Organizational Cryptography for Access Control

Masahito Gotaishi[1] and Shigeo Tsujii[1]

Research and Development Initiative, Chuo University,
1-13-27 Kasuga, Bunkyo-ku, Tokyo, Nippon,
gotaishi@tamacc.chuo-u.ac.jp

Abstract. A cryptosystem for granting/rescinding access permission is proposed, based on elliptic curve cryptography. The 'Organizational Cryptosystem' grants access permission not by giving secret (decription) key to the corresponding user but by converting the ciphertext so that the user can decript with their secret key. The 'conversion key' for the document, which is created from the secret key which the ciphertext has been originally encrypted for, the public key of the member who shall be permitted to read the ciphertext, and a part of the ciphertext. Therefore it is not possible to decrypt the ciphertext with the conversion key. Nor, for the administrator who issues the conversion key, to obtain any information about the plaintext.

Keywords: access control, elliptic curve, ElGamal, Cramer-Shoup, IND-CPA, IND-CCA

1 Introduction

1.1 Cryptography for Data Protection

Access Control in computer systems is usually done by identifying each user and their access permission to a given resource. If the user requests read access to a document file, OS or middlewares check the access control list (ACL) for the username. Then if the user is included in the ACL, the access is permitted, or otherwise the request is refused. This security feature is supported in most of systems including Windows security model and various major systems including Unix/Linux.

However, since access permission is based on user authentication, it is often compromised by impersonating the legitimate user, such as by stealing passwords. Additionally, it is reputed that the security of cloud systems is not so strong against internal attackers as network attackers, because in many cases users have no method to know which system administrator is in charge of the system, or even the geologically precise location of the information. As well as preventing the insiders' attack, it is also quite difficult pursuing suppliers for responsibility in case of security breaches, because of jurisdiction.

Therefore "Malicious Insiders" are counted among the top threats to cloud computing [8]. Therefore it is claimed that the data stored in cloud systems should be protected with cryptosystems whose keys are managed outside the cloud, -namely, users should not rely on the organization managing the cloud system.

1.2 Data protection with Cryptosystems

Data protection with cryptosystems has been employed since the early days of personal comupting, such as the Encrypting File System (EFS) of Microsoft Windows [17]. Apparently they are designed to protect the data against physical attacks. Cryptosystems are generally used to encrypt files, directories or whole hard disks.

However, it would be efficient and reasonable supplementing the access control by authorizing access vectors with the one by cryptosystem, according to the principle of "multi-layer security." When read-access of a document, which is already encrypted with some key such as administrator's, is requested by Alice, the system allows access to the document after converting the document encrypted with Alice's (public) key. Then even if an illegitimate user steals Alice's password, he/she could not read the encrypted document unless Alice's secret key is stolen.

However, in most of the existing cryptography, it is impossible to transform a ciphertext which was encrypted with a certain key into the one encrypted with a different key. In order to do that, the ciphertext should be decrypted and encrypted again, with another key. It would be insecure because there is an instant when the ciphertext is decrypted. Even if the decrypted document is deleted as soon as it is re-encrypted, the process where the ciphertext and its decryption key exist together would be dangerous. In order to realize the access control with cryptography, the above problems should be overcome.

1.3 Structure of the Paper

This paper is structured as follows:

In the chapter 2, we briefly survey the existing cryptosystems used for access control, especially in the cloud systems. Next, in the chapter 3, The concept of 'Organizational Cryptography' as the solution for this kind of situation is explained. Then the two kinds of Organizational Cryptography are proposed in 3.2 and 3.3, the former with IND-CPA and the latter with IND-CCA2 security. Each of their security is discussed in the chapter 4. Finally, conclusion and our future plan of its development and application to other kind of access controls is discussed in the last chapters.

2 Existing Cryptosystems for Cloud Security/Access Control

Several kinds of cryptosystems for access control have been proposed before. Among them, most well-known ones would be proxy re-encryption schemes.

