Security Capabilities Of Fine Grained Two Factor Access Control In Web Based Cloud Computing Services

Ganesh[1], Asha[2]

[1] *M.Tech Student, Department of Computer Science & Engineering, Dr. Ambedkar institute of Technology,Bangalore-560056*
[2] *Associate Professor, Department of Computer Science & Engineering, Dr. Ambedkar institute of Technology,Bangalore-560056*

---***---

Abstract – *For Web Based Cloud Computing Services We Introduced Fine Grained Two Factor Access Control .The Basic Concept Behind The Fine Grained Two Factor Access Control Is Getting The Permission from Two Parties In This Case We Consider Two Parties As User Secret Key And Light Weight Device. In Two Factor Access Control System an Attribute Based Control Mechanism Is Implemented From The Help Of Uesr Secret Key And Lightweight Security Device. User Must Satisfied With This Two For Getting Access To System. If Any One Fails User Can't Get The Access To The System. The Access Control System Denies The Access Of The User To The System If Multiple User Have Same Attribute set value.*

***Key Words*: Security, Cloud computing, , Performance analysis, Access Control,SEM**

1.INTRODUCTION

Cloud computing may be a virtual host system that enables enterprises to purchase for, lease, sell, or distribute software associated different digital resources over the web as an on demand service. It now not depends on a server or variety of machines that physically exist, because it may be a virtual system. There area unit several applications of cloud computing, like information sharing information storage big information management , medical data system etc. End users access cloud-based applications through a wb browser, skinny consumer or mobile app whereas the business package code and user's information area unit keep on servers at an overseas location.

The benefits of web-based cloud computing services area unit huge, that add the convenience of accessibility, reduced prices and capital expenditures, enhanced operational efficiencies, scalability, flexibility and immediate time to promote. Though the new paradigm of cloud computing provides great benefits, there area unit meantime additionally issues regarding security and privacy particularly for web-based cloud services. As sensitive information is also keep within the cloud for sharing purpose or convenient access; and eligible users may additionally access the cloud system for numerous applications and services, user authentication has become a essential element for any cloud system. A user is needed to login before working the cloud services or accessing the sensitive information keep within the cloud. There are a unit 2 issues for the normal account/password based system. First, the normal account/password-based authentication isn't privacy-preserving. However, it is well acknowledged that privacy is a necessary feature that has to be thought-about in cloud computing systems. Second, it is common to share a laptop among completely different individuals. It may be easy for hackers to put in some spyware to find out the login password from the web-browser.

A recently planned access control model known as attribute-based access management may be a smart candidate to tackle the primary drawback. It not solely provides anonymous authentication however conjointly additional defines access control policies supported completely different attributes of the requester, environment, or the information object. In associate attribute-based access control system,1 every user includes a user secret key issued by the authority. In apply, the user secret secret is keep within the personal laptop. after we contemplate the higher than mentioned second drawback on web-based services, it's common that computers is also shared by several users particularly in some large enterprises or organizations. for instance, allow us to think about the following 2 scenarios:
1)during a hospital, computers area unit shared by completely different workers. Dr. Alice uses the pc in area A once she is on duty within the daytime, while Dr. Bob uses a similar computer within the same area once he's on duty at nighttime.2)in a very university, computers within the college man research laboratory area unit usually shared by completely different students.

Consider an organization – industrial, government, or military – wherever all staff (referred to as users) have bound authorizations. we tend to assume that a Public Key Infrastructure (PKI) is offered and every one users have digital signature, as well as en/de-cryption, capabilities. within the course of performing arts routine everyday tasks, users cash in of secure applications, like email, file transfer, remote log-in and internet browsing. Now suppose that a trusty user (Alice) will one thing that warrants immediate revocation of her security privileges. for instance, Alice could be dismissed, or she may suspect that her personal key has been compromised. Ideally, straightaway following revocation, the key holder, either Alice herself or associate degree assailant, ought to be unable to perform any security operations and use any secure applications. Specifically, this might mean:

– The key holder cannot scan any secure email. This includes encrypted email that already resides on Alice's email server (or native host) and potential future email erroneously encrypted for Alice. though encrypted email could also be delivered to Alice's email server, the key holder ought to be unable to rewrite it.

