Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9677942, 5 pages
https://doi.org/10.1155/2022/9677942

Research Article

Application Research in Computer Vision Signature Encryption System of Enterprise Contract Economic Management

Feifei Sun[1,2] **and Guohong Shi**[2]

[1]*School of Economy and Management, Changzhou Institute of Industry Technology, Changzhou 213000, China*
[2]*School of Management, Jiangsu University, Zhenjiang 212013, China*

Correspondence should be addressed to Feifei Sun; lec1811507@163.com

Received 20 March 2022; Revised 19 April 2022; Accepted 22 April 2022; Published 13 May 2022

Academic Editor: Kalidoss Rajakani

This paper mainly proposes a nonrepudiation digital signature authentication scheme based on an enterprise contract computer vision system. Based on the RSA algorithm and MD5 message-digest algorithm, a digital signature scheme of enterprise economic contract based on IC card is proposed. The method realizes digital signature and ensures the confidentiality of the transmitted data. Finally, this paper proposes a concrete digital signature experiment system implemented in Java. The results of this study show that the computer vision signature encryption system helps to improve the reliability of the system.

1. Introduction

Ensuring the security of personal and business information transmitted over the Internet has become a pressing issue. Ensuring its security is the primary problem that enterprise economic applications must solve. The structure of this paper is the research on how the enterprise constructs the computer vision signature encryption system to promote the enterprise contract economic management positively. At the beginning of this paper, this research still has certain limitations because the system is only applicable to enterprises and may not have universal adaptability to other industries. How to better apply to all industries is for future research. Contract management is involved in the day-to-day operation of an enterprise. Various intentional or unintentional errors in the financial process of an enterprise can lead to inconsistent information obtained by transaction parties [1]. This will affect the entire transaction process. Ensuring the integrity of data is the basis for developing enterprise economic applications. Therefore, we must prevent arbitrary modification of the data. The traditional way of the transaction is to limit the behavior of transaction parties through signatures and seals, avoiding delays and denials. However, the corporate economy now based on electronic media must find electronic alternatives. Only in this way can reliable identification of the parties to the transaction be provided. Digital signature technology can guarantee the integrity, reliability, nonrepudiation, and identifiability of information transmitted in the network.

2. Implementation Principle of Digital Signature Technology

Digital signature technology is based on public key encryption technology. Therefore, we first must introduce general key encryption technology and its specific application in the digital signature.

2.1. Public Key Encryption Technology. Public key encryption (PKE) is asymmetric cryptography. It solves the fundamental problem that the traditional encryption method is challenging to overcome in the application. It greatly simplifies the key distribution and management process [2]. Its basic idea is to transform the secret storage and distribution in traditional encrypted communication into the underground hold of a key and the public distribution of a “lock” in a public key encryption system. When using a public key encryption system, recipient A first generates a pair of mathematically related but not identical. A public key k_a is used for encryption (equivalent to a lock). A secret key $k_a{}'$ is used for decryption. This process is called vital formulation. $k_a{}'$ is kept by the recipient A himself. We expose k_a in various ways. This process makes it available to anyone who wants to communicate with the recipient. This process is called public key distribution. Of course, the disclosure of k_a cannot compromise the security of $k_a{}'$. If the B party needs to send the plaintext m to the A party confidentially, we can encrypt the plaintext to be sent with the public key k_a of A ; we found

$$C = E_{k_a}(m). \tag{1}$$

After receiving the ciphertext C, A decrypts C with the secret key $k_a{}'$ that only A knows:

$$m = D_a^{k'}(C). \tag{2}$$

Since the encryption key k_a is different from the decryption key $k_a{}'$, public key cryptography is also called asymmetric cryptography. This can be different from traditional symmetric cryptography [3]. Even if any third party intercepts the ciphertext C, it cannot be achieved. Since it has no decryption key $k_a{}'$ and $k_a{}'$ cannot be derived from the public encryption key k_a, we cannot recover the plaintext m. Compared with the traditional encryption system, the public key encryption system has obvious advantages.

2.2. Digital Signature Technology Using the Public Key Encryption System. What is used more on the Internet is that the information itself does not need to be kept secret, because if the encryption key E_A of A in the public key file is tampered with the legend E_C of C, the consequence is that all the ciphertext C sent to A can be decrypted [4]. The purpose of digital signature technology is to ensure the integrity and authenticity of the information.

