

Figure 13.1 Generic Model of Digital Signature Process

- Bob can sign a message using a digital signature generation algorithm. The inputs to the algorithm are the message and Bob's private key. Any other user, say Alice, can verify the signature using a verification algorithm, whose inputs are the message, the signature, and Bob's public key.

Dispute

1. Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share.
2. John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

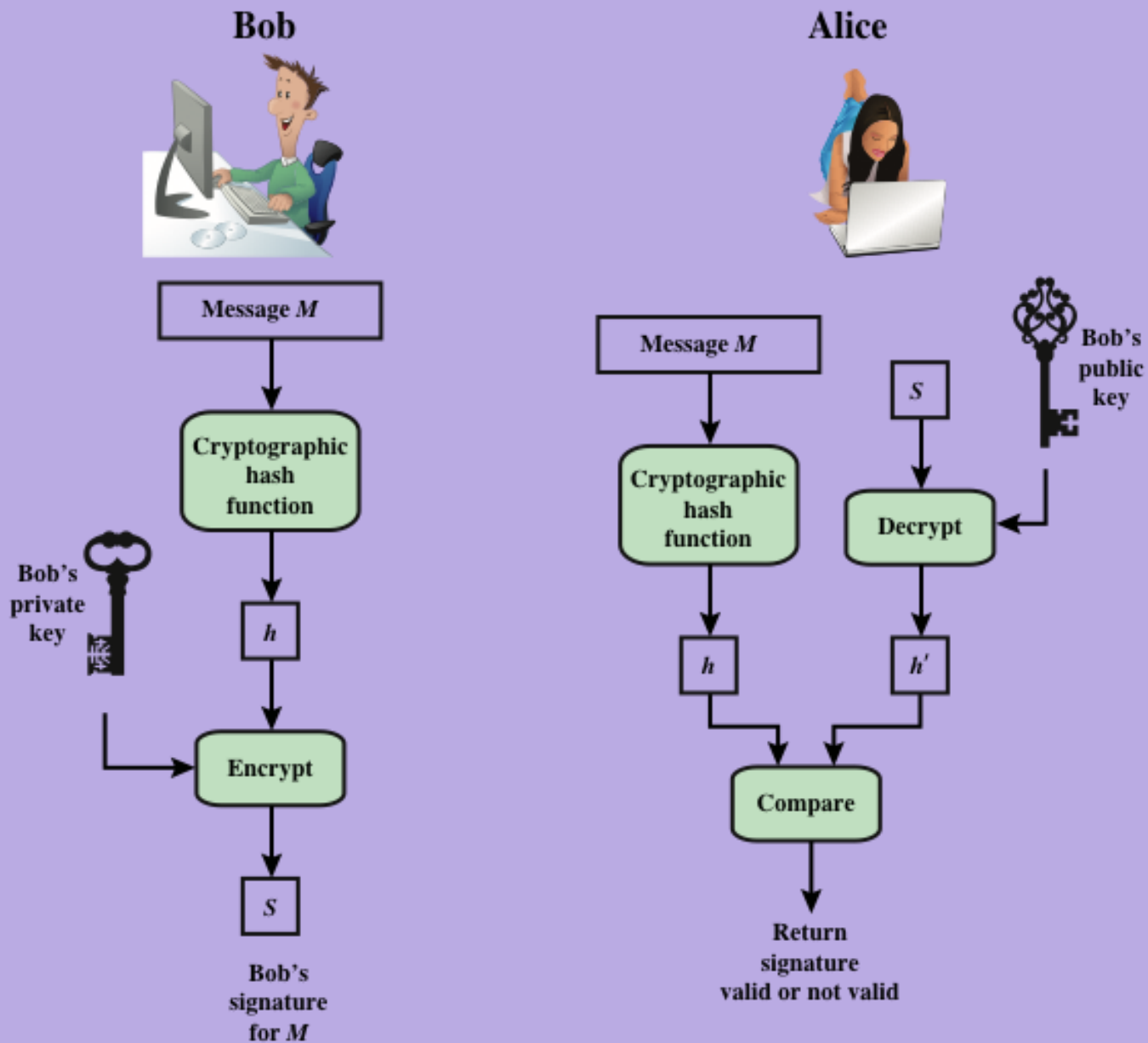
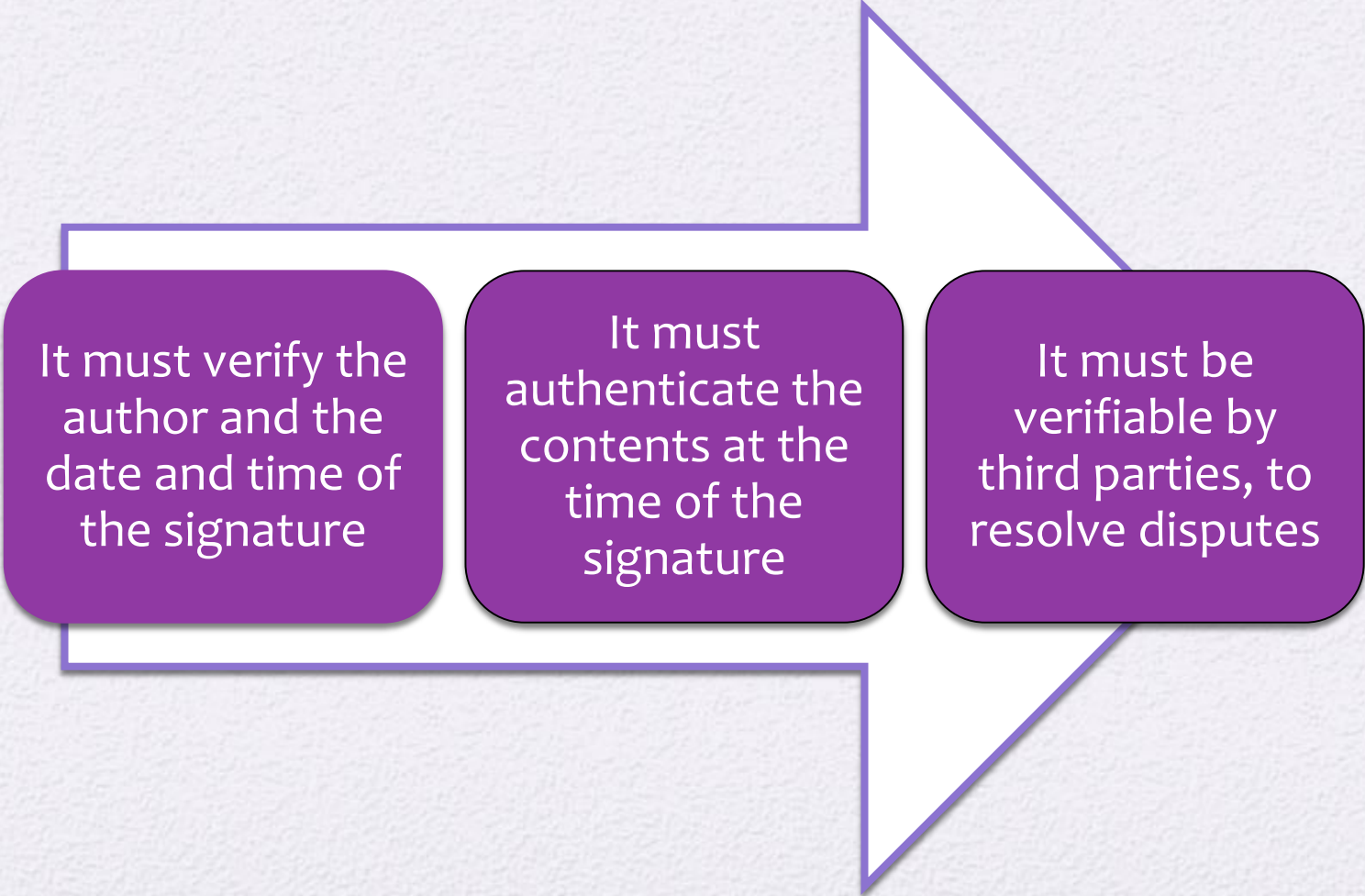