– The key holder cannot generate valid digital signatures on any longer messages. However, signatures generated by Alice before revocation might have to stay valid.

– The key holder cannot evidence itself to company servers (and alternative users) as a legitimate user. Throughout the paper, we have a tendency to use email as associate degree example application. whereas it's a popular mechanism for all-purpose communication, our explanation conjointly applies to alternative secure suggests that of knowledge exchange.

To provide immediate revocation it's natural to initial think about ancient revocation techniques. several revocation ways are proposed; they'll be roughly classified into 2 outstanding types: 1) express revocation structures such as Certificate Revocation Lists (CRLs) and variations on the theme, and 2) real time revocation checking like the web Certificate standing Protocol (OCSP) and its variants. In each cases, some trusty entities ar ultimately responsible of verificatory user certificates. However, the higher than necessities for immediate revocation are not possible to satisfy with existing techniques. this can be primarily as a result of they do not give fine-grained enough management over users' security capabilities. Supporting immediate revocation with existing revocation techniques would result in significant performance value and extremely poor measurability, as mentioned in Section eight.As since every revocation technique exhibits a singular set of pros and cons, the factors for selecting the most effective technique ought to be supported the specifics of the target application surroundings. quick revocation and fine-grained control over users' security capabilities ar the motivating factors for our work. However, the requirement for these options is clearly not universal since several computing environments (e.g., a typical university campus) ar comparatively "relaxed" and do not warrant using quick revocation techniques. However, there ar lots of government, company and military settings wherever quick revocation and fine-grained control ar important.

2. RELATED WORK

Though the new paradigm of cloud computing provides nice blessings, there ar in the meantime conjointly issues regarding security and privacy particularly for web-based cloud services. As sensitive information is also hold on within the cloud for sharing purpose or convenient access; and eligible users may additionally access the cloud system for varied applications and services, user authentication has become a important element for any cloud system. A user is needed to login before victimisation the cloud services or accessing the sensitive information hold on within the cloud. There ar 2 issues for the normal account/password based mostly system.

•First, the normal account/password-based authentication isn't privacy-preserving. However, it's well acknowledged that privacy is an important feature that has to be thought of in cloud computing systems.

•Second, it's common to share a pc among totally different individuals. it should be straightforward for hackers to put in some spyware to find out the login countersign from the web-browser.

•In existing, even supposing the pc could also be bolted by a countersign, it will still be presumably guessed or taken by undiscovered malwares.

To avoid these problems We propose a fine-grained two-factor access management protocol for web-based cloud computing services, employing a light-weight security device. The device has the subsequent properties: (1) it will work out some light-weight algorithms, e.g. hashing and exponentiation; and (2) it's tamper resistant, i.e., it's assumed that nobody will burgled it to induce the key info keep within.Advantages of Proposed System:

1)Our protocol provides a 2FA security 2)Our protocol supports fine-grained attribute-based access that provides a good flexibility for the system to line completely different completely different} access policies in step with different eventualities. At a similar time, the privacy of the user is additionally preserved.

3. RECENT METHODS

We visit our approach because the SEM design. the essential plan is as follows:

We introduce a replacement entity, stated as a SEM (SEcurity Mediator): associate degree online semi-trusted server. To sign or rewrite a message, a consumer should initial get a message-specific token from its SEM. while not this token, the user cannot accomplish the meant task. To revoke the user's ability to sign or rewrite, the security administrator instructs the SEM to prevent supplying tokens for that user's future request. At that instant, the user's signature and/or decipherment capabilities are revoked. For quantifiability reasons, one SEM serves several users.We stress that the SEM design is clear to non-SEM users, i.e., a SEM is not concerned in encoding or signature verification operations. With SEM's facilitate, a SEM consumer (Alice) will generate normal RSA signatures, and rewrite normal ciphertext messages encrypted together with her RSA public key. while not SEM's facilitate, she cannot perform either of those operations. This backwards compatibility is one in every of our main style principles.Another notable feature is that a SEM isn't a totally sure entity. It keeps no consumer secrets and every one SEM computations ar checkable by its purchasers. However, a SEM is partly sure since every signature admirer implicitly trusts it to own checked the signer's (SEM's client's) certificate standing at signature generation time. Similarly, every encryptor trusts a SEM to see the decryptor's (SEM's client's)certificate standing at message decipherment time. we have a tendency to contemplate this level of trust cheap, especially since a SEM serves a large number of purchasers associate degreed so represents an organization (or a group).

