IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10719

A Collaborative Key Management Protocol in Ciphertext Policy Attribute-Based Encryption for Cloud Data Sharing

Kavyashree N [1] VaniShree [2] Vaishnavi. K [3] Veena O M [4]

Asst Professor of Dr AIT,Dept of MCA, Bangalore-560056[1]

Dr AIT,Dept of MCA, Bangalore-560056[2-4]

Abstract: Cryptography based on ciphertext policy attributes (CPABE) is a promising cryptographic method that can be used to fine-tune access to outsourced knowledge in the cloud. However, flaws in key management make your application difficult to detect. A central theme in the written agreement. Please note that buyers' front-end devices (such as smartphones) often have limited data protection. Therefore, if it is completely controlled by a non-public key, consumers may leak a key that is almost unrecognizable, but initially in the previous version. explore. Use ABE wisely. In the course of this work, we proposed the collaborative key management protocol in CPABE (CKMCPABE). Our design performs the distributed generation, release and storage of private keys without adding additional infrastructure. When the key is updated, immediate and detailed attribute revocation is provided. The cooperation mechanism not only effectively solves the core problem of the written agreement, but also effectively solves the problem of key leakage. At the same time, it helps to significantly reduce the workload of consumer decryption. A comparison with various representative CPABE themes shows that our model performs slightly better in terms of external knowledge sharing in the mobile device cloud. Finally, we provide proof of security for the proposed protocol.

Index terms: cloud data exchange, CPABE, key management, security, efficiency.

I.INTRODUCTION

As computer technology and large-scale networks became more profitable, it became more convenient to exchange data with others. computing and cloud storage also make digital assets easier to maintain . Because data exchange in the cloud requires remote infrastructure shared by some organizations, remote storage damages the privacy of data owners to a certain extent.

This work was partially supported by the Wireless Broadband and Sensor Network Core Laboratory of Nanjing University of Posts and Telecommunications. G. Lin's work was partially supported by the National Natural Science Foundation of China (61672299, 61373135). H. Hung's work was partially supported by the National Natural Science Foundation of China (61170276). Z Sun's work was partially supported by the National Natural Science Foundation of China (60973140). G. Lin, H. Hong, Z.Sun arbeitet am Key Laboratory of Wireless Broadband and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003 (E-Mail: 2015070249@njupt.edu.cn; 2014070244@njupt.edu.cn; sunzx@njupt.edu.Com)

Sensitive and confidential data stored in the cloud is extremely important. The simultaneous participation of many users requires detailed access control for data exchange. Attribute encryption (ABE) is a promising encryption primitive that provides an interesting solution for secure and flexible data exchange. .ABE has the single-person feature, that is, one key can decrypt different ciphertexts, or different keys can decrypt the same ciphertext. There are two types of ABE: ciphertext ABE policy (CPABE) and key ABE policy (KPABE). , The access policy is embedded in the ciphertext, and the attribute set is embedded in the private key. With KPABE, the access policy is embedded in the private key, and the attribute set is embedded in the ciphertext. CPABE enables data owners to define their own access policies. Anyone who wants to retrieve data must first match the set access policy attributes. This attribute makes CPABE ideal for the security and fine-grained access control of cloud communications.

However, there are still many unresolved problems related to the actual implementation of ABE , especially in private key management. For many previous ABE schemes, the key authorization must be completely reliable because they can all decipher the ABE scheme. Use the ciphertext of the private key generated by without the owner's permission. This is often referred to as the key store problem, which is an inherent defect that compromises user privacy. The underlying trend of cloud computing. The current research has little evidence that mobile devices with an interface (such as

DOI 10.17148/IJARCCE.2021.10719

smartphones) are more susceptible to data protection than servers. Therefore, the weak point of the protection of the private key can easily lead to the disclosure of the key to unauthorized users. [thirty]. In addition, the current ABE key management scheme also requires many bilinear pair , power and multiplication, and calculations, especially in the decryption step. The resulting runtime may be very unacceptable.

