Certificate Management Client System for E-transactions on Internet

Jeom goo Kim[1], Yoochan Ra[2], and Jaehwan Lim[3]

[1] Dept. of Computer Science Namseoul Univ., Korea
jgoo@nsu.ac.kr
[2] Dept. of E&IC Eng. Namseoul Univ., Korea
[3] Dept. of Multimedia Namseoul Univ., Korea

Abstract. In the paper we propose to design and implement an efficient certificate management client system. A basic function of the system authorizes client oneself, and manages to a certificate safely using the Rijndael algorithm. Also, this proposed system solves to the efficiency of private key and the vulnerability problem of certificate management in all kinds of E-transactions on Internet environment.

1 Introduction

While a diffusion and utilization on the Internet network were generalized, information distribution of various forms that used the Internet and a financial transaction, E-commerce became generalization. A certificate is the strongest means to prove an individual and an organization to use these E-transactions service. However, various problems have that an individual uses a certificate with direct management. It is hard to keep a certificate safely. The reason is because some certificate management system is not holding carelessness of a developer or standard observance to embrace, and it is containing a problem on security [1][2][9]. A certificate is a very important means to confirm personal identity in electronic transactions, Internet banking, electronic documents exchange and the same business on the Internet. Therefore, use is convinced of gradually, but it is embracing a problem on security that realistic use subject manages a certificate directly. We propose to design and implement that the client system could save a certificate safely and management that makes use of the Rijndael algorithm for a public key authentication method. The system have been implemented which applied authentication about client oneself and a dual authentication mechanism of safe private key save in order to improve vulnerability of the existing certificate management system.

2 Related Work

Various products were developed for Internet various user authentication in a lot of enterprises. We are considerate a characteristic about Magic PKI v2.0 of DreamSecurity company, UniCERT of Baltimore company, Managed PKI of

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2660, pp. 596–604, 2003.

VeriSign company and MIT CA [11][12][13],and does in order to be comparable with a system to have been implemented in this paper. Magic PKI v2.0 ensures reliability of a mutual inter-working and service. It is based on wire and wireless integrated certificate issue function,and solves a memory load it uses a multi-thread method, and to occur in the existing fork method and a problem about delay of processing speed and implements the most suitable processing. Magic PKI v2.0 disconnected a key for issue and a key for an operator that raised safety of authentication system which applied certificate management business to various organization systems and maximized flexibility of systems operation. However, the disadvantage where Magic PKI v2.0 uses The RSA and the ECDSA cryptographic algorithm which is implementing a digital signature and an encoding function, and a speed was slowly [13]. The UniCERT reduced bottleneck phenomenon which applied a fabricated design,and it was constructed by hardware for established each department and a business burden about a part of a company in order to disperse duly. The UniCERT use is easy and flexible,and it is usable properly according to scale and there is the merit that compatibility is outstanding. Also,The UniCERT supported strong security, multiple security policy and carried various applications. However,The UniCERT had disadvantage capacity of a system grew larger for large compatibility, high expensive, and a lot of encoding and decoding time waste [11][13]. Managed PKI is suitable for demand of a place to use that increased and is composing. Therefore, it is hard to discuss because each characteristic is each different. However, Managed PKI is using ID and a password in a CA authentication method and has a weakness point problem of transmission layer. MIT CA does not provide application of a certificate and an up-to-date menu in a client, and application and an update of a certificate, all functions about distribution are performed through a MIT software distribution page of a MIT CA system. MIT CA is supporting verification of a certificate, safekeeping, and a function about a reading. However, development of all software cannot support the newest algorithm, and there is a security problem. The reason is because it was performed with construction of a Korbers system in the later half of 1990's [12].

3 Certificate Management Client System

This certificate management client simplifies a complicated process related to public key authentication and carries out efficient certificate management which lets encoding of a message coming and going in authentication of a client and an interval with certificate server and encoding save of a key be efficient and establishes a system in order to be safe. In order to be so, safe communication channel must be established between certificate server with the client which stands up simplification of appropriate business and an instant answer about a request of a client must be possible in the range that does not injure reliability of public key-based authentication system. Also, the system can verify a public key pair generation for a certificate generation and effectiveness true or no of a certificate with a client self-enemy that accesses an open treasure house which a certificate and an inquiry of a certificate disuse list must be possible for certificate utilization and management to be efficient.

