Digital Signature

APPLICATIONS SUCH AS banking, stock trading, and the sale and purchase of merchandise are increasingly emphasizing electronic transactions to minimize operational costs and provide enhanced services. This has led to phenomenal increases in the amounts of electronic documents that are generated, processed, and stored in computers and transmitted over networks. This electronic information handled in these applications is valuable and sensitive and must be protected against tampering by malicious third parties (who are neither the senders nor the recipients of the information). Sometimes, there is a need to prevent the information or items related to it (such as date/time it was created, sent, and received) from being tampered with by the sender (originator) and/or the recipient.

Traditionally, paper documents are validated and certified by written signatures, which work fairly well as a means of providing authenticity. For electronic documents, a similar mechanism is necessary. Digital signatures, which are nothing but a string of ones and zeroes generated by using a digital signature algorithm, serve the purpose of validation and authentication of electronic documents. Validation refers to the process of certifying the contents of the document, while authentication refers to the process of certifying the sender of the document.

Conventional and digital signature characteristics

A conventional signature has the following salient characteristics: relative ease of establishing that the signature is authentic, the difficulty of forging a signature, the non-transferability of the signature, the difficulty of altering the signature, and the nonrepudiation of signature to ensure that the signer cannot later deny signing.

A digital signature should have all the aforementioned features of a conventional signature plus a few more as digital signatures are being used in practical, but sensitive, applications such as secure e-mail and credit card transactions over the Internet. Since a digital signature is just a sequence of zeroes and ones, it is desirable for it to have the following properties: the signature must be a bit pattern that depends on the message being signed (thus, for the same originator, the digital signature is different for different documents); the signature must use some information that is unique to the sender to prevent both forgery and denial; it must be relatively easy to produce; it must be relatively easy to recognize and verify the authenticity of digital signature; it must be computationally infeasible to forge a digital signature either by constructing a new message for an existing digital signature or constructing a fraudulent digital signature for a given message; and it must be practical to save copies of the digital signatures in storage for arbitrating possible disputes later.

To verify that the received document is indeed from the claimed sender and that the contents have not been altered, several procedures, called authentication techniques, have been developed. However, message authentication techniques cannot be directly used as digital signatures due to inadequacies of authentication techniques. For example, although message authentication protects the two parties exchanging messages from a third party, it does not protect the two parties against each other. In addition, elementary authentication schemes produce signatures that are as long as the message themselves.

Basic notations and terminology

Digital signatures are computed based on the documents (message/ information) that need to be signed and on some private information held only by the sender. In practice, instead of using the whole message, a hash function is applied to the message to obtain the message digest. A hash function, in this context, takes an arbitrary-sized message as input and produces a fixed-size message digest as output. Among the commonly used hash functions in practice are MD-5 (message digest 5) and SHA (secure hash algorithm). These algorithms are fairly sophisticated and ensure that it is highly improbable for two different messages to be mapped to the same hash value.

There are two broad techniques used in digital signature computation—symmetric key cryptosystem and public-key cryptosystem (cryptosystem broadly refers to an encryption technique). In the symmetric key system, a secret key known only to the sender and the legitimate receiver is used. However, there must be a

unique key between any two pairs of users. Thus, as the number of user pairs increases, it becomes extremely difficult to generate, distribute, and keep track of the secret keys. A public key cryptosystem, on the other hand, uses a pair of keys: a private key, known only to its owner, and a public key, known to everyone who wishes to communicate with the owner. For confidentiality of the message to be sent to the owner, it would be encrypted with the owner's public key, which now could only be decrypted by the owner, the person with the corresponding private key. For purposes of authentication, a message would be encrypted with the private key of the originator or sender, who we will refer to as A. This message could be decrypted by anyone using the public key of A. If this yields the proper message, then it is evident that the message was indeed encrypted by the private key of A, and thus only A could have sent it.

