*Manuscript (including abstract)
Click here to view linked References

PRTA: a Proxy Re-encryption based Trusted Authorization Scheme for Nodes on CloudIoT

Mang Su[a], Bo Zhou[b], Anmin Fu[a], Yan Yu[a], Gongxuan Zhang[a]

[a] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Jiangsu, Nanjing.*
[b] *Department of Computer Science, Liverpool John Moores University, Byrom Street, Liverpool, UK, L3 3AF.*

Abstract

In CloudIoT platform, the data is collected and shared by different nodes of Internet of Things(IoT), and data is processed and stored based on cloud servers. It has increased the abilities of IoT on information computation. Meanwhile, it also has enriched the resource in cloud and improved integration of the Internet and human world. All of this offer advantages as well as the new challenges of information security and privacy protection. As the energy limitation of the nodes in IoT, they are particularly vulnerable. It is much easier to hijack the nodes than to attack the data center for hackers. Thus, it is a crucial and urgent issue to realize the trusted update of authorization of nodes. When some nodes are hijacked, both of the behaviors to upload data to servers and to download information from servers should be forbidden. Otherwise, it might cause the serious damage to the sensitive data and privacy of servers. In order to solve this problem, we proposed a Proxy Re-encryption based Trusted Authorization scheme for nodes on CloudIoT(PRTA). PRTA is based on the proxy re-encryption (PRE), and the cloud server will play the roles of data storing and re-encrypting, which would reach the full potential of cloud computing and reduce the cost of nodes. The node's status is taken as one of the parameters for data re-encryption and it is under the authorization servers' control, which

Email addresses: `sumang@njust.edu.cn` (Mang Su), `B.zhou@ljmu.ac.uk` (Bo Zhou), `fuam@njust.edu.cn` (Anmin Fu), `yuyan@njust.edu.cn` (Yan Yu), `zhanggongxuan@njust.edu.cn` (Gongxuan Zhang)

could ensure the security and reliability of the data and be beneficial for the privacy protection in CloudIoT. Also, the authorization servers are divided into the downloading and uploading kinds, which will make the application range much wider.
Keywords:
CloudIoT; Proxy re-encryption(PRE); Trusted update of authorization; Data downloading; Data uploading; Privacy protection.

1. Introduction

Internet of Things (IoT) has been proposed by the International Telecommunication Union (ITU) in 2005. Recent advances in sensing technologies and smart chips have promoted the progress of IoT. Based on various sensors and devices, IoT could collect the information of different things communicating with Internet. The communications by Internet are changing from computers to computers to Man-to-Machine or Machine-to-Machine(M2M). In a word, IoT integrates various sensors, objects and smart nodes that are capable of communicating with each other without human intervention[5]. The development of IoT has been blurring the boundaries among the physical, social, and cyber worlds and fueling the astonishing number of Internet-connected devices, which has been increasing from 15 billion in 2014 to 17.6 billion in 2016 and will be 30 billion by 2020[9][1]. In recent years, a variety of applications based on IoT with different areas have been developed, such as logistics, manufacturing, healthcare, industrial surveillance, and etc[33][25].

Meanwhile, a number of corresponding techniques, such as intelligent sensors, wireless networks, big data analysis and mining[40], have been developed to realize the potential of the IoT with different intelligent systems[6][30]. Cloud computing is one of them. The cloud provides flexible, scalable and customized computing service and storage service with lower entry barriers and less cost. More and more users choose cloud to obtain the resource, such as information, software, hardware and platform. In general, the framework of the IoT is consisted of three layers, including the perceptual layer, the transport layer and the intelligent application layer [35]. The perceptual layer is based on various sensors and is responsible for data collection. The transferring layer is based on the current common protocols, such as IP, and is responsible for data transmission. The intelligent application layer is designed for different users' requirements and is responsible for data processing of the layers above. Cloud is suitable for the third layer of IoT for its massive computing and storing capacity. Thus, a novel paradigm where cloud and IoT merged together is proposed, which is called CloudIoT[3]. IoT could benefit from the virtually unlimited capabilities and resources of cloud to compensate its technological constraints (e.g., storage, processing, communication). CloudIoT has given birth to a new set of smart services and applications, which can strongly impact human's daily life. Many applications are beneficial from the M2M communications when things need to exchange information among themselves and not only send them to cloud. From 2008, the number of papers dealing with cloud and IoT shows an increasing tendency. The characteristics of cloud and IoT are often complementary, which is the main reason why many researchers have proposed and are proposing their integration, generally to obtain benefits in specific application scenarios [4][2][8]. Meanwhile, many Internet application vendors,such as Microsoft, IBM, Google, Alibaba and Tencent have developed the cloud platforms which could support the IoT applications. They provide the application programming interfaces (API) for the nodes definitions, simulations and configurations.

The emerging CloudIoT is foreseen as one of the great developments of IoT and cloud, as the users could obtain the convenience of both cloud and IoT. However, the new problems are also brought to the security of CloudIoT. Firstly, the nodes of IoT are numerous, so the data uploaded to cloud and shared by cloud will be increased sharply. The amount of private and confidential data will become more and more as well. For instance, the cameras for smart homes could collect and upload the real records of their owner's daily life, which will concern the privacy of users. Thus, it is important to protect the confidentiality and privacy of the data from such nodes. The security scheme is designed not only to prevent the access by illegal users, but also to avoid the analysis of cloud service providers. Only the authenticated users have privileges to obtain the information,and unauthorized accesses are prevented from tampering the data. Secondly, cyber attacks are becoming more pervasive[39][41]. As the limitation of computation ability, the nodes are weaker in resistant to attack than the common computers. The hackers could attack the CloudIoT by hijacking a node or faking a device[26],and they could obtain information in cloud servers or upload the malicious data to a server by attacking the nodes. Thus, how to revoke the authentication of nodes when they are hacked is a serious problem to be solved.

For the problems above, researchers have done a plenty of work, such as privacy protection[7][38], integrity verification[43], access control, secure storage of data in IoT environment. In order to ensure the data is accessed by the authenticated users, many efforts have been taking place to apply traditional methods of access control to IoT scenarios [29][19]. And there are some new approaches to access control mechanisms in IoT at the same time by describing the parameters of devices, e.g. device ID[32]or combing with some famous security protocols, e.g. Kerberos and RADIUS[24]. Due to the limitation of IoT sensors, some lightweight schemes also have been proposed [36] . As same as the common cloud service, the CloudIoT also requires the data encryption, thus, the cryptography based access control will be needed, e.g. Diffie-Hellman[22] or ECC [20][10]. All the works above contributed a great deal to the data protection of IoT, but they did not discuss the corresponding access control scheme for CloudIoT or how to deal with the data authentication when the nodes are hacked. Although, some of them have talked about the lightweight, but they did not try to take advantage of cloud service. PRE has played an important role in cloud access control and data protection. The proxy server could finish some work of data sharing. For the characteristics of PRE, it also could be applied to CloudIoT. If the cloud server is responsible for the work of re-encryption proxy, the computing cost of individual users and nodes will be much less.

Summing up, the current references have discussed a lot about how to keep the sensitive data security, but it is still a serious problem that how to update the authorization of the hacked nodes to prevent their downloading and uploading information. It means revoking the compromised nodes from the system assuredly is still one of the hottest topics for IoT security. And the main technical challenges are as followed:

(1)The applications of the IoT is various. Some nodes will collect and upload the information to server, some nodes will download the data for configuration and some will both upload and download the data. For example, some nodes for the smart healthcare solution will collect data of patient physical conditions. They focus on the data uploading quickly and correctly. Some nodes for the smart cars will download the information for navigation or speed control. They require to download the information conveniently. And for the smart homes, some nodes will both download and upload the information,such as intelligent entrance guard. Thus the scheme for the nodes revoking should consider the different applications.

(2)The cloud servers usually play the role for data storing, and they are used less in data encryption or decryption. For data security, the IoT servers and nodes will finish the work of encryption and decryption. It will be a great cost for nodes and IoT servers. Also the cloud servers are not fully used.

(3)When the nodes are compromised, some current schemes could update the key for such node. However if the node has already stored the old key, it will be a threat to the system.

Thus, we have done some research on this issue and its corresponding technologies and proposed a PRE based Trusted Authorization Scheme for Nodes on CloudIoT platform (PRTA). Firstly, we analyze the related work and proposed the system model. Secondly, we explain the system processes and algorithms based PRE. Finally, we discuss the properties including the security and efficiency issues. The main contributions of the paper are threefold:

(1) We defined the processes of data downloading and data uploading for nodes, and the permissions are designed for each process respectively. The

downloading permission is managed by the downloading authentication server and uploading permission is managed by the uploading authentication server, which will be more suitable for the various IoT applications. Some applications only need the node to collect data, some only need the nodes to share information of data server and some need both downloading and uploading. The users could be free to deploy the CloudIoT with downloading authentication server or uploading authentication server or both of them for different requirements. This is for the challenge (1).