The digital signature system is the product of the combination of public key encryption technology and message decomposition function (MDF). The packet decomposition function is a one-way irreversible function that can extract a set of information into a string of check digits. First, we use the packet decomposition function to distil the file to be signed into a very long number. We call this the packet decomposition function value. The signer encrypts the value of the message decomposition function with the private key held by the individual [5]. At this point, the system generates a so-called "digital signature."

The recipient authenticates the digital signature after receiving the digitally signed document. To obtain the message decomposition function value, the recipient decrypts the "digital signature" with the signer's public key [6]. Then, they recalculate the packet decomposition function value for the file. We compare the results of the two. If the two are precisely the same, the file's contents' completeness and correctness and the signature's authenticity can be guaranteed. If the content of the file is tampered with or someone forges the signature, the authentication of the digital signature will fail. The authentication process of digital signature and principle of digital signature are shown in Figure 1.

If the sender denies it later, the recipient can submit the digital signature and the original document to a third party for verification. A third party can repeat the above digital signature generation process using the sender's public key. They compare the results with existing digital signatures [7]. This makes it easy to tell if the digital signature originated from the sender. Conversely, if the recipient attempts to forge a digitally signed document, he cannot present the digital signature corresponding to the generated document in front of a third party.

2.3. Message Decomposition Function (MDF). The packet decomposition function mentioned above is a hash function. Any length of information m can be compressed into a fixed-length number after the operation of the hash function.

There are many packet decomposition functions in specific applications. These functions can extract information of any length into a number. But from the perspective of cryptography, its security and reliability are pretty different.

Less secure is the CRC (cyclic redundancy check) code function. It is usually used as an error checking function. The CRC function can represent information as a 16-bit or 32-bit CRC code. It is widely used in computer communication to check for errors in data transmission. Although it can detect the vast majority of random errors in data transmission communications, it has minimal ability to deal with deliberate tampering of information by humans. Because it is easy to tamper with the file's content and add some characters to make its checksum still unchanged. The higher security is the MD5 function. The packet decomposition function is used by the famous PGP encryption system for digital signature [8]. It can convert any information length into an almost unique 128-bit large number. At the same time, MD5 is a one-way mapping function. Although some information can generate the corresponding MD5 code, the same information as the original cannot be recovered from the MD5 code.

3. Digital Signature Technology Application Agreement

An agreement is when two or more parties go through a series of prespecified steps in a particular order to accomplish a task. The practical application of digital signature technology also involves various needs [9]. It is suitable for application protocols in different environments.

3.1. Cryptographic Protocol. We can use the following protocols for digital signatures using RSA public key cryptography:

(1) A first signs the message m with its secret decryption key to obtain $s = D_A(m)$

