Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Enhancing Key Management and Public Key Cryptography Integration: A Novel Approach with Matrices and Digital Signatures

[1]M. John Basha, , [2]B.Rasina Begum, [2]T.Sheik Yousuf ,[1]S.Annamalai,Selvaraj saravanakumar

[1]School of CSE, Jain (Deemed to be University), Bengaluru, India

[2]Department of computer science and engineering, Mohamed Sathak engineering college, India

Abstract: The concept of securing messages through cryptography has a long history. Indeed, Julius Caesar is credited with creating one of the earliest cryptographic systems to send military messages to his generals. In this paper, we propose a practical and Static key management scheme based on the public key system and a set of matrices with canonical matrix multiplication that provides advanced secure feature for smart card along with other authentication schemes. Throughout history, however, there has been one central problem limiting widespread use of cryptography problem is key management. The term key management refers to the secure administration of keys to provide them to users where and when they are required. A major advance in cryptography occurred with the invention of public-key cryptography. The primary feature of public-key cryptography is that it removes themed to use the same key for encryption and decryption. The public portion of the key pair can be distributed in a public manner without compromising the private portion, which must be kept secret by its owner. As a step towards the systematic application of authenticated public key cryptography, this article proposes an extension to the Java framework to integrate public key cryptography with the implementation of Digital signatures. The process of digitally signing starts by taking a mathematical summary (called a hash code) of the message. This hash code is a uniquely identifying digital Fingerprint of the message. If even a single bit of the message changes, the hash code will dramatically change. The next step in creating a digital signature is to sign the hash code with your private key. This signed hash code is then appended to the message. The recipient of your message can verify the hash code sent by you, using your public key. Public-key encryption is used to solve the problem of delivering the symmetric encryption key in a secure manner. To do so, you would encrypt the symmetric key using the receiver"s public key. Since only the receiver has the corresponding private key, only the receiver will be able to recover the symmetric key and decrypt the message. It is our belief that such an extension would help speed up the Public Key Cryptography.

Key words: Public Key management, Matrix multiplication, Digital Signature, Cryptography, Public Key Cryptography

I. INTRODUCTION

[1]The access control problem in an arbitrary partially ordered user hierarchy is defined below. In an organization, the users and their authorized data are organized into a group of disjoint sets of security classes, and each user is assigned to a certain security class called his security clearance. Let $C1$, $C2$, …, Cn, be n disjoint security classes and „≤" be a binary partial-order relation over the set $C=\{C1,C2,., Cn\}$.For the set $(C, \leq)$, $Cj \leq Ci$ (i, j) means that the users in the security class Ci have a security clearance higher than or equal to that in the security class Cj. In other words, the users in security class Ci can read or store the data held by the users in security class Cj, but the opposite is not allowed. Fig.1 shows an example of a partial-order hierarchy. Note that the arrowhead in Fig.1

means the higher level security classes have a security clearance higher than that of the lower level classes. For the relation *Cj*≤*Ci*, *Ci* is called a predecessor of *Cj*, and *Cj* a successor of *Ci*. Further, if *Cj*≤*Ci* and if there is no other security class *Ck* such that *Cj*≤*Ck*≤*Ci*, then*Ci* is said to be an immediate predecessor of *Cj*, and *Cj* an immediate successor of *Ci*. Generally speaking, each user in security class *Ci* is assigned a secret key *Ki*. When he wants to store a data *x* into the database or broadcast it to the network, he first encrypts *x* by his secret key *Ki* to obtain $x'=E_k(x)$□ , then stores or broadcasts *x′*. Only users in possession of *Ki* are able to retrieve *x* by calculating $x=D_k(x'')$ *E* and *D* are called the encryption and decryption algorithms, respectively. However, the key *Ki* is only used for encrypting or decrypting the database entitled to security class *Ci*. That means when a user in security class *Ci*, with a higher clearance than *Cj*, would like to retrieve data encrypted by *Kj*, he should get the right key *Kj* first. In the real world, it is not difficult to conceive that examples of hierarchical access control are required. One is the personnel of achain of department stores, where employees are grouped by their ranks into a partial-order hierarchy. Similar situations abound in other areas, particularly in the government and the military, are easily envisaged. Moreover, consider a secure distributed system where hosts operate at different security levels and the encrypted data are broadcast to the network without concern for misrouting since the unintended recipients would be unable to decrypt the data.