In this article, we propose a new encryption key management protocol (CKMCPABE) based on ciphertext policy attributes to improve the security and efficiency of key management in cloud communications. Through the interaction between the key management center, the cloud server, and the client who wants to access the data, the private key is generated, released, and stored in a distributed manner, and secure key management can be realized without additional physical infrastructure, which is easier to implement than The previous multi-authority system. 2) We introduce attribute groups to create a private key update algorithm. Each attribute group that contains clients with the same attributes is assigned a unique attribute group key. The key to update the attribute group provides detailed and immediate revocation of the attribute. 3) We would like to point out that not only is the key storage problem, but the leakage of the key will also endanger the confidentiality of the private key, which has hardly been considered in previous studies. Two problems with shared key management. Finally, we provide evidence of the security of the proposed protocol. The collaboration mechanism helps to significantly reduce the client's decryption overhead by using the decryption server to perform most of the decryption without leaving any information about the information.
The rest of paper is organized as follow: We review previous work on ABE primitives and their applications in Section II, give essential preliminaries in Section III, present the CKM-CP-ABE model for cloud data sharing in Section IV, propose main key management algorithm in Section V, provide comprehensive analysis on the security and efficiency of CKM-CP-ABE in Section VI, and draw conclusion and highlight our future work in Section VII.

II. RELATED WORK

In this article, we propose a new encryption key management protocol (CKMCPABE) based on ciphertext policy attributes to improve the security and efficiency of key management in cloud communications. Through the interaction between the key management center, the cloud server, and the client who wants to access the data, the private key is generated, released, and stored in a distributed manner, and secure key management can be realized without additional physical infrastructure, which is easier to implement than The previous multi-authority system. 2) We introduce attribute groups to create a private key update algorithm. Each attribute group that contains clients with the same attributes is assigned a unique attribute group key. The key to update the attribute group provides detailed and immediate revocation of the attribute. 3) We would like to point out that not only is the key storage problem, but the leakage of the key will also endanger the confidentiality of the private key, which has hardly been considered in previous studies. Two problems with shared key management. Finally, we provide evidence of the security of the proposed protocol. 4) The collaboration mechanism helps to significantly reduce the client's decryption overhead by using the decryption server to perform most of the decryption without leaving any information about the information. The rest of the article is structured as follows: In Section II, we introduced the previous work on ABE primitives and their applications, in Section III we provided necessary background information, and in Section IV we introduced the use of cloud data The exchanged CKMCPABE model, which is a basic key management recommendation. The algorithm proposed in Section 5 is a comprehensive analysis of the security and effectiveness of CKMCPABE in Section 6, and a conclusion is drawn and emphasized in Section 7 For our future work.

Green et al. Point out that the size of the ciphertext and the cost of decryption are the main disadvantages of using ABE in practice. To overcome these problems, they proposed a new ABE with side decryption (ODABE) for CPABE and KPABE, which sets up a proxy for most decryption calculations. In the decryption process, the data receiver transmits the conversion key and ciphertext to the proxy server, and receives the ciphertext in ElGamal style. The plain text can then be retrieved from the data recipient through a very simple invoice. In view of the potential trend of the development of mobile cloud services, the external application decryption solution greatly optimizes the user experience. Lai et al. Proposed attribute-based encryption and verifiable outsourced decryption (VODABE). In ODABE, an untrusted proxy server risks the confidentiality of the ciphertext and conversion key. In addition, incorrect calculations can make the entire system unusable. His research shows that a security requirement called outsourced verification needs to be proposed for ODABE. Lin et al. Provides an improved VODABE based on the Key Encapsulation Engine (KEM). Their experiments show that the size of the ciphertext and the cost of encryption and decryption are almost half of those designed by Lai et al.

