International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 9 Issue 1 –JULY 2014.

Security and Privacy Challenges Mona in Cloud Computing Using Signature Generation

G. Mahalakshmi [#1] and S. Kumaravel [*2]

[#]*Research Scholar, Mahendra Arts and Science College, Kalipatti, Namakkal (Dt), Tamilnadu*

[*] *Head of the Department (CSE), Mahendra Arts and Science College, Kalipatti, Namakkal (Dt), Tamilnadu*

Abstract— Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in the cloud. In order to address this new problem and further to achieve a secure and dependable cloud storage service, in this paper a new cryptosystem is developed for fine-grained sharing of encrypted data that we call Cipher-Policy Attribute-Based Encryption (CP-ABE) with verifiable outsourced decryption. In our cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. This scheme also eliminates the overhead caused due to the large number of pairing operations in decryption. Attribute-based encryption (ABE) has been envisioned as a promising cryptographic primitive for realizing secure and flexible access control. So, in order to check the correctness of the transformation done and introduce the concept of verifiability. The proposed design consists of efficient methods that enable on-demand data correctness verification. ABE [Attribute based Encryption] extends the ASBE algorithm with a hierarchical structure to improve scalability and flexibility while at the same time inherits the features of fine-grained access control of ABE. A traditional Cloud security concept ensures the Identity based Cryptography, which results in securing outsourcing of cloud data.

Index terms- Access control, cipher-policy based Attribute-based encryption, outsourced decryption, verifiability.

I. INTRODUCTION

Cloud computing is a computing environment, where resources such as computing power, storage, network and software are abstracted and provided as services on the internet in a remotely accessible fashion. With the rapid development of the Internet and Cloud computing, there are more and more network resources. Sharing, management and on-demand allocation of network resources are particularly important in Cloud computing. The Cloud has become a new vehicle for delivering resources such as computing and storage to customers on demand. One of the main efficiency drawbacks of the most existing ABE schemes is that decryption is expensive for resource-limited devices due to pairing operations, and the number of pairing operations required to decrypt a cipher text grows with the complexity of the access policy. At the cost of security, only proven in a weak model (i.e., selective security), there exist several expressive ABE schemes [10], [11] where the decryption algorithm only requires a constant number of pairing computations. Recently, Green *et al.* [12] proposed a remedy to this problem by introducing the notion of ABE with outsourced decryption, which largely eliminates the decryption overhead for users. Based on the existing ABE schemes, Green *et al.* [12] also presented concrete ABE schemes with outsourced decryption. In these schemes (refer to Fig. 1 below), a user provides a un-trusted server, say a proxy operated by a cloud service provider, with a transformation key TK that allows the latter to translate any ABE cipher text CT satisfied by that user's attribute or access policy into a simple cipher text CT', and it only incurs a small overhead for the user to recover the plaintext from the transformed cipher text CT'. The security property of the ABE scheme with outsourced decryption guarantees that an adversary (including the malicious cloud server) be not able to learn anything about the encrypted message; however, the scheme provides no guarantee of the correctness of the transformation done by the cloud server. In the cloud computing setting, cloud service providers may have strong financial incentives to return incorrect answers, if such answers require less work and are un-likely to be detected by others.

Consider a cloud based electronic medical record system in which patients' medical records are protected using ABE schemes with outsourced decryption (e.g., [12]) and are stored in the cloud. In order to efficiently access patients' medical records on her mobile phone, a doctor generates and delegates a transformation key to a proxy in the cloud for outsourced decryption; Given a transformed cipher text from the proxy, the doctor can read a patient's medical record by just performing a simple step of computation. If no verification of the correctness of the transformation is guaranteed, however, the system might

run into the following two problems: 1) for the purpose of saving computing cost, the proxy could return a medical record transformed previously for the same doctor; 2) due to system malfunction or malicious attack, the proxy could send the medical record of another patient or a file of the correct form but carrying wrong information. The consequence of treating the patient based on incorrect information could be very serious or even catastrophic. With the out-sourced decryption, we shift this burdensome task of the mobile device to the proxy, which results in a significant reduction on computing cost for the mobile device. As a consequence, de-crypting the cipher text took approximately 180 milliseconds on the ARM-based device.