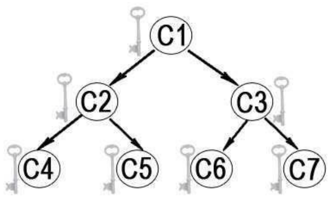


Fig.1 A partial-order hierarchy.

II. THE PROPOSED KEY MANAGEMENT METHOD

The system chooses a public key from the Partial Ordering, encrypt it and embed into an image, andthere is a central authority (CA) that generates and assigns a key to each user in a hierarchy. The Key what so received from the CA is subjected to Encryption and for embedding. All secret parameters are managed by the CA. We assume that the partially ordered binary relationship of " ≤"and *i ID* is the identity number of user U_i. First, we will state some notations and terminology used in our scheme. There are inheritance relationships among the nodes (users) in a hierarchy, the son U_i of a node U_j is defined as a direct child node of user U_i and U_j is defined as a direct parent node of U_i. Furthermore, the inheritance relation is transitive, that is, if node U_i is the son of U_j and U_j is the son of node U_k, then *U* is called as an indirect child node of U_k and U_k is called as indirect parent node of U_i. In order words, if it has relations $U_i \le U_j$ and $U_j \le U_k$, then it provides the relation $U_i \le U_k$. That is, a high-level node in the inheritance relation is transitive, that is, if nodeU_j is the son of U_j and U_j is the son of node U_k , then U_j is called as an indirect child node of U_k and U_k is called as indirect parent node of U_j . In order words, if it has relations $U_i \le U_j$ and $U_j \le U_k$, thenit provides the relation $U_i \le U_k$. That is, a high-level node in the hierarchy can derive the keys of its direct or indirect child nodes.

Though the goals of authenticated encryption have long been studied, most of the researchers have mainly focused in the context of secret-key cryptography and message authentication code that is a symmetric-key equivalent to signature. Zheng [2] considered the problem in the context of public-key cryptography, with signcryption. The main problem considered in the paper [2] was how to design encryption and signature so that their concatenation maximizes savings of computing resources. A security model of parallel signcryption was defined recently for one-to-one user setting in [3]. However, with the rapid development of e-commerce, the conventional cryptographic schemes designed in the literature are unable to deal with the complex situation under a mode of multi-user setting.

Motivated by the above-mentioned security requirements, this paper elaborates the merits of Key Management encryption for a user in the group (t, n).The proposed scheme enables a signer cooperatively to produce a valid authenticated cipher text on behalf of the original group while less than or equal to $t-1$ cannot. This paper on the basis of cryptography work only for text with in alphabets [A-Z,a-z] and in the case of special characters .[dot] is accepted.

III. RESEARCH ON KEY MANAGEMENT

The scheme by Jeng [1] is a representative authenticated encryption scheme with Rabin Public Key System, given its efficiency and performance; therefore, this scheme is used as an example to introduce earlier research. This scheme has Practical and Dynamic Key Management scheme, Key Generation Algorithm, Key Derivation Algorithm – as mentioned below,

A. *Practical and Dynamic Key Management Scheme*

Before describing our key management scheme, we introduce Rabin public-key system (Rabin, 1979) first. The key point of Rabin system is to select a secret pair of large prime numbers (p, q)and to publish the number m where m=pq. The encryption procedure E is given by

$$c = E(m) = M(M+b)(\bmod m) \tag{1}$$

M is a plaintext, C is a cipher text, and b is a public integer. The decryption procedure D is to getsolutions M from the following congruence:

$$M2 + Mb - C = 0(\bmod m) \tag{2}$$

Since the number m is the product of two large prime numbers p and q, solving Eq.(2) is equivalentto solve both of the following congruence"s:

$$M2 + Mb - C = 0(\bmod p) \tag{3}$$

$$M2 + Mb - C = 0(\bmod q) \tag{4}$$

B. *Key generation algorithm*

1) Suppose there are n, nϵN, security classes in a user hierarchy over the partially ordered relation. A central authority (CA) publishes the value b and dominates a pair of distinct prime numbers pi and qi for security class Ci where 1≤i≤n.