Chase et al. a brand new ABE-based multi-authorization key generation rule is planned. They assume that the central authority within the EBA is trustworthy by many previous authoritative there foreurces, so it will extract plain text while not the permission of alternative establishments and users. The proposed algorithm needs interaction between the central authority and other authorities to collectively generate the key and its main secret to resolve the key storage problem. Pletea et al. The idea of universal hierarchic attributes supported the world attribute set is introduced, and therefore the multi-authority hierarchical mechanism of CPABE is proposed. once the user defines the access structure and requests knowledge encryption, every key generation center (KGA) can generate the corresponding access policy and its level of personal key, in order that though the leaked KGA is concerned, the protection of key management will be secure sex. The key issue. Xu et al. A multi-authorization proxy re-encryption theme for ABE is planned to realize economical and correct revocation. so as to manage the keys within the architecture, user-defined access policies are appointed to a weighted access list, that reduces the process work involved in generating and issue private keys. Zhang et al. Take the lead in implementing a 3rd infrastructure for one organization at KPABE to resolve crucial storage issues. The key generation communication overhead is far not up to the previous unreserved multi-authorization scheme. Hur planned a secure and economical attribute-based communication system. They offer a brand new answer to the key storage downside in a very single management system by implementing two-component computing (2PC) between the key generation center and therefore the knowledge storage center while not adding any further infrastructure. Introduce key management infrastructure (KMI) for private health records (PHR) within the cloud. you've got outlined a non-public domain and a property right to share PHR with completely different users. within the personal domain, multi-authority CPABE is introduced to expeditiously management the access of an outsized variety of users. Users rely upon their skilled role. within the public domain, KPABE is suburbanized so as to delegate access to many users regarding the individual PHR holder. The KMI they planned aims to use superior Oualha et al to resolve the matter of attribute request and user recall. it's seen that though ABE needs Brobdingnagian computing resources, it will complete large-scale calculations in advance. in sight of the ability limitations and calculations of nodes in the web of Things, they planned a brand new CPABE, that introduces a pre-calculation methodology that calculates and stores some key parts before finishing encryption. though period calculations are greatly reduced, your design needs trustworthy objects to store items. A reliable channel is additionally required to securely transmit these elements to the desired nodes.

As mentioned above, previous schemes of key management in attribute-based information sharing system primarily focuses on key update, proxy re-encryption and outsourced decryption. Some analysis incontestable untrusted key authority might cause key written agreement downside and provided corresponding solutions. However, very little research notices that if authority is untrusted, front-end devices, particularly mobile ones should be way more untrusted than it as a result of their inherently prone to black-market access. If private keys are still entirely hold on in front-end devices, a worse problem known as key exposure happens, threatening confidentiality of personal keys. In addition, most of attribute-based data sharing schemes increased security of key management at the value of cryptography overhead of information receivers. Therefore, we have a tendency to don't seem to be glad about previous schemes of key management in terms of either security or efficiency.

III. PRELIMINARIES

A. Bilinear Pairings

1) BiLinearity
2) Non-degeneracy:

Let G1 and G2 be two multiplicative cyclic groups with prime order *p* , and let *g* be a generator.

B. Linear Secret Sharing Scheme

Definition 1 (access structure): Let $p=\{p_1,p_2,\ldots p_m\}$ be a group of participants. If for any B and C, we have B$\in$ A and B$\subseteq$ C, and C $\in$ A, then the set is monotonic. A monotonic access structure is a monotonic set A$\in 2^P \setminus \{\emptyset\}$ We call the set in the authorized set A, and the unauthorized set as the unauthorized set A

Definition 2 (Split Secret Linear Scheme): Let P be a set of participants and M be a matrix with m rows and d columns. Map: $\rho:\{1,2,3,\ldots..m\}\rightarrow P$

Assign a subscriber to each line for marking. The secret exchange scheme on P used to access the structure A Π is * the linear secret partition scheme in Z_p^*, denoted by (M, ρ) if it consists of two polynomial time algorithms.

IV. MODEL OF CKM-CP-ABE FOR CLOUD DATA-SHARING

1) Customers. The client (CL) is a user who wants to access data in cloud storage through a front-end device. In view of the potential trend of mobile cloud services, mobile devices constitute most of the front-end equipment. If the attribute set of CL satisfies the ciphertext access policy, CL can obtain plaintext. We believe that the performance of most mobile devices will be poor, putting CL at risk of key leakage.

2) Key Organ-Key Organ (KA) is an integral part of the system. KA is responsible for most of the mathematical operations, including key generation, key update, etc. You are curious about the value of the plaintext, but you will not change it.

3) Cloud server: The cloud server (CS) is responsible for managing cloud storage. All transmitted data is under the control of CS. We assume that every COP is completely reliable.

4) Decryption server: The decryption server (DS) has powerful computing capabilities. Perform and isolate most (but not all) decryption tasks. We assume that the DS is partially trusted, and the DS access channel is insecure, because this is enough for CKMCPABE to keep the data secure, as shown in Section IV.

5) Data Owner: Data Owner (DO) is an authorized user who owns the data to be downloaded in the system. DO defines its own explicit access strategy so that only required CLs can receive plain text.