(2) We designed the algorithms based on PRE, and the permissions assignment according to the re-encryption keys. The cloud server will be responsible for data re-encryption. The IoT data server and nodes will cost less for data accessing and collection. This is for the challenge (2).

(3)It is worth mentioning that there are two kinds of re-encryption algorithms, one is for downloading($ReEnc_1$) and the other is for uploading($ReEnc_2$). $ReEnc_1$ will generate the ciphertext for nodes from the IoT data server, and $ReEnc_2$ will generate the ciphertext for the IoT data server from the nodes. $ReEnc_1$ /$ReEnc_2$ have divided the parameters of re-encryption keys generation, one part is submitted to cloud servers, the other is under the control of downloading or uploading authentication server. When updating the authorization, downloading or uploading authentication server delete that part, the re-encryption keys will not be able to generated for parameters missing. The authorization is updated assuredly.This is for the challenge (3).

Organizations: The rest of this paper is organized as follows: The related works and preliminaries is in section 2, and the system models, main processes and algorithm of it explained in Section 3. Security proof is presented in Section 4, and Properties and efficiency analysis are in Section 5, The concluding remarks are in Section 6.

2. Related works and Preliminaries

2.1. Access control and authentication scheme for CloudIoT

There are a plenty of schemes designed for IoT, some of them are based on the tractional access control models, such as role based access control(RBAC) and attribute based access control(ABAC). Paper [29]is one of them, which focused on the dynamic characteristics of IoT and proposed an access control model based on attribute and role to solve the scenarios of large scale dynamics users. The model has put forward a policy language of attribute rules and a method to solve the policy conflict and redundancy. In IoT, different people visit the nodes of things to obtain the service, the interactions between the people and nodes occur very frequently. Thus, the relation between node and user is one of the vital factors for access control for IoT. But the schemes based on tractional models are lack of discussion on this issue. Thuan et al. [32]proposed a user centric identity management system that incorporates user's identity, device's identity and the relations between them. The proposed system is user centric and allows device authentication and authorization based on the user's identity. In order to make the scheme suitable for the fine-grained managing requirements, Pereira et al. [24] gave a CoAP-based framework for service-level access control on low-power devices. The framework allows fine-grained access control on a per service and method basis. For example, by using this approach a device can allow reading and writing accesses to its services in one group of users while only allow reading access in another group. Users without the right credentials are not even allowed to discover available services.

The works mentioned above have improved the access control schemes for IoT, but they did not discuss the corresponding algorithms for ciphertext protection, eg. how to encrypt/decrypt the data for the nodes or how to authenticate the nodes and servers. Thus, some researchers proposed the schemes based on cryptography for access control of IoT. Mahalle et al.[22] proposed an authentication scheme based on the Diffie-Hellman algorithm for the secret key generation for IoT along with protocol evaluation. Liu et al.[20] proposed authentication mechanism by using simple and secure key establishment method based on ECC. Since IoT nodes are not as powerful as the common computers, they are limited by the energy and not able to deal with large amounts of computational work.Some researchers have paid attentions to the lightweight schemes, Yang et al.[36]proposed a lightweight break-glass access control (LiBAC) system, which simultaneously supports two types of access control patterns: attribute based access control for normal circumstance and break-glass access for emergency circumstance. LiBAC is lightweight since very few calculations are executed by devices in the healthcare IoT network, and the storage and transmission overheads are low.

The schemes above are designed for the application layer. For transferring layer, there also exist some works, which focus on the components security of IoT. As one of the enabling components of IoT, wireless sensor networks (WSNs) have found applications in a wide range of fields, in which outside users could interact with sensors to obtain sensed data directly . However, WSNs are vulnerable to various attacks over wireless links, such as eavesdropping and tampering. Jiang et al.[10]has put forward a privacy aware two-factor authentication protocol based on elliptic curve cryptography for WSNs.This work faced to the challenge of how to ensure that sensitive or critical information is only available to legal users and proposed the two-factor authentication protocol. Paper [23]is also based on the WSN. The work has paid more attentions to the WSNs for military sensing and tracking, target tracking and environment monitoring. It is an important task to design an access control scheme that can authorize, authenticate and revoke a user to access the WSN. The work proposed a heterogeneous signcryption scheme to control the access behavior of the users. An important characteristic of this scheme is to allow a user in a certificateless cryptography (CLC) environment to send a message to a sensor node in an identity-based cryptography (IBC) environment. And Luo et al. [21]proposed a more secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the IoT, which allows an Internet user in a CLC environment to communicate with a sensor node in an IBC environment with different system parameters.

With the development of the cloud and IoT, more and more applications are designed on the combination of them(CloudIoT). The current works mentioned above could be build a base in both theory and practice for further research on access control of CloudIoT. Some researchers tried to design the schemes for secure data-sharing at the edge of cloud connected IoT smart devices[12],which has utilized both secret key encryption and public key encryption. In this scheme, all security operations are offloaded to nearby edge servers, thereby, greatly reducing the processing burden of smart devices.

2.2. Proxy Re-encryption(PRE)

PRE is based on the public-key system, and similar to the common public-key system, PRE system will provide a key pair(public/private key). In a PRE system, user A encrypts the message with his public key(pk_A) and generates the ciphertext(C_A) at first. Then A submits C_A to the PRE server. The PRE server will re-encrypt C_A with the re-encrypted key($rk_{A\to B}$) and generate the re-encrypted ciphertext($C_{A\to B}$) for user B. By PRE, A only needs to finish the first encryption of the message for sharing, and the PRE server will be in charge of other work of re-encryption. The PRE server can only obtain the ciphertext, thus the message is security. For the process above, PRE is beneficial for information sharing in cloud, which can reduce the requirement for personal users.

PRE cannot be applied in cloud independently. It needs to be combined with other cryptographic technologies. Identity based encryption(IBE) is one of those technologies, which constructs the keys based on the users' identities. Xu et al.[34] proposed a conditional identity-based broadcast PRE (CIBPRE) and formalized its semantic security, which is the PRE scheme combined with IBE. CIBPRE allows a sender to encrypt a message to multiple receivers by specifying these receivers' identities, and the sender can delegate a re-encryption key to a proxy so that he can convert the initial ciphertext into a new one for a new set of intended receivers. Moreover, the re-encryption key can be associated with a condition such that only the matching ciphertexts can be re-encrypted, which allows the original sender to enforce access control over his remote ciphertext in a fine-grained manner.

For the complicated environment, such as cloud, identity is not enough for authorization. The user's role, location or other information might be the factors for authorization, thus attribute based encryption(ABE) was proposed based on IBE,which is more flexible than IBE and widely used in cloud, for example, the flexible and fine-grained attribute-based data storage in cloud computing [16]and full verifiability for outsourced decryption in attribute based encryption[13][14]. There are two main forms to realize the ABE, the one is key-policy attribute-based encryption(KP-ABE)[17],the other one is ciphertext-policy attribute-based encryption(CP-ABE). CP-ABE is more suitable for the fine-grained access control and permission assignment in cloud. In order to combine the PRE with CP-ABE, Zhang et al.[42] tackled the aforementioned challenge for the first time by formalizing the notion of anonymous CP-ABPRE and giving out a concrete construction. The work proposed a novel technique called match-then-re-encrypt, in which a matching phase is additionally introduced before the re-encryption phase. This technique uses special components of the proxy re-encryption key and ciphertext to anonymously check whether the proxy can fulfill a proxy re-encryption or not.

Certificate are used to describe the attribute for authorization, which is more convenient and secure. Li et al. [18]proposed the formal definition and security model of certificate-based conditional proxy re-encryption. Further, the work combined the conditional proxy re-encryption with a certificate-based encryption and presented a certificate-based conditional proxy re-encryption scheme.

Yang et al. [37] presented a ciphertext-policy attribute based conditional proxy re-encryption (CPRE) scheme, together with a formalization of the primitive and its security proof. Su et al.[28] gave the PRE scheme based access control conditions, by paying more attentions to the generation of the re-encrypted key based on conditions.

For the requirement of fine-grained ciphertext management, Tang et al. [31] defined the ciphertext into different types, and gave the PRE scheme based on type.

In order to apply the PRE in IoT, Kim[11]gave the PRE scheme for IoT nodes to manage the data. The work designed the PRE server for data uploading.

All the researches above have improved the PRE schemes, but very few of them were targeting IoT environment. We hope to use PRE in IoT in order to reduce the cost of the node for encrypting and decrypting.