In order to experiment and gain sensible expertise, we have a tendency to prototyped the SEM architecture victimization the popular OpenSSL library. SEM is enforced as a daemon process running on a secure server. On the consumer aspect, we have a tendency to engineered plug-ins for the Eudora and Outlook email purchasers for linguistic communication outgoing, and decrypting incoming, emails. each of those tasks ar performed with the SEM's facilitate. Consequently, signing and decipherment capabilities may be simply revoked. It is natural to raise whether or not a similar practicality may be obtained with additional ancient security approaches to fine-grained management and quick certification revocation, such as Kerberos. after all, has been alive since the mid-80s and tends to figure o.k. in corporate-style settings. However, Kerberos is awkward in heterogeneous networks like the Internet; its inter-realm extensions are troublesome to use and need a definite quantity of manual setup. moreover, Kerberos doesn't inter-operate with trendy PKI-s and doesn't offer universal origin authentication offered by public key signatures. On the opposite hand, the SEM design is totally compatible with existing PKI systems. additionally, the SEM is barely to blame for revocation. not like a Kerberos server, the SEM cannot forge user signatures or rewrite messages meant for users. As we have a tendency to discuss later in the paper, our approach isn't reciprocally exclusive with Kerberos-like intra-domain security architectures. we have a tendency to assert that the SEM design may be viewed as a collection of complementary security services.

We currently describe in additional detail however cryptography and digital signature generation are performed within the SEM architecture:

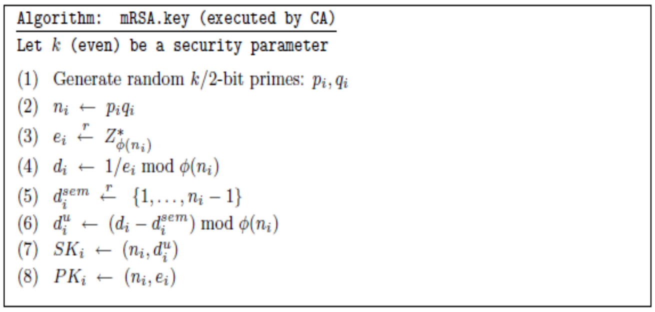
– Decryption: suppose that Alice needs to rewrite associate email message mistreatment her private key. Recall that public key-encrypted email is sometimes composed of 2 parts: (1) a brief preamble containing a per-message key encrypted with Alice's public key, and (2) the body containing the particular email message encrypted mistreatment the per-message key. To decrypt, Alice initial sends the preamble to her SEM. SEM responds with a token that permits Alice to finish the cryptography of the per message key and, ultimately, to browse her email. However, this token contains no information helpful to anyone aside from Alice. Hence, communication with the SEM doesn't have to be compelled to be secret or attested. Also, interaction with the SEM is absolutely managed by Alice's email reader and doesn't need any intervention on Alice's half. If Alice needs to browse her email offline, the interaction with the SEM takes places at the time Alice's email shopper downloads her email from the mail server.

– Signatures: suppose that Alice desires to sign a message mistreatment her personal key. She sends a (randomized) hash of the message to her SEM that, in turn, responds with a token (also mentioned as a half-signature) facultative Alice to come up with the signature. like cryptography, this token contains no helpful info to anyone other than Alice.

