

Digital Signature

Dr. B C Dhara

Department of Information Technology

Jadavpur University

Objectives

- Define a digital signature
- Define security services provided by a digital signature
- Define attacks on digital signatures
- Discuss some digital signature schemes including
 - RSA, ElGamal, Schnorr, DSS, and elliptic curve
- Describe some applications of digital signatures

Introduction

- Signature on a document is a sign of authentication
 - Document is authentic
- A sender ‘X’ sends a message to person ‘Y’
 - The receiver needs to check the authenticity of the sender
 - Receiver needs to be sure that message comes from right person, not from third person
 - The signature procedure can be done electronically, which is known as digital signature

Conventional vs. Digital signature

Conventional signature

- *A conventional signature is included in the document, it is part of the document*
- *For a conventional signature, when the recipient receives a document, compares the signature on the document with the signature on file*

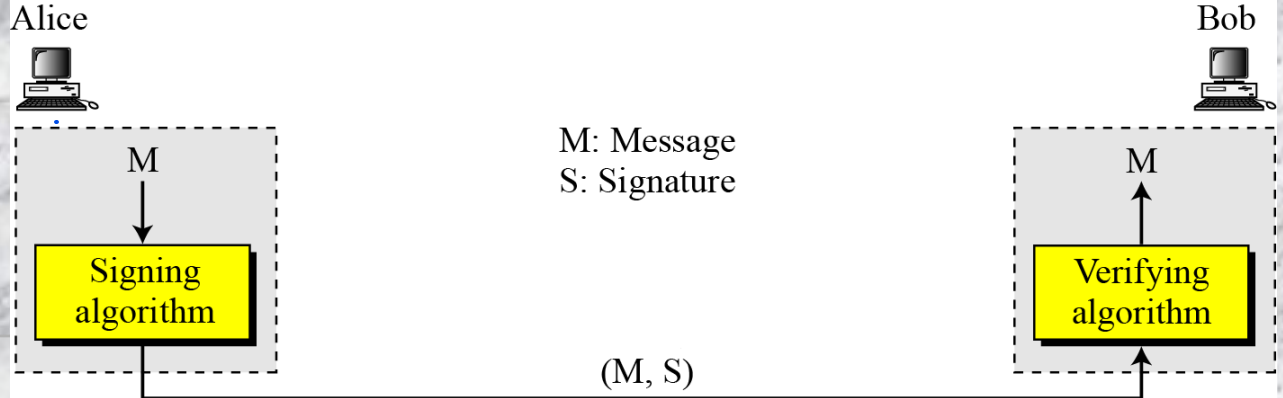
Digital signature

- *When we sign a document digitally, we send the signature as a separate document*
- *For a digital signature, the recipient receives the message and the signature*
 - *The recipient needs to apply a verification technique to the combination of the message and the signature to verify the authenticity*

Conventional vs. Digital signature (contd...)

- *For a conventional signature, there is normally a one-to-many relationship between a signature and documents*
- *In conventional signature, a copy of the signed document can be distinguished from the original one on file*
- *For a digital signature, there is a one-to-one relationship between a signature and a message*
- *In digital signature, there is no such distinction unless there is a factor of time on the document*

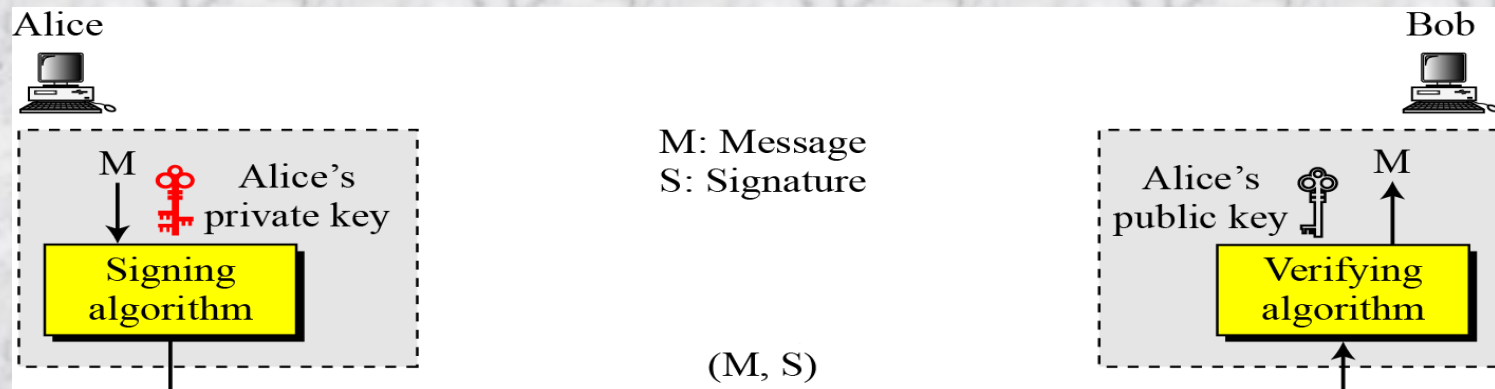
Process



- *The sender uses a signing algorithm to sign the message*
- *The message and the signature are sent to the receiver*
- *The receiver receives the message and the signature and applies the verifying algorithm to the combination*
 - *If the result is true, the message is accepted; otherwise, it is rejected.*

