Cloud Data Protection Using Identity-Based Remote Data Integrity Checking

CHILUKALA VIVEK VARDHAN REDDY[1], C. MD. GULZAR[2]
[1]PG Scholar, Dept of CSE, Dr.K.V.Subba Reddy Institute of Technology, Kurnool, AP, India.
[2]Associate Professor, Dept of CSE, Dr.K.V.Subba Reddy Institute of Technology, Kurnool, AP, India.

Abstract: Remote data integrity checking(RDIC) is a very important technology in cloud computing. It is a challenging part in cloud-server to secure all essential data that are used in many applications for end users. It's is not at all trust worthy to store entire private data in cloud server as the client won't have all the required data locally so need to rely on the cloud storage provider. To overcome this, Some RDIC protocols have been projected by many research developers. Many of the remote data integrity checking (RDIC) protocols are developed based on Public Key Infrastructure (PKI) and ID-Based Cryptography (IBC). However, each protocol has its own benefits and drawbacks. This paper presents a detailed study on RDIC protocols in a cloud environment and confers the taxonomy of RDIC protocols such as Provable Data Possession (PDP), Proof of Retrievability (POR), Proof of Ownership (POW) and ID-based RDIC protocols. We also examine and compares the current RDIC approaches based on the parameters such as integrity checking method, cryptographic model, auditing mode and data recovery. In conclusion, it throws light on the open issues such as research directions in scheming RDIC protocol.

Keywords*:* Data Integrity, Data Security Over Cloud, Cloud Storage, Identity Based Protocol, RDIC.

I. INTRODUCTION

Cloud storage [6], [7], [14], [10] is a model of networked storing system where data is kept in meres of storage which are usually hosted by others. There are many advantages to utilize cloud storage. The most prominent is data accessibility. Data that is stored in the cloud server can be retrieved at anytime from anywhere if there is internet network availability. Storage keep tasks, such as purchasing extra storing capacity, which can be relieve of to the accountability of a data-service-provider. The advantage of cloud data storage is data sharing among data-users. For example, when the data is spread, the more locations it is stored the complex risk it comprehends for illegal physical access to the data. By sharing storage and networks with various other users it is also possible for other unauthorized users to access your data. This may be due to wrong actions, defective equipment, or sometimes because of illegal intent. Downloading the file to checkered the integrity might be exorbitant in terms of bandwidth cost and very impractical as the size of the cloud-data is enormous. Moreover, traditional cryptographic primitives for data integrity checking such as authorisation code (MAC), hash functions cannot apply here right away due to being short of the original file copy in confirmation. However, it is a challenging task of RDIC for cloud storage which is highly desirable. Presently, a common existing RDIC constructions rely on PKI where a digital certificate is used to assurance the genuine of a user's public key. The structures experience composite key management procedures subsequently certificate-generation, certificate-storage, certificate-update and certificate-revocation are inefficient and exclusive. There is a variability of standards, say the Internet X.509 PKI certificate policy and certification practices framework (RFC 2527), that cover features of PKI. However, it privations predominant governing body to impose these standards. In spite of a certificate authority (CA) is often regarded as important, drawbacks in the security procedures of several CAs have endangered trust in the entire PKI on which the Internet relies on. For instance, after noticing more than 100 fake certificates, web browser sellers were forced to blacklist all the certificates delivered by DigiNotar, a Dutch CA, in 2011. Another tactic to using a certificate to validate public key is identity-based cryptography [30], in which where the public key of a data-user is only his individuality, say his names, emails or IP addresses. A righthand key distribution centre (KGC) produces a secret key for every data-user consistent to his individuality. When each and every data-users have their secret keys given by KGC, separate public keys convert obsolete, thus eradicating the need for explicit certification and all the associated burdens.

These structures make the identity-based paradigm particularly appealing for use in conjunction with organization-oriented PDP. For example, a university purchases the cloud storage service for the guides and students, who have a valid E-mail addresses allotted by the department of the university. All the people of the university can have their secret key providing by the KGC, say the department. The people of the university can store their data together with the meta-data of the file to upload to the cloud. To safeguard the data are stored correctly, an auditor, a staff of department can squared the integrity for any people with his E-mail addresses only, which can release the composite key management caused by PKI. The primary ID-based PDP was projected in [31] which rehabilitated the ID-based aggregate signature due to Gentry [32] to an ID-based PDP protocol. Wang [33] anticipated another identity-based provable data possession in multi-cloud storage. However, their security model called unforgeability for identity-based PDP is not strong enough for capturing the property of soundness in the sense that, the challenged data blocks are not allowed for TagGen queries in this model, which indicates that the adversary cannot access the tags of those blocks. This is clearly not consistent with the real cloud storage where the cloud server is, in fact, storing the tags of all data blocks. Moreover, the concrete identity-based PDP protocol in [33] fails to achieve soundness, a basic security requirement of PDP schemes. The reason is that, the hashed value of each block is used for generating a tag of the block, as a consequence, a malicious cloud server can keep only the hash value of the blocks for generating a valid response to a challenge.

