Journal of Ambient Intelligence and Humanized Computing (2018) 9:1267–1273
https://doi.org/10.1007/s12652-017-0602-5

ORIGINAL RESEARCH

IoT device security based on proxy re-encryption

SuHyun Kim[1] · ImYeong Lee[2]

Received: 13 March 2017 / Accepted: 14 October 2017 / Published online: 30 November 2017

Abstract
It appears that interest in the Internet of things (IoT) has recently reached its peak, with a great deal of focus from both the private and public sectors. IoT, a technology that enables the exchange of data through linkage among all objects surrounding the user, can create new services. Data communication among objects is not limited to personal information, but can also deliver different data types, such as sensing information collected from the surrounding environment. When such data is collected and used maliciously by an attacker, it is more vulnerable to threats than in conventional network environments. Security of all data transmitted in the IoT environment is therefore essential for preventing attacks. However, it is difficult to apply the conventional cipher algorithm to lightweight devices. Therefore, we propose a method for sharing and managing data using the conventional cipher algorithm on lightweight devices in various circumstances. This method implements proxy re-encryption in order to manage data with fewer encryptions, and provides a data sharing function to supplement the insufficient capacity of lightweight device networks.

Keywords Internet of things · Lightweight device · Proxy re-encryption · Data sharing

1 Introduction

The Internet of things (IoT) refers to a network in which information from all connected devices can be collected, processed, and modified to provide new services. The IoT can be used in various ways, and the data transmitted during network communication can take many forms, ranging from personal data to sensing information gathered from the environment. If an attacker were to collect such data and use it maliciously, greater security threats would naturally arise in the IoT than in the existing environment. Passive attacks, such as spam messages sent via refrigerators or smart TVs, may result in damage to these devices, and more aggressive attacks may even threaten the user's life; for example, by hacking vehicle communication systems or medical devices. Furthermore, user data collected on IoT platforms can result in privacy invasion; for example, electricity consumption pattern analysis using a smart meter could expose a person's lifestyle. Although users can enjoy the benefits of data being collected to provide customized services, some may not wish to expose their personal data to service providers. Considering the factors mentioned above, data transmission security in the IoT environment is vital for managing multiple forms of personal data and avoiding damage caused by security threats.

Nevertheless, it may not be possible to apply existing security systems to devices used in the IoT environment. This is because security solutions that implement the existing encryption algorithms are difficult to use in downsized, lightweight devices, and intrusion paths are continually becoming more diversified, with complicated network structures composed of countless nodes.

We therefore propose a method that enables data sharing and management using existing encryption algorithms, in an environment consisting of multiple lightweight devices. In this study, data is managed by decreasing the number of encryption and decryption counts for each device and using proxy re-encryption schemes. Proxy re-encryption furthermore provides a data-sharing function that allows the system to make optimal use of the limited data storage capacity of a network device.

✉ ImYeong Lee
imylee@sch.ac.kr

SuHyun Kim
kimsh@sch.ac.kr

[1] IoT Security and Privacy Research Center, Soonchunhyang University, Asan, South Korea

[2] Department of Computer Software Engineering, Soonchunhyang University, Asan, South Korea

Moreover, we propose a safe and effective data-sharing method, by applying attribute-based encryption and considering untrusted data storage.

2 Related works

Various encryption techniques have been studied for protecting user data stored on a cloud server from untrusted administrators or attackers. However, one disadvantage of existing data encryption technologies is that they cannot be easily applied to data-sharing services among many users over cloud storage. To address this problem, data can be managed by means of the most basic scheme, which is the encryption of stored data. However, existing simple encryption schemes exhibit problems with the access management of data stored in the cloud environment. That is, a large number of users may wish to access data in the cloud server simultaneously, or various functions may be required, including access control according to a user's rank. Existing public or symmetric key-based encryption schemes can neither solve the key management problem nor satisfy the access control requirements.

2.1 Apache Hadoop distributed file system

The Hadoop distributed file system (HDFS) is quite similar to existing distributed file systems, and made to be executable in existing hardware. However, it differs in various aspects, such as its effective fault recovery functions and the fact that it is designed to be applicable to low-priced hardware. The HDFS is most commonly utilized for cloud computing platforms of diverse IT companies, such as EC2 of Amazon and Yahoo.