We assume that all parties involved in the data exchange will not cooperate with each other for illegal access to the data, otherwise the model will be unavailable and meaningless. In our model, the attributes are verified by KA. All the provided attributes are represented by a set of random elements contained in the common parameters generated by KA and CS in cooperation. Public parameters. When DO intends to share data, it uses the sent parameters to encrypt the data to form the original CT-Init ciphertext and load it into KA. KA re-encrypts the original ciphertext, forms the final CT ciphertext, sends it and stores it in CS. According to CL s attribute set, $S=\{\theta_1,\theta_2,\theta_3,\ldots..\}$, the key management protocol helps to secretly generate three different private key components at the same time, namely CPK_1, CPK_2 and CPK_3, each of which belongs to KA One, CS or CL. After requesting the data stored in the cloud, DS received CPK_1 and CPK_2s through the latest CT-to-CT′ form. Finally, CL extracts the plain text M and its CPK_3 from CT′. For our proposed CKMCPABE for cloud data exchange, the plaintext can only be extracted from the final ciphertext by combining the three components of the private key.

V. MAIN KEY MANAGEMENT ALGORITHM

For clarity, we provide some notation from : Define the universe of all system CLs as $U=\{u_1,u_2,u_3,\ldots,u_{n}.\}$and the universe of all granted attributes as $L=\{\theta_1,\theta_2,\theta_3,\ldots..\}$. Let $G_t \subseteq U$ be an attribute group of CLs who share θ_t.

Let $K_t \in Z_P^*$ be an attribute group key associated with G_t. Let $G=\{ G_t \mid \forall\ \theta_t \in L \}$be the collection of all attribute groups and $K=\{ K_t \mid \forall\ \theta_t \in L \}$ be the collection of all attribute group keys. Our CKM-CP-ABE scheme consists of six algorithms whose details follow.

A. Setup

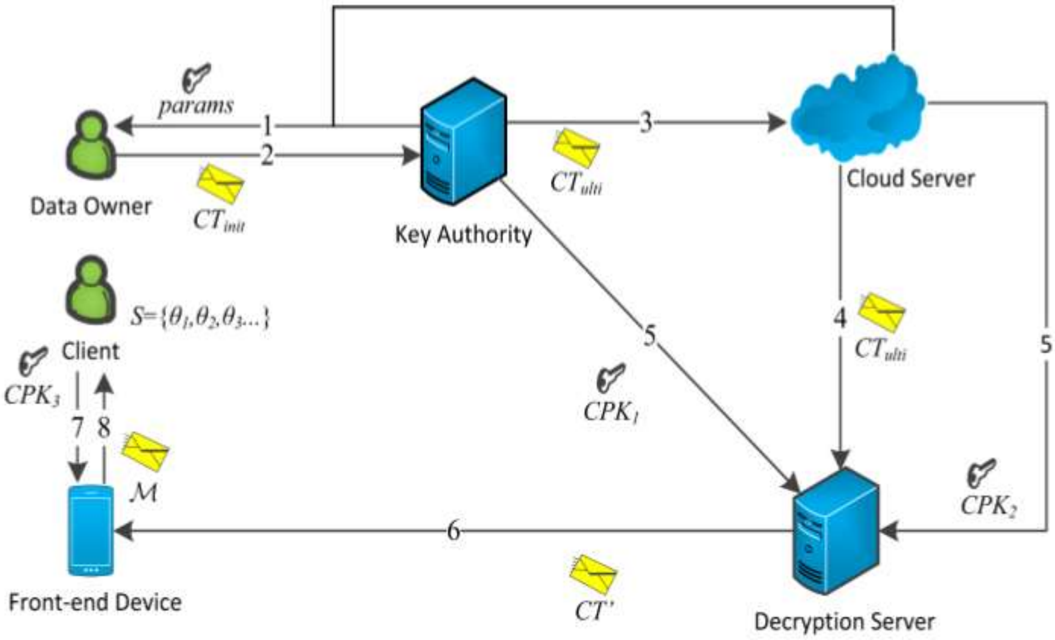
1) Trust Setup L

2) KA Setup:

3) CS Setup:

B. Key Generation

The key generation algorithm executed by KA takes the public parameter PP and a set of CL attributes as input. Then the spacecraft chooses a stochastic index $\tau \in_R Z^*_p$. Then the key generation algorithm generates the initial key value as follows: $PK_{init} =(g^{\tau}, \forall x \in S : h^{\tau}_{x})$ KA stores the initial key for subsequent update of the private key.