It can try to divide a module related to an authentication function of a client by a basic module for a certificate generation and management which an expansion module for certificate utilization greatly. A basic module can try to divide by an initial registration/authentication module for an initial generation of a certificate, EE initialization module, and a key pair generation module. An expansion module is a module to verify effectiveness of a certificate, the certificate update module that used key pair update, a certificate disuse request module and a certificate reading modules.

3.1 Client System Model

A certificate management client inter-works with certificate server, and it is asked authentication relation of a user and makes a request message. Then it takes charge of role which it receives a message of certificate server responded to in transmission and a request to authentication server, and to transmit to a user.

Client model structure: A certificate management client is composed of a request process module, a message switching module, a Kerberos module and a private key save module. A client as above compares with a small scale system administered on current construction and supports safe safe-keeping of private key used by electronic signature, and block off illegal message, a change message coming and going of a client between authentication server.

The key save module: As keep the private key used in encoding and decoding of a received document very important in authentication system. If it has been lost private key then basic reliability collapses in authentication system. Also a relevant certificate cannot use and future problems to occur remain. This paper used the Raijdael algorithm with encoding and decoding for private key safe-keeping. The Raijndael algorithm is the safest algorithm to be able to raise security intensity recently.

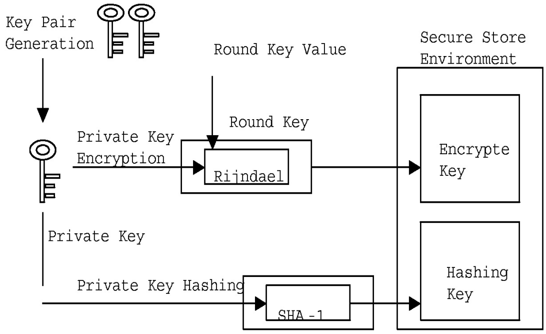

Fig. 1. The private key save module using the Raijndael algorithm

The Rijndael algorithm is block-cipher appointed the DES by ANSI in place of in 2000. A characteristic of the Rijndael algorithm is the 128 bit, 196 bit and 256 bit that block length and key length are variable, and the later cancer that carried out round algorithm of 14 times of 10 and 12, gets encoding and decoding data, and it is byte sub, shift row, key addition, mix column with round function for cryptography basically. A round for encoding carries out four above conversion processes in order, and round function for decoding passes through the process when reciprocal transformation holds a round for encoding. As for the Rijndael algorithm, it is fast as stream-cipher algorithm 128-bit AES, 256-bit SHA-1 and cryptology enemy intensity of the same degree. Fig. 1 shows the model that implemented the private key save modules.

Message exchange module: If it is exchanged a message for a client in the Internet where certificate server was opened to, it is exposed for message effluence and a security menace element as a change in a provincial office of hackers. A message related to a certificate is safe with encoding message format defined by RFC 2511, RFC 2797 and a PKCS #7 standard. However, it is important to establish safe communication channel between certificate servers with the client which stands up in order to raise security intensity more.

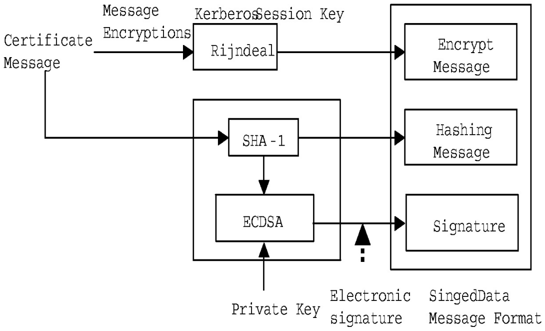

Fig. 2. The message encoding

1. Message encoding is to do an attack of a lot of form with developer and administrator who doing authentication to authorize about specific service to a user under distribution environment in order to be able to stop attack. There is a process a Kerberos authentication mechanism encrypts a message exchanged through established communication channel to transmit. A message to generate in a client or authentication server is composed to an encrypted message part and signature section. Fig 2 illustrates that describes a process generating the message which passes through the message encoding process that was encrypted. Key value of the Rijndael algorithm can exchange a message in a safe channel with what use the later exchanged confidential key which passed through a Kerberos authentication process in