The particulars derived for HDFS design and implementation are almost the same as those of the Google file system (GFS), although the HDFS is superior in that it uses Java and therefore exhibits excellent portability between diverse platforms. The use of the Java language, with its high portability, provides HDFS with the advantage that it can be driven in diverse servers that support Java.

2.2 Attribute-based encryption

To address the problems of existing encryption schemes, methods suitable for distributed storage servers have been actively studied in recent years. As an exemplary study, Sahai and Brent (2005) proposed attribute-based encryption (ABE), which is an extended form of the ID-based encryption(IBE) concept. ID-based encryption method was first proposed by Shamir (1984) in to solve the certificate problem in public-key encryption. ABE is based on the ID-based encryption method, and creates a public key by using attributes rather than user IDs. User attributes can be composed of multiple aspects; for example, decryption of data may only be possible when the attribute values computer science department and professor exist, for the affiliate and position attributes, respectively.

ABE is a technology that provides control mechanisms for access to encrypted data, based on decrypting keys and attributes assigned to encrypted data or access policies. Types of ABE include ciphertext policy attribute-based encryption (CP-ABE or ciphertext-policy ABE), which determines access structures when documents are encrypted, and key policy attribute-based encryption (KP-ABE or key-policy ABE), which determines access structures when user keys are created.

2.3 Previous studies on data sharing in the cloud environment

Thatikayala et al. (2014) discussed the problem that may arise in cloud servers storing personal health records (PHRs). To counter this, attribute-based encryption was used in the process of encrypting and storing PHRs in the servers. In this scheme, multiple encrypted data owners were allowed to reference the same data value. Furthermore, efficient user attribute addition and deletion was supported. The system also implemented scalability and sharing of PHRs over cloud computing by means of attribute-based encryption. However, in this scheme, an encryption key was updated to remove a user's data-sharing right. The efficiency of the re-encryption process of encrypted data stored in the cloud may decrease as a result.

Liang et al. (2014) applied proxy re-encryption to ABE, based on ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE). The authors proposed a measure for efficiently reducing the number of unavoidable encryption processes when changing ciphertext attributes encrypted based on ABE. However, the drawback of this method was increased computation and the large number of encryption processes required during re-encryption of ciphertext using the re-encryption key.

Kumar et al. (2015) defined security requirements for outsourcing data, as well as describing considerations for data sharing in the cloud environment. Stored data should be protected from unauthorized access. There is a need to protect data and data access from system administrators (untrusted administrators). Access control mechanisms, and existing encryption and decryption schemes, are not sufficient for the above scenario. Existing simple encryption technologies are not sufficient when considering data sharing and collaboration. Appropriate key management is required to activate data sharing and collaboration in the cloud. Considerations for data sharing in the cloud environment. Data owners should be able to designate a data-sharing group. Members

of the group should be able to access data without intervention by the data owner. Users other than the data owner and group members, including cloud service providers, should not be able to access data. The data owner should be able to cancel the data access rights of group members (revocation of rights). The data owner should be able to add a member to the group. A group member should not be able to cancel the rights of other members or authorize new user additions.

Li et al. (2015) argued that although searchable encryption provides an efficient solution for supporting keyword-based searches, it may create a sharing key problem between authorized users during data sharing. Li et al. then proposed a single structure for safe data sharing, which performed keyword-based searches on encrypted data without sharing the secret key. It provided two-layered access control for unauthorized users' access to shared data; that is, only authorized users could perform keyword-based searches on encrypted documents.

Marimuthu et al. (2014) discussed group resource sharing problems among cloud users. Dynamic group member changes and data sharing among multiple users over untrusted cloud networks are significant challenges. In this paper, we propose a data-sharing scheme among multiple users in a dynamic group over the cloud. In this scheme, we use a group signature along with dynamic broadcasting encryption technology. The cost of storage method overheads and encryption calculation is independent of the number of revoked users. To solve this, a one-time password (OTP) is used in our research.