2.1 Access Control with Cryptosystem

There exist variations of cryptosystems for access control. Akl and Taylor proposed a cryptographic solution to realize hierarchical access control [3]. Chung, et al. proposed a system to implement this solution with elliptic curve cryptosystem [7]. Their system realized dynamic access control, which enables the system to accept new participants and eliminate obsolete members from the ACL. This cryptosystem also enable to create a new hierarchy in the organization structure. -It seemed a flexible and convenient system.

However, Kuo et al. [15] examined its security and claimed that it is vulnerable to various attacks which were not discussed in their original paper.

Generally, cryptosystems are widely used in data protection. Nonetheless, currently most of the security systems using cryptography seem to have difficulty in dynamic access cotrol. Although it is possible to grant access permission to new participant, it is considerably difficult to revoke the dismissed member's access permission without decrypting and re-encrypting the documents.

2.2 Proxy Re-encryption

The concept of proxy re-encryption has been proposed by Blaze et al [6] and Mambo, et al [16]. This is a system which enables to convert a ciphertext encrypted for Alice to decrypt into the one decryptable by Bob, without decrypting it. In practice, Alice issues the 're-encryption key' and hands it in to a proxy. The proxy re-encrypts the ciphertext for Bob and forward it to him.

Apparently it has been proposed as a system for a manager to delegate the subordinates to cope with the job instead of him. Since the delegated personnel must read documents about the job, access permission to related documents shall be also delegated.

It shall be maintained that, although decryption by Bob is enabled and the proxy must be able to re-encrypt the ciphertext, the proxy shall not be able to cryptanalyze the ciphertext with the re-encryption key. Nor it shall be able to re-encrypt the ciphertext for any person except Bob; it shall not be able to re-encrypt the ciphertext for Chris.

Several schemes of proxy re-encryption have been proposed before [4][13][12].

One of this kind of technologies has been implemented in the access control of cloud storages. Toshiba [2] has once operated a security service entitled "Digital Safe-Deposit Box." This system is designed to share personal (encrypted) files such as photographs and videos on a cloud storage without sharing decryption key. The security service keeps the proxy re-encryption key for each participant and therefore they are not decrypted in converting the personal file of the owner readable for the member. Files are kept encrypted in downloading. If the member leaves the file-sharing group, the owner disables him to decrypt the files by deleting the re-encryption key for him. Japanese 'National Institute of Information and Communication Technology' (NICT) has proposed a security system called "PRINCESS" [20] for managing the security of medical information stored in cloud storages. This system uses ID-Based cryptography, with ID and key-pairs generated for each project.

Although the proxy re-encryption system enables conversion of ciphertext for another decryption key, it would be rather inconvenient in controlling access permission. With this scheme, as long as re-encryption key for Bob is issued, proxy can re-encrypt every document encrypted with Alice's key; read permission of every document is delegated to Bob. However, in the real world, responsibility and power shall be delegated only partially: The general manager would delegate sales and marketing matter to the sales manger, but not the technical matter.

Tsujii et al. proposed a cryptography based on multivariate public key cryptosystem, along the similar concept as Organizational Cryptosystem [19]. Their idea is to give a hierarchical structure to the private (decryption) key, such as "the ciphertext decryptable by a subordinate is also decryptable by the boss, but not vice versa.

3 Organizational Cryptosystems

3.1 Concept of Organizational Cryptosystems

The concept of Organizational Cryptosystems (OC) was structured from the needs of public bodies, especially of departments managing personal data. There actual services for inhabitants are operated by pople on the fields, but assignment of operations and areas change very quickly, because most of operations are performed by part-timers. Therefore managers must change assignment of jobs and manage access permission to information, especially personal ones. However, although managers are responsible for management, they do not perform actual operation, and so they should not access or read the information. Therefore an access control system where the administrator can assign job and access to relevant data without reading it.

The OC data control system prototype was developed in 2014 and several field tests have been performed in several public bodies [18]. Subsequently the OC system was applied for patent in 2015 [10] and its algorithm was published in 2016 [11].