C.Encryption

The encryption algorithem,which is run by the Do, takes as input the public parameters params=(PP,PP_{ka},PP_{cs}), an access structure A and plaintext M . Meanwhile, the DO generates a random vector v={s,y_2,y_3,……y_d} and a linear secret sharing scheme (M,ρ)associated with A to calculate λ_i= v.M_i for each $u_i \in$ U . Then, the DO outputs the initial ciphertext:
CT_{init} =((M,ρ),C=M.e(g,g)^{as} ,
C=g^s $\forall\theta_t \in$ A:C_t^*=$g^{q\lambda_i}h^{-s}{}_{\rho(tj)}$)

D. Re-encryption

The re-encryption algorithm, which is run by the CS, takes as input the public parameters params =(PP ,PP_{ka}, PPcs) , the initial ciphertext CT $_{init}$, and the collection of attribute groups G . We adopt the attribute group-based algorithm of to re-encrypt the initial ciphertext. On receiving and CT $_{init}$, G , the CS selects two random exponents $\mu,\gamma,\epsilon R$ ZP ∗ and generates a set of attribute group keys K . Each CL $u_k \in$ G has a unique constant identity code, labeled ID_k $\epsilon\{0,1\}$ regardless of attribute set changes. Subsequently, CS .

E. Private Key Update

The private key update algorithm, which is the principal innovation of CKM-CP-ABE, takes as input the parameters) params =(PP ,PP_{ka}, PPcs) and the initial key CT_{init} PK . For this algorithm, the collaborative key management protocol is implemented to generate and distribute three different private key components. The protocol consists of two sub-protocols. The flows of the first sub-protocol are presented in Fig. 2.

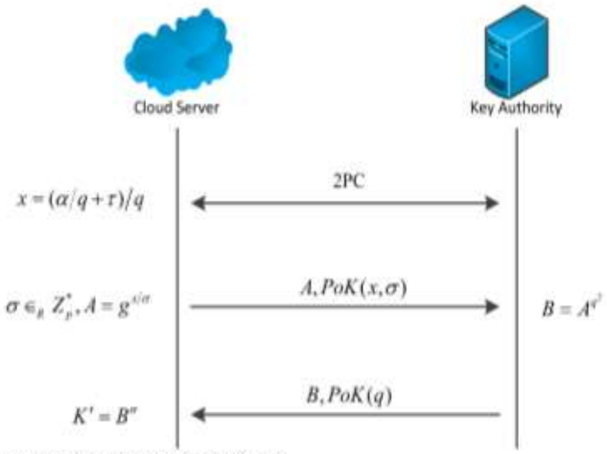


Fig. 2. The first sub-protocol flows.

DOI 10.17148/IJARCCE.2021.10719

Theorem 1. *The above computation for generating private key component* $g^{a+q\tau}$ *is a secure protocol, assuming the underlying arithmetic 2PC and zero knowledge proofs are secure.*

Proof. We provide the proof of theorem 1 in Section VI.
Note that after the above process, the CS holds $K`$ and the KA possesses the initial key *init PK* . These two facts are crucial to the proposed second sub-protocol that involves the KA, CS, and CL. We present flows of the second sub-protocol..
As depicted, the computation is as follows:

1) The CS randomly selects $\mu\,1, \gamma 1, \epsilon R$ ZP $*$
as its private input,
and the KA randomly selects $\mu\,1, \gamma 1, \epsilon R$ ZP $*$
as its private
input.

2) The protocol returns private output $y = (r_1 + r_2)\pi_1\pi_2$ to both KA and CS.

3) The CS selects random exponent $\varepsilon \epsilon {}_R Z_p^*$ and returns $X_{1=}(K`)^{y/}\varepsilon$ to the KA.

4) The KA returns Y_1
to the CS.

5) The CS outputs
$K = Y_1 / \pi^2{}_2$ as its component of the private
key.

6) The KA selects random exponent $\zeta \in {}_R Z_p{}^*$
and returns
$X_2 = (g^\tau)^{y/\zeta}$ and $\forall x \in S$: $Xx = (h^\tau{}_x)^{y/\zeta}$
to the CS.

