Digital signatures

S.R. SUBRAMANYA AND BYUNG K. YI

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. In this article, the terms document and message are used interchangeably.

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 nontransferability 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 retcopies 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 notions 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

0278-6648/06/$20.00 © 2006 IEEE

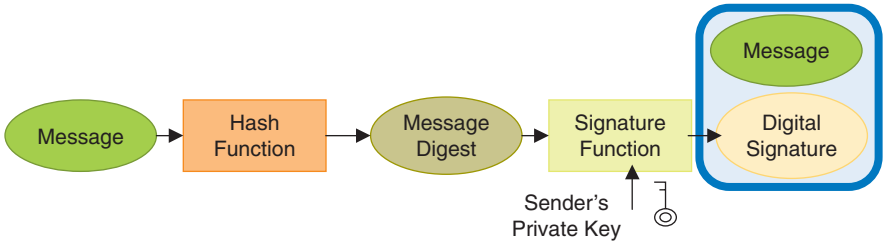

Fig. 1 Creating a digital signature

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.

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.

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.

Direct and arbitrated digital signature

A variety of modes have been proposed for digital signatures that fall into two basic categories: direct and arbitrated. The direct digital signature involves only the communicating parties, sender and receiver. This is the simplest type of digital signature. It is assumed that the recipient knows the public key of the sender. In a simple scheme, a digital signature may be formed by encrypting the entire message or the hash code of the message with the sender's private key. Confidentiality can be provided by further encrypting the entire message plus signature with either the receiver's public key encryption or the shared secret key, which is conventional encryption. A sender may later deny sending a particular message by claiming that the private key was lost or stolen and that someone else forged his signature. One way to overcome this is to include a time stamp with every message and requiring notification of loss of key to the proper authority. In case of dispute, a trusted third party may view the message and its signature to arbitrate the dispute.

In the arbitrated signature scheme, there is a trusted third party called the arbiter. Every signed message from a sender A to a receiver B goes first to an arbiter T, who subjects the message and its signature to a number of tests to check its origin and content. The mes-