3.2 Organizational Cryptosystem based on EC-ElGamal

OC is realized with elliptic curve cryptosystem (EC) [14]. The commonest one would be the EC-ElGamal. Additionally, there is a securer cryptosystem, EC-Cramer-Shoup [9], which is secure against chosen ciphertext attacks. Both can be used to construct OCs. Here the operation of OC based on EC-ElGamal is explained first.

EC-ElGamal Cryptography An elliptic curve over Z_p, where p is a prime number, is the set of solutions (x, y) to the equation $E_p(a, b) : \ y^2 = x^3 + ax + b \pmod{p}$ where $a,\ b \in Z_p$, and $(4a^3 + 27b^2) \bmod p \neq 0$. Then a finite group G is defined based on the set $E_p(a,\ b)$, where addition and multiplication by an iteger of an element is defined. If $k \in Z_p$ and $P \in G \subset E_p(a,\ b)$ is given, it is relatively easy to compute $Q = kP$. But it is hard to compute P from Q and k. This is used as the trapdoor. EC ElGamal cryptosystem works as follows:

1. Secret Key:
   $k \in Z_p$
2. Public Key:
   Finite group G based on the set $E_p(a,\ b)$, $P \in G,\ K := kP \in G$. P is called 'base point.'

Encryption:

1. The plain text message M is expressed as an element of G
2. A random number $r \in Z_p$ is generated.
3. M is encrypted by computing $M' := M + rK (= M + rkP)$, simultaneously rP is computed.
4. $(rP, M') \in G^2$ is the ciphertext.

The ciphertext is twice as long as the plaintext.

Decryption:

1. $M' - krP (= M + rkP - krP) = M$ is computed.

Organizational Cryptosystem based on EC-ElGamal Cryptography
All documents, which are kept by the Proxy, are encrypted with the Administrator's public key K_A.

Each time User2 requests a document M, which is encrypted by Administrator's key $C := Enc(M)$, Proxy checks his permission to read-access. If his permission is confirmed, Proxy asks Administrator to issue a conversion key for User2 ($T_{C,A\to 2}$). Then Proxy converts the document decryptable with User2's secret key k_2 with $T_{C,A\to 2}$. Subsequently User2 receives the converted document and opens it with his secret key.

- Environmental Parameters:
  Finite cyclic group Z_p and Finite group G based on the set $E_p(a,\ b)$, $P \in G$
- Secret Keys of Administrator, User 1, and User 2:
  $k_A,\ k_1,\ k_2 \in Z_p$
- Public Keys corresponding to the private keys:
  $K_A := k_A P,\ K_1 := k_1 P,\ K_2 := k_1 P$

Encryption:
When User 1 saves a message M, it is encrypted with Administrator's public key K_A. Encryption is done along the procedure of ordinary EC-ElGamal.

1. $C := (M + rK_AP, rP)$. $r \in Z_p$ is a random number generated by User1.

Conversion:
When User2, who has access permission to M (plaintext corresponding to C, requests read access to it, Administrator issues transformation key $T_{C,A\to 2}$.

1. Administrator receives, rP, the second element of the ciphertext C.
2. Administrator issues $T_{C,A\to 1}$ as follows:

$$T_{C,A\to 1} := (T_1, T_2) := (-k_A(rP) + sK_2, sP) \tag{1}$$

   $s \in Z_p$ is a new random number issued by Administrator. Then $T_{C,A\to 2}$ is sent to Proxy.
3. With $T_{C,A\to 2}$, Proxy transforms the ciphertext C into C', which is decryptable with User2's secret key k_2.

$$C' := (M + rK_A + T_1, T_2) = (M + rK_A + (-k_ArP + sK_2), sP) = (M + sK_2, sP) \tag{2}$$

4. The converted ciphertext C' is sent to User2.

Random numbers r, s and k_a, the secret key of Administrator, are disclosed exclusively to the Administrator. Hence, although Proxy can transform C so that User2 can decrypt it, he/she cannot draw any information about the plaintext M from C, C', or the transformation key $T_{C,A\to 2}$. Nor he/she cannnot transform it decryptable for any other member than User2.

