Two-Factor Data Security Protection Mechanism for Cloud Storage System

CHILAKA HARI KRISHNA[1], DEVARAPALLI BHANUDHAR[2], BURAGA RAKESH[3]
[1,2,3] *Department of Computer Science & Engineering, Vasireddy Venkatadri Institute of Technology (VVIT), Guntur, Andhra Pradesh, India-522508*

Abstract -- Here, we propose a two-factor data security protection mechanism with factor revocability for cloud storage system. Our system allows a sender to send an encrypted message to a receiver through a cloud storage server. The sender only needs to know the identity of the receiver but no other information (such as its public key or its certificate). The receiver needs to possess two things in order to decrypt the cipher text. The first thing is his/her secret key stored in the computer. The second thing is a unique personal security device which connects to the computer. It is impossible to decrypt the cipher text without either piece. More importantly, once the security device is stolen or lost, this device is revoked. It cannot be used to decrypt any cipher text. This can be done by the cloud server which will immediately execute some algorithms to change the existing cipher text to be un-decrypt able by this device. This process is completely transparent to the sender. Furthermore, the cloud server cannot decrypt any cipher text at any time. The security and efficiency analysis show that our system is not only secure but also practical.

I. INTRODUCTION

CLOUD storage [10] is a model of net-worked storage system where data is stored in pools of storage which are generally hosted by third parties. There are many benefits to use cloud storage. The most notable is data accessibility. Data stored in the cloud can be accessed at any time from any place as long as there is network access. Storage maintenance tasks, such as purchasing additional storage capacity, can be offloaded to the responsibil-ity of a service provider. Another advantage of cloud storage is data sharing between users. If Alice wants to share a piece of data (e.g., a video) to Bob, it may be difficult for her to send it by email due to the size of data. Instead, Alice uploads the file to a cloud storage system so that Bob can download it at any time.

Despite its advantages, outsourcing data storage also increases the attack surface area at the same time. For example, when data is distributed, the more locations it is stored the higher risk it contains for unauthorized physical access to the data. By sharing storage and networks with many other users it is also possible for other unauthorized users to access your data. This may be due to mistaken actions, faulty equipment, or sometimes because of criminal intent. A promising solution to offset the risk is to deploy encryption technology. Encryption can pro-tect data as it is being transmitted to and from the cloud service. It can further protect data that is stored at the ser-vice provider. Even there is an unauthorized adversary who has gained access to the cloud, as the data has been encrypted, the adversary cannot get any information about the plaintext. Asymmetric encryption allows the encrypt or to use only the public information (e.g., public key or identity of the receiver) to generate a cipher text while the receiver uses his/her own secret key to decrypt. This is the most convenient mode of encryption for data transition, due to the elimination of key management existed in symmetric encryption. In a normal asymmetric encryption, there is a single secret key corresponding to a public key or an identity. The decryption of cipher text only requires this key. The key is usually stored inside either a personal computer or a trusted server, and may be protected by a password. The security protection is sufficient if the computer/server is isolated from an opening network. Unfortunately, this is not what happens in the real life. While being associated with the world through the Internet, the PC/server may experience the ill effects of a potential hazard that programmers may barge in into it to trade off the mystery key without letting the key proprietor know. In the physical security viewpoint, the PC putting away a client decoding key might be utilized by another client when the first PC client (i.e. the key proprietor) is away (e.g., when the client goes to latrine for some time without locking the machine). In an endeavor or school, the sharing use of PCs is likewise normal. For instance, in a school, an open PC in a copier room will be imparted to all understudies remaining at a similar floor. In these cases, the mystery key can be bargained by a few aggressors who can get to the casualty's close to home information put away in the cloud framework. In this way, there exists a need to improve the security assurance.

A similarity is e-keeping money security. Numerous e-managing an account applications require a client to utilize both a secret key and a security gadget (two variables) to login framework for cash exchange. The security gadget may show a one-time watchword to give the client a chance to type it into the framework, or it might be expected to associate with the PC (e.g., through USB or NFC). The motivation behind utilizing two elements is to improve the security assurance for the entrance control.