2) For each security class Ci, CA constructs a secret matrix Vi= (vst) n.3, where the jth row vector of Vi is equal to [pjpi qjqi IDj] for all Cj≤Ci where IDj is the identifier of Cj; otherwise, it is [0 0 0] if Cj is not a successor of Ci.

3) CA distributes (pi, qi, IDi, Vi) to security class Ci secretly.

C. *Key derivation algorithm*

Assume a user *ui* in the security class *Ci* wants to access the encrypted data held by the user *uj* inone of his successor classes *Cj*, *ui* can derive the secret key *Kj* of *uj* by the following steps:

1) Get *b*, *Xj* from the public database and the *j*th row vector of his own secret vector ***Vi***.

2) Compute $pj=vj1/pi$.

3) Compute $qj=vj2/qi$.

IV. PROPOSED METHOD

In this paper the proposed Method to implement static key management scheme is described below.

1) Signer u_i (i=1,2,…t) randomly selects k_i $\{1,2,.\in n-1\}$ and calculates the Length of the given String N and find the square root of N where N=square root(String) and make N*N Canonical Matrix

2) While (i*j) > nEvaluate

$$\sum_{i=0}^{N} \sum_{j=0}^{N} str[i*j] > N \quad (5)$$

Where N is a String

3)

To find the Value for x

$$X = 26/2$$

Find the corresponding alphabets based on the value of x.

4)

While(N != NULL)

$$\sum_{i=0}^{R} \sum_{j=0}^{C} mat[i*j] = X \quad (6)$$

5)

To find the value for Y Evaluate

$$Y = \sum_{i=0}^{R} \sum_{j=0}^{C} mat[i][j]\%10 + mat[i+1][j]\%10 \quad (7)$$

6)

To find the value for new Matrix (Encrypt)

$$\sum_{i=0}^{R} \sum_{j=0}^{C} newmat[i][j] = oldmat[i][j] + Y \quad (8)$$

7)

To Decrypt the Key value for the matrix

$$\sum_{j=0}^{C}\sum_{i=0}^{R} newmat[i][j] = oldmat[i][j] + key$$ (9)

V. SIGNCRYPTION GENERATION PHASE

[4]Assume that t signers {u_1, u_2,…., u_t} agree jointly to sign a message m. The message can be divided into t connected message blocks {m_1, m_2, …., m_t},in which m_1 $\in$ $[1,p-1]$ and i=1,2,3,…t. Besides, each signer u_i ($i=1$, 2, ..,t) individually generates his signature block for the message block m_i, according to the following procedure.

1) Collaborate with the other participants to divide the message m into t readable message blocks {m_1, m2... m_t}, and share responsibility for examining and signing the allotted message block m_i;

2) Randomly select an integer $b_i \in [1, q-1]$ to compute Bi as follows;

$B_i = b_i G = (x_B, y_B)$

x_B represents the x-coordinate and y_B the y-coordinate of the elliptic-curve point B_i, the othervalues are determined by analogy;

3) Compute Z_i using the random integer b_i, the value x_B, and the verifier"s public key Y_v, as follows.

$Z_i = (x_B, b_i) Y_v = (x_{zi}, y_{zi})$

4) Send both B_i and Z_i to the otherparticipants via a secure channel;

5) Compute B and Z using all received B_i and Z_i (i=1,2,…,t) as follows

$$B = \sum_{i=1}^{t} B_i = (x_b, y_b) \quad (10)$$

$$Z = \sum_{i=1}^{t} Z_i = (x_z, y_z) \quad (11)$$

Where Z is the common session key of the signer group and the specified verifier U_v;

6) Compute the individual signature block (r_i, s_i) for the message block m_i and declare it to the otherdomestic participants, as follows.

i. Choose a random integer l_i{0,1}and Compute a=h(m_i || l_i), where "||"denotes the concatenationoperator;

ii. Form an instance of a (2,2) Shamir secret sharing scheme with polynomial: F(x)=(a || l_i)+m_i x mod p. Define two shares: S_{i1}=F(1) and S_{i2}=F(2).

iii. Compute transform $k_{i1} = S_{i1} \partial(S_{i2})$ and $k_{i2} = S_{i2} \wp(k_{i1})$

iv. In parallel, calculate

$r_i = k_{i1} .h(i||x_z) \bmod p$

$$s_i = X_{Bi}.b_i - k_{i2}.f(x_i) \prod_{j=1, j\neq i}^{t} \frac{0 - x_j}{x_i - x_j} \bmod q \quad (12)$$

A. *Sign Verification phase*

After receiving the others" individual signature blocks, any one of the t participants, uc, mayperform the following procedure.