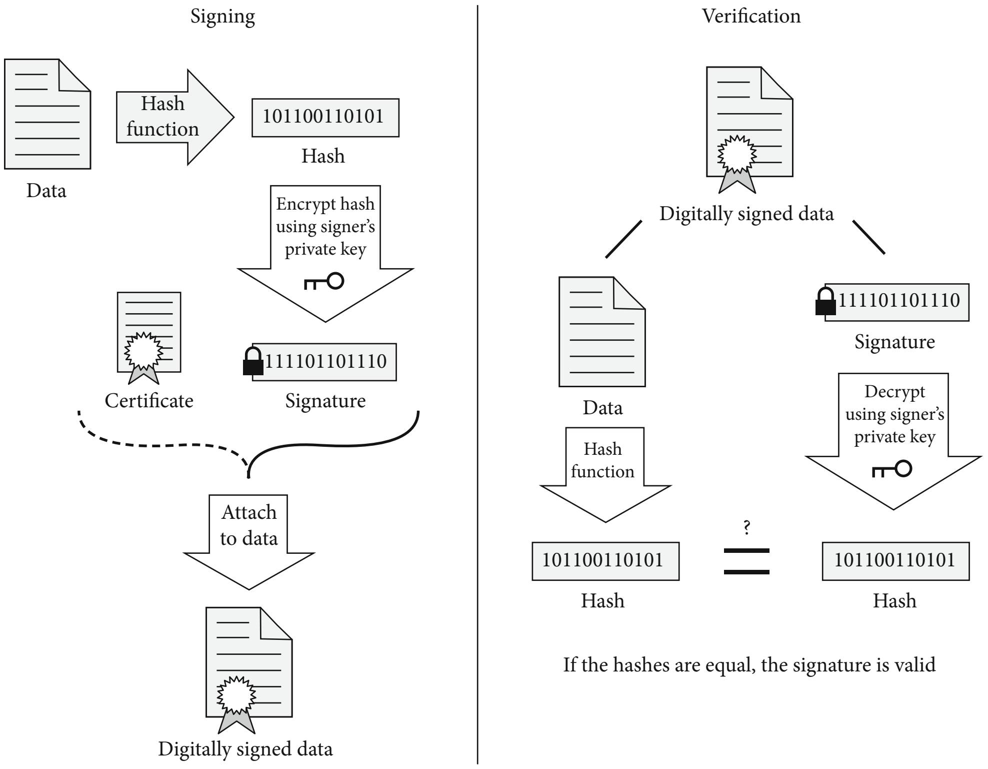

Figure 1: Principle of digital signature and digital signature authentication.

(2) A encrypts s with B's public encryption key to obtain ciphertext $c = E_B(s) = E_B(D_A(m))$ and transmits ciphertext c to B

(3) B first decrypts the received ciphertext c with his decryption key to obtain the following formula:

$$D_B(c) = D_B(E_B(s)) = s = D_A(m) \tag{3}$$

(4) B then encrypts s with s public encryption key:

$$EA(s) = EA(DA(m)) = m \tag{4}$$

In this way, B can confirm that the received information m is indeed sent by A. Because s is derived from a decryption key, only A can.

(5) Based on the obtained information m, send the following ciphertext to A:

$$c' = E_A(D_B(m)) \tag{5}$$

(6) After receiving the ciphertext c', A performs the following processing:

$$E_B\left(D_A\left(c'\right)\right) = m' \tag{6}$$

A then compares m' with the m it sent to B. If m indicates that B has indeed received the information $m = m'$ sent by A in total. For example, C intercepts the ciphertext $c_1 = E_B(D_A(m))$ sent by A to B. We can divide it into two cases: (1) processed and (2) no processing. In both cases, the sender of C to the ciphertext is A. The recipient is certainly B.

In the first case, C decrypts the ciphertext c_1 with A's public key to obtain $E_A(c_1) = E_B(m)$. They then "pseudo-sign" it with their secret key and send $c_2 = D_C(E_B(m))$ to B in C name. After B receives c_2, it will act on it according to the protocol: $E_C(D_B(c_2)) = E_C(D_B(D_C(EB(m))))$. In this way, the plaintext m can also be obtained. The following action is that B sends $c_3 = E_C(D_B(m))$ to C. So, C can get the original text m. In this case, since what B recovers from the received ciphertext is the plaintext under normal conditions, B cannot determine whether the sender of the ciphertext is C or not.

In the second case, C sends the intercepted ciphertext c_1 to B as C without processing. Do the following mechanically per-protocol B: $E_C(D_B(c1)) = E_C(D_B(E_B(D_A(m)))) = E_C(D_A(m)) = m'$, and return $E_C(D_B(m')) = E_C(D_B(E_C(D_A(m))))$ to C. In this way, C can obtain the original text m. In this case, as long as B receives the ciphertext $E_B(D_A(m))$, it does not simply and mechanically execute the "receipt" step but first observes whether the decrypted result is a meaningful plaintext. If so, accept m and further perform the "receipt" operation. They send $E_A(D_B(m))$ or $E_C(D_B(m))$ to the sender (A or C), otherwise they reject it. In this way, unsafe factors can be avoided.

3.2. Shamir Agreement. We use the Shamir agreement. The two parties of any pair of secret communication do not need to establish a public key or a private key to obtain a secure transmission with digital signature properties [10]. The encryption key E_A and the decryption key D_A are kept secretly by Party A itself. E_B is also supported by Party B itself.