They will become more sensitive and important, as if the e-banking analogy. Actually, we have noticed that the concept of two-factor encryption, which is one of the encryption trends for data protection,[1] has been spread into some real-world applications, for example, full disk encryption with Ubuntu system, AT&T two factor encryption for Smart-phones,[2] electronic vaulting and druva—cloud-based data encryption.[3] However, these applications suffer from a potential risk about factor revocability that may limit their practicability. Note we will explain it later. A flexible and scalable two-factor encryption mechanism is really desirable in the era of cloud computing. That motivates our work.

II. RESEARCH ELABORATIONS

In this paper, we propose a novel two-factor security protection mechanism for data stored in the cloud. Our mechanism provides the following nice features:

1) Our system is an IBE (Identity-based encryption) based mechanism. That is, the sender only needs to know the identity of the receiver in order to send an encrypted data (cipher text) to him/her. No other information of the receiver (e.g., public key, certificate etc.) is required. Then the sender sends the cipher text to the cloud where the receiver can download it at any time.

2) Our system provides two-factor data encryption protection. In order to decrypt the data stored in the cloud, the user needs to possess two things. First, the user needs to have his/her secret key which is stored in the computer. Second, the user needs to have a unique personal security device which will be used to connect to the computer (e.g., USB, Bluetooth and NFC). It is impossible to decrypt the cipher text without either piece.

3) More importantly, our system, for the first time, provides security device (one of the factors) revocability. Once the security device is stolen or reported as lost, this device is revoked. That is, using this device can no longer decrypt any cipher text (corresponding to the user) in any circumstance. The cloud will immediately execute some algorithms to change the existing cipher text to begun-decryptableby this device. This process is completely transparent to the sender.

4) The cloud server cannot decrypt any cipher text at any time.

We provide an estimation of the running time of our prototype to show its practicality, using some benchmark results. We also note that although there exist some naive approaches that seem to achieve our goal, we have discussed in Section 1.1 that there are many limitations by each of them and thus we believe our mechanism is the first to achieve all the above mentioned features in the literature

III. CONSTRUCTION

We have two diverse encryption innovations: one is IBE and the other is conventional Open Key Encryption (PKE). At first we enable a client to produce at first level figure message under a collector's personality. The fisrt level figure content will be additionally changed into a moment level figure content comparing to a security gadget. The subsequent figure content can be unscrambled by a legitimate collector with mystery key and security gadget. Here, one may question that our development is an insignificant and clear blend of two distinct encryptions. Shockingly, this isn't valid because of the way that we have to additionally bolster security gadget revocability. A unimportant mix of IBE and PKE can't accomplish our objective. To help revocability, we utilize re-encryption innovation with the end goal that the piece of figure content for an old security gadget can be refreshed for another gadget if the old gadget is denied. Then, we have to produce an extraordinary key for the above figure content transformation. By getting to the exceptional key, the old figure content and the refreshed figure message, the cloud server can't accomplish any learning of message. We additionally utilize hash-signature technique to "sign" figure content with the end goal that once a segment of figure content is tempered by enemy, the cloud and cipher text collector can tell. From the above introductions, we can see that our two-factor assurance framework with security gadget revocability can't be acquired by inconsequentially joining an IBE with a PKE. We present the system description as follows.

1) Setup phase: the setup phase generates all public parameters and master secret key used throughout the execution of system.

2). The SDI finally delivers the security device to a user ID.

3) First-level cipher text generation phase: a data sender encrypts a data under the identity of a data receiver, and further sends the encrypted data to the cloud server.

4) Second-level cipher text phase: after receiving the first level cipher text of a data from the data sender, the cloud server generates the second-level cipher text

5) Device updated phase: Once a device of a user needs to be updated due to some incidences (e.g., it is either lost or stolen), the user first reports the issue to the SDI. The SDI then issues a new device for the user.

