COMPUTER SECURITY
(CSE 4105)

Digital Signatures

Digital Signature

- A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature.
- Typically 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.
- The most important development from the work on public-key cryptography is the digital signature.
- The digital signature provides a set of security capabilities that would be difficult to implement in any other way.

Digital Signature

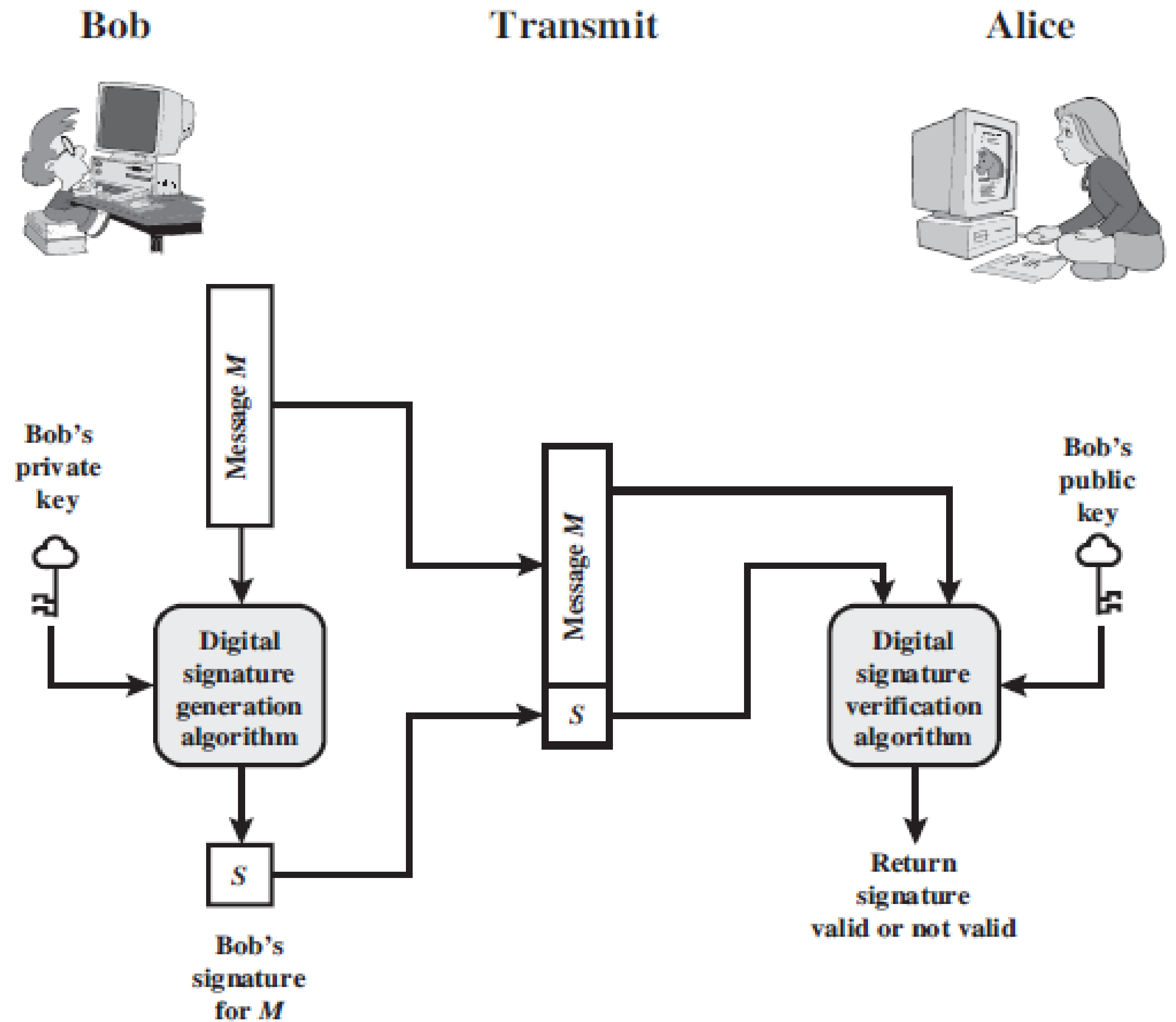

Figure 13.1 Generic Model of Digital Signature Process

Digital Signature

- 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.
- In simplified terms, the essence of the digital signature mechanism is shown in Figure 13.2.