(1) A signs the message m with its encryption key to get $E_A(m)$ and sends it to B

(2) After receiving $E_A(m)$, B uses its encryption key to perform encryption operation to obtain $E_B(E_A(m))$ and returns it to A

(3) After A receives the $E_B(E_A(m))$ sent by B, he decrypts it with the decryption key he owns to obtain $D_A(E_B(E_A(m))) = E_B(m)$ and sends $E_B(m)$ to B

(4) After receiving $E_B(m)$, B decrypts it with the decryption key mastered by B to obtain $D_B(E_B(m)) = m$

When A encrypts the information m with its encryption key, the ciphertext received by B is equivalent to A's unique imprint. Unlike the digital signature B in the usual sense, A's identity cannot be confirmed immediately [11]. The received ciphertext is encrypted with B's encryption key and sent to A . Only after the encrypted ciphertext is partially deencrypted by A can B obtain the plaintext information m through its decryption key.

3.3. Electronic Certificate. The identity authentication technology based on electronic certificates is an identity authentication scheme suitable for multiple security domains. In this scheme, our user's identity and public key are encapsulated in a certificate and digitally signed by the CA by a trusted or certifiable third party. At present, the X.509 certificate standard of ITU-T is generally recognized by the industry. In this way, the trust in the user's claimed identity is guaranteed by faith CA. As long as trust can be established between CA in different security domains [12], then we can build trust in each other's users. When a user in a security domain requests services from other disciplines, as long as the user certificate CA is recognized by the target security domain, the user's identity can be authenticated. The management of users by the certificate-based authentication method is the responsibility of trusted or certifiable CA distributed in different security domains. As long as the user can be authenticated by a CA, it can be shown by any security domain that can deliver the CA. We do not require the user to register with the authentication server for every security domain he may request.

Suppose the confidentiality of the data transmitted by both parties is relatively high. We can store the public key in the directory server as an electronic certificate after being authenticated by the certificate server for identity authentication. We use secondary key, signature, and message digest technology to ensure the integrity and confidentiality of information based on guaranteeing public key security.

The public key of the CA is known in advance by all communicating parties. The communicating party A has a public key PKA. Its electronic certificate has the following forms:

$$\mathrm{Cert}(A) = (\mathrm{ID}(A), \mathrm{PK}_A, \mathrm{sig}_{\mathrm{CA}}(\mathrm{ID}(A), \mathrm{PK}_A)). \quad (7)$$

Before each party communicates, we obtain the electronic certificate of the other party. It receives the public key of the other party after authentication. Then, we encrypt our randomly generated conversation key with the other party's public key and send it with a signature generated with our private key. The other party decrypts and verifies its validity after receiving the session key. It then also issues its own randomly generated and signed session key. At this time, both communication parties have two session keys. We combine them into the actual session key K used. The process is as follows:

(1) $A\text{–>}\mathrm{DS}(\mathrm{ID}(B), \mathrm{sig}_A(\mathrm{ID}(B)), A$ requests directory server DS for an electronic certificate, including B public key. It also includes its signature

(2) $\mathrm{DS}\text{–>}A(\mathrm{Cert}(B), \mathrm{sig}_{\mathrm{DS}}(\mathrm{Cert}(B)))$. Directory server B returns DS an electronic certificate to verify DS signature with the known DA public key after being received by A

(3) $A\text{–>}B(E_{\mathrm{PK}B}(R_{A0}, \mathrm{ID}(A)), \mathrm{sig}_A(E_{\mathrm{PK}B}(R_{A0}, \mathrm{ID}(A))), A$ generates a random number$RA0$ and its own identity $\mathrm{ID}(A)$, encrypts it with B's public key, and sends it to B. It also includes its signature

(4) After $B\text{–>}\mathrm{DS}(\mathrm{ID}(A), \mathrm{sig}_B(\mathrm{ID}(A))), B$ receives the application for exchanging session keys from A, it decrypts the received information with its own secret key to obtain the identification $\mathrm{ID}(A)$ of R_{A0} and A. We now query DS for A public key

(5) $\mathrm{DS}\text{–>}B(\mathrm{Cert}(A), \mathrm{sig}_{\mathrm{DS}}(\mathrm{Cert}(A)))$. It is similar to step (2).

(6) $B\text{–>}A(E_{\mathrm{PK}A}(R_{B0}, \mathrm{ID}(B)), \mathrm{sig}_B(E_{\mathrm{PK}A}(R_{B0}, \mathrm{ID}(B))))$. It is similar to step (3).