Meanwhile, it is an important problem to revoke the permission for the current cryptographic schemes used in cloud and IoT. But it is also a difficult problem to revoke some users' permissions without affecting the other users. For example, every attribute in ABE may be shared by multiple users and each user holds multiple attributes, any single-attribute revocation for someone might affect the other users with the same attribute in the system. Some researchers have tried to solve this problem, Li et al.[15]present a user collusion avoidance ciphertext-policy ABE scheme with efficient attribute revocation for the cloud storage system. The problem of attribute revocation is solved efficiently by exploiting the concept of an attribute group.

The permission revocation is also a serious problem for PRE, unfortunately, the corresponding schemes are not enough. In the earlier work[27], we proposed a PRE based trusted update scheme of authorization for nodes on IoT cloud platform (PRE-TUAN). It could realize trusted update of authorization for the nodes in the updating application scenarios preliminarily, but the downloading scenarios have not been discussed and the details have not been presented. Therefore, we will try to propose the scheme improved from [27] and design the scheme for both the updating and downloading scenarios for CloudIoT.

2.3. Bilinear Groups and hardness assumption

Bilinear Groups: G_1 ,G_2 and G_3 are three multiplicative cyclic groups of prime order p, and g is a generator of G_1 . $e : G_1 \times G_2 \rightarrow G_3$ is a computable bilinear map with the following properties:

(1)Bilinear: For all $a, b \in Z_p^*$, if $g \in G_1$ and $h \in G_2$we have $e(g^a, h^b) = e(g, h)^{ab}$.

(2)Non-degenerate: if $g \in G_1$ and $h \in G_2$, $e(g, h) \neq 1$.

(3)Computability:For all $g \in G_1$ and $h \in G_2$, we have an algorithm,which can obtain $e(g, h)$ in polynomial time.

DBDH(Decisional bilinear Diffie-Hellman) Problem: Given $\langle g, g^a, g^b, g^c, e(g, g)^{abc} \rangle$ for some $a, b, c \in Z_p^*$,$z = abc \ mod \ p$, a polynomial- time algorithm A has advantage ε in solving the DBDH problem, if and only if $|Pr[A(g, g^a, g^b, g^c, e(g, g)^{abc}) = 0] - Pr[A(g, g^a, g^b, g^c, e(g, g)^z) = 0]| \leq \varepsilon$.

3. PRTA Scheme

3.1. Goals and Preconditions

Our scheme will face to the application scenario in Fig. 1. The first part of Fig. 1 shows the common framework for the IoT. The nodes will collect and upload the data to IoT data server. Also the nodes might download the data from the data server, including data or configure commands. In order to increase the computing and storing abilities of IoT, the cloud server is applied in the IoT, and CloudIoT appeared. For the first step of our research, we will deploy a proxy server for re-encryption in cloud. And then, we will realize the trusted authentication updating when the nodes hacked at the second step.

In our paper, the PRTA will aim at the following goals:

(1) To realize the authorization of trusted update, including the data downloading and uploading. If a node is hacked, the IoT authorization servers could prevent the downloading behavior and refuse the data collected by that node assuredly.

(2) Our scheme could be realized in the existing IoT system. That means the framework of the current system will not be changed. Our scheme is still based on the 3-layer model of IoT and is designed for the application layer.

(3) The cloud computing will play a larger role in the CloudIoT. We will make the cloud server to finish more work of re-encryption, instead of individual

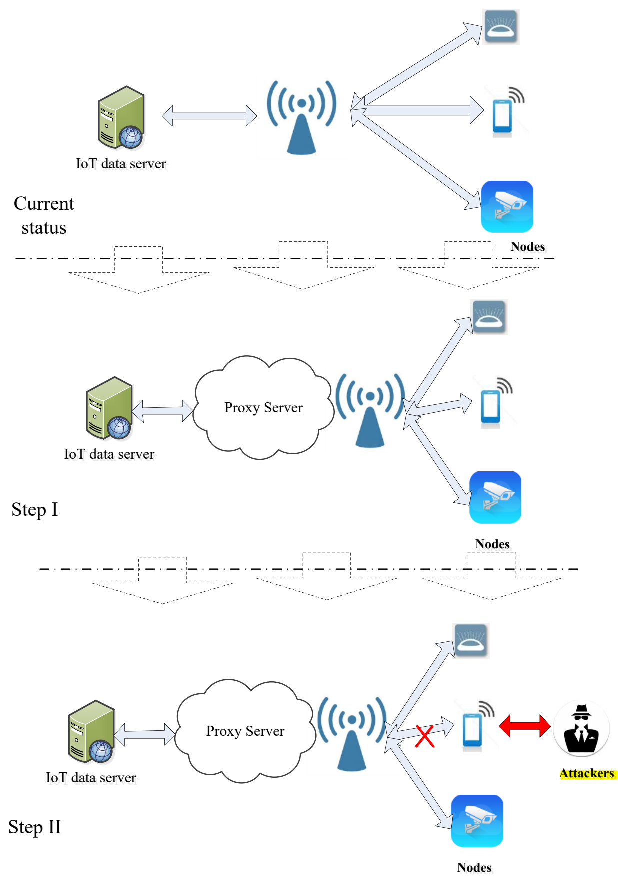

Figure 1: application scenario of PRTA.

user or nodes.

(4) PRTA is designed to resist the attacks such as cryptographic analysis, database injection, and the nodes hijacking. If the node has been hijacked already, our scheme will stop its work and revoke its permission.

PRTA system will be constructed under the following preconditions.

(1)All the nodes will connect to IoT data server by the network (WSN or others). They can communicate with the cloud servers to download and upload the information.

(2)IoT data server and authorization servers are trusted parties.

(3)The cloud servers are semi-trusted. They will finish the work honestly, but be curious about the privacy.

(4)The nodes will decrypt the ciphertext based on the parameters from the authorization servers and will delete the parameters after decryption.

3.2. Conceptions

The notations of system model (see Fig.2) are defined as follows:

(1)(pk_i,sk_i): the key-pairs of node i or server i.pk_i is the public key and sk_i is the secret key.

(2)$rk_{i \to j}$: the re-encryption key from entity i to j. If i is the node and j is IoT data server, the $rk_{i \to j}$ is for data uploading. If i is the IoT data server and j is the node, $rk_{i \to j}$ is for data downloading.

(3)Φ:the parameters list for data downloading, and it is managed by the downloading authentication server.

(4)(α_i, β_i):the items of list Φ, here i is the ID of node, and $i = 1...n$, n is the number of the nodes. α_i is also the parameter for the node to decrypt the re-encrypted ciphertext.

(5)Ψ:the parameters list for data uploading, and it is managed by the uploading authentication server.

(6)(φ_i, η_i):the items of list Ψ, here i is the ID of node, and $i = 1...n$, n is the number of the nodes. φ_i is also the parameter for the node to encrypt the collected data and is generated based on the status of the node.

(7) $(M)_K$:the ciphertext generated by the symmetric method based on the symmetric key K.

(8)$E(K)_{pk_i}$:the ciphertext generated by the public method based on public key pk_i of i, which could be decrypted by sk_i.

3.3. System Entities

There are seven kinds of entities in the system model of PRTA.

(1) IoT data server(IoT-DS): there are two functions of IoT-DS, the first one is to store the data collected by nodes, the second one is to generate the data or configuration commands for nodes.

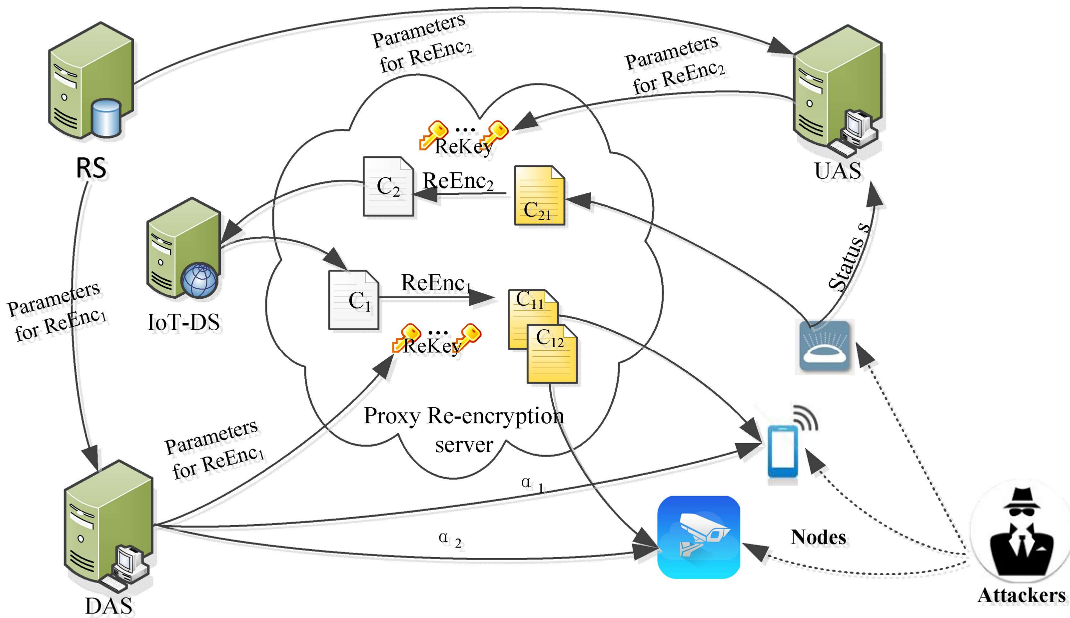

Figure 2: System model of PRTA.