II. OVERVIEW

Here, we analysis some introductory knowledge used in this paper, plus bilinear pairings and zero-knowledge proof.

A. Bilinear Pairing: A bilinear pairing [30] maps a pair of group elements to another group element. Specifically, let S1, S2 be two cyclic groups of order p. s1 and s2 denote generators of S1 and S2 respectively. A function e: S1 x S1 -> S2 is called a bilinear pairing if it has the following properties:

Bilinearity. For all u, v € S1 and x, y € Zp, $e(a^x, b^y) = e(a,b)^{xy}$ holds.
Non-Degeneracy. $e(s1, s2) \neq 1_{S2}$ where 1_{S2} is the identity element of S_2.
Effective Computation. e(a, b) can be computed efficiently (in polynomial time) for all a,b € G_1.

B. Equality of Discrete Logarithm: Let S be a finite cyclic group such that |S| = r for some prime r, and s1; s2 be generators of S. The following protocol [39] enables a prover P to prove to a verifier V that two elements X1; X2 have equal discrete logarithm to base s1 and s2 respectively.

Commitment. P randomly chooses ρ € Zq, computes $T1 = s^{p}_1$,$T2 = s^{p}_2$ and sends T1, T2 to V.
Challenge. V randomly chooses a challenge c € $\{0,1\}^{\lambda}$ and sends c back to P.
Response. P computes z = ρ - cx (mod q) and returns z to V.
Verify. V accepts the proof if and only if $T_1 = s^{z}_1 X^{c}_1 \wedge T2 = s^{z}_2 X^{c}_2$ holds.

This protocol can be converted into a more efficient non-interactive version, which is denoted as $POK\{(k): Y1 = s^{k}_1 \wedge X2 = s^{k}_2\}$, by replacing the challenge with the hash of the commitment, that is, c = H(T1||T2), where H is a secure hash function.

C. ID-based Signature:

An identity-based signature (IDS) scheme [40], [41] consists of four polynomial-time, probabilistic algorithms described below.

Setup(k). This algorithm takes as input the security parameter k and outputs the master secret key msk and the master public key mpk.
Extract (msk, ID). This algorithm takes as input a user's identity ID, the master secret key msk and generates a secret key usk for the user.
Sign (ID, usk, m). This algorithm takes as input a user's identity ID, a message m and the user's secret key usk and generates a signature _ of the message m.
Verify (ID, m, σ, mpk). This algorithm takes as input a signature σ, a message m, an identity ID and the master public key *mpk*, and outputs if the signature is valid or not.

III. SYSTEM AND SECURITY MODEL

Here, we try to explain the system and security model of identity-based RDIC protocols.

1. ID-based RDIC System Typically, data owners themselves can check the integrity of their cloud data by running a dual-party RDIC protocol. However, the inspecting result from either the data owner or the cloud server might be observed as biased in a dual-party situation. The RDIC protocols with public verifiability allow anybody to audit the integrity of the outsourced data. To make the description of the publicly certifiable RDIC protocols clearly, we undertake there exists a third-party auditor (TPA) who has proficiency and abilities to do the verification work. With this in observance, the ID-based RDIC architecture is demonstrated in Fig 1. Four dissimilar entities namely the KGC, the data-cloud-user, the data-cloud-server and the TPA are involved in the system. The KGC produces secret keys for all the users conferring to their identities. The data-cloud-user has great number of files to be kept on cloud without having a local copy, and the data-cloud-server has important storing space and computation properties and provides data storing services for cloud users. TPA has expertise and abilities that cloud users do not have and is trusted to verify the integrity of the cloud data on behalf of the data-cloud-user upon request. Every entity has their own responsibilities individually. The data-cloud-server could be selfish, and for his own benefits, such as to uphold a good reputation, the data-cloud-server might even decide to hide data corruption incidents to data-cloud-users. However, we assume that the cloud server has no inducements to reveal the hosted data to TPA because of guidelines and financial inducements. The TPA's task is to achieve the data integrity checking on behalf the data-cloud-user, but the TPA is also curious in the sense that he is willing to learn some information of the user's data during the data integrity checking procedure.