Creating and verifying a digital signature

A simple generic scheme for creating and verifying a digital signature is shown in Figs. 1 and 2, respectively. A hash function is applied to the message that yields a fixed-size message digest. The signature function uses the message digest and the sender's private key to generate the digital signature. A very simple form of the digital signature is obtained by encrypting the message digest using the sender's private key. The message and the signature can now be sent to the recipient. The message is unencrypted and can be read by anyone. However, the signature ensures authenticity of the sender (something similar to a circular sent by a proper authority to be read by many people, with the signature attesting to the authenticity of the message). At the receiver, the inverse signature function is applied to the digital signature to recover the original message digest. The received message is subjected to the same hash function to which the original message was subjected. The resulting message digest is compared with the one recovered from the signature. If they match, then it ensures that the message has indeed been sent by the (claimed) sender and that it has not been altered

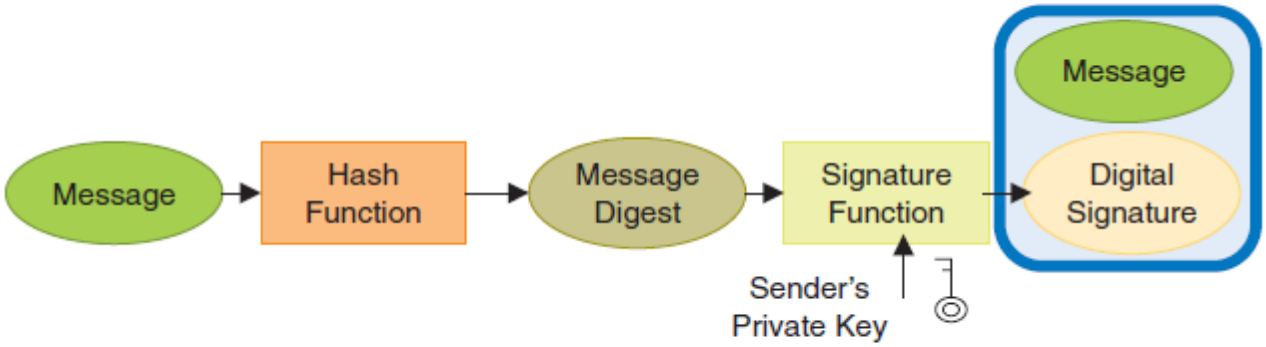

Fig. 1 Creating a digital signature

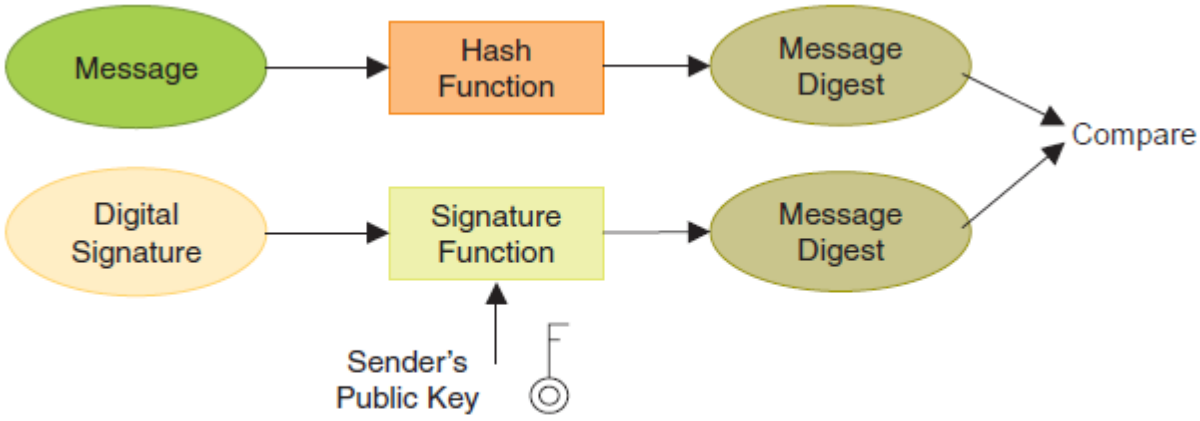

Fig. 2 Verifying a digital signature

Creating and opening a digital envelope

A digital envelope is the equivalent of a sealed envelope containing an unsigned letter. The outline of creating a digital envelope is shown in Fig. 3. The message is encrypted by the sender using a randomly generated symmetric key. The symmetric key itself is encrypted using the intended recipient's public key. The combination of the encrypted message and the encrypted symmetric key is the digital envelope. The process of opening the digital envelope and recovering the contents is shown in Fig. 4. First, the encrypted symmetric key is recovered by a decryption using the recipient's private key. Subsequently, the encrypted message is decrypted using the symmetric key.