6) Cipher text updated phase: The SDI notifies the cloud server to update the cipher text of the user by sending a special piece of information.

7) Data recovery phase. A data receiver uses a decryption key and a device to recover the data as follows.

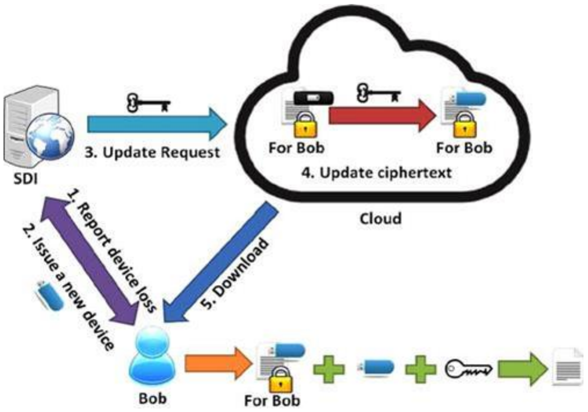


Fig 1: Update cipher text after issuing a new security device.

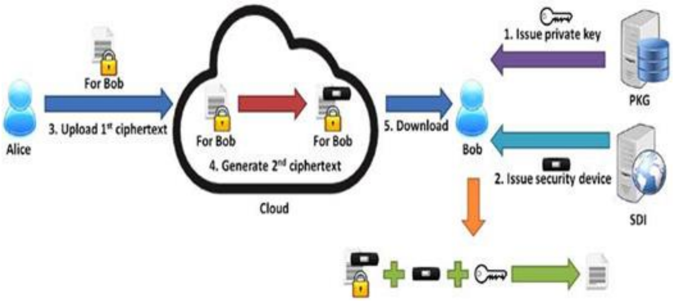


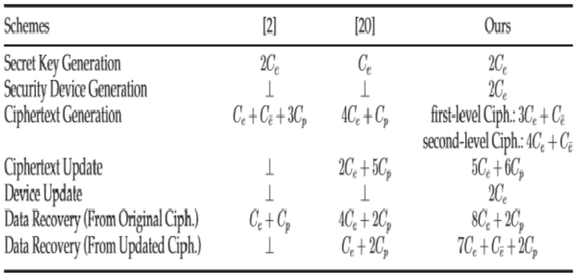
Fig 2: Ordinary data sharing.

IV. SYSTEM EVALUATION

4.1 Security Analysis

We separate two security levels for our scheme: one is allowing an adversary to achieve the secret key of user but not the corresponding secure device, and the other is the reversed case. For Type-I Security. Here we allow an adversary to obtain the secret key of a user but not the corresponding security device. We analyze the security of our scheme under the model of Type-I. Practical analysis: An adversary A now is given the secret key skIDi of user IDi. We show thatAcannot recover the underlying message by only leveraging knowledge of skIDi as follows.

Table 1 Computer Comparison



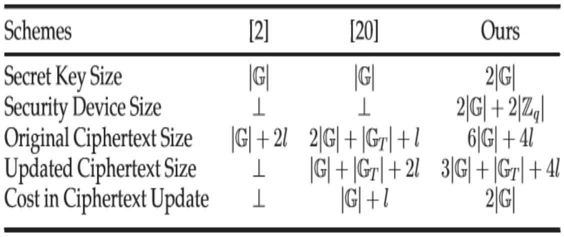
Schemes	[2]	[20]	Ours
Secret Key Generation	$2C_e$	C_e	$2C_e$
Security Device Generation	$\perp$	$\perp$	$2C_e$
Ciphertext Generation	$C_e + C_{\bar{e}} + 3C_p$	$4C_e + C_p$	first-level Ciph.: $3C_e + C_{\bar{e}}$ second-level Ciph.: $4C_e + C_{\bar{e}}$
Ciphertext Update	$\perp$	$2C_e + 5C_p$	$5C_e + 6C_p$
Device Update	$\perp$	$\perp$	$2C_e$
Data Recovery (From Original Ciph.)	$C_e + C_p$	$4C_e + 2C_p$	$8C_e + 2C_p$
Data Recovery (From Updated Ciph.)	$\perp$	$C_e + 2C_p$	$7C_e + C_{\bar{e}} + 2C_p$