II. RELATED WORK

Sahai et al. proposed a scheme for ensuring data storage security in untrusted cloud [1]. Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. The straightforward and trivial way to support these operations is for users to download all the data from the cloud servers and re-compute the whole parity blocks as well as verification tokens. The user can always ask servers to send back blocks of the rows specified in the challenge and regenerate the correct blocks by erasure correction.

Lewko et al. Proposed a scheme for enabling public verifiability and storage dynamics for cloud computing [5]. They have proposed a general formal PoR model with public verifiability for cloud data storage, in which both block less and stateless verification. The challenge-response protocol can both determine the data correctness and locate possible errors. They employ authenticated skip list data structure to authenticate the tag information of challenges or updated blocks first. It provides integrity verification under different data storage systems, the problem of supporting both public verifiability and data dynamics has not been fully addressed. The verification algorithm accepts when interacting with the valid prover (e.g., the server returns a valid response) and it is sound if any cheating server that convinces the client it is storing the data file is actually stored that file.

V. Goyal et al. Developed a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE) [2]. In our cryptosystem, cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. It has limited applicability to access control of data, our primary motivation for this work. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level. We demonstrate the applicability of our construction to the sharing of audit-log information and broadcast encryption. Our construction supports delegation of private keys which subsumes Hierarchical Identity-Based Encryption (HIBE). As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites (i.e., giving another party your private key). The cryptosystem of Sahai and Waters allowed for decryption when at least k attributes overlapped between a cipher text and a private key.

Pairing delegation enables a client to outsource the computation of parings to another entity. However, the schemes proposed in [17], [18] still require the client to compute multiple exponentiations in the target group for every pairing it outsources. Most importantly, when using paring delegation in the decryption of ABE cipher texts, the amount of computation of the client is still proportional to the size of the access policy. Tsang et al. [19] consider batch pairing delegation. However, the scheme proposed in [19] can only handle batch delegation for pairings in which one of the points is a constant and it still requires the client to compute a pairing.

Proxy Re-encryption: In ABE with outsourced decryption, a user provides the cloud with a transformation key that allows the cloud to translate an ABE cipher text on message into a simple cipher text on the same, without learning anything about. This is reminiscent of the concept of proxy re-encryption [20], Proxy re-encryption allows a proxy, using a re-encryption key, to transform an encryption of under Alice's public key into an encryption of the same under Bob's public key without the proxy learning anything about the encrypted message. We emphasize that in the model of proxy re-encryption, verifiability of the proxy's transformation cannot be achieved. This can be briefly explained as follows. A proxy could replace the encryption of under Alice's public key with the encryption of another message under Alice's public key and then use its re-encryption key to transform the latter into an encryption of under Bob's public key. Obviously, without interaction with Alice, Bob cannot detect this malicious behavior of the proxy.

III. STEPS IN CIPHER TEXT-ATTRIBUTE BASED ENCRYPTION

A CP-ABE scheme consists of the following four algorithms for the process of attribute-based encryption.

A. Setup (λ, U) takes as input a security parameter λ and an attribute universal description U. It outputs the public parameter PK and master secret key MSK.

B. Keygen (PK, MSK, S) takes as input the public parameter PK, master secret key MSK, and a set of attributes S. It outputs a private key, SK.

C. Encrypt (PK, M, A) takes as input the public parameters PK, the message M, and an access structure A. It outputs a cipher text CT.

D. Decrypt (PK, SK, CT) takes as input the public parameter PK, private key SK for S and a cipher text CT. It outputs the message M.

In order to verify the correctness we have to check whether the following holds,

If a set of attributes S satisfies the attribute structure A, then M←Decrypt (PK, SK, CT).
Otherwise Decrypt (PK, MSK, CT) outputs the error message.