We currently describe intimately however a SEM interacts with shoppers to get tokens. The SEM design is predicated on a variant of RSA that we have a tendency to decision mediate RSA (mRSA). the most plan is to separate every RSA personal key into 2 components mistreatment easy 2-out-of-2 threshold RSA . One half is given to a shopper and therefore the different is given to a SEM. If the shopper and its SEM work, they use their various half-keys in a manner that's functionally cherish commonplace RSA. the actual fact that the personal key's not control in its completeness by anyone party is transparent to the surface world, i.e., to the those that use the corresponding public key. Also, information of a half-key can't be wont to derive the whole personal key.

Therefore, neither the shopper nor the SEM will decode or sign a message while not mutual consent. (Recall that one SEM serves many purchasers.) The mRSA technique consists of 3 algorithms: mRSA key generation, mRSA signatures, and mRSA cryptography. mRSA Key Generation Similar to RSA, every consumer Ui includes a distinctive public key and personal key. The public key PKi includes Ni and ei, wherever the previous could be a product of 2 giant distinct primes (pi,qi) associated ei is an whole number comparatively prime to (ni) = (pi − 1)(qi − 1).

Logically, there's conjointly a corresponding RSA non-public key SKi = (ni, di) where di ei = one mod (ni). However, as mentioned higher than, nobody party has possession of di. Instead, di is effectively split into 2 parts: dui and dsem i that area unit on the QT held by the consumer Ui and a SEM, severally. the connection among them is: di = dsem i + dui mod (ni) Unlike plain RSA, a private consumer Ui cannot generate its own mRSA keys. Instead, a sure party (most probably, a CA) initializes and distributes the mRSA keys to shoppers. The policy for authenticating and authorizing clients' key generation requests isn't mentioned during this paper. Once a client's request is received and approved, a CA executes the mRSA key generation rule represented below. mRSA Key Setup. CA generates a definite set: for Ui. The first four values area unit generated as in customary RSA. The fifth worth, dsemi , is a random whole number within the interval [1, ni], wherever Ni = piqi. The last worth is about as: dui = di − dsem i mod (ni). we tend to show the protocol in Figureone.



```
Algorithm:  mRSA.key (executed by CA)
Let $k$ (even) be a security parameter
(1) Generate random $k/2$-bit primes: $p_i, q_i$
(2) $n_i \leftarrow p_i q_i$
(3) $e_i \overset{r}{\leftarrow} Z^*_{\phi(n_i)}$
(4) $d_i \leftarrow 1/e_i \bmod \phi(n_i)$
(5) $d_i^{sem} \overset{r}{\leftarrow} \{1, \ldots, n_i - 1\}$
(6) $d_i^u \leftarrow (d_i - d_i^{sem}) \bmod \phi(n_i)$
(7) $SK_i \leftarrow (n_i, d_i^u)$
(8) $PK_i \leftarrow (n_i, e_i)$
```

Fig. 1. mRSA Key Generation Algorithm

4. PROPOSED WORK

We propose a fine-grained two-factor access management protocol for web-based cloud computing services, employing a light-weight security device. The device has the subsequent properties: (1) it will work out some light-weight algorithms, e.g. hashing and exponentiation; and (2) it's tamper resistant, i.e., it's assumed that nobody will burgled it to induce the key info keep within.Advantages of Proposed System:1)Our protocol provides a 2FA security 2)Our protocol supports fine-grained attribute-based access that provides a good flexibility for the system to line completely different|completely different} access policies in step with different eventualities. At a similar time, the privacy of the user is additionally preserved.