Key in DS

- A digital signature uses the private key of the sender in the signing algorithm
- The receiver (or verifier) uses the public key of the signer in the verifying algorithm to verify the document



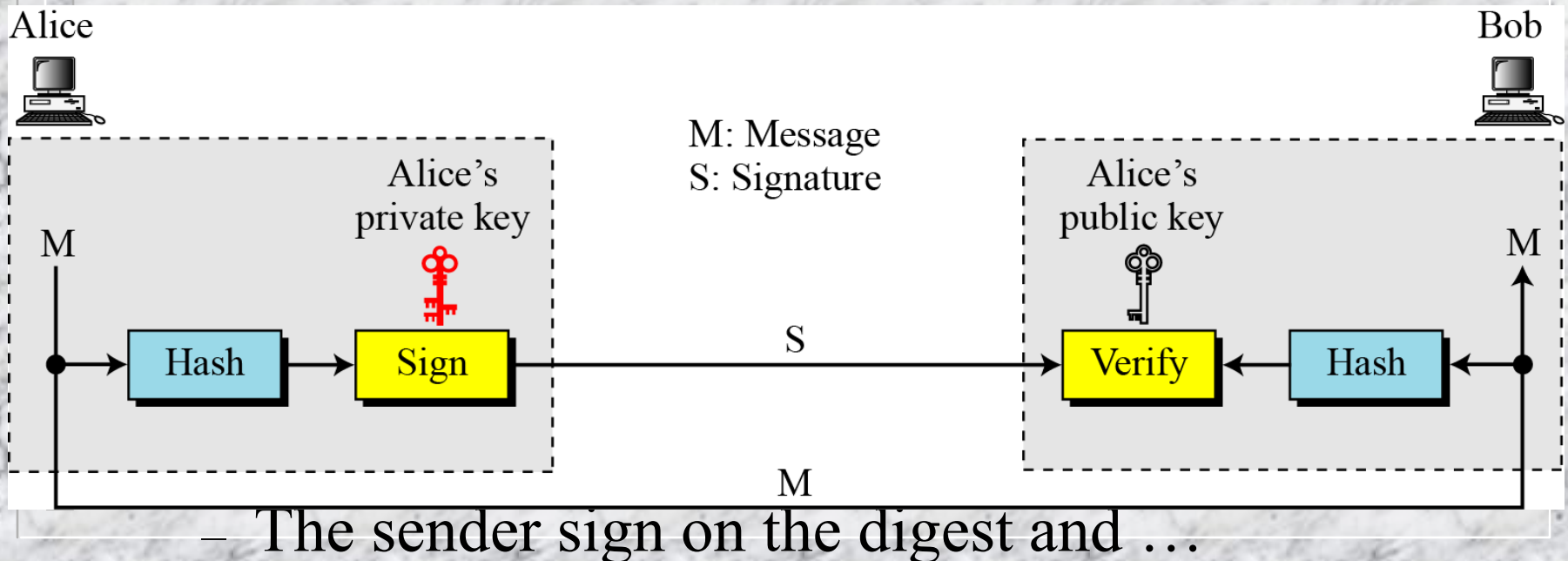
Signing the Digest

- Can we use symmetric key in digital signature?
- If yes, then for different pair of users, different keys are needed
- Also, for different session we may need different key

A cryptosystem uses the private and public keys of the receiver: a digital signature uses the private and public keys of the sender.

Signing the Digest (contd...)