2.4 Proxy re-encryption

Proxy re-encryption schemes function by converting encrypted data that can be deciphered with the private key of user A so that it can also be deciphered with that of user B. The proxy does not decode the encrypted data from user A and encrypt it with user B's public key. Instead, it converts the ciphertexts encrypted with user A's public key into ciphertexts encrypted with that of user B. In this case, user A creates the re-encryption key and sends it to the proxy (2005) (Ateniese et al. 2006; Blaze et al. 1998; Hui and Sherratt 2017; Ivan and Dodis 2003; Keegan et al. 2016; Maity and Park 2016).

3 Proposed scheme

3.1 Data management based on proxy re-encryption

In this study, we propose an IoT network environment in which information is gathered and processed from dozens of sensor nodes. In this environment, each node shares its information with nodes in different locations via communication with the server, which acts as a gateway. Existing public key encryption schemes require the number of encryptions to correspond to the number of nodes (n). For example, if 50 sensor nodes are allocated, each node must be encrypted 50 times during data transmission. If the proxy re-encryption schemes are used as proposed, each node performs an encryption creating n re-encryption keys, and sends these to the proxy server. The server then generates ciphertexts that allow other nodes to decipher the texts, decreasing the encryption calculation burden for each node (Fig. 1).

The proposed method uses the following system coefficients:

- p: Prime number
- G: Additive group of order p
- q: Generator of G
- e: Bilinear mapping, $G \times G \rightarrow G_T$
- sk*: Private key of *
- pk*: Public key of *

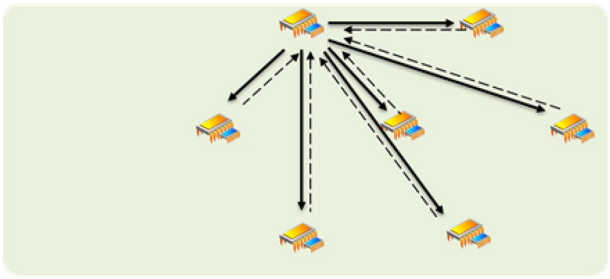
1. Key generation

Each sensor node generates and maintains its key pair as follows:

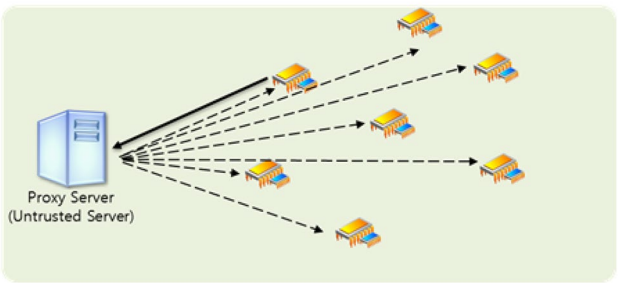
random $x \in Z_q$

$sk_a = x$

$pk_a = g^x (= g^{sk_a})$.



(a) Sharing and management of data in existing networks



(b) Proposed method

Fig. 1 Comparison of encrypted communication process of an individual node

2. Data encryption

Each node generates data to be transmitted as follows:

random $r \in Z_q$

$A = pk_a{}^r$

$B = e(g, g)^{sk_a \cdot r}$

$C = e(g, H(pk_a))^r \cdot m$

$E = (A, B, C)$.

3. Re-encryption key generation

Each node generates a re-encryption key in order to share its data with other nodes. In order for node **a** to share data with node **b**, a re-encryption key is generated using the private key of **a** and the public key of **b**, and sent to the proxy server with the encrypted text.

In the proposed scenario, each node creates re-encryption keys for all nodes other than itself:

$A' = pk_b{}^r$

$rk_{a \to b} = (A', pk_b{}^{-sk_a})$.

4. Re-encryption

The proxy server replaces A′ received from node **a** with A, and conducts re-encryption using the re-encryption key, target ciphertexts, and public key, to create re-encryption ciphertexts that can be deciphered by node **b** as follows:

$$\begin{aligned} B' &= e(A, rk_{a \to b}) \\ &= e(A, pk_b{}^{-sk_a}) = e(A, g^{sk_b/sk_a}) \\ &= e(g^{sk_a \cdot r}, g^{sk_b/sk_a}) = e(g, g)^{sk_b \cdot r} \end{aligned}$$

$E = (A', B', C)$.