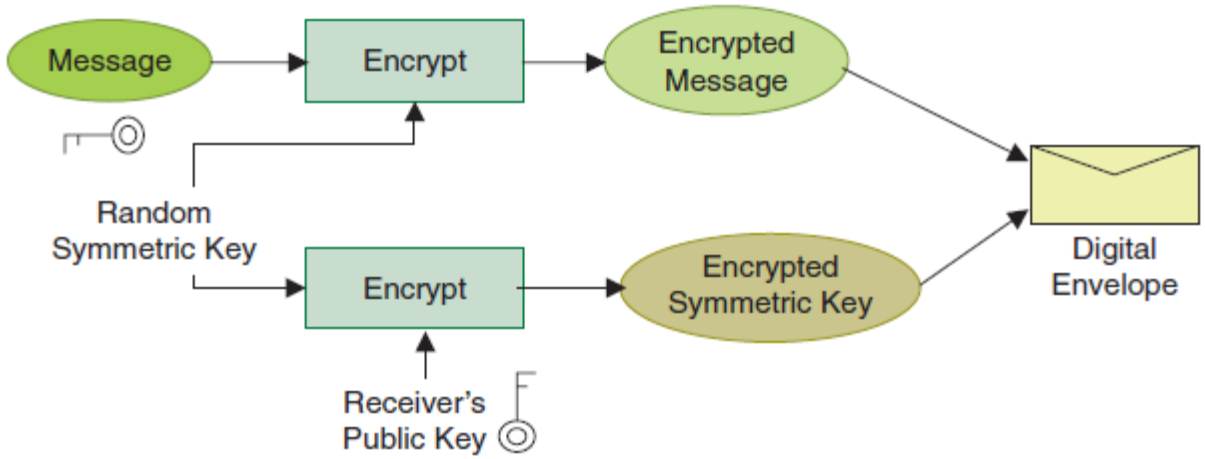

Fig. 3 Creating a digital envelope

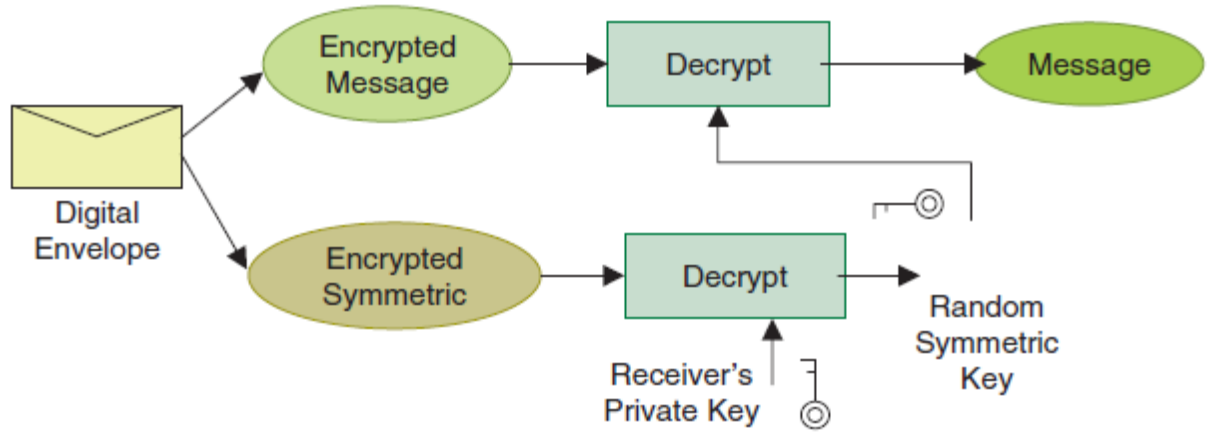

Fig. 4 Opening a digital envelope

Creating and opening digital envelopes carrying signed messages

The process of creating a digital envelope containing a signed message is shown in Fig. 5. A digital signature is created by the signature function using the message digest of the message and the sender's private key. The original message and the digital signature are then encrypted by the sender using a randomly generated key and a symmetric- key algorithm. The symmetric key itself is encrypted using the recipient's public key. The combination of encrypted message and signature, together with the encrypted symmetric key, form the digital envelope containing the signed message. Figure 6 shows the process of opening a digital envelope, recovering the message, and verifying the signature. First, the symmetric key is recovered using the recipient's private key. This is then used to decrypt and recover the message and the digital signature. The digital signature is then verified as described earlier.