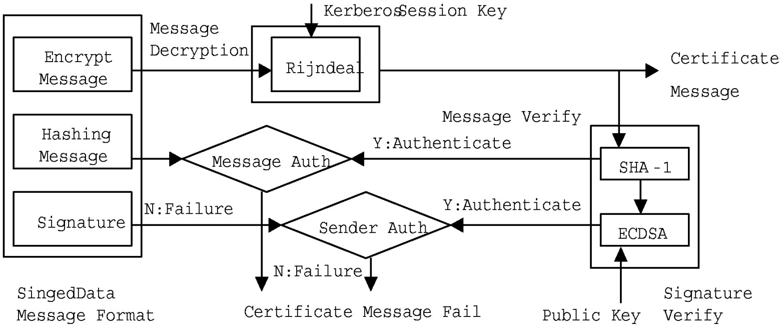

Fig. 3. The message decoding

a message-encoding model. The following is technology about a process to encrypt a message.

(a) It uses the confidential key which it got in a Kerberos authentication process when it uses Rijndael algorithm and encrypts a message.
(b) It uses the SHA-1 algorithm, and a message does hashing.
(c) It signs Hash value with private key of a message transfer.

2. Message decoding is the certificate management message which was received to a client or authentication server verifies decoding and signature to each part. Fig. 3 illustrates that describes these process.
(a) Decoding does a message with the confidential key which it uses the Rijndael algorithm which got a received message in a Kerberos authentication process.
(b) Hashing does the message which it uses the SHA-1 algorithm, and was restored.
(c) It uses the ECDSA algorithm which verifies signature with a public key of a message transfer.

Clint authentication module: A client system uses the Kerberos for safe message exchanging, but an adaptation of a Kerberos total system makes a certificate management system complicatedly that it is increased an unnecessary management element. Because it was so, it applied a Kerberos model of a simplified form in this paper and implemented.

As for the Kerberos which applied to this paper, an AS (Authentication Server) module was deleted. Even if authentication about ID and password of a client user operating in AS and an access ticket about the TGS deletes, it does not have a large influence on a total system. Clint and process that CA gets approval for message exchange can summarize as following.

1 step: ClientCA -> TGS: IDv || Tackett's || AuthenticatorclientCA
2 step: TGS -> ClientCA: EKc, tgs [Kc, v || IDv || TS2 || Ticket
3 step: ClientCA -> CA: Ticket || AuthenticatorClientCA
4 step: CA -> ClientCA: Ec, v [TS4+1]

(a) A client is ID of a user in order to order a service-approval ticket, ID of service to request, and transmits the message which included a ticket-approval ticket to TGS.
// User ID and a time-stamp creation to ask a service-approval ticket byte [ ] A = pbeCipher2.doFinal(ID); byte [ ] t = pbeCipher2. do Final (ts);
// Transmits a created ticket-approval ticket to TGS and requests a user and service authentication //
Con1.sendInt (cipher _Ticket.length); Con1.sendBytes (cipher _Ticket);

(b) TGS decode the ticket which was received, decides on success of decoding whether relevant ID exists and checks effectiveness time. The TGS compares a user ID and a network address with information to have come for confirmation of a user. If access of authentication server is admitted, the TGS issues a ticket-approving interface in the service that it requested.
// A service approval ticket issue about demand of a client byte [ ] cipher_Ticket = pbeCipher2.doFinal (Ticket);
// Guarantee value transmission to allow access of a client and
certificate server//
Out.writeInt (cipher_Kab.length); Out. write (cipher_Kab, 0,
cipher_Kab.length);
Out.writeInt (cipher_Nbr.length); Out. write (cipher_Nbr,
0, cipher_Nbr.length);
// Transmits a ticket approving interface from a client to the service
that requested//
Out.writeInt (cipher_Ticket.length); Out. write (cipher_Ticket, 0,
cipher_Ticket.length);

