Digital Signatures

Introduction

- A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.
- The signature is formed by taking the hash of the message and encrypting the message with the creator's private key.
- The signature guarantees the source and integrity of the message
- a signature on a document, when verified, is a sign of authentication – the document is authentic
- When Alice sends a message to Bob, Bob needs to check the authenticity of the sender; he needs to be sure that the message comes from Alice and not Eve. Bob can ask Alice to sign the message electronically.
- In other words, an electronic signature can prove the authenticity of Alice as the sender of the message. - this type of signature is referred to as a **digital signature**.

Differences between conventional signatures and digital signatures

- **Inclusion**-A conventional signature is included in the document; it is part of the document
- But when we sign a document digitally, we send the signature as a separate document.
- The sender sends two documents: the message and the signature.
- The recipient receives both documents and verifies that the signature belongs to the supposed sender.
- If this is proven,the message is kept; otherwise, it is rejected.

Verification Method

- Conventional signature
  - when the recipient receives a document, she compares the signature on the document with the signature on file. If they are the same, the document is authentic.
  - The recipient needs to have a copy of this signature on file for comparison.
- Digital signature
  - the recipient receives the message and the signature
  - Copy of the signature is not stored anywhere.
  - The recipient needs to apply a verification technique to the combination of the message and the signature to verify the authenticity.

Relationship

Conventional signature

- there is normally a one-to-many relationship between a signature and documents.
- A person uses the same signature to sign many documents.
- For a digital

Digital signature

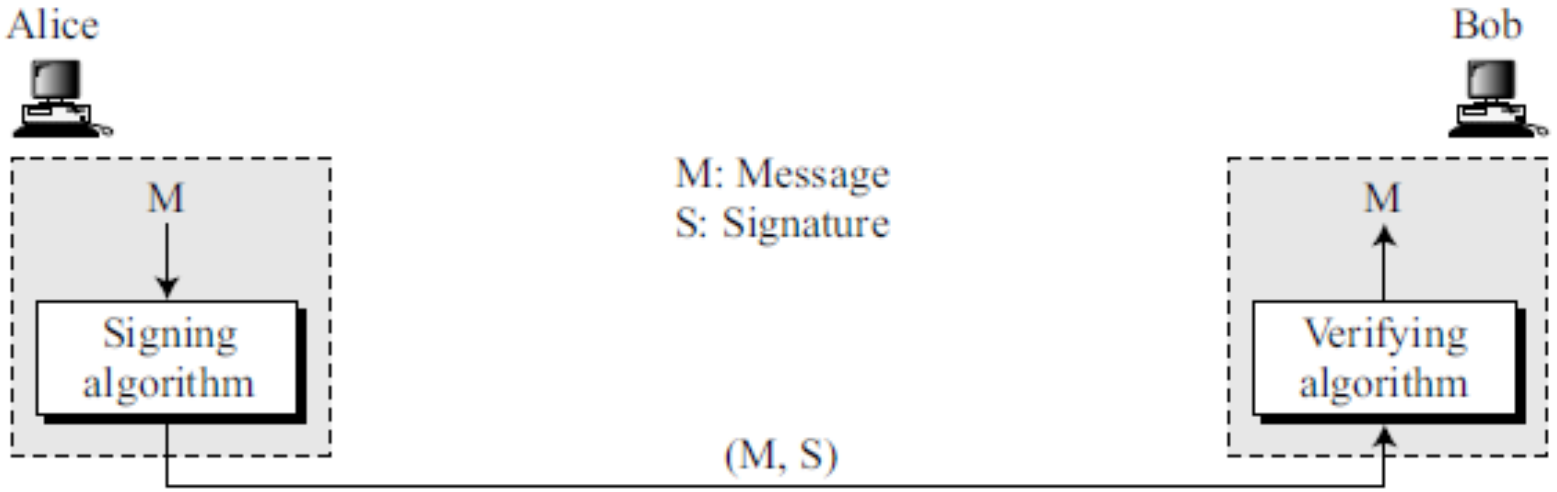
- there is a one-to-one relationship between a signature and a message.
- Each message has its own signature.
- The signature of one message cannot be used in another message.
- If Bob receives two messages, one after another, from Alice, he cannot use the signature of the first message to verify the second. Each message needs a new signature