- Asymmetric key is useful technique in digital signature
 - Asymmetric key inefficient for long message



SERVICES

- *Important concerns of this subject are*
 - *message confidentiality*
 - *message authentication*
 - *message integrity and*
 - *Nonrepudiation*
- *A digital signature can directly provide the last three; for message confidentiality we still need encryption/decryption.*

Message authentication

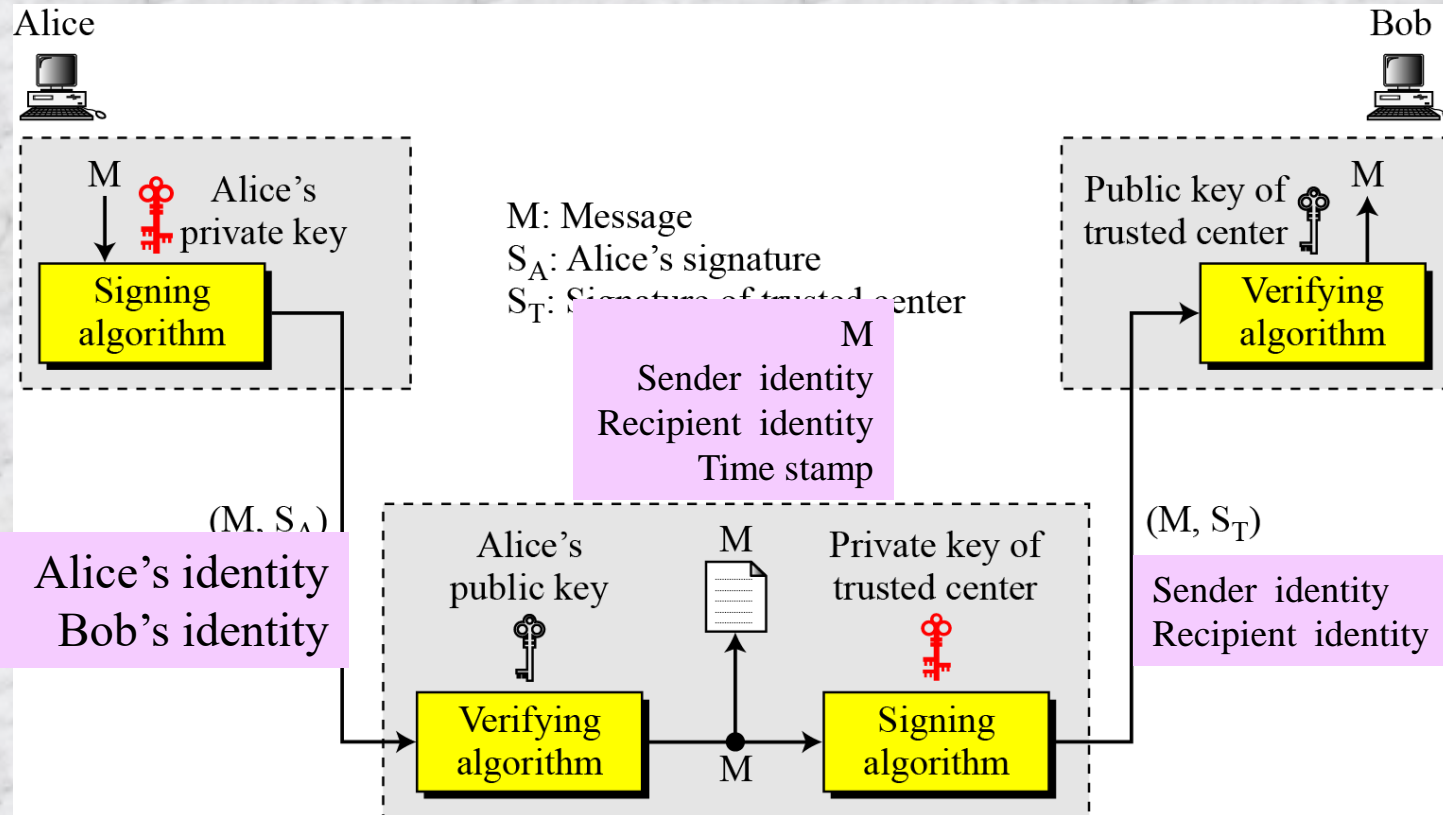
- A secure signature scheme (designed with private key of sender) can provide message authentication (data-origin authentication)
- Authenticity is verified by the public key of the sender
 - Public key of the third person cannot verify the signature

Message Integrity

- *The integrity of the message is preserved even if we sign the whole message because we cannot get the same signature if the message is changed.*

A digital signature provides message integrity.