EC-ElGamal cryptosystem is widely used and, as long as decision Diffie-Hellman assumption holds, has IND-CPA (indistinguishability against chosen plaintext attack) security. However, it is vulnerable to chosen ciphertext attacks (CCA):

If a plaintext M is encrypted with EC-ElGamal with a plublic key K and a random number r, the ciphertext C is $(M + rK, rP)$. Then the adversary can forge a ciphertext $C' := (M + rK + M', rP)$ with an arbitrary text M' and ask the decryption oracle to decrypt it to obtain the decryption $M + M'$, from which the adversary can easily recover M. Hence it would be necessary to construct a non-malleable OC cryptosystem. It can be achieved by using Cramer-Shoup cryptography.

3.3 Organizational Cryptosystem Based on EC-Cramer-Shoup Cryptography

Original Cramer-Shoup cryptography should be described first.

Cramer-Shoup Cryptosystem

- Environmental Parameters:
  Finite cyclic group Z_p, Finite group G based on the set $E_p(a,\ b)$, $P_1,\ P_2 \in G$, and collision-resistant hash function H
- Secret Keys:
  $x_1,\ x_2,\ y_1,\ y_2,\ z \in Z_q$
- Public Keys:
  $c := x_1P_1 + x_2P_2,\ d := y_1P_1 + y_2P_2,\ Z := zP_1$

Encryption:
A message $M \in G$ is encrypted as follows:

1. $r \in Z_q$ is randomly chosen.
2. Hash $\alpha := H(rP_1, rP_2, rZ + M)$ is computed.
3. The ciphertext is $(u_1, u_2, e, v) = (rP_1, rP_2, rZ + M, rc + r\alpha d)$

Therefore the ciphertext is 4 times as long as the plaintext.

Decryption:
1. Given a ciphertext (u_1, u_2, e, v), $\alpha = H(u_1, u_2, e)$ is computed.
2. It is checked whether $u_1(x_1+y_1\alpha)+u_2(x_2+y_2\alpha)$ is equal to $v(= r(x_1P_1+x_2P_2)+r\alpha(y_1P_1+y_2P_2)$.
3. If the result of 2. is false, the ciphertext is REJECTED.
4. Otherwise, the ciphertext is decrypted: $m = e - zu_1(= rZ + M - z(rP_1))$.

It is checked in the procedure 2. whether the ciphertext is forged from another ciphertext. Therefore, among the private keys, a_1 and a_2 are not used to decrypt the ciphertext but to know whether the ciphertext is a forged one.

Organizational Cryptosystem based on Cramer-Shoup
Like in the OC based on EC-ElGamal, all documents are encrypted so that only Administrator can decrypt before they are saved. In the environment where OC is operated, two kinds of chosen ciphertext attacks are possible:

- Ciphertexts, generated by a malicious user, forging from other ones
- A transformation key falsified by an eavesdropper, to transform an existing ciphertext into a forged one

Proxy is responsible for checking ciphertexts and transformation keys, as well as for transforming ciphertexts.

- Environmental Parameters:
  Finite cyclic group Z_p, Finite group G based on the set $E_p(a,\ b)$, $P_1,\ P_2 \in G$, collision-resistant hash function H
- Secret Keys:
  Administrator: $z_A \in Z_q$
  User1: $z_1 \in Z_p$
  User2: $z_2 \in Z_p$
  Proxy: $x_1,\ x_2,\ y_1,\ y_2 \in Z_p$
- Public Key:
  Administrator: $Z_A := z_AP_1$
  User1: $Z_1 := z_1P_1$
  User2: $Z_2 := z_2P_1$
  Proxy: $c := x_1P_1 + x_2P_2,\ d := y_1P_1 + y_2P_2$

All users including User 1 save documents after encrypting them in Cramer-Shoup with Administrator's and Proxy's public keys.

Encryption:
User 1 encrypts a message M by Administrator's public key and sends it to Proxy after encrypting it.