5. Decryption

Node **b** deciphers the ciphertexts received from **a** using its private key, as follows:

$$\begin{aligned} m &= C / e(A', H_2(pk_a))^{-sk_b} \\ &= \frac{e(g, H_2(pk_a))^r \cdot m}{e(A', H_2(pk_a))^{-sk_b}} \\ &= \frac{e(g, H_2(pk_a))^r \cdot m}{e(pk_b{}^r, H_2(pk_a))^{-sk_b}} = \frac{e(g, H_2(pk_a))^r \cdot m}{e(g^{sk_b \cdot r}, H_2(pk_a))^{-sk_b}} \\ &= \frac{e(g, H_2(pk_a))^r \cdot m}{e(g, H_2(pk_a))^{sk_b \cdot r \cdot -sk_b}} = m. \end{aligned}$$

3.2 Data sharing scheme based on attribute re-encryption

The overall cloud computing environment used in this study was designed based on the Apache HDFS, and the basic concept of the proposed method is based on CP-ABE. A user who satisfies specific attributes in a group may obtain the rights to acquire a decryption key. In this scheme, a user may obtain final data decryption rights of data according to their rights rank. When user A wishes to share data with user B, if both users have the same attributes, they will have the same decryption rights. However, if their attributes differ, those of the other user must be revoked.

In the proposed data-sharing scheme, there is no need for the revocation of user attributes. User A transfers the encryption key to user B, who wishes to access the shared data, by re-encrypting the encryption key that can decrypt the encrypted data, using its own attributes based on the attribute public key of user B. Therefore, the proposed method eliminates unnecessary processes, thereby providing increased efficiency and safer data sharing.

- n: Number of participants
- P: Set of participants $P_i (1 \leqslant i \leqslant n)$ in a secret distribution
- q: Fraction
- k: Secret information $\in Z_q$
- K: Set of secret information k
- s_{P_i}: Pieces of secret $\in Z_q$
- S_{P_i}: Set of secret pieces s_{P_i} possessed by individual participantsP_i
- k: Security parameter
- i: Attribute value
- L: Attribute set
- $a_i, \hat{a}_i, a_i^* \in Z_P^*$: Correspond to attribute i
- $W = [W_1, W_2, \ldots, W_n]$: Access structure
- $r \in Z_p{}^*$: Random value
- M: Plaintext

1. Setup

Enter security parameter k to output the public key, PK, and master key, MK, corresponding to the value of the parameter.

1. $G = [p, G, G_T, g \in G, e]$
   Create random values$w \in Z_P^*$.
2. Select random values $a_i, \hat{a}_i, a_i^* \in Z_P^*$ corresponding to attribute $i, (1 \leqslant i \leqslant n)$.
3. Calculate $Y = e(g, g)^w$ and $A_i = g^{a_i}, \hat{A}_i = g^{\hat{a}_i}, A_i^* = g^{a_i^*}$.
4. PK is $< Y, p, G, G_T, g, r, (A_i, \hat{A}_i, A_i^*)_{1 \leqslant i \leqslant n} >$ and MK is $< w, (a_i, \hat{a}_i, \hat{a}_i)_{1 \leqslant i \leqslant n} >$.

2. KeyGen

This is an algorithm for entering MK and attribute set L in order to output the secret key SKL corresponding to the access structure.

1. Enter attribute set $L = \lfloor L_1, L_2, ..., L_n \rfloor$ to create a secret key.
2. Select $s_i \in Z_P{}^*$ randomly and calculate $s = \sum_{i=1}^{n} s_i$, $D_0 = g^{w-s}$.
3. If $L_i = 1$, calculate $[D_i, D_i{}^*] = [g^{\frac{s_i}{\hat{a}_i}}, g^{\frac{s_i}{a_i{}^*}}]$
4. The secret key is $SK_L =< D_0, (D_i, D_i{}^*)_{1 \leqslant i \leqslant n} >$.