Figure 13.2 Simplified Depiction of Essential Elements of Digital Signature Process

Digital Signature Properties

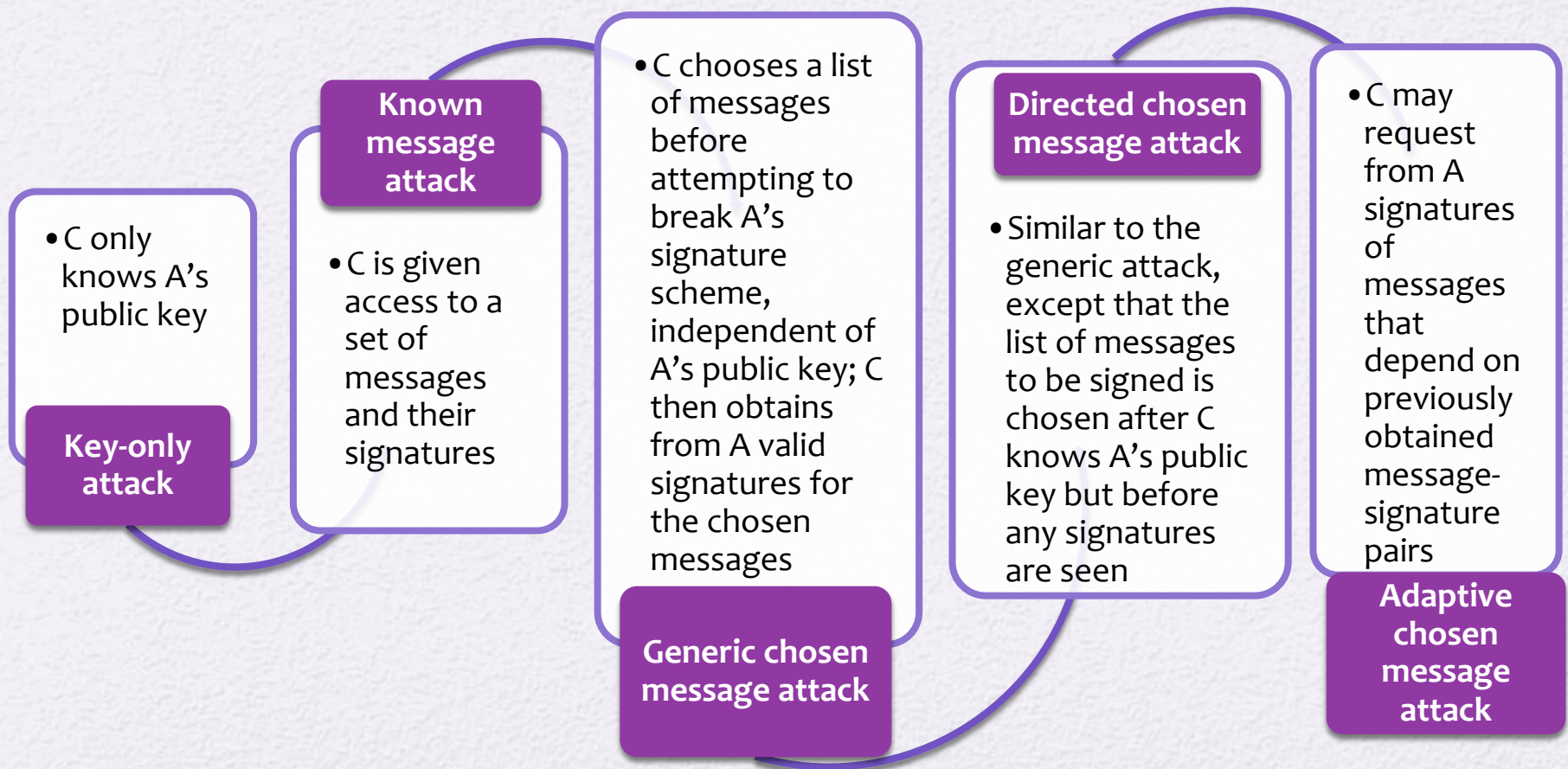