A CP-ABE scheme is a CCA-secure if all polynomial time adversaries have at most a negligible advantage in this security game.
CPA Security: We say that a CP-ABE scheme is *CPA-secure* (or secure against chosen-plaintext attacks) if the adversary cannot make decryption queries.
Selective Security: We say that a CP-ABE scheme is *selectively secure* if we add an Init stage before Setup where the adversary commits to the challenge access structure A.
In the original model defined in [12], a CP-ABE scheme with outsourced decryption consists of five algorithms: Setup, Encrypt, Keygen, Transform, and Decrypt. A trusted party uses the algorithm Setup to generate the public parameters and a master secret key, and uses Keygen, to generate a private key and a transformation key for a user. Taking as input the transformation key given by a user and a cipher text, the cloud can use the algorithm Transform to transform the cipher text into a simple ciphertext if the user's attribute satisfies the access structure associated with the cipher text; then the user uses the algorithm Decrypt to recover the plaintext from the transformed cipher text. Note that in the definition of Green *et al.* [12], the input to the algorithm Decrypt includes only the private key of the user and the transformed cipher text, but does not include the original cipher text. Because of this omission of the original cipher text, it is not possible to construct a CP-ABE scheme with verifiable outsourced decryption under the definition of [12]. This can be explained as follows. A malicious cloud could replace the cipher text, it supposes to transform with a cipher text of a different message, and then transforms the latter into a simple cipher text using its transformation key. Obviously, the user cannot detect this malicious behavior of the cloud since the input to the algorithm Decrypt does not include the original cipher text required to be transformed. In order to achieve verifiability, we need to modify the model of CP-ABE with outsourced decryption defined in [12]. We now formally describe our new model.

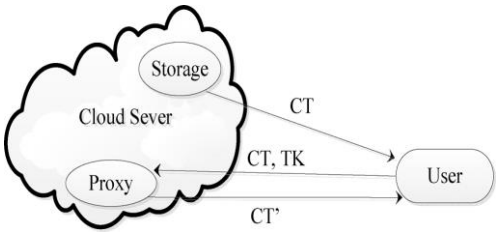

Fig 1. System model which constitutes an ABE system with verifiable outsourced decryption

In our new model, the algorithms and Setup, Encrypt and Decrypt constitute a traditional CP-ABE scheme. The input to the algorithm Decrypt includes the original cipher text and the transformed cipher text. In fact, in our concrete scheme, a user only needs to know a small part of the original cipher text to verify the correctness of the transformation done by the cloud in the algorithm Decrypt. In addition, in our model, using the algorithm GenTK, and his private key, the user generates the transformation key by himself, not by the trusted party as in [12]. Having either the trusted party or the user generate the transformation key does not have an effect on the security of the scheme. However, it is more flexible if we let the user himself generate the transformation key. On the other hand, if the trusted party is responsible for the generation of transformation keys, the user is required to reinitialize the system for outsourced decryption.
Now, we formally describe the security and verifiability requirements of a CP-ABE scheme with outsourced decryption. Informally, security ensures that an adversary (including a malicious cloud) is not able to learn anything about the encrypted message and verifiability allows a user to check on the correctness of the transformation done by the cloud.
Security. Since the traditional notion of security against adaptive chosen-cipher text attacks (CCA) does not allow any bit of the cipher text to be altered, similar to [12], we adopt a relaxation called repayable CCA (RCCA) security, which allows modifications to the cipher text provided they cannot change the underlying message in a meaningful way.

IV. PROPOSED SCHEME WITH VERIFIABLE OUTSOURCED DECRYPTION

Here we first propose a new CP-ABE scheme utilizing Waters' CP-ABE scheme [1], which is proven to be selectively CPA-secure. Then, based on the scheme, we

propose a CP-ABE scheme with outsourced decryption and prove that it is selectively CPA-secure and verifiable in the standard model.