Duplicity

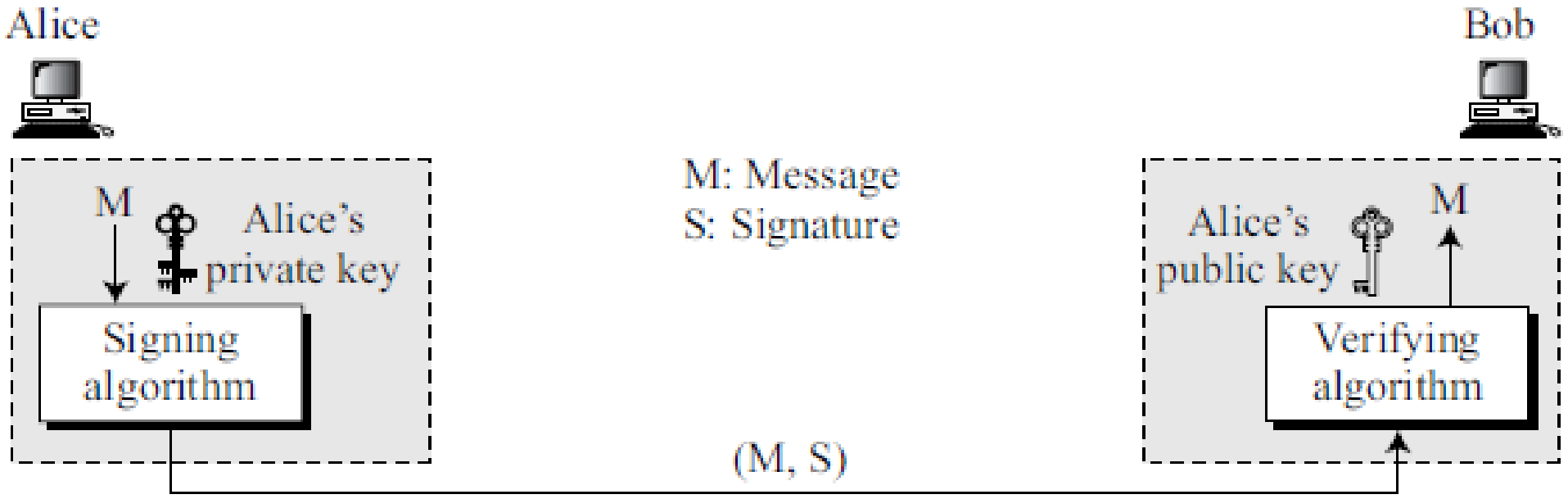
- conventional signature
  - a copy of the signed document can be distinguished from the original one on file.
  - In, there is no such distinction unless there is a
- digital signature
  - no such distinction unless there is aactor of time (such as a timestamp) on the document.
  - For example, suppose Alice sends a document instructing Bob to pay Eve. If Eve intercepts the document and the signature, she can replay it later to get money again from Bob.

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.



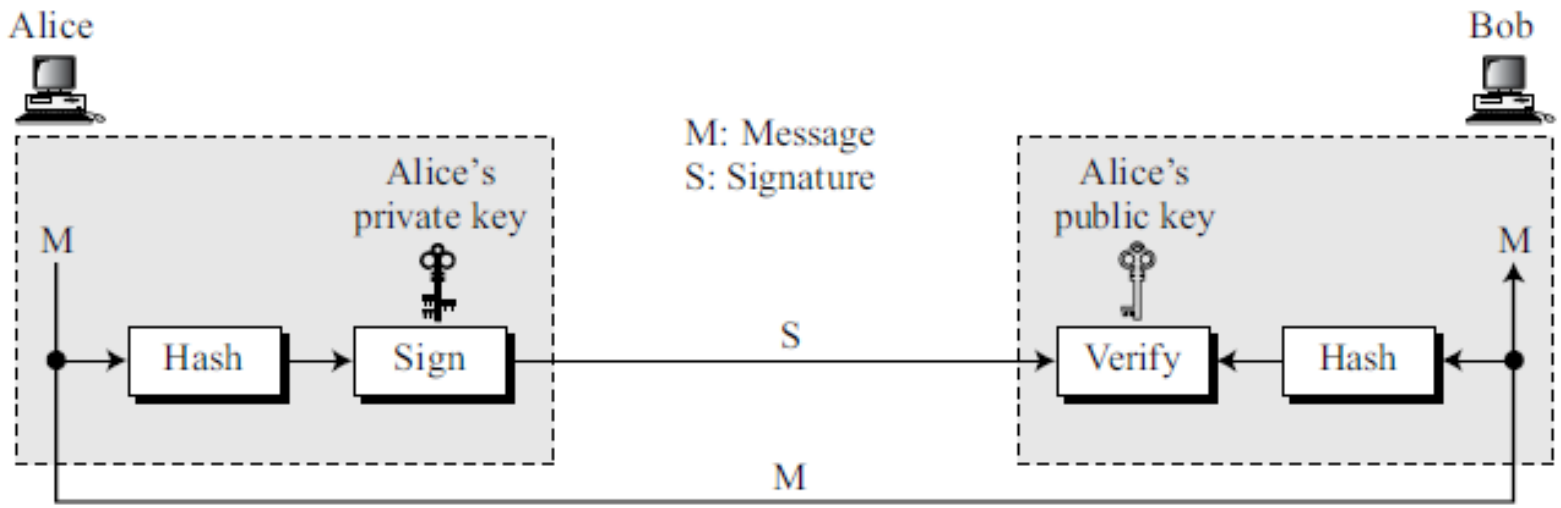
- the signer uses her private key, applied to a signing algorithm, to sign the document.
- The verifier, uses the public key of the signer, applied to the verifying algorithm, to verify the document



Can symmetric key be used to both sign and verify a signature?