It must verify the author and the date and time of the signature

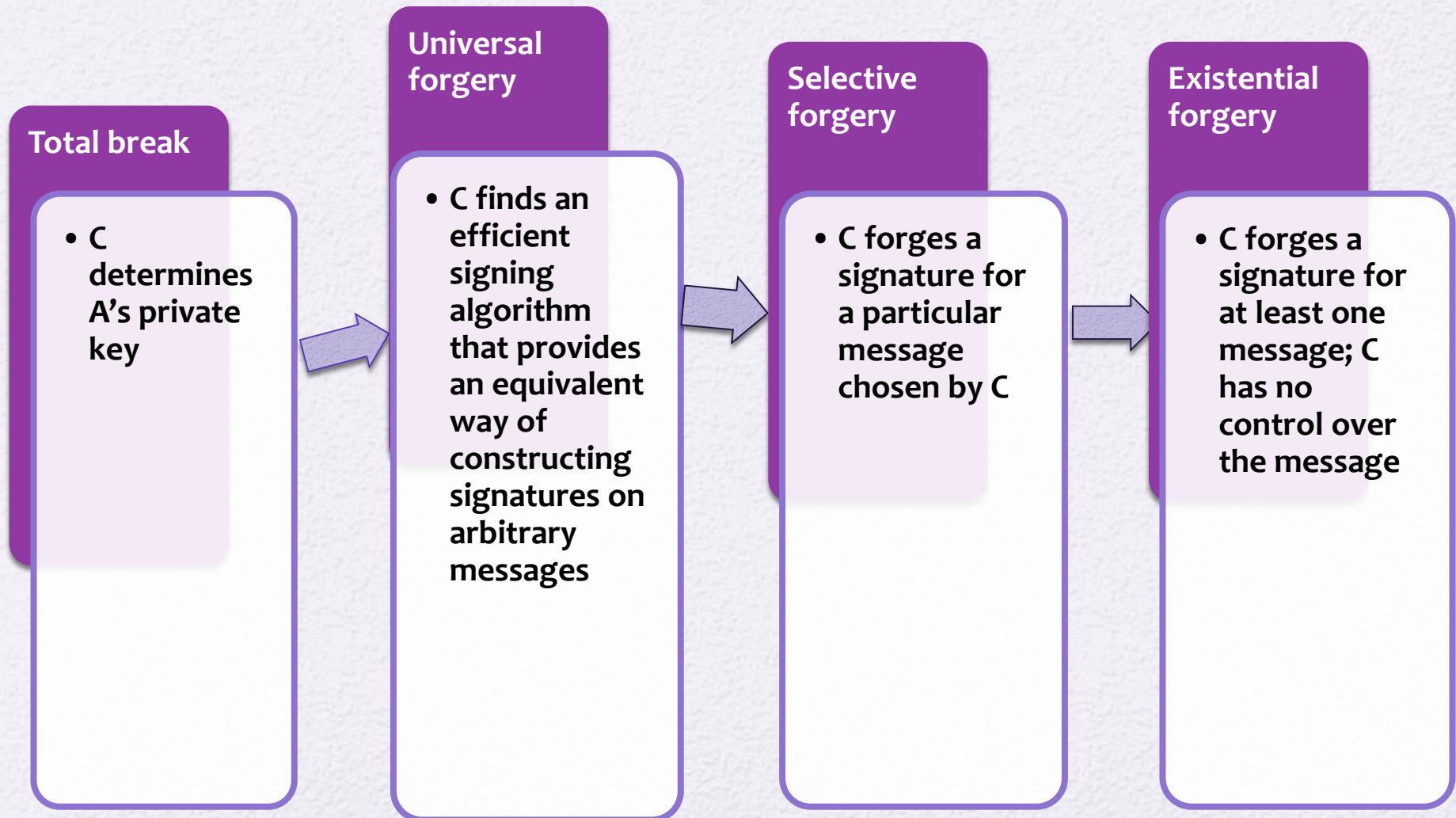
It must authenticate the contents at the time of the signature