1. Random number r is generated. $C = (u_1, u_2, e, v) := (rP_1, rP_2, M + rZ_A, rc + (r + \alpha)d)$, where $\alpha = H(u_1, u_2, e)$. Then User 1 sends it to Proxy.
2. Proxy computes $\alpha := H(u_1, u_2, e)$ and checks whether $(x_1 + y_1\alpha)u_1 + (x_2 + y_2\alpha)u_2)(= rx_1P_1 + rx_2P_2 + \alpha ry_1P_1 + \alpha ry_2P_2)$ is equal to v.
3. If the result of 2. is false, Proxy REFUSES to receive the ciphertext. Otherwise Proxy receives and stores it as the one encrypted along EC-ElGamal:

$$(rP_1, M + rZ_A)(= (u_1, e)) \tag{3}$$

Transformation:

Administrator receives the first element of the ciphertext and generates the transformation key for User 2. Proxy checks the legitimacy of the transformation key.

1. Proxy sends rP_1 of $C := (rP_1, M + rK_A)$ to Administrator.
2. Administrator generates a new random number s and computes $\beta := H(sP_1, sP_2, -k_A(rP_1) + sX_2)$.
3. Administrator generates the transformation key:

$$\begin{aligned} T_{C,A\to 2} = &\ (T_1, T_2, T_3, T_4) \\ := &\ (sP_1,\ sP_2,\ -k_A rP_1 + sK_2,\ sc + (s+\beta)d) \end{aligned} \tag{4}$$

   and sends it to Proxy.
4. Proxy computes $\beta := H(T_1, T_2, T_3) = H(sP_1, sP_2, -k_A rP_1 + sK_2)$ and checks whether $(T_1(x_1 + y_1\beta) + T_2(x_2 + y_2\beta))$ is equal to T_4.
5. If the result of 4. is false, Proxy REFUSES to receive the transformation key.
6. Otherwise Proxy receives the key and transforms the ciphertext as

$$(T_1, M + rK_A + T_3) := (sP_1, M + rK_A - rk_A P_1 + sK_2) = (sP_1, M + sK_2) \tag{5}$$

   The resulting ciphertext (in ElGamal) is decryptable with User2's secret key.

Considering the operation of OC, Cramer-Shoup encryption and validation could be performed only in sending ciphertexts to Proxy and transforming ciphertexts and the stored ciphertext could be in ordinary ElGamal.

4 Discussion of Security

4.1 Security of EC-ElGamal and Cramer-Shoup Cryptography

Security proof of ElGamal is based on the assumption that decisional Diffie-Hellman (DDH) problem is hard: The tuple (xP, yP, xyP) (P is a base point) is computationally indistinguishable from (xP, yP, zP). ($x, y, z \in Z_p$ are chosen randomly).

Theorem 1. *If the decisional Diffie-Hellman problem is hard, ElGamal cryptography is IND-CPA secure.*

Its proof would be found in several lecture notes including [1]. IND-CCA2 security is also proved by Cramer, et al.[9]

Theorem 2. *If the decisional Diffie-Hellman is hard and the hash function $H(\cdot,\cdot,\cdot)$ is collision-resistant, ElGamal is IND-CCA2 secure.*

Additionally, IND-CPA security of multi-recipient ElGamal and IND-CCA2 of multi-recipient Cramer-Shoup cryptography was proved by Bellare, et al.[5]. They have given a more general theorem:

Theorem 3. *Let $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme that is polynomially-secure against chosen-plaintext (resp. chosen-ciphertext) attack in the single-user setting. Then $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is also polynomially-secure against chosen plaintext (resp. chosen-ciphertext) attack in the multi-user setting.*

4.2 Security of the Organizational Cryptosystem Based on ElGamal Cryptography

The following game is considered in discussing the IND-CPA of OC based on EC-ElGamal.

Definition 1. *There is Administrator, Proxy, and n users User* 1,. . . ,*User n. There are a pair of messages* m_0, m_1 ($|m_0| = |m_1|$), *and* m_b, *which is either* m_0 *or* m_1, *is encrypted by Administrator's key* K_A*:* $m_b \rightarrow c_b := Enc(K_A, m_b)$.