(2) Register server(RS):the server for the node register, key generation, and re-encryption parameter generation.

(3) Downloading authorization server(DAS): the server to manage the authorization of data downloading.

(4) Uploading authorization server(UAS): the server to manage the authorization of data uploading.

(5) Nodes: the IoT devices, such as cameras, sensors for physical phenomena detection and so on.

(6) Proxy Re-Encryption server(PRES):the server for re-encryption.

(7) Attackers: they might attack the system and database by cryptographic analysis, brute-force password-cracking, and hijack attacking.

3.4. System Processes

There are five main processes of the PRTA.

(1)System setup and node registration

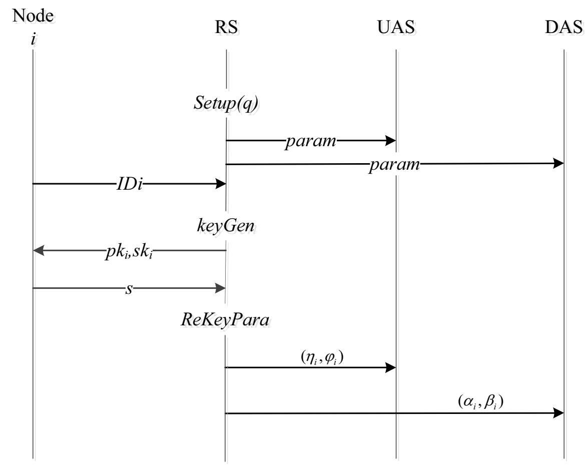

Figure 3: Working principles for system setup and node registration of PRTA.

The working principles for system setup and node registration of PRTA is shown in Fig3. The detail will be shown as follows.

Step1: RS selects a large security parameter q, and invokes the algorithm $Setup(q)$ belonging to the algorithm level to generate system parameters $params$, and sends the $params$ to DAS and UAS.

Step2: Node i sends its ID ID_i for registration.

Step3: RS gets i's requirement and invokes the algorithm $KeyGen$ belonging to the algorithm level to generate the key pairs for node.

Step4: Node i sends its original system status s to RS and UAS.

Step5: RS invokes the algorithm $ReKeyGen$ belonging to the algorithm level to generate the parameters (α_i, β_i) and (φ_i, η_i) and sends to the DAS and UAS respectively.

(2)data downloading(authorized)

The working principles for authorized node to download data is shown in Fig4. The detail will be shown as follows.

Step1: IoT-DS generates the message M for the authorized nodes and encrypts M by the symmetric method. The ciphtext is $(M)_K$, and symmetric key is K.

Step2: IoT-DS invokes the algorithm $IoTEnc$ belonging to the algorith-

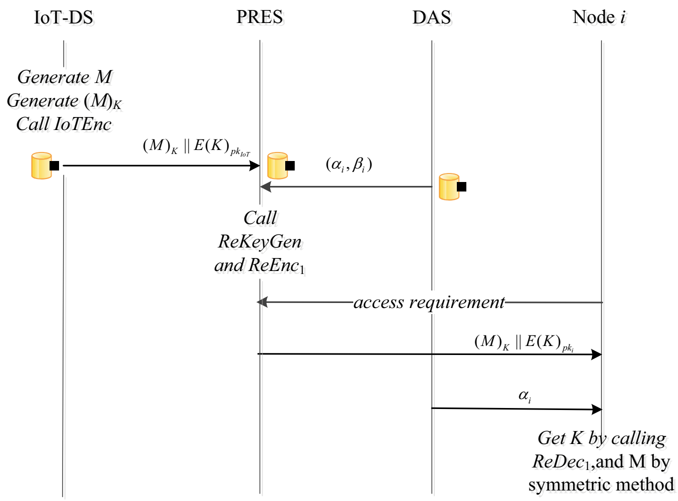

Figure 4: Working principles for authorized data downloading of PRTA.

m level to generate ciphertext $E(K)_{pk_{IoT}}$, and sends the $(M)_K||E(K)_{pk_{IoT}}$ to PRES.

Step3: DAS sends (α_i, β_i) to PRES for re-encryption.

Step4: PRES invokes the algorithms $ReKeyGen$ and $ReEnc_1$ belonging to the algorithm level to generate the re-encryption key and $E(K)_{pk_i}$ for node i.

Step5: Node i sends the access requirement to PRES and gets the $(M)_K||E(K)_{pk_i}$.

Step6: Node i gets α_i and invokes the algorithm $ReDec_1$ belonging to the algorithm level to obtain K.

Step7: Node i decrypt the $(M)_K$ by the symmetric method based on K.

(3)data downloading(unauthorized)

The working principles for unauthorized node to download data is shown in Fig5. If the node i is hijacked by the attacker, DAS will delete the parameter (α_i, β_i). The node i could not obtain the K without α_i according to the algorithm $ReDec_1$.

(4)data uploading(authorized)

The working principles for authorized node to upload data is shown in Fig.6. The detail will be shown as follows.

Step1: UAS invokes the algorithm $IoTEnc$ belonging to the algorithm level

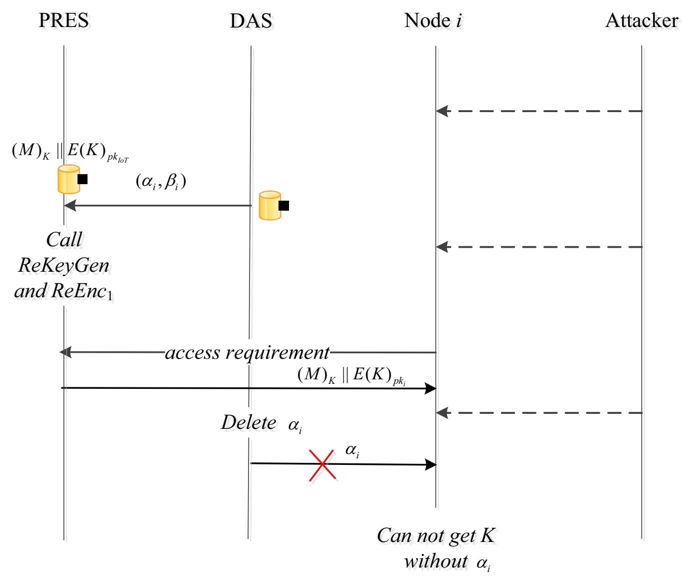

Figure 5: Working principles for unauthorized data downloading of PRTA.

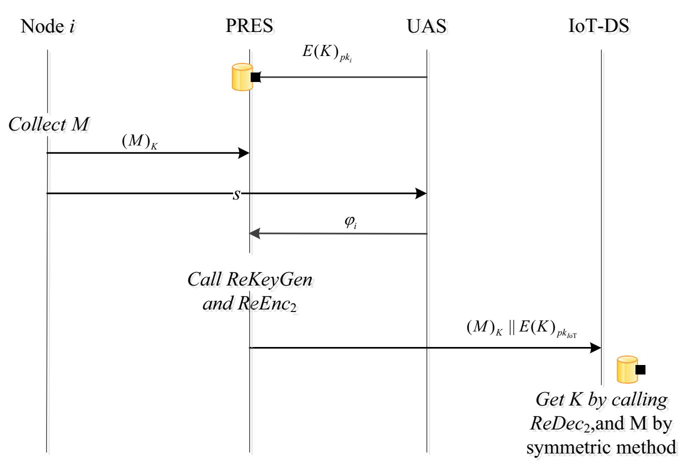

Figure 6: Working principles for authorized data uploading of PRTA.

to generate ciphertext $E(K)_{pk_i}$ and sends it to PRES.

Step2: Node i collects the data M and encrypts it by the symmetric method. The symmetric key K is initialized at the system setup phrase.