- No.
- a secret key is known by only two entities (Alice and Bob, for example). So if Alice needs to sign another document and send it to Ted, she needs to use another secret key.
- Creating a secret key for a session involves authentication, which uses a digital signature.
- Bob could use the secret key between himself and Alice, sign a document, send it to Ted, and pretend that it came from Alice.

Signing the Digest



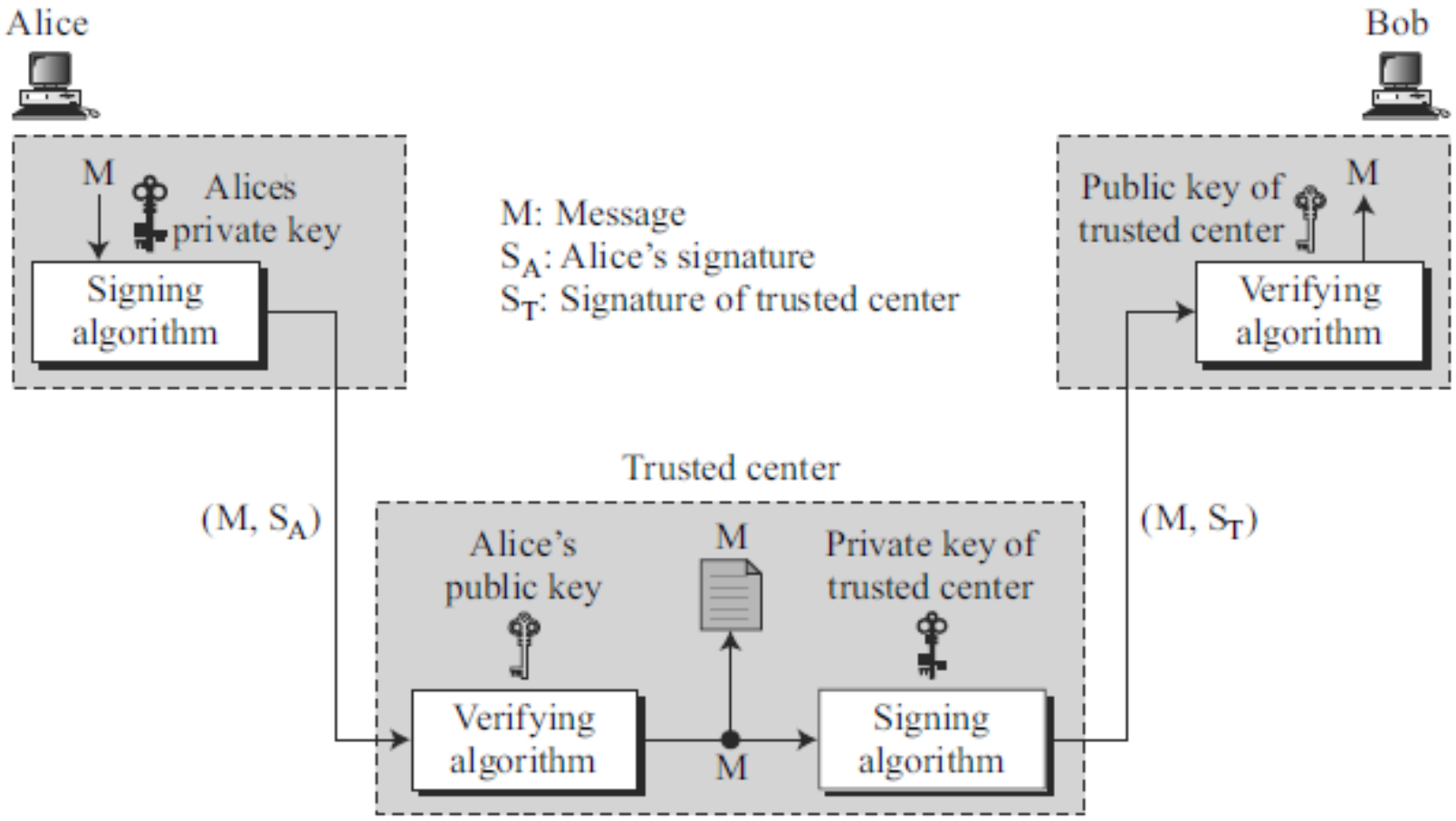
SERVICES

- A digital signature can directly provide
  - message authentication
  - message integrity
  - nonrepudiation.
- A secure digital signature scheme like a secure conventional signature can provide message authentication (also referred to as data-origin authentication).
- Bob can verify that the message is sent by Alice because Alice’s public key is used in verification.
- Alice’s public key cannot verify the signature signed by Eve’s private key.