7) The CS returns $Y_2 = X^{1/}\pi^2{}_1$
and $\forall x \in S$:$Y_x = X_x{}^{1/}\pi\,2|1$
to the
KA.

8) The KA outputs
$D_2 = Y_2\,\zeta/^2\pi_2$
and $\forall x \in S$: $D_2 = Y_2\,\zeta/^2\pi_2$.

9) The protocol secretly sends $(r_1 + r_2)/\pi_1\pi_2$ to the CL as its component of the private key.

Thus, the components of the private key are
As shown above, CPK_1 and CPK_2 are, respectively, kept by the KA and CS, while CPK_3 is held by the CL. All private key component generation and transmission is executed under the collaborative key management protocol. Neither KA nor CL can decrypt ciphertext on their own due to the proposed collaborative key management protocol. Thus, not only does our scheme avoid the key escrow problem, but it also helps prevent the key exposure.

A. Security Requirements

1) Data confidentiality

For the CKMCPABE model, we assume that all entities participating in data exchange are semi-verified, but will not conflict with each other for illegal access to data. Since KA and CS are semi-trusted but interested in plain text, shared data should be kept secret from them. And unauthorized CL. In our scheme, KA receives the original ciphertext from DO and generates the initial key corresponding to the CL attribute set, similar to what the key generation center in and should do. However, KA cannot use the initial key to decrypt the initial ciphertext, because CS stores important secrets that other CAs do not know, so the CA itself cannot obtain the plaintext. It also cannot extract the ciphertext data itself. DS converts the final ciphertext into a semi-hard ciphertext with a constant size and shorter than the final ciphertext. During the conversion process, DS cannot obtain any data information, so even if the DS is semi-trusted, the DS access channel is insecure and data security is guaranteed.In addition, it is almost guaranteed that external visitors cannot access

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10719

plain text. Any CL without authorized attributes will not be able to obtain the appropriate attribute group keys to obtain 0T and 1T. Therefore, such a CL cannot receive the value in sec. Therefore, decryption using less than three different private key components will not work properly.

2) Backward and forward secrecy

The first ABE key **lock** mechanism was proposed by Betencourt et al. **It is recommended that... the timing** mechanism **allocates verification** time **for** each attribute. **However, it** is impractical to **assign an** exact expiration time to each attribute of each CL. **The synchronization** mechanism **does not** guarantee **confidentiality back** and **forth** . **Green** et al. **Mainly focused** on decryption **efficiency, without considering the design of the callback** block. **Based** on **the method of Bethencourt** et al. Hur and Xiong et al. **introduced** an attribute group mechanism **for instant retrieval. For** CKMCPABE, we **propose** a scheme inspired by to **ensure forward and** backward **confidentiality. As soon as** CL **receives** a new attribute, KA **will update** the corresponding attribute group key. **, Embed** the newly **reconstructed main message** in the ciphertext. **Finally,** all CLs in the new attribute group obtain **the new** private key component.Even if **the** CL **has an encrypted ciphertext** before **receiving** this attribute, **the CL cannot decrypt it correctly. This will ensure** backward secrecy. **If** some CL **loses** an attribute, KA will **immediately update** the attribute group list. **The** cloud server immediately updates the **key set corresponding to the** attribute **group, and re-encrypts** the **cipher text by re-creating** the **main message. The shared key** management protocol **then distributes the updated** private key component. **This** attribute **cannot** extract information from the cipher **text in any way. Therefore,** forward secrecy can be guaranteed.

D. Security Proof

1) Proof of Theorem 1

We demonstrate the security of the first sub-protocol by examining the cases of a corrupt CS and corrupt KA separately. For a corrupt CS, the simulator ka Sim proceeds as follows: The arithmetic 2PC extracts a from CS in preparation for obtaining x x=(a/q+τ)/q . Then, a random x $\in_R$ $Z^*{}_P$is selected and sent to the arithmetic 2PC simulator. Note that it is correctly distributed, since there exists an q such that x=(a/q+τ)/q for any x, a , and τ . Then, the random x is transmitted to the adversary. Next, ka Sim receives A from the corrupt CS with the corresponding zero knowledge proof. After that, we extract σ from the proof system with the extractor and send a random $B \in_R G_1$ to the corrupt CS. Finally, a is sent to the trusted party for computation of K`, which is returned to the KA.