An adversary, who wants to know whether the plaintext of c_b *is* m_0 *or* m_1, *can do following operations:*

- *Encrypting arbitrary plaintexts by Administrator's public key and sending them to Proxy.*
- *Asking Administrators to transform existing ciphertexts for any of User*1,. . . ,*User n.*
- *Reading any of the ciphertexts, transformation keys, and ciphertexts transformed for any of the users.*

Theorem 4. *If multi-recipient EC-ElGamal cryptosystem is IND-CPA secure, advantage for the adversary in the game defined in* ***Definition*** *1 to distinguish wheter* m_b *is* m_0 *or* m_1 *is negligible.*

Proof. If a ciphertext of a certain message M encrypted by Administrator's public key $C := (M + rK_A, rP = (c_1, c_2))$ (r is a random number) and a ciphertext of the same plaintext encrypted by the key of User i ($1 \leq i \leq n$), $C' := (M + sK_i, sP) = (c_1', c_2')$ (s is another random number) are given, a transformation key to convert C into C' is determined. In concrete, $T_{C,A\rightarrow 1} = (-c_1 + c_1', c_2') = (-rK_A + sK_i, sP)$.

Here let $\mathcal{A}_{OCE}^{n-cpa}$ be an algorithm which can distinguish whether m_b is m_0 or m_1 with non-negligible advantage. Then it is possible to structure an algorithm B, which supplies parameters determined by public keys and ciphertexts of multi-recipient ElGamal cryptosystem to $\mathcal{A}_{OCE}^{n-cpa}$. Then B can defeat the multi-recipient ElGamal with non-negligible advantage. This result contradicts to the assumption. □

4.3 IND-CCA2 Security of the Organizational Cryptosystem Based on Cramer-Shoup Cryptography

For Organizational Cryptosystem based on Cramere-Shoup, the following game is considered in discussing the IND-CCA2.

Definition 2. *There are Administrator, Proxy, and n users User* 1,. . . ,*User n. There are a pair of messages* m_0, m_1 ($|m_0| = |m_1|$), *and* m_b, *which is either* m_0 *or* m_1, *is encrypted by Administrator's key* K_A*:* $m_b \rightarrow c_b := Enc(K_A, m_b)$.

An adversary, who wants to know whether the plaintext of c_b *is* m_0 *or* m_1, *can do following operations:*

- *Encrypting arbitrary plaintexts by Administrator's public key and sending them to Proxy.*
- *Asking any User i to decrypt arbitrary ciphertexts except the one encrypted from* m_b.
- *Asking Administrators to transform existing ciphertexts for any of User* 1,. . . ,*User n.*
- *Reading any of the ciphertexts, transformation keys, and ciphertexts transformed for any of the users.*

Proxy and all Users are used as the decryption oracle.

Theorem 5. *If multi-recipient EC-Cramer-Shoup cryptosystem is IND-CCA2 secure, advantage for the adversary in the game defined in* ***Definition*** *2 to distinguish wheter m_b is m_0 or m_1 is negligible.*

Proof. If a ciphertext of a certain message M encrypted in Cramer-Shoup by Administrator's public key $C := (rP_1, rP_2, M + rZ_A, rc + r\alpha d) = (c_1, c_2, c_3, c_4)$ (r is a random number) and a ciphertext of the same plaintext encrypted by the key of User i ($1 \leq i \leq n$), $C' := (sP_1, sP_2, M + sK_i, sc + s\beta d) = (d_1, d_2, d_3, d_4)$ (s is another random number) are given, a transformation key to convert C into C' is determined. In concrete, $T_{C,A\to i} = (d_1, d_2, -c_3 + d_3, d_4)$.

Therefore, as proved in the ***Theorem*** 4, The advantage of the adversary is negligible. □

5 Conclusion

Organizational Cryptosystem, which has been created to improve the security in access control, is based on Elliptic-Curve Cryptography. This system was highly evaluated in the field tests of the public bodies. Although the proposed technology seemed simple, or because of its simplicity, it is as secure as the underlying cryptographies.