We seek advice from our approach because the SEM design. the essential plan is as follows: We introduce a brand new entity, mentioned as a SEM (Security Mediator): associate online semi-trusted server. To sign or decode a message, a consumer should 1st get a message-specific token from its SEM. while not this token, the user cannot accomplish the meant task. To revoke the user's ability to sign or decode, the security administrator instructs the SEM to prevent issue tokens for that user's future request. At that instant, the user's signature and/or cryptography capabilities are revoked. For quantifiability reasons, one SEM serves several users. We stress that the SEM design is clear to non-SEM users, i.e., a SEM is not concerned in cryptography or signature verification operations. With SEM's facilitate, a SEM consumer (Alice) will generate customary RSA signatures, and decode customary cipher text messages encrypted together with her RSA public key. while not SEM's facilitate, she cannot perform either of those operations. This backwards compatibility is one in every of our main style principles.
Another notable feature is that a SEM isn't a completely trustworthy entity. It keeps no consumer secrets and every one SEM computations area unit checkable by its shoppers. However, a SEM is part trustworthy since every signature supporter implicitly trusts it to possess checked the signer's (SEM's client's) certificate standing at signature generation time. Similarly, every encryptor trusts a SEM to examine the decryptor's (SEM's client's) certificate standing at message cryptography time. we have a tendency to think about this level of trust cheap, especially since a SEM serves a large number of shoppers associated so represents an organization (or a group). In order to experiment and gain sensible expertise, we have a tendency to prototyped the SEM architecture exploitation the popular OpenSSL library.

SEM is enforced as a daemon process running on a secure server. On the consumer aspect, we have a tendency to designed plug-ins for the Eudora and Outlook email shoppers for sign language outgoing, and decrypting incoming, emails. each of those tasks area unit performed with the SEM's facilitate. Consequently, signing and cryptography capabilities may be simply revoked.

It is natural to raise whether or not constant practicality may be obtained with additional ancient security approaches to fine-grained management and quick written document revocation, such as Kerberos. Kerberos [25], after all, has been breathing since the mid- 80s and tends to figure fine in corporate-style settings. However, Kerberos is awkward in heterogeneous networks like the Internet; its inter-realm extensions are tough to use and need a definite quantity of manual setup. moreover, Kerberos doesn't inter-operate with fashionable PKI-s and doesn't give universal origin authentication offered by public key signatures. On the opposite hand, the SEM design is totally compatible with existing PKI systems. additionally, the SEM is merely answerable for revocation. not like a Kerberos server, the SEM cannot forge user signatures or decode messages meant for users. As we have a tendency to discuss later in the paper, our approach isn't reciprocally exclusive with Kerberos-like intra-domain security architectures. we have a tendency to assert that the SEM design may be viewed as a group of complementary security services.

Authority It is responsible to generate user secret key for each user according to their attributes. Authority which performs the function like Upload File And Provide Download Permission Cloud Server: It provides services to anonymous authorized users. It interacts with the user during the authentication process.

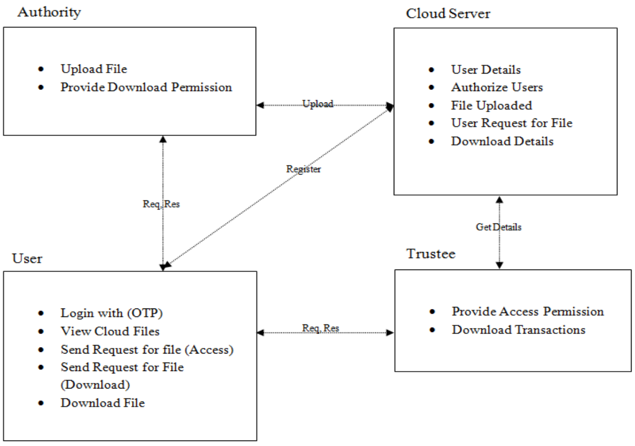


Fig:2 Architecture Diagram

Cloud Server which performs the function like User Details ,Authorize Users, File Uploaded, User Request for File, Download Details User: It is the player that makes authentication with the cloud server. Each user has a secret key issued by the attribute-issuing authority and a security device initialized by the trustee. User which performs the function like Login with (OTP),View Cloud Files, Send Request for file (Access),Send Request for File ,Download File.Trustee:It is responsible for generating all system parameters and initializes the security device. Trustee which performs the function like Provide Access Permission, Download Transactions