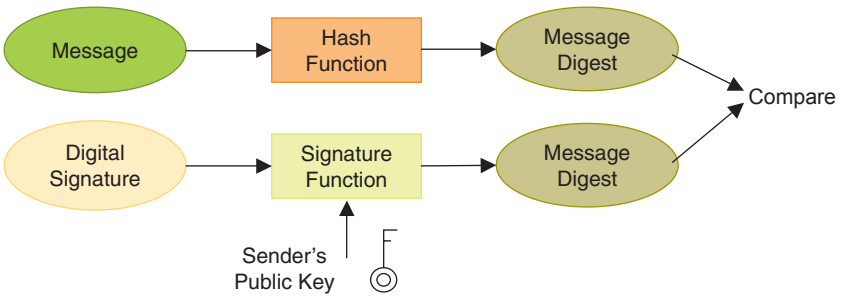

Fig. 2 Verifying a digital signature

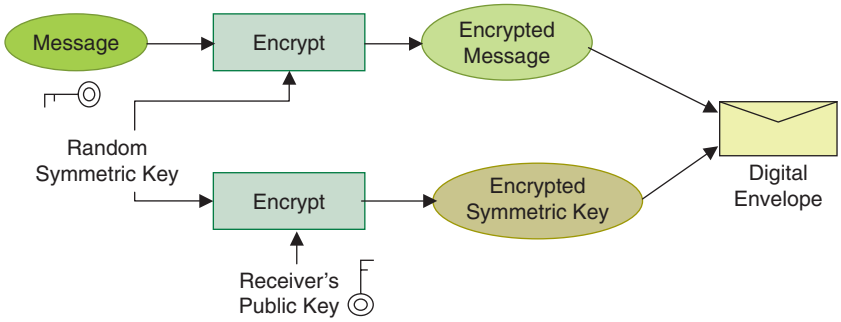

Fig. 3 Creating a digital envelope

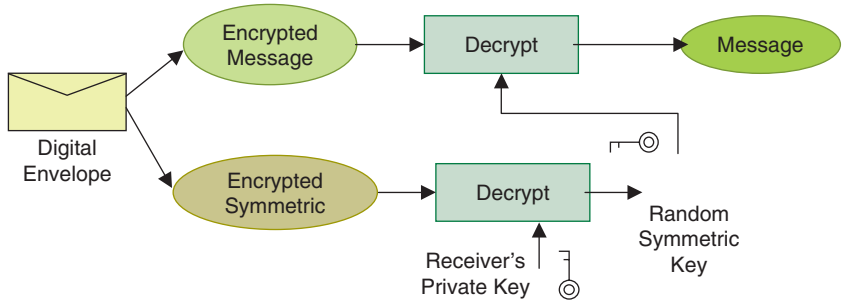

Fig. 4 Opening a digital envelope

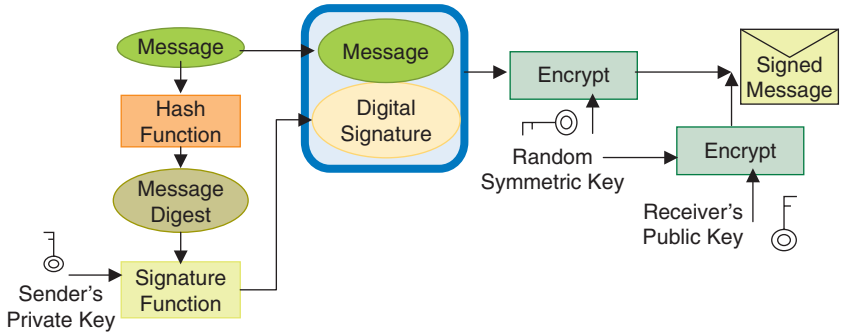

Fig. 5 Creating a digital envelope carrying a signed message

sage is then dated and sent to B with an indication that it has been verified to the satisfaction of the arbiter. The presence of T solves the problem faced by direct signature schemes, namely that A might deny sending a message. The arbiter plays a sensitive and crucial role in this scheme, and all parties must trust that the arbitration mechanism is working properly. There are many variations of arbitrated digital-signature schemes. Some schemes allow the arbiter to see the messages, while others don't. The particular scheme employed depends on the needs of the applications. Generally, an arbitrated digital-signature scheme has advantages over a direct digital-signature scheme such as the trust in communications between the parties provided by the trusted arbiter and in the arbitration of later disputes, if any.

A public versus a private approach to digital signatures

Another way of classifying digital signature schemes is based on whether a private-key system or a public-key system is used. The public-key system based digital signatures have several advantages over the private-key system based digital signatures. The two most popular and commonly used public-key system based digital signature schemes are the RSA (named after Rivest, Shamir, and Aldeman, the inventors of the RSA public-key encryption scheme) and the digital signature algorithm (DSA) approaches. The DSA is incorporated into the Digital Signature Standard (DSS), which was published by the National Institute of Standards and Technology as the Federal Information Processing Standard. It was first proposed in 1991, revised in 1993, and further revised with minor changes in 1996.

RSA is a commonly used scheme for digital signatures. In a broad outline of the RSA approach, the message to be signed is input to a hash function that produces a secure hash code of fixed length. This hash code is then encrypted using the sender's private key to form the signature. Both the signature and the message are then concatenated and transmitted. The recipient takes the message and produces a hash code. The recipient also decrypts the signature using the sender's public key. If the calculated hash code matches the decrypted signature, the signature is accepted as valid. This is because only the sender knows the private key, and thus only the sender could have produced a valid signature. The signature generation and verification using RSA is identical to the schemes shown in Figs. 1 and 2, respectively.

The signing process in DSS (using DSA) is shown in Fig. 7. The DSA approach also makes use of a hash function. The hash code is provided as input to a signature function together with a random number generated for this particular signature. The signature function also uses the sender's private key and a set of parameters known to a group of communicating parties, referred to as global public key. The output signature consists of two components. The signature verification process is shown in Fig. 8. At the receiving end, the hash code of the incoming message is generated and input to a verification function, together with the two components of the signature. The verification function uses the global public key as well as sender's public key and recreates (one of the two components of) the original digital signature. A match between the recreated and the original signature indicates the authenticity of the signature. The signature function is such that it assures the recipient that only the sender, with the knowledge of the private key, could have produced the valid signature.

The basis of the RSA scheme is the difficulty of factoring of large prime numbers. That of the DSA scheme is the difficulty of computing discrete logarithms. The DSA provides only the signature function where as the RSA scheme could additionally provide encryption and key exchange. The signature verification using the RSA scheme is about 100 times faster than a DSA scheme. The signature generation is slightly faster in the DSA scheme.