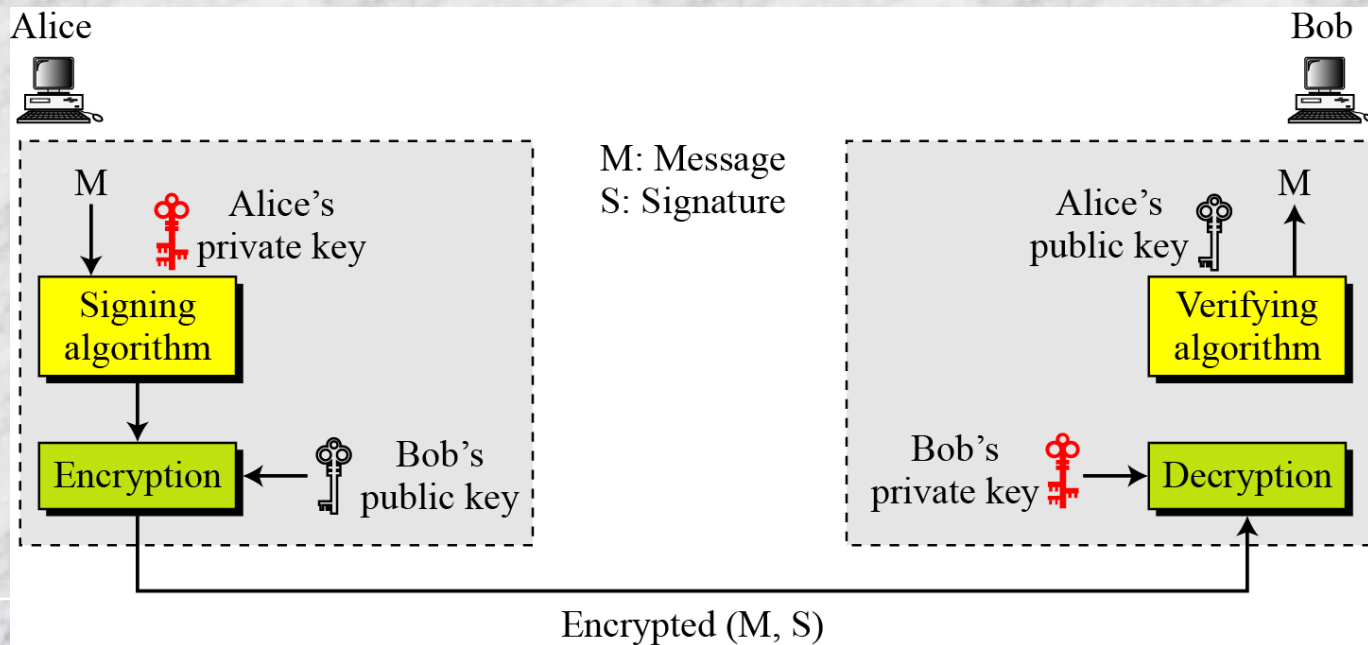
Nonrepudiation



Nonrepudiation can be provided using a trusted party

Confidentiality

- ❑ A digital signature does not provide privacy
- ❑ If there is a need for privacy, another layer of encryption/decryption must be applied



ATTACKS ON DIGITAL SIGNATURE

□ *This section describes some attacks on digital signatures and defines the types of forgery*

□ Attack types:

- Key only attack
- Known message attack
- Chosen message attack

□ Forgery types:

- Existential forgery
- Selective forgery

Attacks on digital signatures

- Key only attack, attacker can only access public information released by sender. This attack is same as the ciphertext-only attack
- Known message attack, attacker has access one or more message-signature pairs. This is similar to the known-plaintext attack
- Chosen-message attack, this is similar as chosen-plaintext attack