Now consider a hybrid simulator ka Hyb_{ka} that takes as input q and τ from the KA. It computes x x=(a/q+τ)/q using the arithmetic 2PC simulator as described above. *If* (a/q+τ)/q corresponding to a is correctly computed, the 2PC simulator runs as the real 2PC does. In the real protocol, q and σ are randomly picked so that x is uniformly distributed over Z_p^* and B is uniformly distributed over G_1 . Thus, the x and B generated in ka Sim are distributed identically to those in ka Hyb . Assuming security of the proof of knowledge is guaranteed, H yb should be indistinguishable from ka Sim .

For a corrupt KA, the simulator cs Sim proceeds as follows: The arithmetic 2PC simulator calculates (a/q+τ)/q and extracts τ . Then, a random value $A \in_R G_1$ is chosen and sent to the corrupt KA. When cs Sim receives B from the adversary, the arithmetic 2PC extracts and sends q to the trusted party, which computes k`=$g^{q+q}\tau$ . Finally, K` is transferred to the CS as private output. Now consider a hybrid simulator cs Hyb that takes as input the secret . It first runs the arithmetic 2PC simulator to compute x. If this protocol provides the correct output value of q and σ , the simulator executes all steps as in the real protocol. This is totally indistinguishable from the real process aided by arithmetic 2PC security. Regardless of whether the real protocol or the simulator is used, A is distributed uniformly over 1 since a is chosen at random. Thus, the values of x and A generated by cs Sim are distributed identically to those in cs Hyb . Assuming the proof of knowledge is secure, cs Hyb should be indistinguishable from cs Sim . Thus, we conclude that our first sub-protocol is secure.

VII. CONCLUSION AND FUTURE WORK

Encryption based on ciphertext policy attributes is a promising encryption method for performing fine-grained access control to protect cloud storage. The generation of distributed keys and the issuance and storage of private keys do not require any additional physical infrastructure. We introduce attribute groups to create private key update algorithms for detailed and immediate attribute revocation. The proposed collaboration mechanism not only perfectly solves the key storage problem, but also solves the more serious key leakage problem that has been seldom paid attention to in previous studies. At the same time, it helps to optimize the user experience because they are almost not responsible for decryption. Therefore, the proposed solution works best in a cloud-based communication system that serves the specific performance of front-end equipment in terms of security or efficiency. Our future work will be based on the preliminary results of this work and develop the proposed scheme by reducing the size of the ciphertext. , Encryption cost and decryption cost, these are still open issues that hinder the practical application of attribute data. Considering some specific industry

DOI 10.17148/IJARCCE.2021.10719

scenarios, such as B. Clinics that control access to personal history, it is also necessary to improve the information value of the access strategy.