4. CONCLUSIONS

In this paper, we've given a brand new 2FA (including both user secret key and a light-weight security device) access control system for web-based cloud computing services. Bas ed on the attribute-based access management mechanism, the planned 2FA access system has been known to not solely enable the cloud server to limit the access to those users with an equivalent set of attributes however additionally preserve user privacy. Detailed security analysis shows that the planned 2FA access control system achieves the specified security necessities. Through performance analysis, we tend to incontestable that the construction is "feasible". we tend to leave as future work to any improve the potency whereas keeping all nice options of the system.
new approach to certificate revocation and fine-grained management over security capabilities. instead of revoking the client's certificate our approach revokes the client's ability to perform cryptographical operations like signature generation and cryptography. This approach has many blessings over ancient certificate revocation techniques: (1) revocation is quick – once its certificate is revoked, the shopper will not decipher or sign messages, (2) with binding signature semantics, there's no got to validate the signer's certificate as a part of signature verification, and (3) our revocation technique is clear to the peers since it uses commonplace RSA signature and cryptography formats.

REFERENCES

[1] M. H. Au and A. Kapadia, "PERM: Practical reputation-based blacklisting without TTPS," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Raleigh, NC, USA, Oct. 2012, pp. 929–940.
[2] M. H. Au, A. Kapadia, and W. Susilo, "BLACR: TTP-free blacklistable anonymous credentials with reputation," in *Proc. 19th NDSS*, 2012, pp. 1–17.
[3] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic *k*-TAA," in *Proc. 5th Int. Conf. SCN*, 2006, pp. 111–125.
[4] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 233–244, Apr./Jun. 2015.
[5] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Proc. 12th Annu. Int. CRYPTO*, 1992, pp. 390–420.
[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.
[7] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2004, pp. 41–55.
[8] D. Boneh, X. Ding, and G. Tsudik, "Fine-grained control of security capabilities," *ACM Trans. Internet Technol.*, vol. 4, no. 1, pp. 60–82, 2004.
[9] J. Camenisch, "Group signature schemes and payment systems based on the discrete logarithm problem," Ph.D. dissertation, ETH Zurich, Zürich, Switzerland, 1998.
[10] J. Camenisch, M. Dubovitskaya, and G. Neven, "Oblivious transfer with access control," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, Chicago, IL, USA, Nov. 2009, pp. 131–140.
[11] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," in *Proc. 3rd Int. Conf. Secur. Commun. Netw. (SCN)*, Amalfi, Italy, Sep. 2002, pp. 268–289.
[12] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2004, pp. 56–72.
[13] Y. Chen, Z. L. Jiang, S. M. Yiu, J. K. Liu, M. H. Au, and X. Wang, "Fully secure ciphertext-policy attribute based encryption with security mediator," in *Proc. ICICS*, 2014, pp. 274–289.
[14] S. S. M. Chow, C. Boyd, and J. M. G. Nieto, "Security-mediated certificateless cryptography," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 3958. Berlin, Germany: Springer-Verlag, 2006, pp. 508–524.
[15] C.-K. Chu, W.-T. Zhu, J. Han, J.-K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 50–57, Oct./Dec. 2013.
[16] R. Cramer, I. Damgård, and P. D. MacKenzie, "Efficient zero-knowledge proofs of knowledge without intractability assumptions," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 1751, H. Imai and Y. Zheng, Eds. Berlin, Germany: Springer-Verlag, 2000, pp. 354–373.
[17] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-insulated public key cryptosystems," in *Proc. EUROCRYPT*, 2002, pp. 65–82.
[18] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 3386, S. Vaudenay, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 416–431.
[19] M. K. Franklin, in *Proc. 24th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 2004.
[20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13^th^ ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[21] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 11, pp. 2150–2162, Nov. 2012.
[22] X. Huang *et al.*, "Cost-effective authentic and anonymous data sharing with forward security," *IEEE Trans. Comput.*, vol. 64, no. 4,
pp. 971–983, Apr. 2015.
[23] J. Hur, "Attribute-based secure data sharing with hidden policies in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2171–2180,Nov. 2013.
[24] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.
[25] T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. Liu, "TIMER: Secure and reliable cloud storage against data re-outsourcing," in *Proc. 10th Int. Conf. ISPEC*, 2014, pp. 346–358.
[26] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *Proc. WPES*, 2005, pp. 61–70.