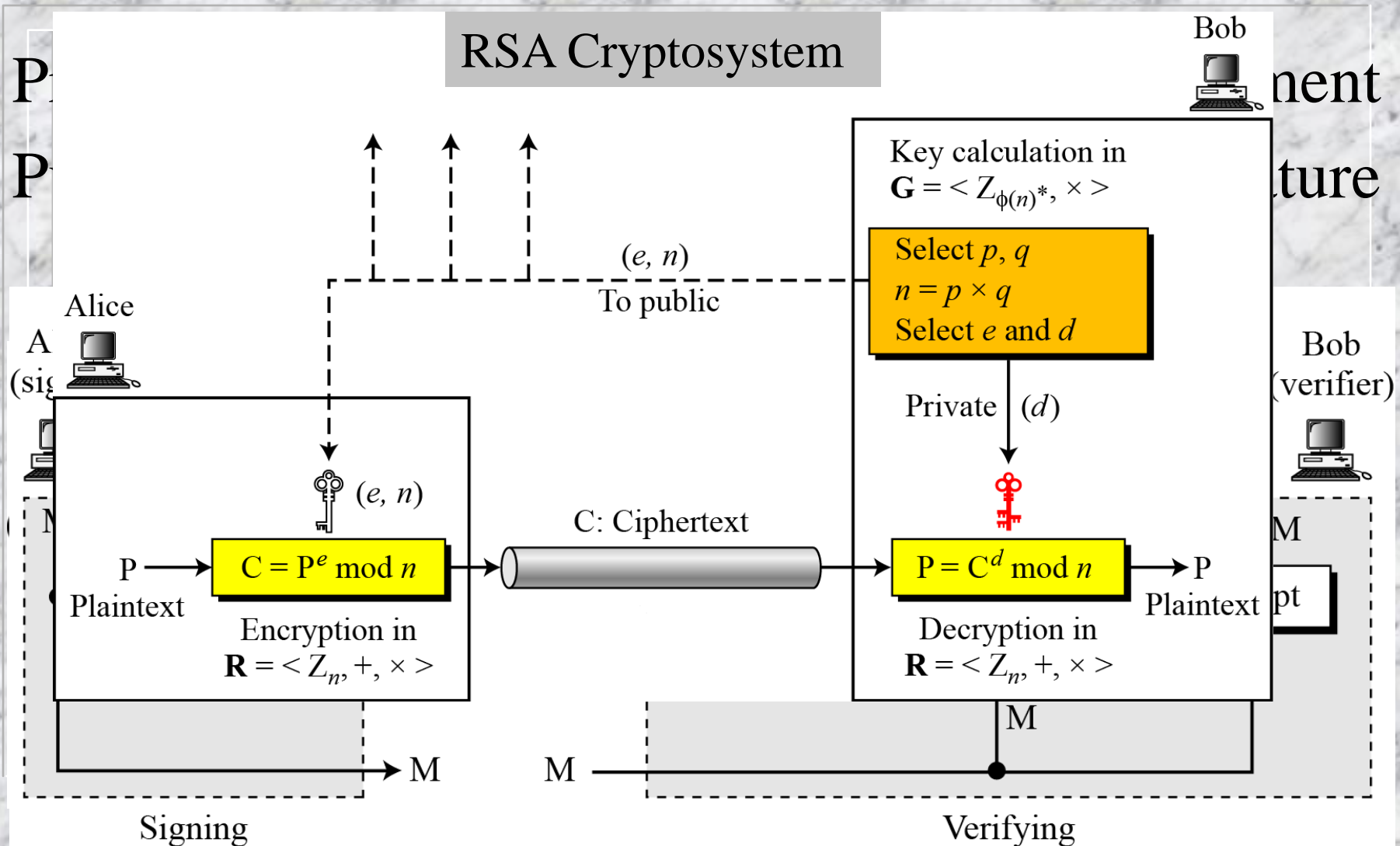
Types of forgery

- If attack is successful, result is a forgery
- Existential Forgery: the attacker may be able to compute valid message-signature pair, but cannot be used as content of the document randomly calculated
 - Message could be syntactically or semantically unintelligible
- Selective Forgery: the attacker may be able to forge sender's signature of a message
 - This may be very harmful to sender

Digital signature schemes

- ❑ RSA Digital Signature Scheme
- ❑ ElGamal Digital Signature Scheme
- ❑ Schnorr Digital Signature Scheme
- ❑ Digital Signature Standard (DSS)
- ❑ Elliptic Curve Digital Signature Scheme

RSA Digital Signature Scheme



Example: RSA Digital Signature Scheme

As a trivial 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:

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

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

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

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

Attacks on RSA Signature

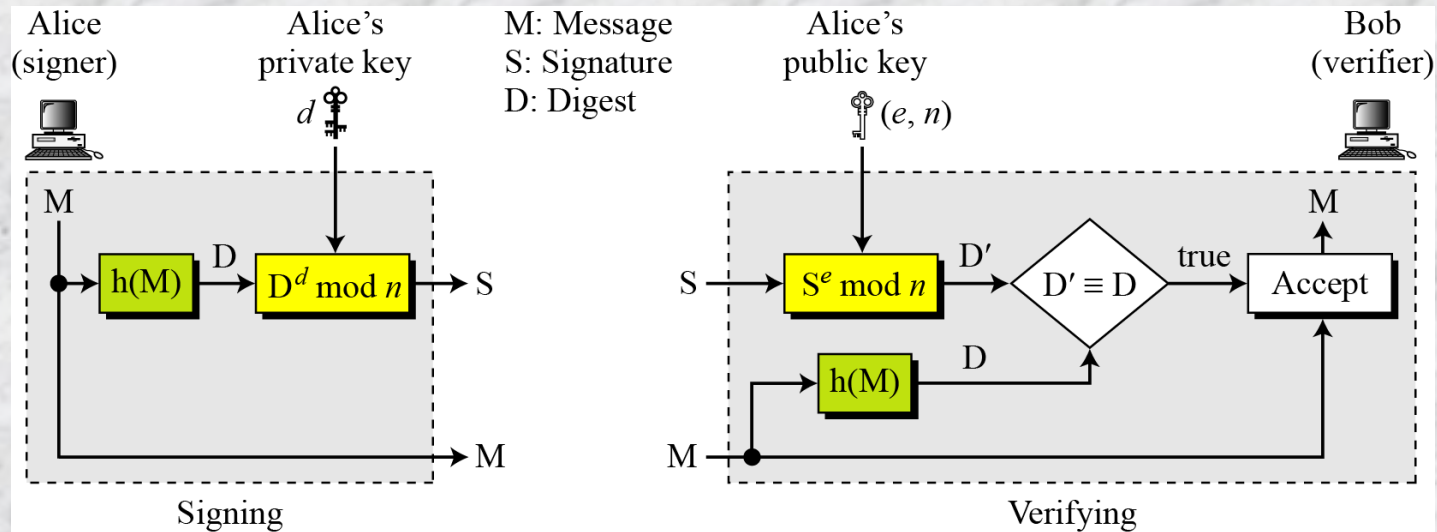
- Key only attack: attacker can intercepts (M, S) and try to find another message M' such that $M' \equiv S^e \pmod{n}$
 - This problem is as difficult to solve discrete logarithm problem
- Known-message attack: two message-signature pairs $(M1, S1)$ and $(M2, S2)$, computed by same private key, are intercepted and attacker send (M, S) to receiver to fool-ed him
 - $M = M1 * M2 \pmod{n}$ and $S = S1 * S2 \pmod{n}$
 - This existential forgery, and useless
 - This attack known as multiplicative key