It must be verifiable by third parties, to resolve disputes

Attacks



Forgeries



Example

- As a prototypical application, consider a software company that wants to disseminate software updates in an authenticated manner; that is, when the company releases an update it should be possible for any of its clients to verify that the update is authentic, and a malicious third party should never be able to fool a client into accepting an update that was not actually released by the company.

- In the current example, pk could be bundled with the original software purchased by a client.) When releasing a software update m , the company computes a digital signature σ on m using its private key sk , and sends (m, σ) to every client. Each client can verify the authenticity of m by checking that σ is a correct signature on m with respect to the public key pk .

- Both message authentication codes and digital signature schemes are used to ensure the **integrity** of transmitted messages.
- Using digital signatures rather than message authentication codes simplifies key distribution and management, especially when a sender needs to communicate with multiple receivers.

- A qualitative advantage that digital signatures have as compared to message authentication codes is that signatures are **publicly verifiable**.
- Digital signature schemes also provide the very important property of **nonrepudiation**.

- Digital signatures are the public-key counterpart of message authentication codes, and their syntax and security guarantees are analogous. The algorithm that the sender applies to a message is here denoted Sign (rather than Mac), and the output of this algorithm is now called a signature (not tag)

DEFINITION 12.1 A (digital) signature scheme consists of three probabilistic polynomial-time algorithms $(\text{Gen}, \text{Sign}, \text{Vrfy})$ such that:

1. The key-generation algorithm Gen takes as input a security parameter 1^n and outputs a pair of keys (pk, sk) . These are called the **public key** and the **private key**, respectively. We assume that pk and sk each has length at least n , and that n can be determined from pk or sk .
2. The signing algorithm Sign takes as input a private key sk and a message m from some message space (that may depend on pk). It outputs a signature σ , and we write this as $\sigma \leftarrow \text{Sign}_{sk}(m)$.
3. The deterministic verification algorithm Vrfy takes as input a public key pk , a message m , and a signature σ . It outputs a bit b , with $b = 1$ meaning valid and $b = 0$ meaning invalid. We write this as $b := \text{Vrfy}_{pk}(m, \sigma)$.

Adversary Model

1. $\text{Gen}(1^n)$ is run to obtain keys (pk, sk) .
2. Adversary \mathcal{A} is given pk and access to an oracle $\text{Sign}_{sk}(\cdot)$. The adversary then outputs (m, σ) . Let \mathcal{Q} denote the set of all queries that \mathcal{A} asked its oracle.
3. \mathcal{A} succeeds if and only if (1) $\text{Vrfy}_{pk}(m, \sigma) = 1$ and (2) $m \notin \mathcal{Q}$. In this case the output of the experiment is defined to be 1.

$$\Pr[\text{Sig-forge}_{\mathcal{A}, \Pi}(n) = 1] \leq \text{negl}(n).$$

RSA DSS

Let **GenRSA** be as in the text. Define a signature scheme as follows:

- **Gen**: on input 1^n run **GenRSA**(1^n) to obtain (N, e, d) . The public key is $\langle N, e \rangle$ and the private key is $\langle N, d \rangle$.
- **Sign**: on input a private key $sk = \langle N, d \rangle$ and a message $m \in \mathbb{Z}_N^*$, compute the signature

$$\sigma := [m^d \bmod N].$$

- **Vrfy**: on input a public key $pk = \langle N, e \rangle$, a message $m \in \mathbb{Z}_N^*$, and a signature $\sigma \in \mathbb{Z}_N^*$, output 1 if and only if

$$m \stackrel{?}{=} [\sigma^e \bmod N].$$

Digital Signature Requirements

- The signature must be a bit pattern that depends on the message being signed
- The signature must use some information unique to the sender to prevent both forgery and denial
- It must be relatively easy to produce the digital signature
- It must be relatively easy to recognize and verify the digital signature
- It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message
- It must be practical to retain a copy of the digital signature in storage