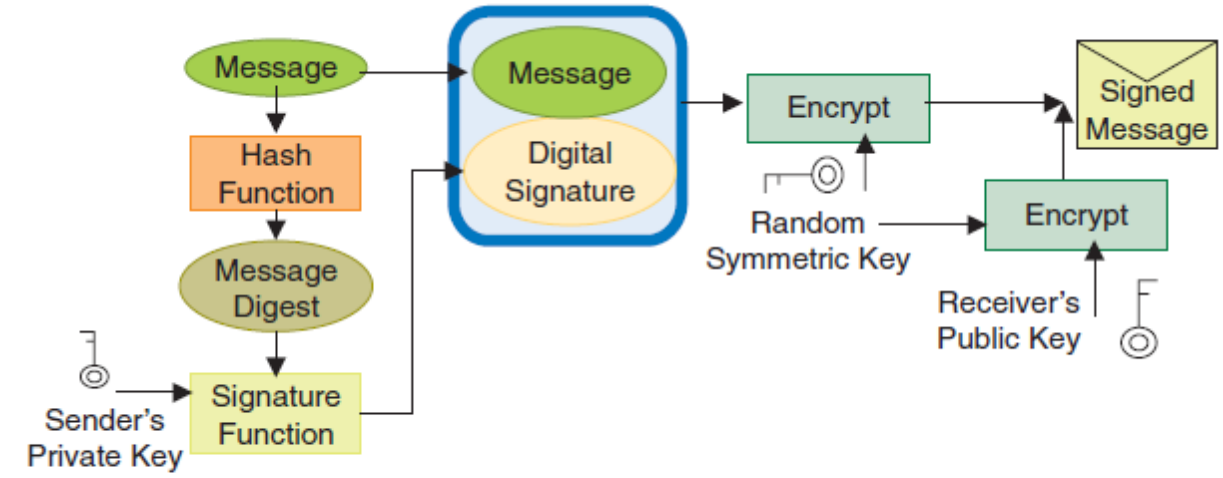

Fig. 5 Creating a digital envelope carrying a signed message

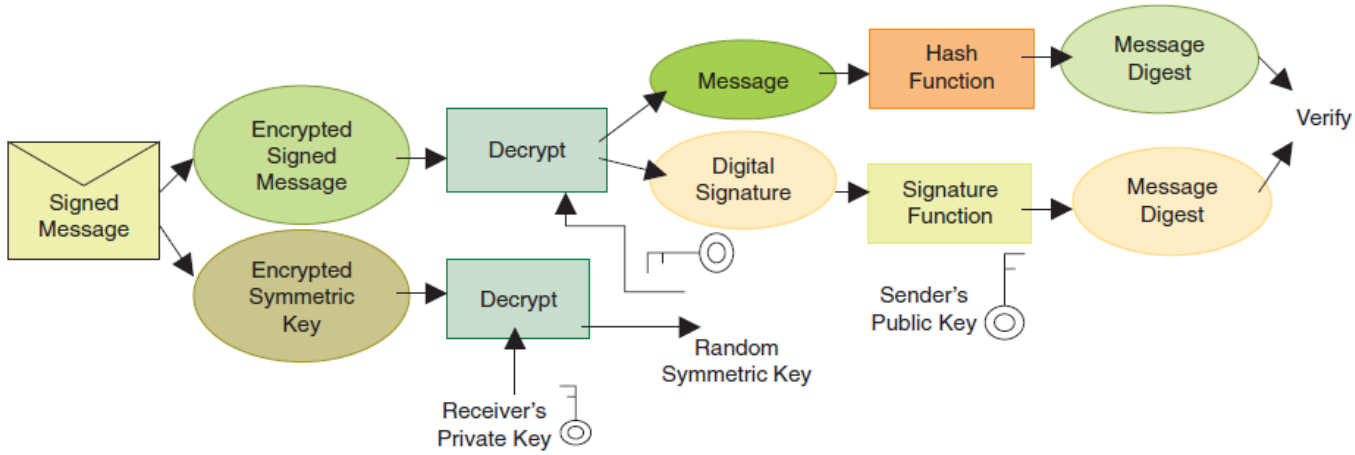

Fig. 6 Opening a digital envelope and verifying a digital signature

Digital signatures in real applications

Increasingly, digital signatures are being used in secure e-mail and credit card transactions over the Internet. The two most common secure e-mail systems using digital signatures are Pretty Good Privacy and Secure/Multipurpose Internet Mail Extension. Both of these systems support the RSA as well as the DSS-based signatures. The most widely used system for the credit card transactions over the Internet is Secure Electronic Transaction (SET). It consists of a set of security protocols and formats to enable prior existing credit card payment infrastructure to work on the Internet. The digital signature scheme used in SET is similar to the RSA scheme