1) Verify the validity of each signature block one at a time, according to the following equation.

$$s_i G + \left(k_{i2} \prod_{j=1, j\neq i}^{t} \frac{0 - x_j}{x_i - x_j} \right) y_i \qquad (13)$$

If the discriminatory equation is satisfied, then the individual signature block (r_i, s_i) can bevalidated;

2) Combine all individual signature blocks into a single group-signature block after validating themas follows.

$$r = \sum_{i=1}^{t} r_i \bmod q \qquad (14)$$

$$s = \sum_{i=1}^{t} s_i \bmod q \qquad (15)$$

3) Send the group-signature block ($r, s, r_1, r_2, \ldots, r_t$) for the whole message to the specified verifier $U_v$over a public channel.

B. *Message recovery phase*

After receiving the group-signature block (r,s, r1,r2,…,r_t), the receiver U_v performs the followingprocedure to recover the message blocks $\{m_1, m_2, \ldots m_t\}$, as follows.

1) Compute the common session key Z shared with U_s, using the received (r, s), the public key Y_s ofthe signer group Us, and the private key x_v, as follows.

$Z = sY_v + (r.X_v)\, Y_s = (x_z, yz)$

2) Recover the message blocks according to the following equation.

i. Compute

$$ut_{i1} = r_i . h(i \,||\, x_z)^{-1} \bmod p \qquad (16)$$

$$ut_{i2} = (y_z - s_i) \bmod q \qquad (17)$$

ii. Compute inverse transform

(18) $$k_{i2} = ut_{i2} \oplus \wp(ut_{i1})$$

and $$k_{i1} = ut_{i1} \oplus \partial(ik_2) \qquad (19)$$

iii. Knowing two points (1,ik_1) and (2,ik_2), use the Lagrange interpolation and find thepolynomial

$$\tilde{F}(x) = a_0 + a_1 x \bmod p \quad (20)$$

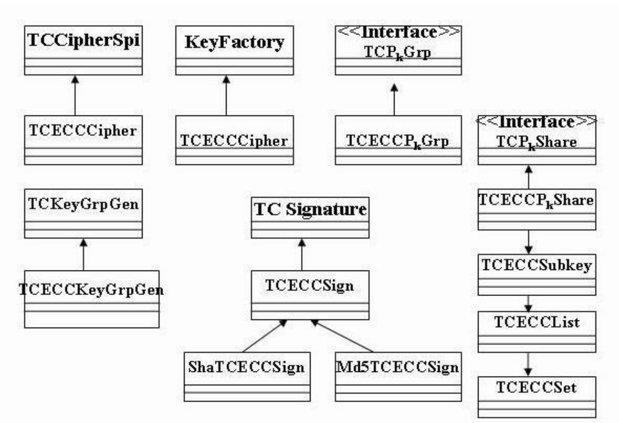
iv. Extract m_i from a_0 as follows $a_0=(m_i||l_i)$

VI. JCA FRAMEWORK EXTENSION

JCA [5] and the Common Data Security Architecture (CDSA) are two cryptographic frameworks

[6] for conventional public key cryptography. Neither of them supports group-oriented cryptography. Recently JCA has been extended to support RSA based group-oriented cryptography.The present work extends the JCA architecture to integrate cryptography, one of the most importanttypes of group-base public key cryptography. The UML class diagram of TC based Java frameworkextension is depicted in fig. Under this extension, various TC providers can be plugged into a security application at runtime. The extension also makes it easy for those JCA-based applicationsto easily be migrated to use Key management cryptography to enhance system security. It is our belief that this integration would speed up the adoption of cryptography for Key Management.

To test the above framework extension, we have implemented an example provider based on the threshold ECC algorithm [7], which can be used for signcryption. The example provider has the following concrete classes:



TCECCCipher inherits the TCCipherSpi and implements the threshold ECC algorithm. It isused in the threshold decryption and in the threshold co-signing as well.