Recently, the first CP-ABE scheme that achieves full security was proposed by Luke *et al.* [5]. Since the underlying structure of the CP-ABE scheme presented by Luke *et al.* [5] Is almost identical to the underlying Waters' CP-ABE scheme [4] we use, one can adapt our construction techniques to the CP-ABE scheme proposed in [5] to achieve fully secure (i.e., RCCA secure) CP-ABE scheme with verifiable outsourced decryption in the standard model. Before presenting our new CP-ABE scheme, we give some intuitions of our construction. Based on Waters' CP-ABE scheme [4], we add to the ciphertext the encryption of an extra random message and a checksum value, which is computed with this random message and the actual plaintext. We regard this checksum value as a commitment of the actual plaintext, which can be used to check if the transformation is done.

V. PERFORMANCE

In order to evaluate the performance of our CP-ABE scheme with verifiable outsourced decryption, we implement our scheme in software based on the libfenc library and using a 224-bit MNT elliptic curve from the Stanford Pairing-Based Crypto library. Although our implementation based the MNT curve implies the use of asymmetric pairing, only a small change need to be made on our scheme of symmetric setting in the implementation. Specifically, suppose that an asymmetric pairing takes elements from and as inputs.

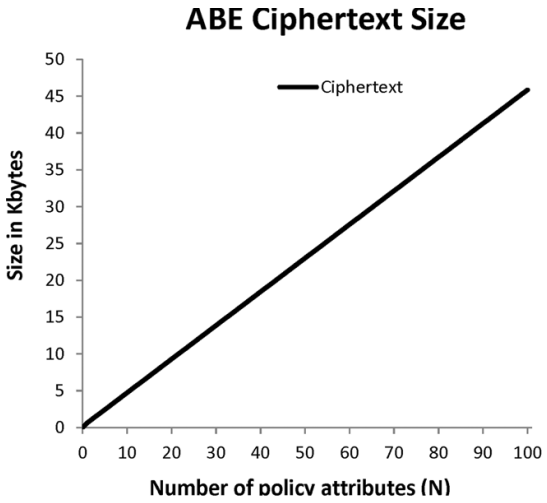

Fig 2. Comparison of the size of ABE Ciphertext in CP-ABE and CP-ABE with verifiable outsourced decryption scheme

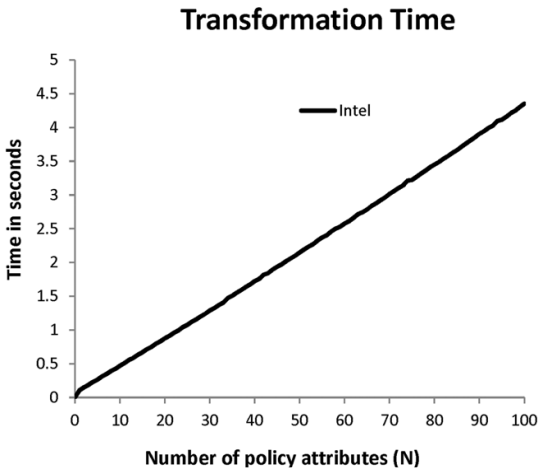

Fig 3. Comparison of transformation time in CP-ABE and CP-ABE with verifiable outsourced decryption.

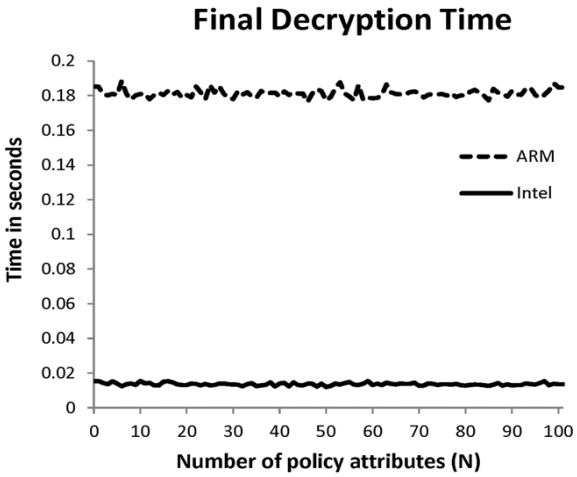

Fig 4. Comparison of decryption time in CP-ABE and CP-ABE with verifiable outsourced decryption.