Step3: Node i sends $(M)_K$ to PRES, and sends s to UAS for uploading authorization.

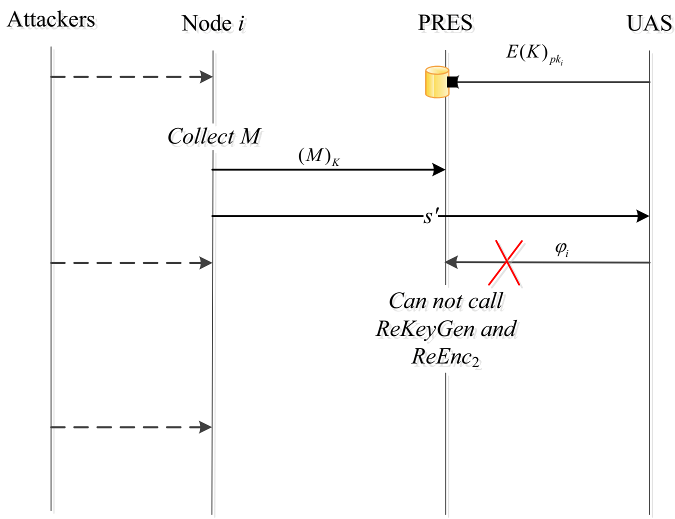

Figure 7: Working principles unauthorized data uploading of PRTA.

Step4: UAS compares the s to the parameter η_i, and sends φ_i to PRES.

Step5: PRES invokes the algorithms $ReKeyGen$ and $ReEnc_2$ belonging to the algorithm level to generate the re-encryption key and $E(K)_{pk_{IoT}}$ for IoT-DS.

Step6: IoT-DS invokes the algorithm $ReDec_2$ belonging to the algorithm level to obtain K and decrypts the $(M)_K$ by the symmetric method based on K.

(5)data uploading(unauthorized)

The working principles for unauthorized node to upload data is shown in Fig.7. If the node i is hijacked by the attacker, UAS will delete the parameter (φ_i, η_i). PRES will not re-encrypt without η_i.

3.5. Algorithms

(1)$Setup(q) \to param$, this algorithm picks a q-bit prime p. G_1,G_2 are multiplicative cyclic groups of prime order p and g is a generator of G_1. There are four hash functions H_1, H_2, H_3, H_4 with $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \to Z_p^*$, $H_3 : G_2 \to \{0,1\}^l$, $H_4 : \{0,1\}^* \to G_1$. It outputs public parameters $param = \{p, G_1, G_2, g, H_i(i = 1, \cdots, 4)\}$.

Let us define the bilinear map $e : G_1 \times G_1 \to G_2$.

(2)$KeyGen(param) \to (sk_i, pk_i)$, this algorithm picks $x_i \in Z_p^*$, and outputs $sk_i = x_i$,$pk_i = g^{x_i}$.

(3)$IoTEnc(m, pk_i) \to C_i$:according to this algorithm, enetity i uses its public key pk_i to encrypt plaintext m. This algorithm picks $f \in G_2$ to compute $r = H_2(m \parallel f)$, and outputs $C_i = (c_1, c_2, c_3, c_4, c_5)$.

$c_1 = g^r$;

$c_2 = k \cdot e(pk_i, H_1(pk_i))^r$;

$c_3 = m \oplus H_3(f)$;

$c_4 = H_1(pk_i)$;

$c_5 = H_4(c_1 \parallel c_2 \parallel c_3 \parallel c_4)^r$.

(4)$ReKeyPara(pk_i, sk_i, pk_{IoT}, sk_{IoT}, r, s) \to rkpara$, this algorithm generates the parameter list for re-encryption key based on node's status s, and output $rkpara = (\alpha_i, \beta_i), (\varphi_i, \eta_i)$. $\beta_i = \{pk_i, pk_{IoT}^{sk_{IoT}}, r\}$; $\alpha_i = \varphi_i = H_1(s)$; $\eta_i = \{pk_{IoT}, pk_i^{sk_i}, r\}$.

(5)$ReKeyGen(rkpara, flag) \to rk$, this algorithm generates the re-encryption key based on $rkpara$, and outputs the re-encryption key rk; if $flag =$ "$download$",then $rk = (pk_i, pk_i^r, H_1(pk_i \parallel \alpha) \cdot H_1(pk_{IoT})^{sk_{IoT}}, g^{-r})$. if $flag =$ "$upload$",then $rk = (pk_{IoT}, pk_{IoT}^r, H_1(pk_{IoT}) \cdot H_1(pk_i)^{sk_i}, g^{-r})$

(6)$ReEnc_1(C_{IoT}, rk) \to C_{IoT \to i}$,this algorithm outputs a ciphertext $C_{IoT \to i}$ based on re-encryption key, where $C_{IoT \to i} = (c_1', c_2', c_3', c_4', c_5')$ if $e(c_1, H_4(c_1 \parallel c_2 \parallel c_3 \parallel c_4)) = e(g, c_5)$, otherwise, outputs the error information. The ciphertext $C_{IoT \to i}$ can be decrypted with sk_i.

$c_1' = c_1$;

$c_2' = c_2 \cdot e(pk_i^r g^{-r}, H_1(pk_{IoT})^{-sk_{IoT}}) \cdot e(pk_i^r, H_1(pk_i \parallel \alpha) \cdot H_1(pk_{IoT})^{-sk_{IoT}})$

$= f \cdot e(pk_i^r, H_1(pk_i || \alpha_i))$;

$c_3' = c_3$;

$c_4' = H_1(pk_i)$;

$c_5' = H_4(c_1', c_2', c_3', c_4')^r$.

(7)$ReEnc_2(C_i, rk) \to C_{i \to IoT}$,this algorithm outputs a ciphertext $C_{i \to IoT}$ based on re-encryption key, where $C_{i \to IoT} = (c_1', c_2', c_3', c_4', c_5')$ if $e(c_1, H_4(c_1 \parallel$

$c_2 \parallel c_3 \parallel c_4)) = e(g, c_5)$, otherwise, outputs the error information. The ciphertext $C_{i \to IoT}$ can be decrypted with sk_{IoT}.

$c_1' = c_1$;

$c_2' = c_2 \cdot e(pk_{IoT}^r g^{-r}, H_1(pk_i)^{-sk_i}) \cdot e(pk_{IoT}^r, H_1(pk_{IoT}) \cdot H_1(pk_i)^{-sk_i})$

$= f \cdot e(pk_{IoT}^r, H_1(pk_{IoT}))$;

$c_3' = c_3$;

$c_4' = H_1(pk_{IoT})$;

$c_5' = H_4(c_1', c_2', c_3', c_4')^r$.

(8)$ReDec_1(sk_i, C_{IoT \to i}, \alpha_i) \to m$, this algorithm recovers m by node $i's$ private key and parameter α_i as follows: If $e(c_1', H_4(c_1' \parallel c_2' \parallel c_3' \parallel c_4')) = e(g, c_5')$, it continues, otherwise, returns errors for integrity.

$f = c_2'/e(c_1', H_1(pk_i||\alpha))^{sk_i}$

$m = c_3' \oplus H_3(f)$;

$r = H_2(m \parallel f)$;

If $c_1' = g^r$ and $c_2' = f \cdot e(pk_i, H_1(pk_i \parallel \alpha))^r$, then outputs m, otherwise, returns error.

(9)$ReDec_2(sk_{IoT}, C_{i \to IoT}) \to m$,this algorithm recovers m by IoT-DS's private key as follows:

If $e(c_1', H_4(c_1' \parallel c_2' \parallel c_3' \parallel c_4')) = e(g, c_5')$, it continues, otherwise, returns errors for integrity.

$f = c_2'/e(c_1', H_1(pk_{IoT}||\alpha))^{sk_{IoT}}$

$m = c_3' \oplus H_3(f)$;

$r = H_2(m \parallel f)$;

If $c_1' = g^r$ and $c_2' = f \cdot e(pk_{IoT}, H_1(pk_{IoT}))^r$, then outputs m, otherwise, returns error.

4. Security proof

4.1. Security Model

We will build the security model of PRTA based ont the DBDH problem. In the security model, adversary $\mathscr{A}$ can query the oracles such as key generation, data creation, ciphertext sharing by nodes, re-decryption and so on. The security model will be described as follows.

Setup: Challenger sets up system parameters $param$.

Phase 1: Adversary $\mathscr{A}$ can query one of the any oracles as follows: $KeyGen$,$IoTEnc$, $ReKeyGen$,$ReEnc_1$, $ReDec_1$,$ReEnc_2$ and $ReDec_2$. During the querying of $IoTEnc$, $ReKeyGen$,$ReEnc_1$, $ReDec_1$,$ReEnc_2$ and $ReDec_2$, $\mathscr{A}'s$ private key is generated by $KeyGen$.