(7) A and B, respectively, calculate the session key for communication:

$$K = f(R_{A0}, R_{B0}) \quad (8)$$

The above process can be simplified as follows:

$$\frac{E_{\mathrm{PK}B}(R_{A0}, \mathrm{ID}(A)), \mathrm{sig}_A(E_{\mathrm{PK}B}(R_{A0}, \mathrm{ID}(A)))}{E_{\mathrm{PK}A}(R_{B0}, \mathrm{ID}(B)), \mathrm{sig}_B(E_{\mathrm{PK}A}(R_{B0}, \mathrm{ID}(B)))}. \quad (9)$$

We use the above method to prevent insertion attacks. R_{A0} typical case of an insertion attack is when C intercepts A and replaces it with R_{C0}. This means that C must replace the original digital signature $\mathrm{sig}_A(E_{\mathrm{PK}B}(R_{A0}, A))$ with a forged digital signature $\mathrm{sig}_A(E_{\mathrm{PK}B}(R_{C0}, A))$. Because C does not have A secret key, it cannot compute A signature on $E_{\mathrm{PK}B}(R_{C0}, A)$. C then receives R_{B0} from B and attempts to replace R_{B0} with R'_{C0}. Likewise, C cannot generate B signature.

3.4. Other Agreements. In general, digital signatures are sent to the receiver after the sender encrypts and signs the information m. Anyone who knows A public key can verify this signature. A nonrepudiation signature requires A participation in its verification. B cannot prove the correctness of the signature to a third party. Another blind protocol applies when A wants B to sign message m but does not let B know the content of m.

4. Conclusion

This paper introduces a new type of enterprise contract economic management. The system uses Java language technology. The core idea is to use the computer vision signature encryption system. The security package contains a series of API functions. It can provide encryption, information fusion, key management, authentication, access control, and digital signature. It allows developers to develop low-level and high-level security applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study was financially supported by major research projects of Philosophy and Social Sciences in Jiangsu Universities, No. 2020SJZDA066, and Industry University Research Project of Jiangsu Province, No. BY2021213.

References

[1] U. Rahardja, S. Sudaryono, N. P. L. Santoso, A. Faturahman, and Q. Aini, “Covid-19: digital signature impact on higher education motivation performance,” *International Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 65–74, 2020.

[2] Y. Shang, “Application of mathematical signature technology in computer information security design,” *Journal of Frontiers of Society, Science and Technology*, vol. 1, no. 4, pp. 141–144, 2021.

[3] S. M. Pournaghi, M. Bayat, and Y. Farjami, “MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4613–4641, 2020.

[4] S. R. Maniyath and V. Thanikaiselvan, “A novel efficient multiple encryption algorithm for real time images,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1327–1336, 2020.

[5] M. B. Kiliç, “Encryption methods and comparison of popular chat applications,” *Advances in Artificial Intelligence Research*, vol. 1, no. 2, pp. 52–59, 2021.

[6] M. Nurullaev and R. D. Aloev, “Software, algorithms and methods of data encryption based on national standards,” *IIUM Engineering Journal*, vol. 21, no. 1, pp. 142–166, 2020.

[7] C. Jin, G. Chen, C. Yu, and J. Zhao, “Deniable authenticated encryption for e-mail applications,” *International Journal of Computers and Applications*, vol. 42, no. 5, pp. 429–438, 2020.

[8] S. Yin, J. Liu, and L. Teng, “Improved elliptic curve cryptography with homomorphic encryption for medical image encryption,” *International Journal of Network Security*, vol. 22, no. 3, pp. 419–424, 2020.

[9] Z. Guan, N. Wang, X. Fan, X. Liu, L. Wu, and S. Wan, “Achieving secure search over encrypted data for e-commerce: a blockchain approach,” *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 1, pp. 1–17, 2021.

[10] I. Maurya, “Image encryption using chaos and scrambling procedure,” *JIMS8I-International Journal of Information Communication and Computing Technology*, vol. 8, no. 1, pp. 409–413, 2020.

[11] J. Liu, Z. Liu, C. Sun, and J. Zhuang, “A data transmission approach based on ant colony optimization and threshold proxy re-encryption in WSNs,” *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 23–31, 2021.

[12] B. P. Kavin and S. Ganapathy, “A new digital signature algorithm for ensuring the data integrity in cloud using elliptic curves,” *International Arab Journal of Information Technology*, vol. 18, no. 2, pp. 180–190, 2021.