Direct Digital Signature

- Refers to a digital signature scheme that involves only the communicating parties
 - It is assumed that the destination knows the public key of the source
- Confidentiality can be provided by encrypting the entire message plus signature with a shared secret key
 - It is important to perform the signature function first and then an outer confidentiality function
 - In case of dispute some third party must view the message and its signature
- The validity of the scheme depends on the security of the sender's private key
 - If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature
 - One way to thwart or at least weaken this ploy is to require every signed message to include a timestamp and to require prompt reporting of compromised keys to a central authority

ElGamal Digital Signature

- Scheme involves the use of the private key for encryption and the public key for decryption
- Global elements are a prime number q and a , which is a primitive root of q
- Use private key for encryption (signing)
- Uses public key for decryption (verification)
- Each user generates their key
 - Chooses a secret key (number): $1 < x_A < q-1$
 - Compute their public key: $y_A = a^{x_A} \bmod q$

1. Generate a random integer X_A , such that $1 < X_A < q - 1$.
2. Compute $Y_A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A ; A's public key is $\{q, \alpha, Y_A\}$.

To sign a message M , user A first computes the hash $m = H(M)$, such that m is an integer in the range $0 \leq m \leq q - 1$. A then forms a digital signature as follows.

1. Choose a random integer K such that $1 \leq K \leq q - 1$ and $\gcd(K, q - 1) = 1$. That is, K is relatively prime to $q - 1$.
2. Compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.
3. Compute $K^{-1} \bmod (q - 1)$. That is, compute the inverse of K modulo $q - 1$.
4. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1)$.
5. The signature consists of the pair (S_1, S_2) .

Any user B can verify the signature as follows.

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$.

The signature is valid if $V_1 = V_2$. Let us demonstrate that this is so. Assume that the equality is true. Then we have

$\alpha^m \bmod q = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$	assume $V_1 = V_2$
$\alpha^m \bmod q = \alpha^{X_A S_1} \alpha^{K S_2} \bmod q$	substituting for Y_A and S_1
$\alpha^{m - X_A S_1} \bmod q = \alpha^{K S_2} \bmod q$	rearranging terms
$m - X_A S_1 \equiv K S_2 \bmod (q - 1)$	property of primitive roots
$m - X_A S_1 \equiv K K^{-1} (m - X_A S_1) \bmod (q - 1)$	substituting for S_2

For example, let us start with the prime field $\text{GF}(19)$; that is, $q = 19$. It has primitive roots $\{2, 3, 10, 13, 14, 15\}$, as shown in Table 8.3. We choose $\alpha = 10$.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 16$.
2. Then $Y_A = \alpha^{X_A} \bmod q = 10^{16} \bmod 19 = 4$.
3. Alice's private key is 16; Alice's public key is $\{q, \alpha, Y_A\} = \{19, 10, 4\}$.

Suppose Alice wants to sign a message with hash value $m = 14$.

1. Alice chooses $K = 5$, which is relatively prime to $q - 1 = 18$.
2. $S_1 = \alpha^K \bmod q = 10^5 \bmod 19 = 3$ (see Table 8.3).
3. $K^{-1} \bmod (q - 1) = 5^{-1} \bmod 18 = 11$.

4. $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1) = 11(14 - (16)(3)) \bmod 18 = -374 \bmod 18 = 4.$

Bob can verify the signature as follows.

1. $V_1 = \alpha^m \bmod q = 10^{14} \bmod 19 = 16.$
2. $V_2 = (Y_A)^{S_1} (S_1)^{S_2} \bmod q = (4^3)(3^4) \bmod 19 = 5184 \bmod 19 = 16.$

Thus, the signature is valid.