4.2 Efficiency Analysis

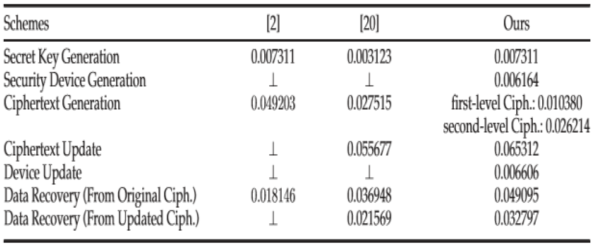
We analyze the efficiency of our mechanism as well as its comparison with [2] (the most efficient two-secret protection system but no revocability) and [20] (the most efficient single secret system with revocability) in terms of computational and communicational cost. We present the theoretical comparison in Tables 2 and 3 for computation and communication complexity, respectively. From Table 2, it can be seen that our system requires additional computation cost in security device generation and update, whereas others do not need any cost. This is because ours supports security device revocability. In cipher text generation, our system does not require any pairings operation, and it is worth of mentioning that the second level cipher text generation cost can be offloaded to a cloud server. Compared to [20] for other metrics, our system only requires slight extra cost; while we just need an additional pairing in cipher text update. A similar phenomenon does exist in Table 3 in the sense that our system needs extra communication cost in delivery of security device. Except for this, our communication complexity is much closer to that of others.

V. CONCLUSION

In this paper, we presented a novel two-factor information security assurance instrument for distributed storage framework, in which an information sender is permitted to encode the information with learning of the personality of a collector just, while the beneficiary is required to utilize the two his/her mystery key and a security gadget to access the information. Our answer upgrades the privacy of the information, as well as offers the revocability of the gadget so once the gadget is repudiated, the relating figure content will be refreshed consequently by the cloud server with no notice of the information proprietor. Besides, we displayed the security confirmation and productivity examination for our framework.



Schemes	[2]	[20]	Ours
Secret Key Size	$\lvert\mathbb{G}\rvert$	$\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert$
Security Device Size	$\perp$	$\perp$	$2\lvert\mathbb{G}\rvert + 2\lvert\mathbb{Z}_q\rvert$
Original Ciphertext Size	$\lvert\mathbb{G}\rvert + 2l$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert + l$	$6\lvert\mathbb{G}\rvert + 4l$
Updated Ciphertext Size	$\perp$	$\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert + 2l$	$3\lvert\mathbb{G}\rvert + \lvert\mathbb{G}_T\rvert + 4l$
Cost in Ciphertext Update	$\perp$	$\lvert\mathbb{G}\rvert + l$	$2\lvert\mathbb{G}\rvert$



Schemes	[2]	[20]	Ours
Secret Key Generation	0.007311	0.003123	0.007311
Security Device Generation	$\perp$	$\perp$	0.006164
Ciphertext Generation	0.049203	0.027515	first-level Ciph.: 0.010380 second-level Ciph.: 0.026214
Ciphertext Update	$\perp$	0.055677	0.065312
Device Update	$\perp$	$\perp$	0.006606
Data Recovery (From Original Ciph.)	0.018146	0.036948	0.049095
Data Recovery (From Updated Ciph.)	$\perp$	0.021569	0.032797

TABLE 4 Computation Comparison (Running Time in Second) II

From Table 4, we see that our running time is nearly the same as that of [20], and meanwhile, our system outperforms [20] and [2] in encryption. In the communication cost, our scheme suffers from the largest price in “Updated Cipher text Size” due to a reason that the scheme outputs a pairing in the update phase. However, we state that the price is only an approximately 50 percent increase from that of [20] in the same metric, which is an acceptable increment.