Attacks on RSA Signature (contd...)

- Chosen plaintext attack: somehow the attacker (Eve) manages sender to sign on two messages $M1$ and $M2$
 - Later, the attacker may claim that sender has signed on $M = M1 * M2$, which is a serious problem
 - This is a selective forgery

RSA Signature on the Message Digest

- Several advantages like: signing and verifying is faster
- If a strong cryptographic hash function is used, attack on signature will be more difficult



Attacks on RSA Signed on Digest

- The attacks like on RSA signature are possible
- When the digest is signed instead of the message itself, the susceptibility of the RSA digital signature scheme depends on the strength of the hash algorithm

ElGamal Digital Signature Scheme

ElGamal Cryptosystem

$d \in G = \langle \mathbb{Z}_p^*, \times \rangle, 1 \leq d \leq p-2$
 e_1 primitive root in $G = \langle \mathbb{Z}_p^*, \times \rangle$

Bob



Alice



Public key: (e_1, e_2, p)

(e_1, e_2, p)



Plaintext $P \rightarrow$
 $C_1 = e_1^r \bmod p$
 $C_2 = (e_2^r \times P) \bmod p$

Encryption

Ciphertext: (C_1, C_2)

Key generation

Select p (very large prime)
Select e_1 (primitive root)
Select d
 $e_2 = e_1^d \bmod p$

Private key: d

d



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

Decryption

Plaintext P

ElGamal Digital Signature Scheme

M: Message

r : Random secret

S_1, S_2 : Signatures

d : Alice's private key

V_1, V_2 : Verifications

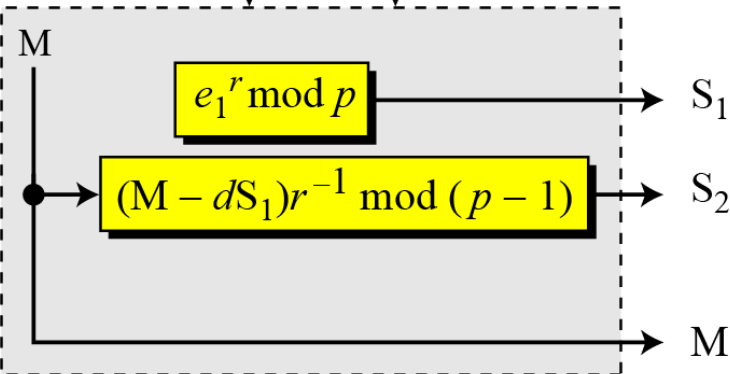
(e_1, e_2, p) : Alice's public key

Alice
(signer)




d 

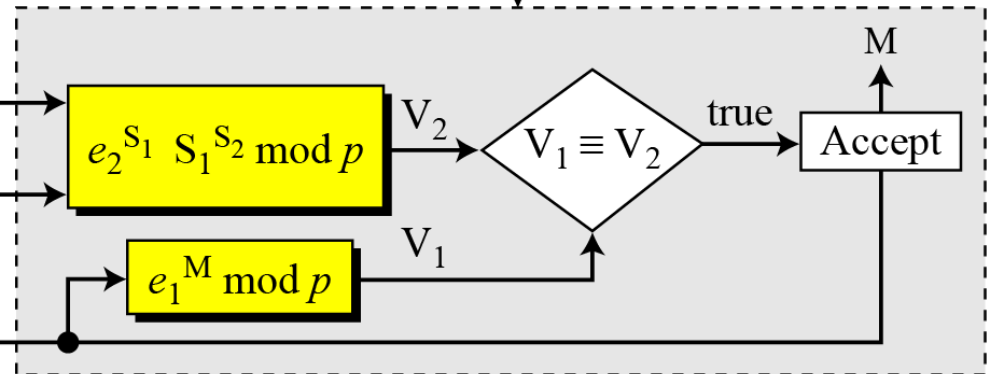
r



Signing

(e_1, e_2, p) 