Challenge: When $\mathscr{A}$ finishes Phase 1, the challenger picks and outputs $m_0, m_1 \in M$, parameter list $rkpara^*$ and a target public key pk^* generated by $KeyGen$. Its corresponding private key is undisclosed. When $\mathscr{A}$ queries $ReKeyGen$ with (pk^*,pk',$rkpara^*$),the private key corresponding with pk' should be undisclosed. Challenger picks $b \in \{0, 1\}$ randomly and computes $C_b = IoTEnc(m_b, pk^*)$ as the challenge to $\mathscr{A}$.

Phase 2: $\mathscr{A}$ is allowed to continue querying the same types of oracles as in Phase 1. At the end of Phase 2, we have the following constraints.

(1)If $\mathscr{A}$ queries $ReKeyGen$ with ($list*$), the corresponding private key is undisclosed.

(2)If $\mathscr{A}$ queries $ReEnc_1$ with (C_b,pk_*,pk',α^*) , the corresponding private key is undisclosed.

(3)If $\mathscr{A}$ queries $ReEnc_2$ with (C_b,pk_*,pk') , the corresponding private key is undisclosed.

(4)$\mathscr{A}$ cannot query $ReDec_1$ and $ReDec_2$ with (C'_b,pk_*) directly.

(5)If $\mathscr{A}$ queries $ReKeyGen$ with (pk^*,pk',$rkpara*$), $\mathscr{A}$ cannot query $ReDec_1$ and $ReDec_2$ with C'_b , where C'_b is valid.

Guess: $\mathscr{A}$ outputs a guess, if $b' = b$, $\mathscr{A}$ will success.

Let us define the advantage of $\mathscr{A}$ to success as ε, where $\varepsilon = |Pr[b' = b] - \frac{1}{2}|$. If ε is negligible, $\mathscr{A}$ will fail. It means that the PRTA is CCA security.

4.2. Proof

Theorem: If DBDH assumption holds in groups (G_1,G_2), then the PRTA is CCA secure based on random oracle model.

Proof sketch:

(1)$\mathcal{G}_0$:Challenger $\mathscr{B}$ faithfully answers the oracle queries from $\mathscr{A}$. At the same time,$\mathscr{B}$ initializes $H_i^{list}(i=1,\cdots,4)$ by choosing $\pi_1,\pi_4 \in G_1, \pi_2 \in Z_p^*, \pi_3 \in \{0,1\}^l$ and setting (pk_i,π_1),(m,k,π_2),(k,π_3), (c_1,c_2,c_3,c_4,π_4) in $H_i^{list}(i=1,\cdots,4)$. Let $\delta_0 = Pr[b'=b]$, then $|\delta_0 - \frac{1}{2}| = \varepsilon$.

(2)$\mathcal{G}_1$:Challenger $\mathscr{B}$ does the same as that in $\mathcal{G}_0$, except the following:

$\mathscr{B}$ randomly picks $\tau \in \{1,2,\cdots,p+1\}$ to query H_1 in τ times. When $\mathscr{B}$ receives $\mathscr{B}'s$ challenge to query H_1, B aborts the game. Therefore, the probability of B to succeed is $\frac{1}{p+1}$ at least . $\delta_1 = Pr[b'=b]$ in $\mathcal{G}_1$, and then $Pr[T_1] = \frac{\delta_1}{p+1}$.

(3)$\mathcal{G}_2$:Challenger $\mathscr{B}$ does the same as that in $\mathcal{G}_1$, except the situation of H_i conflicting. The hash functions are the standard random oracles, so $|Pr[T_1] - Pr[T_2]|$ is negligible.

(4)$\mathcal{G}_3$:Challenger $\mathscr{B}$ does the same as that in $\mathcal{G}_2$, except the query of $ReDec_1$ and $ReDec_2$. In the oracle of $ReDec_1$ querying, if the input is (C,pk^*,α^*) and $\mathscr{A}$ has not queried H_1 with $(pk^* \parallel \alpha^*)$, then $\mathscr{B}$ aborts the game, otherwise $\mathscr{B}$ returns the ciphertext to $\mathscr{A}$. In the oracle of $ReDec_2$ querying, if the input is (C,pk^*) and $\mathscr{A}$ has not queried H_1 with pk^*, then $\mathscr{B}$ aborts the game, otherwise $\mathscr{B}$ returns the ciphertext to $\mathscr{A}$. Since the hash functions are the standard random oracles and all the cryptography algorithms are certain, $|Pr[T_2]-Pr[T_3]|$ is also negligible.

(5)$\mathcal{G}_4$:Challenger $\mathscr{B}$ does the same as that in $\mathcal{G}_3$, except the query of $ReDec_1$ and $ReDec_2$. If $\mathscr{A}$ has not queried H_2 with $m_b \parallel k^*$, there is no differences between $\mathcal{G}_4$ and $\mathcal{G}_3$. Therefore, $|Pr[T_3] - Pr[T_4]|$ is negligible.

(6)$\mathcal{G}_5$:Challenger $\mathscr{B}$ does the same as that in $\mathcal{G}_4$, except the querying of $ReKeyGen$,$ReEnc_1$ and $ReEnc_2$. During the query, $\mathscr{B}$ searches the re-encryption key list with $(pk_i,pk_{IoT},rkpara)$ proposed by $\mathscr{A}$. If there is a result of search , $\mathscr{B}$ will return $rk_{IoT\to i}$ to $\mathscr{A}$, otherwise, $\mathscr{B}$ will continue as follows.

If $flag$ = "$upload$" and node $i's$ private key is corrupted, which means $sk_i = x_i$, then $\mathscr{B}$ computes $rk = (pk_{IoT}, pk_{IoT}^r, H_1(pk_{IoT}) \cdot H_1(pk_i)^{sk_i}, g^{-r})$.

If node $i's$ private key is corrupted, then $\mathscr{B}$ will pick $a \in G_1$, set $sk_i = ax_i$, and compute $rk = (pk_{IoT}, pk_{IoT}^r, H_1(pk_{IoT}) \cdot H_1(pk_i)^{sk_i}, g^{-r})$.

If $IoT-DS's$ private key is corrupted, then $\mathscr{B}$ aborts.

If $flag =$ "$download$" and IoT-DS's private key is corrupted, which means $sk_{IoT} = x_{IoT}$, then $\mathscr{B}$ computes $rk = (pk_i, pk_i^r, H_1(pk_i \parallel \alpha) \cdot H_1(pk_{IoT})^{sk_{IoT}}, g^{-r})$.

If IoT-DS's private key are uncorrupted, then $\mathscr{B}$ will pick $a \in G_1$, set $sk_{IoT} = ax_{IoT}$, and compute $rk = (pk_i, pk_i^r, H_1(pk_i \parallel \alpha) \cdot H_1(pk_{IoT})^{sk_{IoT}}, g^{-r})$.

If node $i's$ private key is corrupted, then $\mathscr{B}$ aborts.

When $ReEnc_1$ is being queried, $\mathscr{B}$ computes the re-encryption ciphertext with (pk_{IoT}, pk_i, C) proposed by $\mathscr{A}$. If it does not hold, $\mathscr{B}$ aborts. Otherwise,$\mathscr{B}$ searches the private keys from private key list and re-encryption key list, and returns ciphertext to $\mathscr{A}$. If pk_i is not generated by $KeyGen$,B aborts. $|Pr[T_4] - Pr[T_5]|$ is negligible.

When $ReEnc_2$ is being queried, $\mathscr{B}$ computes the re-encryption ciphertext with (pk_i, pk_{IoT}, C) proposed by $\mathscr{A}$. If it does not hold, $\mathscr{B}$ aborts. Otherwise,$\mathscr{B}$ searches the private keys from private key list and re-encryption key list, and returns ciphertext to $\mathscr{A}$. If pk_{IoT} is not generated by $KeyGen$,B aborts. $|Pr[T_4] - Pr[T_5]|$ is negligible.

(7)$\mathcal{G}_6$:Challenger $\mathscr{B}$ does the same as that in $\mathcal{G}_5$, excepts the following situations.