3. Encrypt

This is an algorithm that outputs ciphertext CT corresponding to the plaintext, by entering PK and access structure W, as well as plaintext M.

1. Encrypt the access structure $W = [W_1, W_2, ..., W_n]$ and plaintext M.
2. Calculate a random value $r \in Z_p{}^*$, and $\tilde{C} = MY^r, C_0 = g^r$.
3. Calculate the ciphertext that satisfies the following conditions: $C_i : W_i = 1, C_i = A_i{}^r, W_i = 0, C_i = \hat{A}_i^r, W_i =*$, $C_i = A_i{}^{*r}$.
4. The ciphertext is $CT =< \tilde{C}, C_0, (C_i)_{1 \leqslant i \leqslant n} >$.
   $di = H(CT)$
   $A = pk^{di}$
5. $B = e(g, g)^{SK_L \cdot di}$
   $C = e(g, H(pk))^{di} \cdot m$
   $E_a = (A, B, C)$.
6. Store Ea in the cloud storage.

4. ReKey generation

User A transfers the re-encrypted encryption key that can decrypt data, using an attribute public key of user B, with whom data is shared.

$A' = pk_b^{di}$

$rk_{a \to b} = (A', pk_b^{-sk_L})$

$B' = e(A, rk_{a \to b})$

$E_b = (A', B', C)$.

5. Decrypt

This is an algorithm that outputs plaintext corresponding to the ciphertext by entering the recovered secret key SKL and ciphertext CT.

1. $CT = C / e(A', H(pk_a))^{sk_L}$.
2. Decrypt using ciphertext $CT =< \tilde{C}, C_0, (C_i)_{1 \leqslant i \leqslant n} >$ and secret key $SK_L =< D_0, (D_i, D_i{}^*)_{1 \leqslant i \leqslant n} >$.

$$For \quad 1 \leqslant i \leqslant n$$

$$D'_i = \left\{ \begin{array}{ll} D_i & if\ W_i \neq * \\ D_i{}^* & if\ W_i = * \end{array} \right\}$$

$$\frac{\tilde{C}}{e(C_0, D_0) \prod_{i=1}^{n} e(C_i, D'_i)} = \frac{M(e(g,g)^w)^r}{e(g^r, g^{w-s}) \prod_{i=1}^{n} e((g^{a_i})^r, g^{\frac{s_i}{a_i}})}$$

$$= \frac{M(e(g,g)^{wr})}{e(g^r, g^{w-s}) \cdot e(g,g)^{s \cdot r}} = \frac{M(e(g,g)^{wr})}{e(g,g)^{w \cdot r}} = M.$$

4 Analysis of proposed method

4.1 Data management based on proxy re-encryption

4.1.1 Communication traffic

Table 1 provides a comparison of node-to-node communication count and duration in the existing network and the proposed method (Liu et al. 2014). Unlike in the existing scheme, where calculation time increases rapidly with the number of nodes, the proposed method figures display

Table 1 Calculation time according to number of devices

Node count	Comm. count (existing method)	Duration (s)	Comm. count (proposed method)	Duration (s)
2	2	0.22	2	0.22
3	6	0.66	3	0.33
4	12	1.32	4	0.44
5	20	2.2	5	0.55
–	–	–	–	–
50	2450	269.5	50	5.5

linear growth as the node count rises. That is, when the count exceeds 100, the duration increases to more than 11 s. It will therefore be necessary to consider a long-term time frame in future.

4.1.2 Confidentiality

With the proposed method, it is difficult to infer communication contents, even when a malicious third party wiretaps node-to-node communication by means of pairings.

4.1.3 Data sharing efficiency

We use a proxy server that does not need to check reliability, so that secure and efficient data sharing can be carried out between nodes. Moreover, the limited data storage capacity problem of lightweight devices, such as Atmega128, can be solved through efficient data sharing.