- **Message Integrity-**
  - cannot get the same signature if the message is changed.
  - The digital signature schemes today use a hash function in the signing and verifying algorithms that preserve the integrity of the message
- **Non Repudiation**
- If Alice signs a message and then denies it, can Bob later prove that Alice actually signed it?
- Suppose Alice sends a message to a bank (Bob) and asks to transfer some amount from her account to Ted's account, can Alice later deny that she sent this message?
- With this scheme Bob must keep the signature on file and later use Alice's public key to create the original message to prove the message in the file and the newly created message are the same.
  - not feasible because Alice may have changed her private or public key during this time
  - may also claim that the file containing the signature is not authentic.

  Solution- Trusted third party

Trusted third party

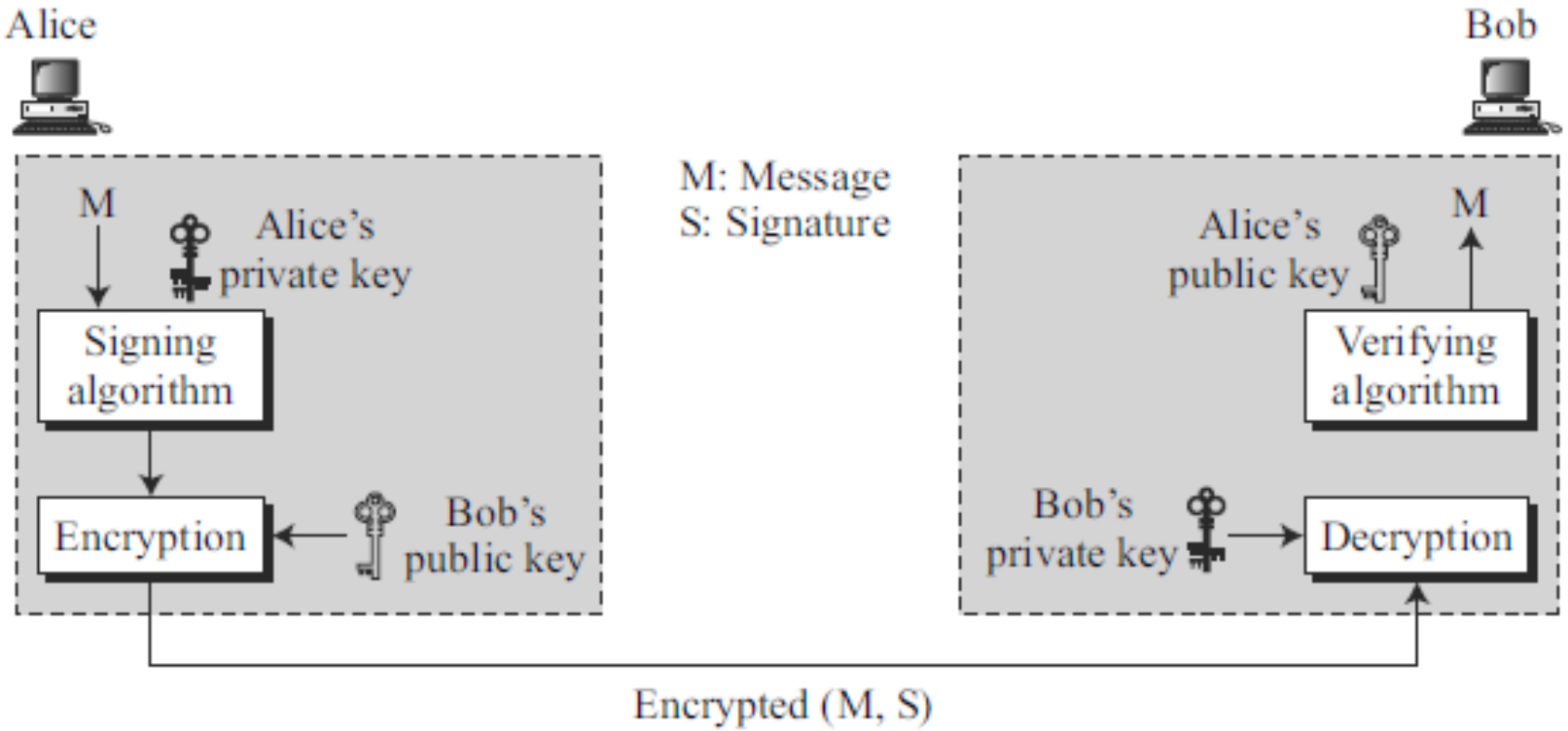


Trusted third party Contd…

- Alice creates a signature from her message (SA) and sends the message, her identity, Bob's identity, and the signature to the center.
- The center, after checking that Alice's public key is valid, verifies through Alice's public key that the message came from Alice
- The center then saves a copy of the message with the sender identity, recipient identity, and a timestamp in its archive.
- The center uses its private key to create another signature (ST) from the message.
- The center then sends the message, the new signature, Alice's identity, and Bob's identity to Bob.
- Bob verifies the message using the public key of the trusted center
- If in the future Alice denies that she sent the message, the center can show a copy of the saved message.
- If Bob's message is a duplicate of the message saved at the center, Alice will lose the dispute.