Bob
(verifier)



Verifying

Example: ElGamal Digital Signature

Here is a trivial example. Alice chooses $p = 3119$, $e_1 = 2$, $d = 127$ and calculates $e_2 = 2^{127} \bmod 3119 = 1702$. She also chooses r to be 307. She announces e_1 , e_2 , and p publicly; she keeps d secret. The following shows how Alice can sign a message.

$$M = 320$$

$$S_1 = e_1^r = 2^{307} = 2083 \bmod 3119$$

$$S_2 = (M - d \times S_1) \times r^{-1} = (320 - 127 \times 2083) \times 307^{-1} = 2105 \bmod 3118$$

Alice sends M , S_1 , and S_2 to Bob. Bob uses the public key to calculate V_1 and V_2 .

$$V_1 = e_1^M = 2^{320} = 3006 \bmod 3119$$

$$V_2 = d^{S_1} \times S_1^{S_2} = 1702^{2083} \times 2083^{2105} = 3006 \bmod 3119$$

Example: ElGamal Digital Signature

Now imagine that Alice wants to send another message, $M = 3000$, to Ted. She chooses a new r , 107. Alice sends M , S_1 , and S_2 to Ted. Ted uses the public keys to calculate V_1 and V_2 .

$$M = 3000$$

$$S_1 = e_1^r = 2^{107} = 2732 \bmod 3119$$

$$S_2 = (M - d \times S_1) r^{-1} = (3000 - 127 \times 2083) \times 107^{-1} = 2526 \bmod 3118$$

$$V_1 = e_1^M = 2^{3000} = 704 \bmod 3119$$

$$V_2 = d^{S_1} \times S_1^S = 1702^{2732} \times 2083^{2526} = 704 \bmod 3119$$

Digital Signature Standard (DSS)

- DSS was adopted by NIST in 1994
- DSS uses a digital signature algorithm (DSA) based on ELGamal scheme with some idea from Schnorr