c) A client requests access to certificate server. A client sends the message that user ID and a service-approval ticket were included in authentication server for this object. Authentication server uses contents of a message to gain in authentication and approves use of service.
// It is decoding with a transmission message of a client byte [ ] decrypt_ID = pbeCipher2.doFinal(ID); byte [ ] decrypt_t =
pbeCipher2.doFinal(t);
// Compares a message and approves service use
if (str.regionMatches (0, str1, 0, decrypt_ID.length))
if ((t1<ts*100) && (ts*100<t2*10000))

(d) It transmits a message to share an established confidential key, and it is used the same confidential key to gain in authentication with a client as a session key.
// Transmits an established session key to a client and shares a session key
Out.writeInt (ra2.length); Out. write (ra2, 0, ra2.length);

It is authorized a client in order to exchange a message for ClientCA proceeded by the fourth step between authentication server, and it is exchanged a confidential key necessary later by encoding.

3.2 Implementation of Certificate Management Client

The core of this client is a management class of a certificate adhering to certificate management protocol, the kerberos class which secures safe communication channel between a KeyStore class and a client and authentication server saves private key safely, and it is a CipherMessage class for safe message switching. Fig. 4 illustrates that is a formation table of each class composing a client.

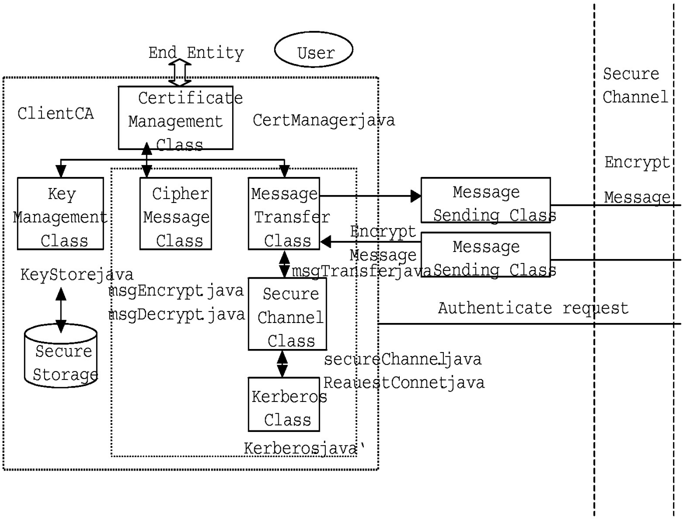

Fig. 4. Composition Class

Private key save: The private key which is used so that decipherment does a signature about a private key save document or a received document must be kept safely. Safekeeping of private key selects a private key save button in a menu of client application. Generally, it appoints safe mobile save media or a directory and database, and save of private key is saved. Safe communication channel establishment: A message exchanged for a safe communication channel establishment between client and authentication server is including personal information of a user. It is a security menace, and, as for these messages which is carried a cow by others when transmitted through an opened network. It has to form a secure data exchange channel between authentication server with a client in order to prevent these problems. This paper proposed the system is certified that used the Kerberos in order to be able to establish safe communication channel, and the system was finished a client and a user.

4 Comparison Performances

The client that implemented in this paper (like the client systems that mentioned) provides application of a certificate to manage a certificate, an update, verification, and a private key save function. And a level can achieve high security requirements with security intensity applying high ellipse curve cryptography algorithm to a module it encrypts a message to transmit each request to, and to verify. Finally, it is coming illegal access and can block off a globe from the outside with using a Kerberos authentication mechanism if an individual client approaches authentication server effectively.