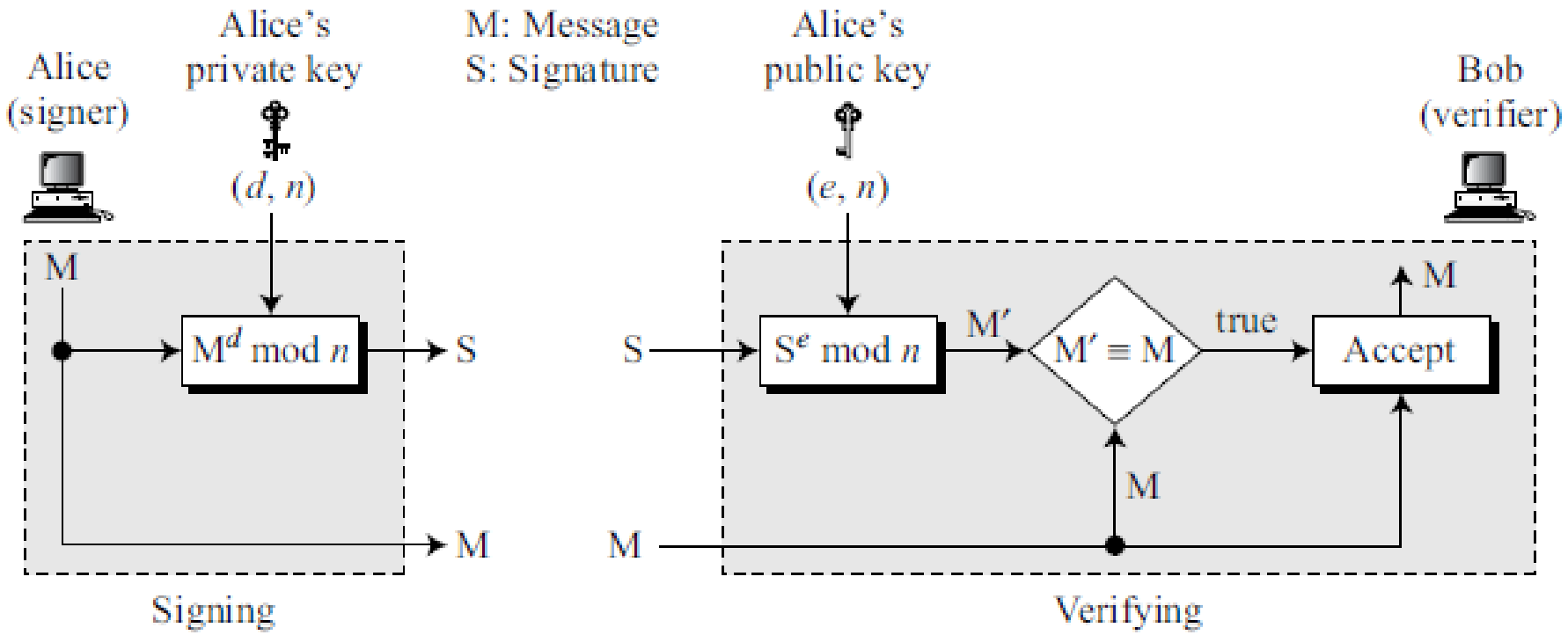
Confidentiality

- A digital signature does not provide confidential communication
- If confidentiality is required, the message and the signature must be encrypted using either a secret-key or public-key cryptosystem



DIGITAL SIGNATURE SCHEMES

- RSA Digital Signature Scheme



Example1

Let p←7,q←13, Choose e←5

Find d

Sign the message m=35 and verify the signature

Example1 -Solution

Let $p \leftarrow 7, q \leftarrow 13$, Choose $e \leftarrow 5$

Find d

Sign the message m=35

$d \leftarrow 29$

$s = m^d \bmod n = 35^{29} \bmod n = 42$

(m,s)=(35,42)

$M' = 42^5 \equiv 35 \bmod n$.