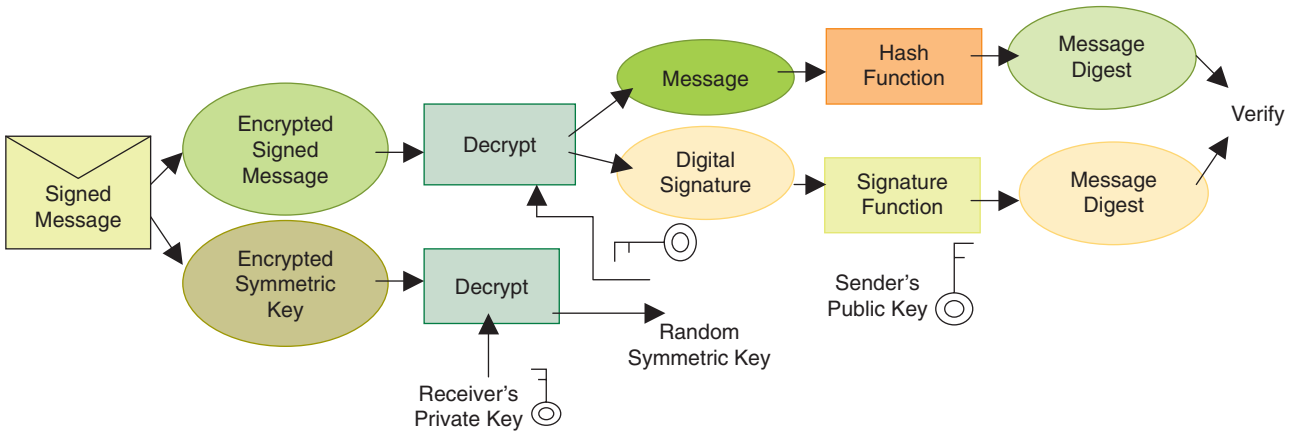

Fig. 6 Opening a digital envelope and verifying a digital signature

Work is underway for several extensions of the basic digital signature scheme such as enabling signatures by multiple parties (group digital signatures), signatures by a hierarchy of signatories, and protocols for simultaneous signing of contracts electronically by two or more signatories, separated by wide distances.

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.

Conclusions

Many traditional and newer businesses and applications have recently been carrying out enormous amounts of electronic transactions, which have led to a critical need for protecting the information from being maliciously altered, for ensuring the authenticity, and for supporting nonrepudiation. Just as signatures facilitate validation and verification of the authenticity of paper documents, digital signatures serve the purpose of validation and authentication of electronic documents. This technology is rather new and emerging and is expected to experience growth and widespread use in the coming years.

Read more about it

- W. Stallings, *Cryptography and Network Security*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- M. Bishop, *Introduction to Computer Security*. Reading, MA: Addison-Wesley, 2005.
- J. Feghhi and P. Williams, *Digital Certificates: Applied Internet Security* 1st ed. Reading, MA: Addison-Wesley, 1999.
- C.P. Pfleeger and S.L. Pfleeger, *Security in Computing*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

About the authors

S.R. Subramanya obtained his Ph.D. in computer science from George washington University where he received the Richard Merwin memorial award from the EECS department in 1996. He received the Grant-in-Aid of Research award from Sigma-Xi in 1997 for his research in audio data indexing. He is a senior research scientist at LGE Modile Research in San Diego. His current research interests include mobile multimedia services and content management. He is the author of over 70 research papers and articles. He is a Senior Member of the IEEE.

Byung K. Yi obtained his Ph.D. in electrical engineering from George Washington University. He is the senior executive vice president of LG Electronics in San Diego. Dr. Yi's previous affiliations include Orbital Sciences Corp., Fairchild, and several high technology companies. He is a Senior Member of the IEEE.

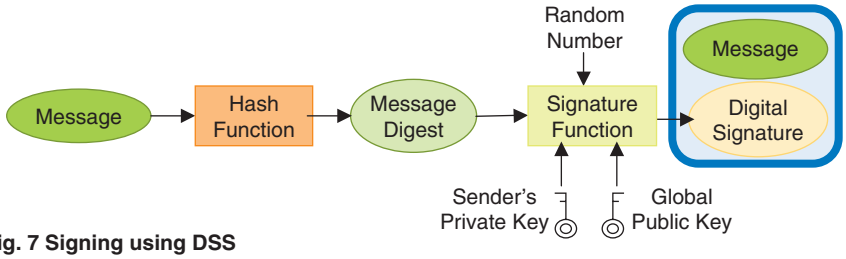

Fig. 7 Signing using DSS

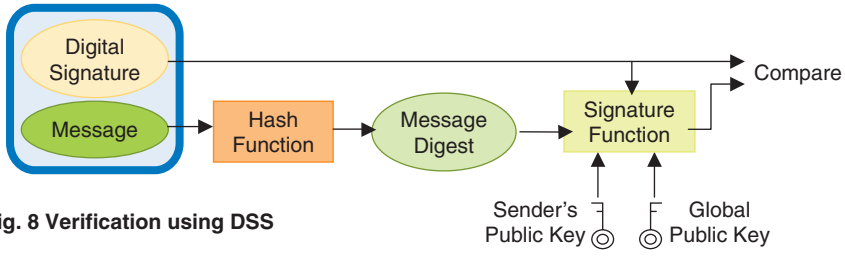

Fig. 8 Verification using DSS