Additionally, according to the test result of the implemented system developed by Saisho et al. [18], its encryption, transformation, and decryption were as quick as ordinary elliptic curve cryptographies.

Authors are planning to implement OC as the security feature of the patients / care receivers monitoring system.

Acknowledgment

This study is supported by the Project 181603006 of Strategic information and COmmunications R & D Promotion programmE (SCOPE) of Japanese Government. The authors appreciate Secure IoT Platform Consortium, which is working on the SCOPE project together with Chuo University, for the fruitful discussions and advices.

References

1. Lecture 13: Asymmetric Encryption and Digital Signatures. `http://isis.poly.edu/courses/cs6903/Lectures/lecture13.pdf`, accessed: 2018-11-18
2. Abe, S., Kaseda, Y., Fujii, Y., Yoshida, T., Okada, K.: A Proposal of a Secure Cloud Storage using a Proxy Re-Encryption . In: The 2012 Symposium on Cryptography and Information Security (2012)
3. Akl, S.G., Taylor, P.D.: Cryptographic Solution to a Problem of Access Control in a Hierarchy. ACM Trans. Comput. Syst. 1(3), 239–248 (Aug 1983)
4. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. ACM Transactions on Information and System Security (TISSEC) 9(1), 1–30 (2006)
5. Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. pp. 259–274 (2000), `https://doi.org/10.1007/3-540-45539-6\_18`
6. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Advances in Cryptology –EUROCRYPT'98, pp. 127–144. Springer (1998)
7. Chung, Y.F., Lee, H.H., Lai, F., Chen, T.S.: Access control in user hierarchy based on elliptic curve cryptosystem. Information Sciences 178(1), 230–243 (2008)
8. Cloud Security Alliance: Top Threats to Cloud Computing V1.0. `https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf` (February 2010)

9. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Advances in Cryptology -CRYPTO'98. pp. 13–25. Springer (1998)
10. Gotaishi, M., Tsujii, S.: Information Processing Devices, Communication System, Information Processing Method and Program Japanese published patent applications 2015-110103 (May 2015)
11. Gotaishi, M., Tsujii, S.: Organizational Cryptosystem - Application of Elliptic-curve Cryptosystem to Dynamic Access Control- . In: The 2016 Symposium on Cryptography and Information Security (2016)
12. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: Applied Cryptography and Network Security. pp. 288–306. Springer (2007)
13. Hayashi, R., Matsushita, T., Yoshida, T., Fujii, Y., Okada, K.: Unforgeability of Re-Encryption Keys against Collusion Attack in Proxy Re-Encryption. In: Iwata, T., Nishigaki, M. (eds.) Advances in Information and Computer Security. pp. 210–229. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
14. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation 48(177), 203–209 (1987)
15. Kuo, W.C., Wuu, L.C.: On the Security of " A Novel Elliptic Curve Dynamic Access Control System ". International Journal of Security and Its Applications 3(2), 37–48 (2009)
16. Mambo, M., Okamoto, E.: Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. IEICE transactions on fundamentals of electronics, Communications and computer sciences 80(1), 54–63 (1997)
17. Microsoft: Encrypting File System overview. `http://technet.microsoft.com/en-us/library/cc759177` (January 2005)
18. Saisho, T., Kondo, T., Shouji, T., Gotaishi, M., Tsujii, S., et al.: New Cryptosystems for Social Organizations and Practical Use of It in Society —Promoting the Personal Data Utilization Securely. Journal of Information Processing Society 56(9), 1868–1876 (2015)
19. Tsujii, S., Gotaishi, M.: Proposal of Organization Cryptosystem based on STS-MPKC . In: The 2011 Symposium on Cryptography and Information Security 2A4-2 (2011)
20. Wang, L., Waseda, A., Nojima, R., Moriai, S.: Proxy Re-encryption with IND-CCA security in Encrypted file Storage System . In: The 2014 Symposium on Cryptography and Information Security (2014)