When $\mathscr{B}$ receives the $\mathscr{A}'s$ challenging (m_0,m_1,$rkpara$), $\mathscr{B}$ will decrypt the ciphertext at first time, and then pick $b \in \{0, 1\}$ to compute $f \in G_2$,$r = H_2(m_b||f)$,$c_1 = g^r$,$c_2 = k \cdot e(pk_i, H_1(pk_i))^r$,$c_3 = m \bigoplus H_3(k)$,$c_4 = H_1(pk_i)$,$c_5 = H_4(c_1||c_2||c_3||c_4)^r$. Therefore, the difference between $\mathcal{G}_6$ and $\mathcal{G}_5$ is whether query H_3 or not. The difficulty of querying H_3 is based on the DBDH problem, so $|Pr[T_5] - Pr[T_6]|$ is negligible. Hash functions are the random oracles, so $Pr[T_6] = \frac{1}{2(p+1)}$. $|Pr[T_1] - Pr[T_6]| = |Pr[T_1] - \frac{1}{2(p+1)}|$ is negligible based on the analysis from (1) to (7), the $\Pr[T_1] = \frac{\delta_0}{p+1}$ and $|\frac{2\delta_0 - 1}{2(p+1)}| = |\frac{\delta_0 - \frac{1}{2}}{(p+1)}| = |\frac{\varepsilon}{(p+1)}|$ is negligible. Therefore, ε is negligible. The proof is finished.

4.3. Correctness Proof

Our scheme have protected the message M with the traditional symmetric algorithm (eg. AES), and protect the symmetric key K with the PRE method. Thus, the correctness proof will be based on two parts. One part is the correctness of the traditional symmetric algorithm which is obvious. The other part is the correctness of the proposed algorithm. The key pairs of node i and IoT-DS are generated by $KeyGen$. And for IoT-DS $K = ReDec_2(sk_{IoT}, ReEnc_2(IoTEnc(K, pk_i), ReKeyGen(rkpara, flag)))$. It means the K is encrypted by node i's public key based on $IoTEnc$, and after $ReEnc_2$, the ciphertext of K could be decrypted by IoT-DS's private key. And for node i, $K = ReDec_1(sk_i, \alpha_i, ReEnc1(IoTEnc(K, pk_{IoT}), ReKeyGen(rkpara, flag)))$. It means the K is encrypted by node IoT-DS's public key based on $IoTEnc$, and after $ReEnc_1$, the ciphertext of K could be decrypted by i's private key.

5. Discussion

5.1. Security Analysis and Properties

5.1.1. Security Analysis

The security of our scheme will be based on two issues. Firstly, we will prove the security of algorithm. We have constructed a security framework based on random oracle model in the section above. In the framework, we can prove our algorithm is CCA-security by the challenge-response method (see Section 5.2). Secondly, we will analyze the possible attacks to prove the security of our system.

1) cryptographic analysis. We will encrypt the message M by symmetric method based on the traditional algorithms. And the symmetric K for M will be protected by the CCA-security algorithm. Thus, we can defense the cryptographic analysis.

2) data leaking of database. The data in the databases is encrypted, thus if attackers steal the information of the database, they will not be able to obtain the original data.

3) collusive attack. The PRE servers, common nodes hijacked by attackers might try to compute the IoT-DS's private key based on the parameters. However the difficulty to recovery the sk_{IoT} from $H_1(pk_{IoT})^{sk_{IoT}}$ is equal to solve the discrete logarithm problem.

4)Hijacking attack. When the node is hijacked by attackers, the status s might be changed. Our scheme takes the status s of the node as one of the parameters for re-encryption key generation. $\alpha = \eta = H_1(s)$, and H_1 is a hash function. If the s is changed, then α and η will be changed obviously. The DAS and UAS will compare the status s of the node when it need to download or upload data, when the s changed, they will refuse the requirement of the node's requirement assuredly by delete the re-encryption key parameters. The attackers will not be able to download or upload the information.

5.1.2. Properties

We will make a comparison between other works with ours(see Table 1). We have analyzed the five schemes [12],[23],[10],[36] and [11]. Firstly, [12],[23],[36]and [11] have considered the cloud computing. Paper [12]has made cloud as the service provider, the host will create and encrypted the data for senors in WSN. Paper [23]proposed a secure data-sharing scheme at the edge of cloud connected IoT smart devices that utilizes both secret key encryption and public key encryption. all security operations are off loaded to nearby edge servers, thereby, reducing the processing burden of smart devices. The cloud will be responsible for data storing and computing.Paper [36] has taken the cloud platform as the storing service provider.Paper [11] has divided the processes into the managing and sharing parts, the cloud will store the data for the managing part and deal with the data for the sharing part. In our scheme, the cloud server provides the proxy re-encryption service, the cost of the nodes will be reduced. Secondly,paper [12],[23], [36] , [11]and our scheme are designed for application layer, and paper[10] is designed for transferring layer.Paper[12], [23] and [11] are for data downloading and sharing scenarios, paper[10]and [36] are focus on the data collecting scenarios. Our scheme have designed the servers for data sharing

Table 1: Properties comparisons between current works and our scheme

Schemes	[12]	[23]	[10]	[36]	[11]	Ours
Cloud-based	$\surd$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$
Role of cloud	Storing	Storing & computing	No	Storing	Storing & computing	Storing & computing
Layers	Application	Application	Transferring	Application	Application	Application
Applications scenarios	Downloading	Downloading	Updating	Updating	Downloading & updating	Downloading & updating
Trusted updating	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$

* $\surd$ means the scheme can support the property. $\times$ means the scheme can not support the property.

and collecting respectively. Finally, our scheme takes the nodes status as one of the parameters for data re-encryption and the parameters are controlled by the UAS/DAS for data uploading/downloading, which is the trusted server. During uploading process, if some node is hacked by attackers, the status of this node will be changed. When it submits the new status to UAS for data uploading, UAS will discover the difference and delete the parameter of this node for data uploading. As a result, the hacked node will not upload the data without the parameter for re-encryption. During downloading process, when the IoT-DS wants to change the permission of the node, it will tell the DAS to delete the node's parameter. The node will not be able to download the information from the IoT-DS without the parameter for re-encryption.The other works have not discussed the authorization assured updating.

5.2. Efficiency analysis

This section compares the efficiency of the proposed scheme with that of the previous ones[12],[23],[10],[36] and [11]. As the setup and registration phase of nodes is executed only once, only the efficiency comparison of the downloading and uploading phases are necessary. To simplify the presentation, the following symbols are defined.

T_E: the cost of exponent computation;

T_P: the cost of pairing computation;

T_a: the other cost of symmetric encryption.

Table 2: Efficiency comparisons between current works and our scheme

Schemes	[12]	[36]	[11]	Ours
Data accessing	$T_E + 6T_P$	$2T_E + 3T_P$	$T_E + T_P$	$5T_E + 2T_P$
Data collecting	–	$2T_E$	$6T_E + 3T_P$	$5T_E + 2T_P$
Nodes' cost	T_P	$2T_E$	$5T_E + 3T_P$	T_a or $T_E + T_P$

Table 2 shows the results of efficiency comparisons among the proposed scheme and the previous ones [12],[36] and [11].([23] has not described the details of the algorithm and [10] is based on ECC which is different from the ideal of other work, thus , we do not analyze the efficiency of them)

It shows that our proposed scheme has better efficiency than [11] for data collection and has better efficiency than [12] at for data accessing. Although the scheme of [36] has slightly better efficiency than ours, it cannot accomplish the trusted revoking of the compromised nodes, as is shown in Table 1. And our scheme divided the nodes into two kinds, the ones are designed for data collection,such as cameras of smart city. They will update the information on time, and they are not good at computing, thus, we can just configure the scheme for data collection by selecting the algorithms including $ReEnc_2$ and $ReDec_2$, the cost of nodes will be T_a. The other ones are designed for data collecting and accessing, such as the nodes for military. They will update the information and download the commands from the servers. They could deal with some work of decryption. Thus, we can just configure the scheme for data collection by selecting the algorithms including $ReEnc_1$ and $ReDec_1$,the cost of nodes will be $T_E + T_P$.

5.3. Further works

(1)How to obtain the status s

In our scheme, we have made the status of node to be one parameter for re-encryption, but we have not discussed the detail of how to construct the status s. In the further, we will do some work on the description of s.

(2)How to detect the attacks

Our scheme is based on the assumption that the system could detect the attacks. In the future, we will design the schemes with the abilities of attack detecting.

(3)How to realize the lightweight algorithm on the nodes

The nodes are limited in the energy, computing and storing, thus we will analyze the cost of the algorithms for data encryption and design the lightweight scheme.