Discussion: The ABE ciphertext size and decryption/transformation time, increase linearly as the ciphertext policy's complexity grows in fig 2. An encryption under a ciphertext policy with 100 attributes results in an ABE ciphertext of nearly 46 KB and it takes about 5 seconds for the Intel platform to decrypt this ciphertext. On the other hand, decryption time degrades considerably on the ARM platform: it requires more than 1 second to decrypt a ciphertext under a policy with one attribute, 5 seconds under a policy with ten attributes and almost 50 seconds under a policy with one hundred attributes. As expected, outsourcing substantially reduces the computation time required for devices with limited computing resource to recover the plaintext. The bulk of the decryption operation is now handled by the proxy. For

each cipher text policy, we repeat our experiment 100 times on the PC and 30 times on the ARM device and we take the average values as the experimental results.

VI. CONCLUSION

In distributed settings with untrusted servers, such as the cloud, many applications need mechanisms for complex access-control over encrypted data. In order to avoid the problem we designed an ABE system with outsourced decryption that largely eliminates the decryption overhead for user and also considered a new requirement of ABE with outsourced decryption which is verifiability. We focus on improving the efficiency of ABE by leveraging a previously overlooked fact, i.e., the often-found hierarchy relationships between the Access control in that are inherent in many Cloud Computing Scenarios. The transformation correctness were checked using the concept of verifiability. We also proposed a concrete ABE scheme with verifiable outsourced decryption and proved that it is secure and verifiable. To assess the practicability of our scheme, we implemented it and conducted experiments in a simulated out-sourcing environment. As expected, the scheme substantially reduced the computation time required for resource-limited de-vices to recover plaintexts.

VII. REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in Proc. EUROCRYPT, page (s): 457–473,2005.

[2] V. Goyal, O. Pandey, A. Sahai, B. Waters "Attribute-basedencryp-tion for fine-grained access control of encrypted data," in Proc. ACM Conf. Computer and Communications Security, page (s): 89–98, 2006.

[3] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in Proc. ACM Conf. Computer and Communications Security, page (s): 195–203,2007.

[4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. Public Key Cryptography, page (s): 53–70, 2011.

[5] A. B. Luke, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in Proc. EUROCRYPT, page(s): 62–91, 2010.

[6] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in Proc. CRYPTO, page(s): 191–208, 2010.

[7] A. B. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in Proc. EUROCRYPT, page(s): 547–567, 2011.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Proc. IEEE Symp. Security and Privacy, page(s): 321–334, 2007.

[9] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in Proc. ACM Conf. Computer and Communications Security, page(s): 456–465, 2007.

[10] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," Theor. Comput. SCI., vol. 422, page (s): 15–38, 2012.

[11] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in Proc. Public Key Cryptography, page (s): 162–179, 2012.

[12] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proc. ACM Conf. Computer and Communications Security, page (s): 62–73, 1997.

[13] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited (preliminary version)," in Proc. STOC, page (s): 209–218, 1998.

[14] K.-M. Chung, Y. T. Kalai, and S. P. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in Proc. CRYPTO, page (s): 483–501, 2010.

[15] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc. STOC, page(s): 169–178, 2009.

[16] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in Proc. EUROCRYPT, page(s): 129–148, 2011.

[17] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in Proc. TCC, page (s): 422–439, 2012.

[18] B. Chevallier-Mames, J.-S. Coron, N. McCullagh, D. Naccache, and M. Scott, "Secure delegation of elliptic-curve pairing," in Proc. CARDIS, page (s): 24–35, 2010.

[19] B. G. Kahn, M. S. Lee, and J. H. Park, "Efficient delegation of pairing computation," IACR Cryptology ePrint Archive, page (s): 259, 2005.

[20] P. P. Tsang, S. S. M. Chow, and S. W. Smith, "Batch pairing delegation," in Proc. IWSEC, page(s): 74–90, 2009.