REFERENCES

[1] A. Sahai and B. Waters, "Encryption based on fuzzy identity", Proc. EuroCrypt, 2005, S. 457473.
[2] J. Bethencourt, A. Sahai and B. Waters, "Encryption based on ciphertext strategy" attribute, "in Proc.IEEE Symp.Secur.Privacy, 2007, S.321334.
[3] H.Attpadung and H. Imai, "Attribute-based encryption and joint diffusion", in Proc.Int.Conf. PairingBased Cryptography, 2009, S. 248265.
[4] B. Waters, "Attribute-based encryption ciphertext strategy: implementation", in Proc. Public Key Cryptography, 2011, p. 5370.
[5] M. Green, S. Honberger and B. Waters, "Outsourcing ABE ciphertext decryption", collection works.USENIX-Sicher. Symp., 2011, S. 34.
[6] J. Lai, RHDeng, C. Guan and J. Weng, "Property encryption with verifiable outsourcing decryption", IEEE Trans.Inf.Forens.Security, vol. .. 8. Ning. 8, S. 13431354, 2013.
[7] S.Lin, R. Zhang, H. Ma, and M. Wang, "Overview of attribute-based encryption with verifiable out-of-band decryption", IEEE Trans.Inf.Forens.Security, Vol. 10. No. 10, p. 21192130, 2015.
[8] M. Chase, S.S.M.Chow, "Enhancing Confidentiality and Security with Multi-Authority Attribute-Based Encryption", in Proc. ACM CCS, 2009, 121130.
[9] G. Zhang, L. Liu und Yu. Liu, "The security of attribute-based encryption schemes. gegen bösartige KGC, "at Proc.TRUSTCOM, 2012, S. 13761380.
[10] J. Hoore, "Improving the Security and Efficiency of Attribute-Based Data Exchange," IEEE Trans.Knowl.Data.Eng., Vol. 2, No. 25, nein. 10, S. 22712282, 2013.
[11] S.P. Chandar, D. Mutkurman, and M. Ratinrai, "Proxy re-encryption access control based on hierarchical attributes in cloud computing", Proc.ICCPCT, 2014, p. 15651570.
[12] XA Wang, J. Ma and F. Xhafa, "Energy Efficient Attribute-Based Encryption, Outsourced Decryption", Proc. 3PGCIC, 2015, p. 444448.
[13] L. Chung and K.Newport, "Provide a secure ABE ciphertext strategy", Proc. ACM CCS, 2007, S. 456465.
[14] J. Hur und DKNoh, "Effective Revocation Attribute Access Control in Data Outsourcing System", IEEE Trans.Parallel Distrib., Bd.-Nr. 22, Ning. 7, S. 12141221, 2011.
[15] M. Pirretti, P. Trainor, P. McDaniel and B.Waters, "Attribute-Based Security System", Proc. ACM CCS, 2006, S. 99112.
[16] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-Based Encryption and Effective Revocation", in Proc. ACM CCS, 2008, S. 417426.
[17] A. Xiong, C. Xu and Q. Gan, "CPABE scheme for system attribute failure in cloud storage", in Proc.ICCWAMIP, 2014, S. 331335.
[18] In.Wu, "General Construction of Encryption Support for Attribute Revocation Based on Ciphertext Policy Attributes", China Communications, Vol. 11. No. 13, p. 93100, 2014.
[19] SSM Chow, "Remove Custody from Identity-Based Encryption", in Proc.Int.Conf. Practice and Theory of Public Key Cryptography, 2009, p.129. 256276.
[20] M. S. Ahmad, N. E.Musa, R. Nadaraja, R. Hassan and NEOtman, "Comparing Android and iOS from a Security Perspective", Proc.CITA, 2013, paragraph 14.
[21] V. Goyal, O. Pandy, A. Sakhai, and B Waters, "Attribute Encryption for Granular Access Control of Encrypted Data", in Proc.ACM CCS, 2006, S. 8998. [22] S. Rafaeli und D. Hutchison, "Overview of Key Management for Security Group Communication", ACM Computational Survey, Vol. 35, number. 3, S. 309329, 2003.
[23] S. Subashini and V. Cavita, "Overview of Security Issues in Cloud Service Delivery Models", Journal of Network and Computer Applications, Vol. 34, number. 1, S. 111, 2011.
[24] HTDinh, C. Lee, D. Niyato, and P. Wang, "Mobile Cloud Computing Overview: Architecture, Applications, and Methods", Wireless and Mobile Computing, Volume 1. 13. Ning. 18, S. 15871611.
[25] H.Gao Bi, JB.D. Joshi and JJ Ahn, "Security and Privacy Challenges in Cloud Computing Environments", IEEE Security and Privacy, Vol. 8. No. 6, p. 2431, 2010.
[26] C. Wang, S. S. M. Chohow, Q.Wang, K. Ren, and W. Lu, "Public Privacy Review of Secure Cloud Storage," IEEE Trans.Comput. 62, no. 2, p. 362375, 2013. [27] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing with encryption attributes", IEEE-Distributed each in parallel.Sister, T. 24, no. 1, p. 131143, 2013.
[28] Q. Liu, G. Wang and J. Wu, "A temporary proxy re-encryption scheme for secure communications in the cloud", Information Science, Vol. 258, p. 355370, 2014

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10719

[29] X. Yao, Z. Chen, and Y. Tian, "Lightweight attribute-based IoT encryption scheme", Future Generation Computer Systems, vol. 49, p. 104112, 2015.
[30] H. Hong, Z. Sun, "Attribute-based encryption scheme with high-performance key isolation without bilinear pairing", SpringerPlus, Volume 5, Issue 4. 1. Page 131, 2016.