6. Conclusion

The IoT brings the convenience to connect everything to Internet. And CloudIoT has increased the abilities of IoT to data storing and processing. However, this integration has brought the new challenges to security of IoT and cloud. Firstly, there are thousands and hundreds nodes of IoT, more and more information is appeared in cloud including the sensitive data and privacy. Secondly, the nodes are limited in storing and computing, which make them weak in defending against attacks. Connected to cloud server will make the problem even more serious. How to realize the trusted authorization updating is one of the serious problems in CloudIoT. In our work, we have proposed a PRE based Trusted Authorization scheme for nodes on CloudIoT (PRTA). Firstly, we have analyzed the motivation and state-of-the-art solutions. Secondly, we have given the system model, processes and algorithms. Finally, we have proved the security and analyzed the properties. In the PRTA scheme, the cloud server will be responsible for data storing and re-encryption, which will reach the full potential of cloud computing and reduce the cost of nodes. We have taken the node's status as one of the parameters for data re-encryption and the parameters are under the authorization servers' control, which could update the authorization assuredly . Also, the authorization servers are divided into the downloading and uploading types, which will be wider in application range, including the IoT for data sharing, IoT for data collecting and both of them.

References

[1] Ironpaper Growth Agency. Internet of things market statistics-2016., 2016.

[2] R. Aitken, V. Chandra, J. Myers, B. Sandhu, L. Shifren, and G. Yeric. Device and technology implications of the internet of things. In *2014 Symposium on VLSI Technology (VLSI-Technology): Digest of Technical Papers*, pages 1–4, 2014.

[3] B. Alessio, D. D. Walter, P. Valerio, and P. Antonio. Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700, 2016.

[4] N. Alhakbani, M. M. Hassan, and M. A. Hossain Alnuem M. and. A framework of adaptive interaction support in cloud-based internet of things (iot) environment. In *International Conference on Internet and Distributed Computing Systems*, pages 136–146, 2014.

[5] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R Moosavi, A. M Rahmani, and P. Liljeberg. On the feasibility of attribute-based encryption on internet of things devices. *IEEE Micro*, 36(6):25–35, 2016.

[6] Z. Bi, L. Xu, and C. Wang. Internet of things for enterprise systems of modern manufacturing. *IEEE Transactions on Industrial Informatics*, 10(2):1537–1546, 2014.

[7] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang. Npp: a new privacy-aware public auditing scheme for cloud data sharing with group users. *IEEE Transactions on Big Data*, pages 1–10, 2018.

[8] M. M. Gomes, R. R. Righi, and A. C. Cristiano. Future directions for providing better iot infrastructure. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 51–54, 2014.

[9] IHS. Internet of things (iot) connected devices installed base worldwide from 2015 to 2025 (in billions), 2017.

[10] Q. Jiang, N. Kumar, J. Ma, J. Shen, D. He, and N. Chilamkurti. A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks. *International Journal of Network Management*, 27(3):1–17, 2017.

[11] S. H. Kim and I. Y. Lee. Iot device security based on proxy re-encryption. *Journal of Ambient Intelligence & Humanized Computing*, 9(4):1267–1273, 2018.

[12] F. Li, Y. Han, and C. Jin. *Practical access control for sensor networks in the context of the Internet of Things*, volume 89-90. Elsevier Science Publishers B. V., 2016.

[13] J. Li, X. Lin, Y. Zhang, and J. Han. Ksf-oabe: outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Transactions on Services Computing*, 10(5):715–725, Sept 2017.

[14] J. Li, Y. Wang, Y. Zhang, and J. Han. Full verifiability for outsourced decryption in attribute based encryption. *IEEE Transactions on Services Computing*, pages 1–12, 2018.

[15] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen. User collusion avoidance cp-abe with efficient attribute revocation for cloud storage. *IEEE Systems Journal*, 12(2):1767–1777, June 2018.

[16] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han. Flexible and fine-grained attribute-based data storage in cloud computing. *IEEE Transactions on Services Computing*, 10(5):785–796, Sept 2017.

[17] J. Li, Q. Yu, Y. Zhang, and J. Shen. Key-policy attribute-based encryption against continual auxiliary input leakage. *Information Sciences*, 470:175–188, 2019.

[18] J. Li, X. Zhao, Y. Zhang, and W. Yao. Provably secure certificate-based conditional proxy re-encryption. *Journal of Information Science & Engineering*, 32(4):813–830, 2016.

[19] J. Liu, Y. Xiao, and C. L. P. Chen. Authentication and access control in the internet of things. In *International Conference on Distributed Computing Systems Workshops*, pages 588–592, 2012.

[20] J. Liu, Y. Xiao, and C. L. P. Chen. Authentication and access control in the internet of things. In *International Conference on Distributed Computing Systems Workshops*, pages 588–592, 2012.

[21] M. Luo, Y. Luo, Y. Wan, and Z. Wang. Secure and efficient access control scheme for wireless sensor networks in the cross-domain context of the iot. *Security and Communication Networks*, pages 1–10, 2018.

[22] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad. Identity establishment and capability based access control (iecac) scheme for internet of things. In *International Symposium on Wireless Personal Multimedia Communications*, pages 187–191, 2012.

[23] M. B. Mollah, M. A. K. Azad, and A. Vasilakos. Secure data sharing and searching at the edge of cloud-assisted internet of things. *IEEE Cloud Computing*, 4(1):34–42, 2017.

[24] P. P. Pereira, J. Eliasson, and J. Delsing. An authentication and access control framework for coap-based internet of things. In *Industrial Electronics Society, IECON 2014 - Conference of the IEEE*, pages 5293–5299, 2016.

[25] K. Pretz. The Next Evolution of the Internet The Internet of Things means everything will be connnected, 2013.

[26] D. Puthal, S. Nepal, R. Ranjan, and J. Chen. Threats to networking cloud and edge datacenters in the internet of things. *IEEE Cloud Computing*, 3(3):64–71, 2016.

[27] M. Su, M. Cao, R. Xie, and A. Fu. Pre-tuan: proxy re-encryption based trusted update scheme of authorization for nodes on iot cloud. *Journal of Computer Research and Development*, 55(7):1479–1487, 2018.

[28] M. Su, G. Shi, R. Xie, and A. Fu. Multi-element based on proxy re-encryption scheme for mobile cloud computing. *Journal on Communications*, 36(11):73–79, 2015.

[29] K. Sun and L. Yin. *Attribute-role-based hybrid access control in the internet of things*. Springer International Publishing, 2014.

[30] W. Tan, S. Chen, J. Li, L. Li, T. Wang, and X. Hu. A trust evaluation model for e-learning systems. *Systems Research & Behavioral Science*, 31(3):353–365, 2014.

[31] Q. Tang. Type-based proxy re-encryption and its construction. In *International Conference on Cryptology in India: Progress in Cryptology*, pages 130–144, 2008.

[32] D. V. Thuan and P. Butkus. A user centric identity management for internet of things. In *International Conference on It Convergence and Security*, pages 1–4, 2014.

[33] International Telecommunication Union. Overview of the internet of things. Technical report, International Telecommunication Union (Itu) internet report., 2012.

[34] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin. Conditional identity-based broadcast proxy re-encryption and its application to cloud email. *IEEE Transactions on Computers*, 65(1):66–79, 2015.

[35] Z. Yan, P. Zhang, and A. V. Vasilakos. A survey on trust management for internet of things. *Journal of Networks & computer Applications*, 42(3):120–134, 2014.

[36] Y. Yang, X. Liu, and R. H. Deng. Lightweight break-glass access control system for healthcare internet-of-things. *IEEE Transactions on Industrial Informatics*, 14(8):3610–3617, 2018.

[37] Y. Yang, H. Lu, J. Weng, Y. Zhang, and K. Sakurai. Fine-grained conditional proxy re-encryption and application. *International Conference on Provable Security*, 8782:206–222, 2014.

[38] S. Yu. Big privacy: challenges and opportunities of privacy study in the age of big data. *IEEE Access*, 4:2751–2763, 2016.

[39] S. Yu, G. Gu, A. Barnawi, S. Guo, and I. Stojmenovic. Malware propagation in large-scale networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(1):170–179, Jan 2015.

[40] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou. Networking for big data: a survey. *IEEE Communications Surveys Tutorials*, 19(1):531–549, Firstquarter 2017.

[41] S. Yu, Y. Tian, S. Guo, and D. O. Wu. Can we beat ddos attacks in clouds? *IEEE Transactions on Parallel and Distributed Systems*, 25(9):2245–2254, Sept 2014.

[42] Y. Zhang, J. Li, X. Chen, and H. Li. Anonymous attribute-based proxy re-encryption for access control in cloud computing. *Security and Communication Networks*, 9(14):2397–2411, 2016.

[43] L. Zhou, A. Fu, S. Yu, M. Su, and B. Kuang. Data integrity verification of the outsourced big data in the cloud environment: a survey. *Journal of Network and Computer Applications*, 122:1–15, 2018.

***LaTeX Source Files**

Click here to download LaTeX Source Files: latex-InS.rar