- TCECCSgn extends the TCSgnSpi and provides the threshold signature service.
- TCECCGrpGen is defined to extend the TCKeyGrpGenSpi and used to generate key shares
- TCECCPvtKeyGroup extends the TCPvtKeyGroup and houses a collection ofTCECCPvtKeyShr
- TCECCKeyFactory extends the KeyFactorySpi class

VII. CONCLUSION

A practical and dynamic cryptographic smart card key management scheme for access control in a hierarchy is proposed. It is ensured that the conspiracy of one‟s successors cannot reveal their predecessor‟s secret key, and similarly cannot generate their sibling‟s secret key. The scheme is proven to be secure.

REFERENCES

[1] A practical and dynamic key management scheme for a user hierarchy- JENG Fuh-gwo, WANG Chung-ming

[2] Y. Zheng, "Signcryption or How to Achieve Cost (Signature & Encryption) << Cost (Signature) +Cost (Encryption)", Crypto '97, LNCS 1294, Springer-Verlag, Berlin, 1997

[3] J. H. An, Y. Dodis, and T. Rabin. "On the Security of Joint Signatures and Encryption", Euro crypt '02, LNCS2332 Springer-Verlag, Berlin, 83-107, 2002.

[4] A Java framework for Static Key Management using Digital Signature – V,Karthick T , Varalakshmi R S.,ThavavelV,Suhhuraj

[5].Sun Microsystem, "Java cryptography architecture API specification&reference",http://java.sun.com/j2se/sdk/1.3/docs/guide/security/CryptoSpec.html, 1999

[6] The Open Group, "Common security:CDSA and CSSM,version2",http://www.opengroup.org/publications/catalog/c914.htm

[7] Faz-Hernández, Armando, Julio López, and Ricardo Dahab. "High-performance implementation of elliptic curve cryptography using vector instructions." *ACM Transactions on Mathematical Software (TOMS)* 45.3 (2019): 1-35.

[8] Abdelfatah, Roayat Ismail. "Secure image transmission using chaotic-enhanced elliptic curve cryptography." *IEEE Access* 8 (2019): 3875-3890.

[9] Kumar, Vinod, Musheer Ahmad, and Adesh Kumari. "A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS." *Telematics and Informatics* 38 (2019): 100-117.

[10] Sekaran, R., Munnangi, A. K., Ramachandran, M., & Gandomi, A. H. (2022). 3D brain slice classification and feature extraction using Deformable Hierarchical Heuristic Model. Computers in Biology and Medicine, 149, 105990-105990.

[11] Ramesh, S. (2017). An efficient secure routing for intermittently connected mobile networks. Wireless Personal Communications, 94, 2705-2718.

[12] Sekaran, R., Al-Turjman, F., Patan, R., & Ramasamy, V. (2023). Tripartite transmitting methodology for intermittently connected mobile network (ICMN). ACM Transactions on Internet Technology, 22(4), 1-18.

[13] Kumari, Adesh, et al. "A secure user authentication protocol using elliptic curve cryptography." *Journal of Discrete Mathematical Sciences and Cryptography* 22.4 (2019): 521-530.

[14] Almajed, Hisham N., and Ahmad S. Almogren. "SE-ENC: A secure and efficient encoding scheme using elliptic curve cryptography." *IEEE Access* 7 (2019): 175865-175878.

[15] Khan, Akber Ali, Vinod Kumar, and Musheer Ahmad. "An elliptic curve cryptography based mutual authentication scheme for smart grid communications using biometric approach." *Journal of King Saud University- Computer and Information Sciences* (2019).

[16] Smart, Nigel P., and Younes Talibi Alaoui. "Distributing any elliptic curve based protocol." *IMA International Conference on Cryptography and Coding*. Springer, Cham, 2019.

[17] Dinarvand, Negin, and Hamid Barati. "An efficient and secure RFID authentication protocol using elliptic curve cryptography." *Wireless Networks* 25.1 (2019): 415-428.

[18] Hayat, Umar, and Naveed Ahmed Azam. "A novel image encryption scheme based on an elliptic curve." *Signal Processing* 155 (2019): 391-402.