Digital Signature

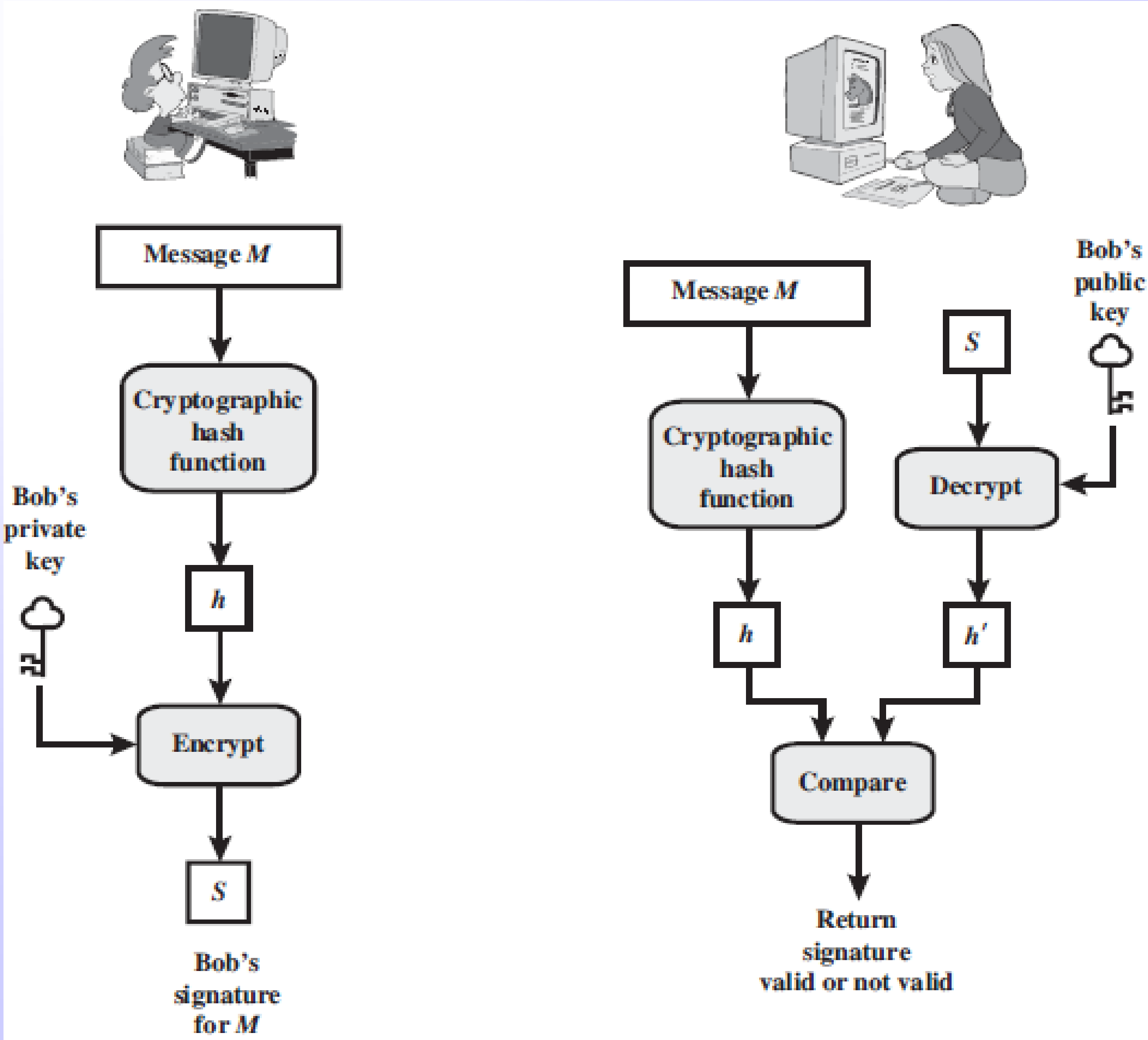

Figure 13.2 Simplified Depiction of Essential Elements of Digital Signature Process

Digital Signature

- Consider the following disputes that could arise:
  - 1. Bob may forge a different message and claim that it came from Alice. Bob would simply have to create a message and append an authentication code using the key that Alice and Bob share.
  - 2. Alice can deny sending the message. Because it is possible for Bob to forge a message, there is no way to prove that Alice did in fact send the message.

Properties of Digital Signature

- In situations where there is not complete trust between sender and receiver, something more than authentication is needed.
- The most attractive solution to this problem is the digital signature.
- The digital signature must have the following 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 and Forgeries

☞ [GOLD88] lists the following types of attacks, in order of increasing severity. Here *A* denotes the user whose signature method is being attacked, and *C* denotes the attacker.

- **Key-only attack:** C only knows A's public key.
- **Known message attack:** C is given access to a set of messages and their signatures.
- **Generic chosen message attack:** C chooses a list of messages before attempting to breaks A's signature scheme, independent of A's public key. C then obtains from A valid signatures for the chosen messages. The attack is generic, because it does not depend on A's public key; the same attack is used against everyone.
- **Directed chosen message attack:** Similar to the generic attack, except that the list of messages to be signed is chosen after C knows A's public key but before any signatures are seen.
- **Adaptive chosen message attack:** C is allowed to use A as an "oracle." This means the A may request signatures of messages that depend on previously obtained message–signature pairs.

Attacks and Forgeries

- [GOLD88] then defines success at breaking a signature scheme as an outcome in which *C* can do any of the following with a non-negligible probability:

- **Total break:** C determines A's private key.
- **Universal forgery:** C finds an efficient signing algorithm that provides an equivalent way of constructing signatures on arbitrary messages.
- **Selective forgery:** C forges a signature for a particular message chosen by C.
- **Existential forgery:** C forges a signature for at least one message. C has no control over the message. Consequently, this forgery may only be a minor nuisance to A.

Digital Signature Requirements

- We can formulate the following requirements for a digital signature.
  - 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.

ElGamal Digital Signature Scheme

- Before proceeding, we need a result from number theory. For a prime number q, if α is a primitive root of q, then:

  $$\alpha, \alpha^2, \ldots\ldots, \alpha^{q-1}$$

  are distinct of (mod q).
- The global elements of ElGamal digital signature are a prime number q and α, which is a primitive root of q. User A generates a private/public key pair as follows.

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 pubic key is $\{q, \alpha, Y_A\}$.

ElGamal Digital Signature Scheme

- 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).

ElGamal Digital Signature Scheme

- 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 $V_1 = V_2$
$\alpha^m \bmod q = \alpha^{X_A S_1}\alpha^{KS_2} \bmod q$	substituting for Y_A and S_1
$\alpha^{m-X_A S_1} \bmod q = \alpha^{KS_2} \bmod q$	rearranging terms
$m - X_A S_1 \equiv KS_2 \bmod (q-1)$	property of primitive roots
$m - X_A S_1 \equiv KK^{-1}(m - X_A S_1) \bmod (q-1)$	substituting for S_2

Anupam Kumar Bairagi, PhD

Thanks for your Attention