4.2 Data sharing scheme based on attribute re-encryption

4.2.1 Security

The proposed method employs pairing during data sharing to prevent malicious third-party users from understanding the communication contents, even if wiretapping is carried out between the client and server. In addition, the $rk_{a\to b} = (A', pk_b{}^{-sk_L})$ re-encryption key generated during re-encryption of user A's encryption key is a one-time key used during data sharing, so that user B, who shares the data, cannot use the key continuously. Thus, this method can also provide backward secrecy.

4.2.2 Computation amount

The proposed system can provide efficient computation during data sharing, by performing lightweight pairing computation over the cloud storage server, which cannot be trusted, through re-encryption. Furthermore, the proposed method is more efficient than existing methods in that it can transfer the encryption key by re-encrypting only the key, as opposed to re-encrypting the data itself or transferring all the data (Table 2). As shown in the Table 1, in which the computation load during data sharing is compared, the proposed method is more efficient than existing methods in terms of the increase in data-sharing delay time (Fig. 2) during computation. Furthermore, the data computation load according to the number of shared members is shown be highly efficient during data sharing among groups.

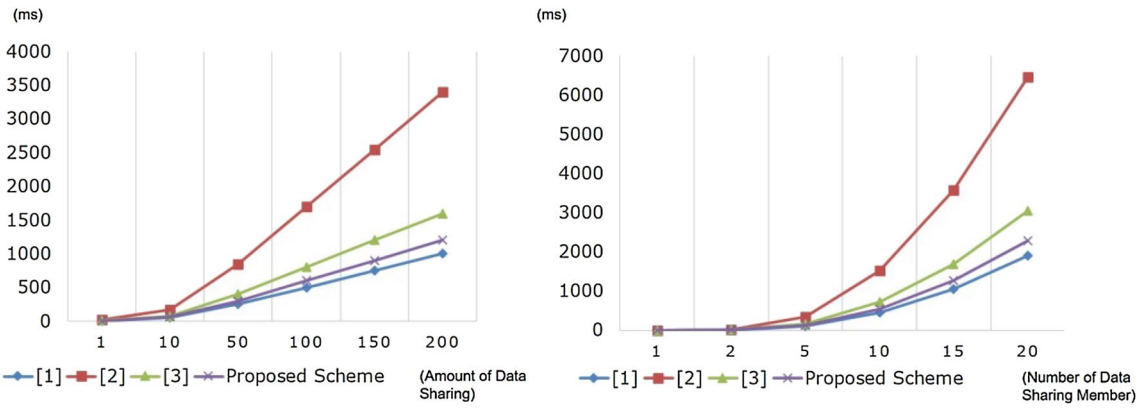


Fig. 2 Comparison of sharing time

Table 2 Comparison of proposed scheme

	Public information	Illegal user tracking	Data sharing computation	Amount of Sharing
Li et al. (2015)	$(X, T_{X,} g_X)$, f(K)	X	m + h + a	O(n(n-1)/2)
Marimuthu et al. (2014)	$H^{[h]} = (R_X{}^{[h]}, T_X{}^{[h]})$	O	{(m3 + m) + 3h + 2a}	O(n(n-1)/2)
Kar et al. (2011)	$(X, T_{X,} g_X)$, f(K), g(K)	O	{2m + h + a}	O(n(n-1)/2)
Proposed scheme	(n. e. P, PK, g_X)	O	m + p	O(n)

m modular exponentiation, *h* hash function, *a* addition operation, *p* pairing operation, *n* number of data-sharing group users

4.2.3 Forward and backward secrecy

There is flexible subscription and unsubscription of users in data sharing among groups. Subscribed group members should not know the secret group key used previously, and unsubscribed members should not know the new secret group key. The proposed method is based on the group signature, and therefore provides safety.

5 Conclusion and future research

This study proposes a method for improved security and efficiency in data sharing and management among multiple nodes in an IoT environment. Compared to existing environments, the proposed method provides more efficient communication settings. Furthermore, our method offers enhanced security features by using elliptic curve cryptosystem-based proxy re-encryption schemes in lightweight devices.

There is a need to study methods for resolving the unidirectional and non-transferable issues of proxy re-encryption schemes. Another problem to be addressed is the fact that the operation time increases in proportion to the number of sensor nodes, although it is more efficient than previous approaches.