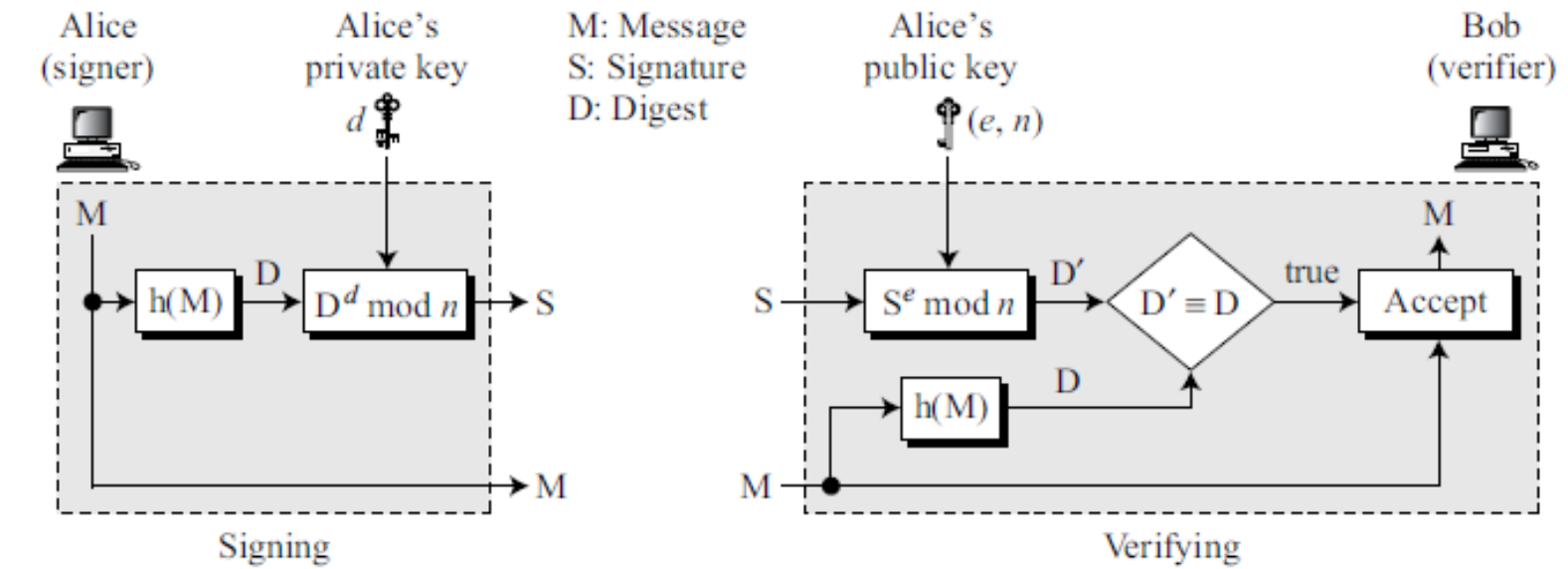
Example2

Let p←53,q←59, Choose e←5

Find d

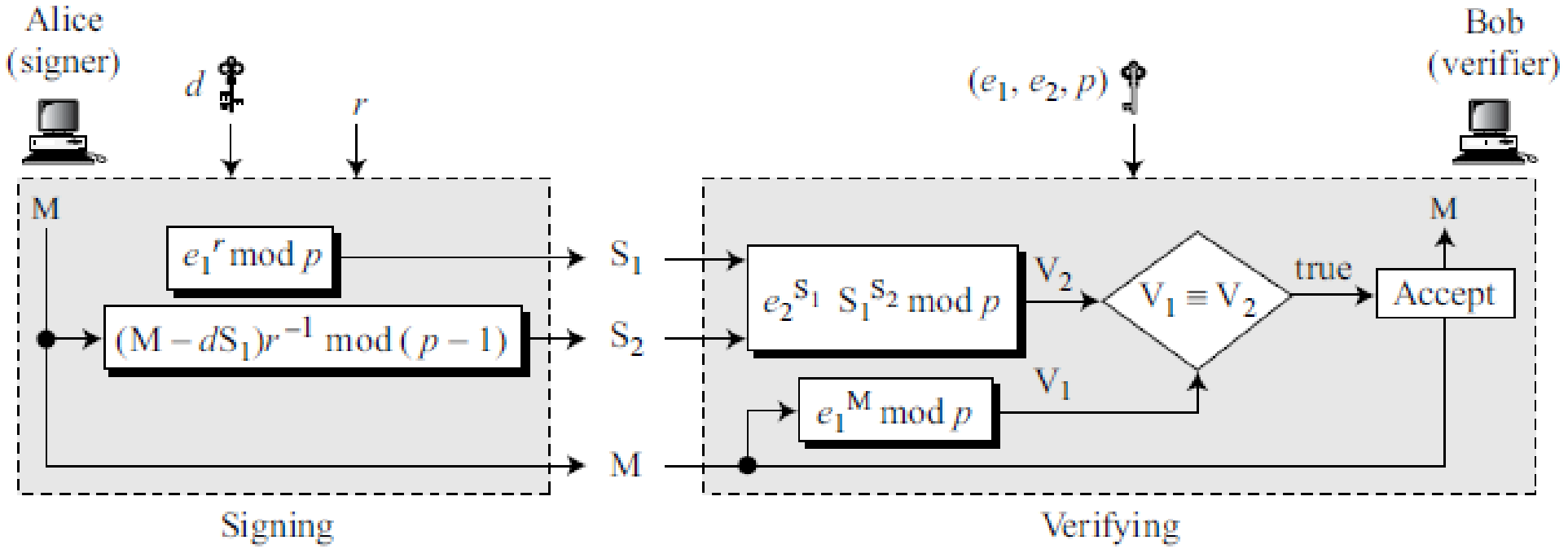
Sign the message m=1394 and verify the signature