REFERENCES

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, “Simultaneous hardcore bits and cryptography against memory attacks,” in Proc. 6th Theory Cryptography Conf., 2009, pp. 474–495.

[2] S. S. Al-Riyami and K. G. Paterson, “Certificateless public key cryptography,” in Proc. 9th Int. Conf. Theory Appl. Cryptol., 2003, pp. 452–473.

[3] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen, “Certificate based (linkable) ring signature,” in Proc. Inf. Security Practice Experience Conf., 2007, pp. 79–92.

[4] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang, “Malicious KGC attacks in certificateless cryptography,” in Proc. 2nd ACM Symp. Inf., Comput. Commun. Security, 2007, pp. 302–311.

[5] M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxy cryptography,” in Proc. Int. Conf. Theory Appl. Cryp-tographic Techn., 1998, pp. 127–144.

[6] A. Boldyreva, V. Goyal, and V. Kumar, “Identity-based encryp-tion with efficient revocation,” in Proc. ACM Conf. Comput. Com-mun. Security, 2008, pp. 417–426.

[7] D. Boneh, X. Ding, and G. Tsudik, “Fine-grained control of secu-rity capabilities,” ACM Trans. Internet Techn., vol. 4, no. 1, pp. 60– 82, 2004.

[8] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” in Proc. 21st Annu. Int. Cryptol. Conf., 2001, pp. 213– 229.

[9] R. Canetti and S. Hohenberger, “Chosen-ciphertext secure proxy re-encryption,” in Proc. ACM Conf. Comput. Commun. Security, 2007, pp. 185–194.

[10] H. C. H. Chen, Y. Hu, P. P. C. Lee, and Y. Tang, “NCCloud: A net-work-coding-based storage system in a cloud-of-clouds,” IEEE Trans. Comput., vol. 63, no. 1, pp. 31–44, Jan. 2014.

[11] S. S. M. Chow, C. Boyd, and J. M. G. Nieto, “Security-mediated certificateless cryptography,” in Proc. 9th Int. Conf. Theory Practice Public-Key Cryptography, 2006, pp. 508–524.

[12] C.-K. Chu, S. S. M. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, “Key-aggregate cryptosystem for scalable data sharing in cloud storage,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 468– 477, Feb. 2014.

[13] C.-K. Chu and W.-G. Tzeng, “Identity-based proxy re-encryption without random oracles,” in Proc. 10th Int. Con. Inf. Security, 2007, pp. 189–202.

[14] R. Cramer and V. Shoup, “Design and analysis of practical publickey encryption schemes secure against adaptive chosen ciphertext attack,” SIAM J. Comput., vol. 33, no. 1, pp. 167–226, Jan. 2004.

[15] Y. Dodis, Y. T. Kalai, and S. Lovett, “On cryptography with auxiliary input,” in Proc. 41st Annu. ACM Symp. Theory Comput., 2009, pp. 621–630.

[16] Y. Dodis, J. Katz, S. Xu, and M. Yung, “Key-insulated public key cryptosystems,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., 2002, pp. 65–82.

[17] Y. Dodis, J. Katz, S. Xu, and M. Yung, “Strong key-insulated signature schemes,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., 2003, pp. 130–144.

[18] L. Ferretti, M. Colajanni, and M. Marchetti, “Distributed, concurrent, and independent access to encrypted cloud databases,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 437–446, Feb. 2014.

[19] C. Gentry, “Certificate-based encryption and the certificate revocation problem,” in Proc. Int. Conf. Theory Appl. Cryptographic Techn., 2003, pp. 272–293.

[20] M. Green and G. Ateniese, “Identity-based proxy re-encryption,” in Proc. 5th Int. Conf. Appl. Cryptography Netw. Security, 2007, pp. 288–306.