S_1, S_2 : Signatures

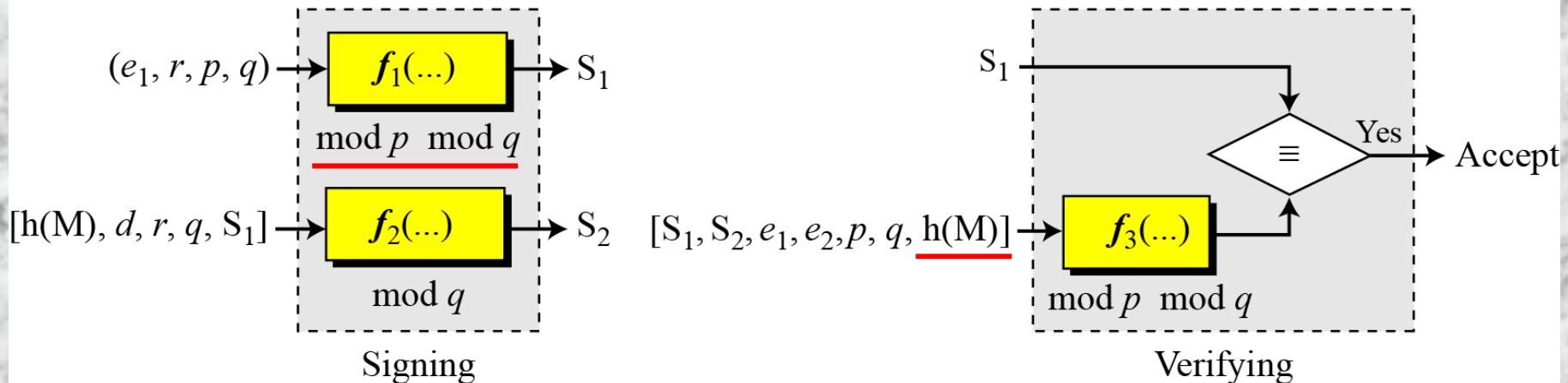
M: Message

(e_1, e_2, p, q) : Alice's public key

d : Alice's private key

r : Random secret

General idea behind DSS scheme



Key Generation

- 1) Alice chooses primes p , between 512 and 1024 bits and multiple of 64
- 2) Choose another prime q of 160 bits and $q \mid (p-1)$
- 3) Alice uses $\langle \mathbb{Z}_p^*, \times \rangle$ and $\langle \mathbb{Z}_q^*, \times \rangle$, second is subgroup of first
- 4) Alice chooses e_1 to be the q th root of 1 modulo p
 - 1) For a primitive element e_0 in \mathbb{Z}_p , $e_1 = e_0^{(p-1)/q} \bmod p$
- 5) Alice chooses an integer, d , as her private key
- 6) Alice calculates $e_2 = e_1^d \bmod p$
- 7) Alice's public key is (e_1, e_2, p, q) ; her private key is (d) .

Verifying and Signing

M: Message

r : Random secret

$h(M)$: Message digest

S_1, S_2 : Signatures

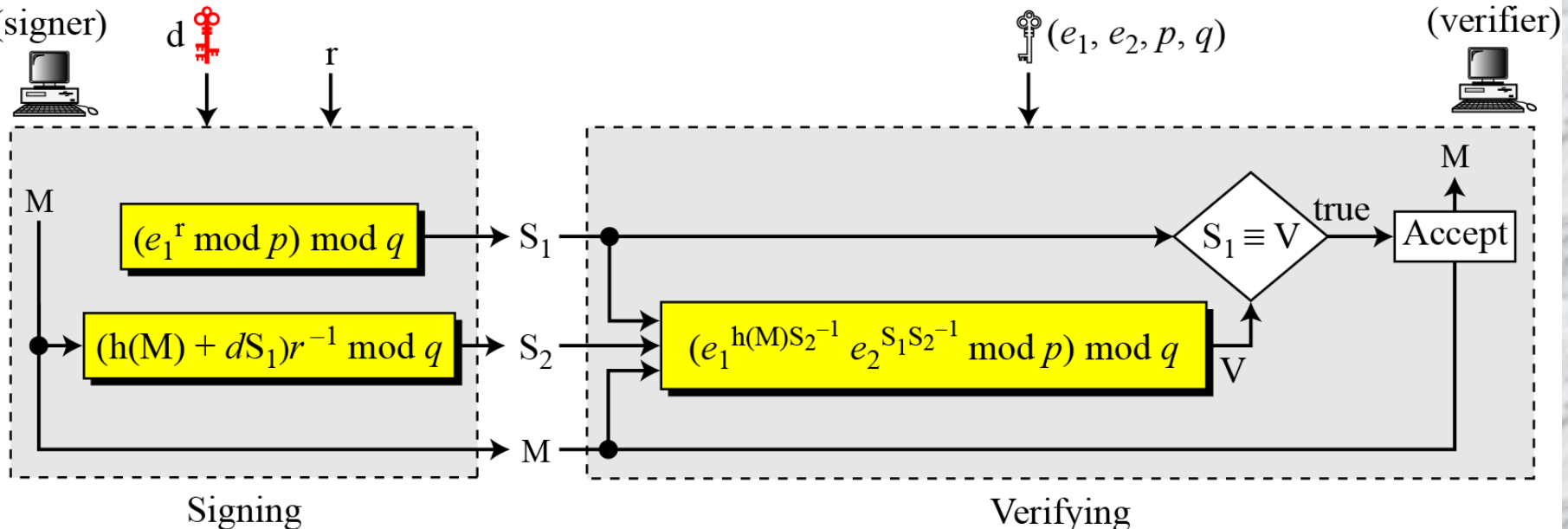
d : Alice's private key

V: Verification

(e_1, e_2, p, q) : Alice's public key

Alice
(signer)

Bob
(verifier)



Example: DSS

Alice chooses $q = 101$ and $p = 8081$. Alice *selects* $e_0 = 3$ and calculates $e^1 = e_0^{(p-1)/q} \bmod p = 6968$. Alice chooses $d = 61$ as the private key and calculates $e_2 = e_1^d \bmod p = 2038$. Now Alice can send a message to Bob. Assume that $h(M) = 5000$ and Alice chooses $r = 61$:

$$h(M) = 5000 \quad r = 61$$

$$S_1 = (e_1^r \bmod p) \bmod q = 54$$

$$S_2 = ((h(M) + d S_1) r^{-1}) \bmod q = 40$$

Alice sends M , S_1 , and S_2 to Bob. Bob uses the public keys to calculate V .

$$S_2^{-1} = 48 \bmod 101$$

$$V = [(6968^{5000 \times 48} \times 2038^{54 \times 48}) \bmod 8081] \bmod 101 = 54$$