The RSA signature on the message digest



ElGamal Digital Signature Scheme

M: Message
S_1, S_2: Signatures
V_1, V_2: Verifications
r: Random secret
d: Alice's private key
(e_1, e_2, p): Alice's public key



Signing

- Alice chooses a secret random number r.
- Though public and private keys can be used repeatedly, Alice needs a new r each time she signs a new message
- Alice calculates the first signature $S1 = e1^{r} \bmod p$.
- Alice calculates the second signature $S2 = (M - d \times S1) \times r^{-1} \bmod (p - 1)$, where r^{-1} is the multiplicative inverse of r modulo p-1.
- Alice sends M, S1, and S2 to Bob.

Verifying

Bob receives M, S1, and S2, which can be verified as follows:

1. Bob checks to see if $0 < S1 < p$
2. Bob checks to see if $0 < S2 < p - 1$
3. Bob calculates $V1 = e1^{M} \bmod p$
4. Bob calculates $V2 = e2^{S1} \times S1^{S2} \bmod p$
5. If V1 is congruent to V2, the message is accepted; otherwise, it is rejected.

Example

q = 19, d=16, r =5, e1=10 sign the message M=14 and verify the same

Example

q = 19, d=16, r =5, e1=10 sign the message M=14 and verify the same

S1 = $e1^r \bmod p = 10^5 \bmod 19 = 3$

E2= $e1^d \bmod p = 10^{16} \bmod 19 = 4$

$r^{-1} \bmod (p-1) = \times 5^{-1} \bmod 18) = 11$

S2 = $(M - d \times S1) \times r^{-1} \bmod (p-1) = (14 - 16 \times 3) \times 11 \bmod (18) = 4$

Sends 14, 3, 4 to Bob

Bob calculates V1 = $e1^M \bmod p = 10^{14} \bmod 19 = 16$

Bob calculates V2 = $e2^{S1} \times S1^{S2} \bmod p = 4^3 \times 3^4 \bmod 19 = 16$

Therefore Signature is valid

Schnorr Digital Signature Scheme

- The problem with the ElGamal digital signature scheme is that p needs to be very large to guarantee that the discrete log problem is intractable in Zp*.
- The recommendation is a p of at least 1024 bits
- This could make the signature as large as 2048 bits
- Schnorr proposed a new scheme based on ElGamal, but with a reduced signature size.

Key Generation

- Before signing a message, Alice needs to generate keys and announce the public ones to the public.

1. Alice selects a prime p, which is usually 1024 bits in length.

2. Alice selects another prime q, which is the same size as the digest created by the cryptographic hash function. The prime q needs to divide (p − 1). In other words, (p − 1) = 0 mod q.