By enabling detailed user access control in cloud environments, sensitive information stored on cloud servers can be managed more safely. The proposed protocol provides a structure by means of which a large capacity of various data, including users' personal information requiring high confidentiality, can be accessed safely and efficiently. We expect the proposed protocol to be widely and efficiently used in the cloud computing environment. However, a disadvantage of this method is the additional computation in the polynomial equation compared to existing attribute-based encryption methods, since it provides more functions. In the future, we will study more efficient and safer methods based on the proposed method.

Acknowledgements This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2015-0-00403) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

Ateniese G et al (2006) Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans Inf Syst Secur (TISSEC) 9.1:1–30. doi:10.1145/1127345.1127346

Blaze M, Bleumer G, Strauss M (1998) Divertible protocols and atomic proxy cryptography. Adv Cryptol EUROCRYPT'98. doi:10.1007/BFb0054122

Hui TKL, Sherratt RS (2017) Towards disappearing user interfaces for ubiquitous computing: human enhancement from sixth sense to super senses. J Ambient Intell Hum Comput. doi:10.1007/s12652-016-0409-9

Ivan A-A, Dodis Y (2003) Proxy Cryptography Revisited. NDSS. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.232.2463. Accessed 27 Sept 2017

Kar J, Majhi B (2011) A novel deniable authentication protocol based on Diffie–Hellman algorithm using pairing technique. In: Proceedings of the 2011 international conference on communication, computing and security. ACM, 2011. doi:10.1145/1947940.1948042

Keegan N et al (2016) A survey of cloud-based network intrusion detection analysis. Hum Centric Comput Inf Sci 6.1:19. https://hcis-journal.springeropen.com/articles/10.1186/s13673-016-0076-z. Accessed 27 Sept 2017

Kumar SN (2015) Cryptography during data sharing and accessing over cloud. Int Trans Electr Comput Eng Syst 3.1:12–18. http://pubs.sciepub.com/iteces/3/1/2/. Accessed 27 Sept 2017

Kwon T et al (2015) Efficiency of LEA compared with AES. JoC 6.3:16–25. http://www.earticle.net/Article.aspx?sn=258798. Accessed 27 Sept 2017

Li J et al (2015) Enabling efficient and secure data sharing in cloud computing. Concurr Comput Pract Exp 26.5:1052–1066. doi:10.1002/cpe.3067

Liang K et al (2014) An adaptively CCA-secure ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. ISPEC. doi:10.1007/978-3-319-06320-1_33

Liu Z et al (2014) Reverse product-scanning multiplication and squaring on 8-bit AVR processors. Int Conf Inf Commun Secur. https://link.springer.com/chapter/10.1007/978-3-319-21966-0_12. Accessed 27 Sept 2017 **(Springer International Publishing)**

Maity S, Park JH (2016) Powering IoT devices: a novel design and analysis technique. J Converg 7:1–18. http://www.manuscriptlink.com/journals/joc/digitalLibrary/2016/7/0/3571. Accessed 27 Sept 2017

Marimuthu K et al (2014) Scalable and secure data sharing for dynamic groups in cloud. In: Advanced communication control and computing technologies (ICACCCT), 2014 international conference on. IEEE, pp 1697–1701. doi:10.1109/ICACCCT.2014.7019398

Sahai A, Brent W (2005) Fuzzy identity-based encryption. In: Annual international conference on the theory and applications of cryptographic techniques, EUROCRYPT 2005: Advances in Cryptology – EUROCRYPT 2005, pp 457–473. https://link.springer.com/book/10.1007/b136415#page=470. Accessed 27 Sept 2017

Shamir A (1984) Identity-based cryptosystems and signature schemes. Crypto 84. https://link.springer.com/book/10.1007/3-540-39568-7#page=53. Accessed 27 Sept 2017

Thatikayala S, Sandhyarani J, Sravanthi J (2014) Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Trans Parallel Distrib Syst 3:4912–4917. http://ijsetr.com/issue.php?issue=ISSUE%2024&volume=Volume3&page=4. Accessed 27 Sept 2017