Table 1. Comparison with the proposed client and commercial software

Division	Proposal System	B Product	V Product	E Product	M Product
– Management function of certificate–					
Apply	Yes	Yes	Yes	Yes	No
Update	Yes	Yes	Yes	Yes	No
Verify	Yes	Yes	Yes	Yes	Yes
Save	Yes	Yes	Yes	Yes	Yes
Reading	Yes	Yes	Yes	Yes	Yes
— Algorithms–					
Key save	Rijndael	3-DES	3-DES	3-DES	DES
Message encoding	Rijndael	RSA	RSA	RSA	RSA
Message hashing	SHA-1	SHA-1	SHA-1	SHA-1	SHA-1
Message signature	ECDSA	ECDSA	ECDSA	DSA	DSA
— Protocols —					
Authentication	Kerberos	ID/PW	ID/PW	ID/PW	ID/PW
Encoding	Authentication Key	SSL	SSL	SSL	SSL

Table 1 compared and analyzed a function and an application algorithm with a client of a different product and the client system that it implemented in this paper. Also, it is same as BBB if it tries to currently compared to the Rijndael algorithm and difference with existing private key algorithm with the 3-DES used as private key save algorithm.

Table 2. Table 2 Block-cipher algorithm comparison

Algorithms	Block size	Key length	Number round	Attack
3-DES	64	168	48	K:2/112/56
RC2	64	8 ~ 1024	18	C:64/64/.(16)
RC5	128	8s, s<256	16	C:83/./., C:123/./.(24)
IDEA	64	128	8,5	C:/56/67/32(3,5)
Rijndael	128	128,192,256	10, 12, 14	?

– An attack way explanation

∗ K:a/b/c: Plain text/cipher teat of 2^a is necessary in Known Plaintext Attack, must enconding of 2^b and needs memory of 2^c.

∗ C:a/b/c: Plain text/cipher teat of 2^a is necessary in Chosen Plaintext Attack, must enconding of 2^b and needs memory of 2^c.

∗ (r) : An attack round is numerical, and a round of algorithm is numerical
if it is a blank
∗ ? : If attack law is unknown

It has a smaller round number than the 3-DES, and the Rijndael can know that the clear attack way that it was made to know does not exist so far as it sees in the above table. That is, it is displaying that it is distinguished than the existing algorithm in the accomplishment speed side and a safety plane.

5 Conclusion

A certificate is carrying a form of electronic documents authorizing a public key of an owner in an organization having trust. And A certificate is including a position organization of an owner, a contact point and the personal information to the certificate. But it is not based on most carelessness of the authentication service product program developers who listened to or standard recommendations excluded a product used in an authorization, it holds weakness point on security because of the cause and it has a problem without certificate effluence changes, hacking and robbery about private key. It applied authentication and a security technology element of encoding to public key authentication system which it designed a safe certificate management client and implemented the plan that can solve these problems in this paper. A system proposed in this paper can be operated in the limit that it does not injure range of a standard document with safely. And specially, it will be usable in a small-scale organization and authentication system construction to use in the organization which had electronic transactions specific object effectively.

References

1. Joan. Daemen, Vincent. Rijmen, “AES Proposal: Rijndael”, 1999.
2. RSA Data Security, Inc., Public Key Cryptography Standards #1-9, June 3, 1991.
3. IETF, “Internet X.509 Public Key Infrastructure Certificate and CRL Profile”, RFC 2459, 1999
4. IETF, “Internet X.509 Public Key Infrastructure Certificate Management Protocols”, RFC2510, 1999.
5. IETF, “Internet X.509 Certificate Request Message Format”, RFC 2511, 1999
6. IETF, “Certificate Management Message over CMS”, RFC 2797, April 2000
7. Baltimore, Inc., “PKI based E-security”, 2000.
8. William Stallings,“Cryptography And Network Security”, Prentice-Hall Inc., 1998.
9. The Advanced Encryption Standard, http://home.ecn.ab.ca/ jsavard/crypto/
10. Carl Ellison, Bruce Schneier, “Ten Risks of PKI: What You’re not Being Told about Public Key Infrastructrue”, Computer Security Journal, Vol. 16 No. 1, 2000.
11. IETF, “The Kerberos Network Authentication Service(V5)”, RFC 1510, 1993.
12. http://www.verisign.com/products/onsite/index.html
13. http://www.pki.or.kr/eng/detail.asp?co=14