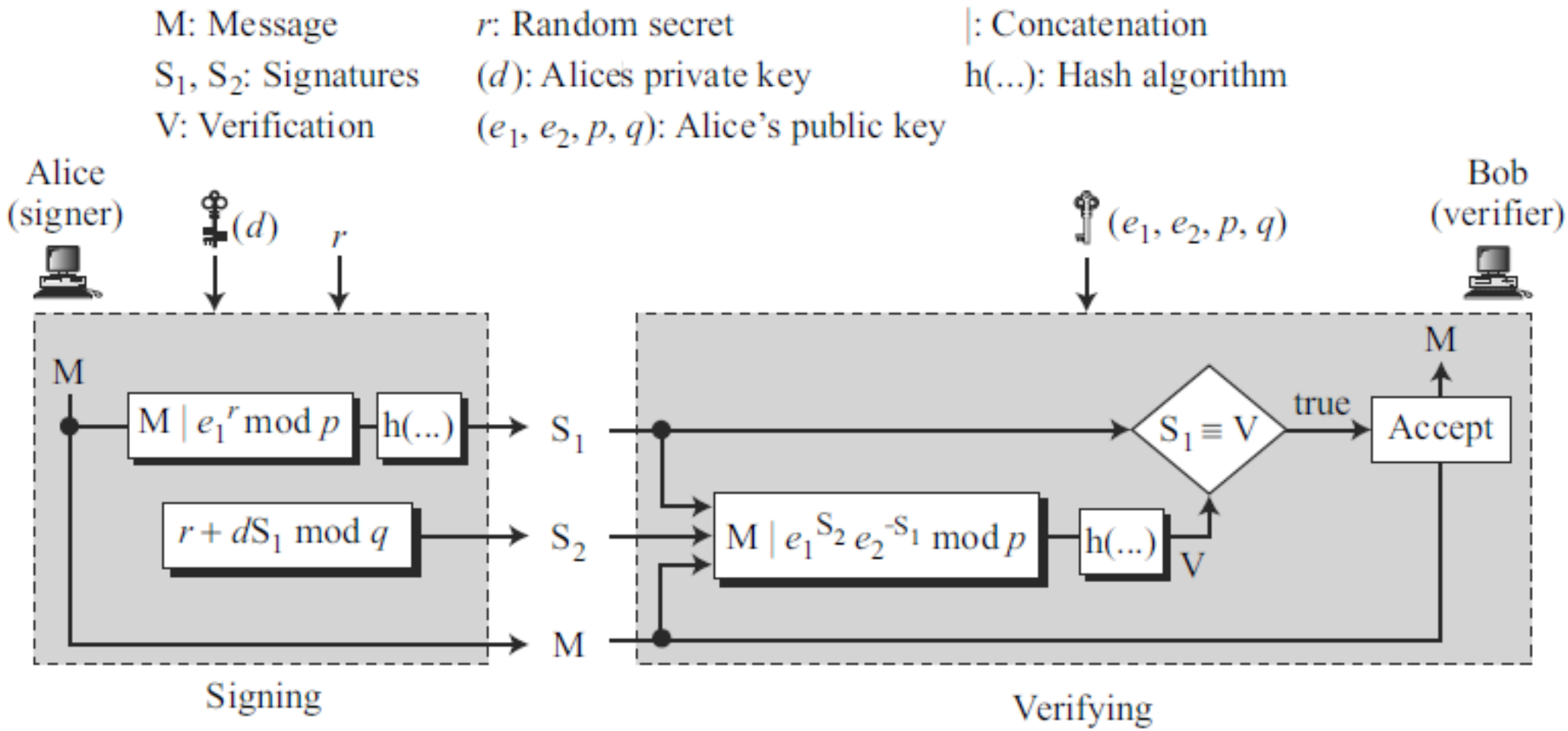
3. Alice chooses e1 to be the qth root of 1 modulo p. To do so, Alice chooses a primitive element in Zp, and calculates $e1 = e0^{(p-1)/q} \bmod p$.

4. Alice chooses an integer, d, as her private key.

5. Alice calculates $e2 = e1^{d} \bmod p$.

6. Alice's public key is (e1, e2, p, q); her private key is (d);

Signing and Verifying



Signing

1. Alice chooses a random number r. Although public and private keys can be used to sign multiple messages, Alice needs to change r each time she sends a new message. Note also that r needs to be between 1 and q.

2. Alice calculates the first signature S1 = h(M| $e1^r$ mod p).

The message is prepended to the value of $e1^r$ mod p then the hash function is applied to create a digest.

Alice calculates the second signature S2 = r + d × S1 mod q.

4. Alice sends M, S1, and S2.

Verifying Message

The receiver, Bob receives M, S1, and S2.

1. Bob calculates $V = h (M \mid e1^{S2} e2^{-S1} \bmod p)$.

2. If S1 is congruent to V modulo p, the message is accepted; otherwise, it is rejected.

1. Alice selects a prime p=31
2. Alice selects another prime q, such that $(p - 1) = 0 \bmod q$. Let q=5
3. Let e0=3; $e1 = e0^{(p-1)/q} \bmod p = 3^{(31-1)/5} \bmod 31 = 16$
4. Alice chooses an integer, d, as her private key. Let d=7
5. Alice calculates $e2 = e1^{d} \bmod p = 16^{7} \bmod 31 = 8$
6. Alice's public key is (e1, e2, p, q); her private key is (d);

- Alice chooses a random number r Let r=3
- $S1 = h(M | e1^{r} \bmod p) = h(14 | 16^{3} \bmod 31) = h(14 | 4 \bmod 31)$ Let it be = 9
- $S2 = r + d \times S1 \bmod q = 3 + 7 \times 9 \bmod 5 = 6$
- Alice sends 14, 9, and 6

Verifying Message

The receiver, Bob receives 14, 9, and 6

1. Bob calculates $V = h(M \mid e1^{S2}\, e2^{-S1} \bmod p) = h(14 \mid 16^{6}\, 8^{-9} \bmod 31) = h(14 \mid 4 \bmod 31) = 9$

2. If S1 is congruent to V modulo p, the message is accepted; otherwise, it is rejected.

References

- Behrouz A. Forouzan and Debdeep Mukhopadhyay – “Cryptography and Network Security”, McGraw Hill, 2nd Edition, 2008.
- William Stallings,“Cryptography And Network Security Principles And Practice”, Fifth Edition, Pearson Education, 2013