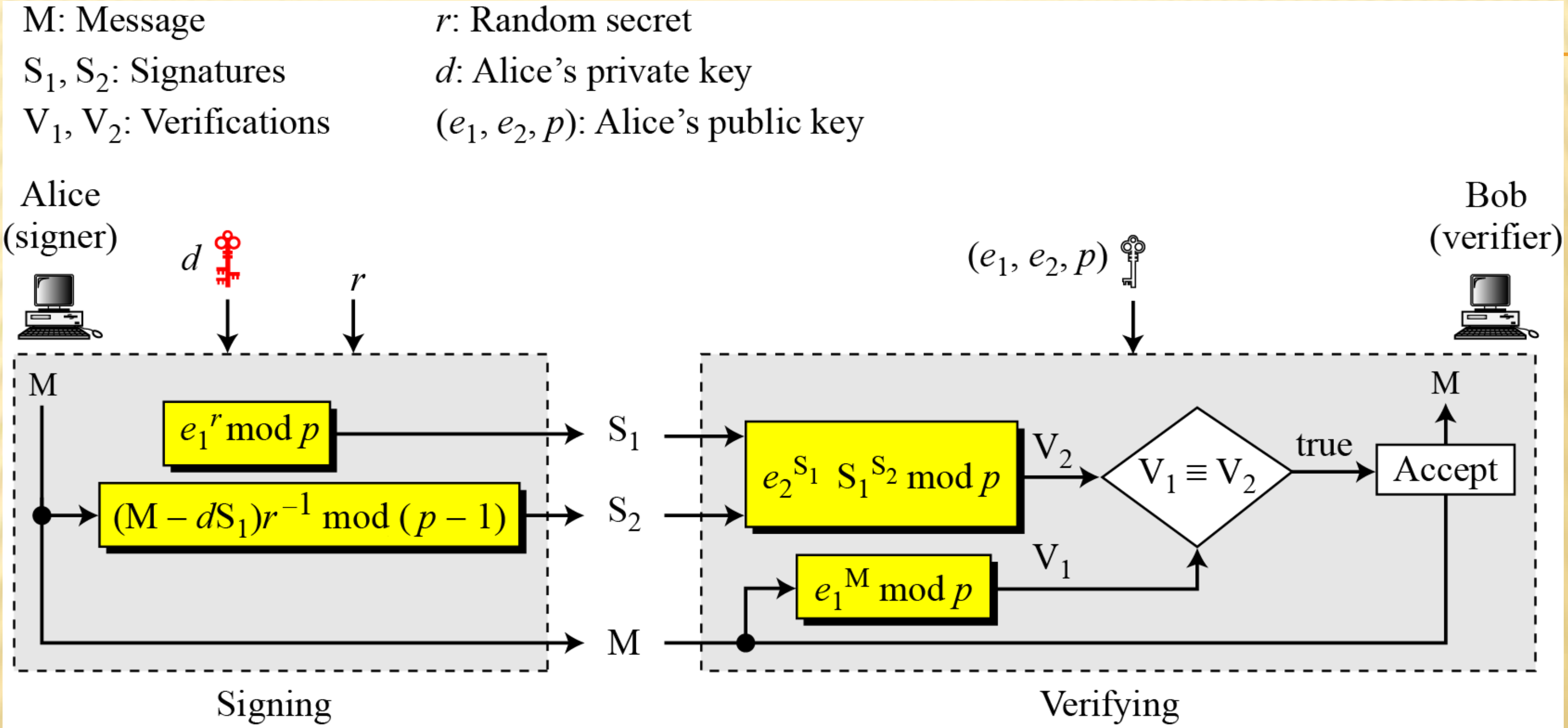


Fig: ElGamal digital signature scheme

SIGNING

- Alice chooses a secret no r.
  - Public and private keys can be used repeatedly.
  - Sender needs a new r each time she signs a new message.
- Alice calculates the first signature $S_1 = e_1^r \bmod p$
- Alice calculates the second signature $S_2 = (M - d \times S_1) \times r^{-1} \bmod (p - 1)$
- Alice sends M, S_1 , and S_2 to Bob

VERIFYING

- Bob checks if $0 < S_1 < p$
- Bob checks if $0 < S_2 < p - 1$
- Bob calculates $V_1 = e_1^M \bmod p$
- Bob calculates $V_2 = e_2^{S1} \times S_1^{s2} \bmod p$

- If V_1 is congruent to $V_{2,}$, the message is accepted, otherwise it's rejected.