We consider three security properties namely completeness, security against a malevolent server (security), and privacy against the TPA (perfect data privacy) in identity-based remote data integrity checking(RDIC) protocols. Following the security ideas due to Shacham and Walters [7], an identity-based RDIC scheme is called safe against a server if there occurs no polynomial-time algorithm that can fraud the TPA with non-negligible probability and there occurs a polynomial-time extractor that can recover the file by running the tasks response protocols multiple times. Comprehensiveness states that when interacting with a valid data-cloud-server, the algorithm of Proof Check will accept the proof. Soundness says that a fraud prover who can convince the TPA it is storing the data file is actually storing that file. We now solemnize the security model of soundness for identity-based remote data integrity checking(RDIC) below, where an adversary who plays the role of the not a trust worthy server and a contender who represents a data owner are involved in process.

- **Setup**: The contender runs the Setup algorithm to obtain the system parameters param and the master secret key msk, and forwards param to the opponent, while keeps msk private.

- **ProofGen**: TagGen query has been made for the file H, the opponent can undertake executions of the ProofGen algorithm by stipulating an identity ID of the data owner and the file name Hn. The contender plays the role of the TPA and the opponent O behaves as the prover during the proof generation. In conclusion, the opponent can get the output of P from the contender when a protocol implementation finishes.
- **Generated Output**: In conclusion, the opponent chooses a file name $H_\eta^\dagger$ and a user identity $ID^\dagger$. $ID^\dagger$ must not have appeared in key extraction queries and there exists a TagGen query with input $H^\dagger$ and $ID^\dagger$. The opponent creates outputs the description of a prover $P^\dagger$ which is €-admissible.

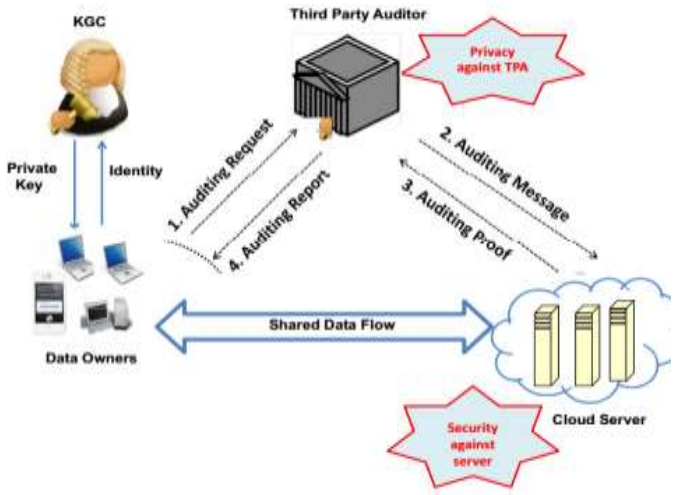

Fig. 1. Identity-based Remote Data Integrity Checking(RDIC) System Model.

IV. MODULES

In data integrity checking with public verifiability, we have following modules:

1. **Data Owner**: In Data Owner module, Initially Data Owner must have to register their detail and KGC will authorize the registration by sending Private Key through email. After successful login data Owner can upload files into cloud server. He/she can view the files that are uploaded in cloud. Data Owner will send audit request to TPA then DO receives the audit report. After receiving the report, he/she can verify the files.
2. **KGC**: In KGC module, KGC can view all the Data owners' details. KGC will authorize data owners also KGC will send the Private Key to the users.
3. **TPA**: In TPA module, TPA can view all the Data owners audit request details. TPA can send challenge to cloud server to generate proof. Then cloud receives the request and sends the proof to TPA. After receiving the proof TPA will send it to the Data Owner.
4. **Cloud Server**: In Cloud Server module, Cloud server can view Audit request details. Cloud server will generate proof and send it to TPA. Cloud Server can also view the files details in the cloud server.

V. IMPLEMENTAION

The implementation was led with pbc-0.5.13 [42] with pbc wrapper-0.8.0 [43] on Intel i7-4700MQ CPU @2.40GHz. The memory is always adequate since the scheme only requires a polynomial space. In our implementation, we made use of parameter a.param, one of the standard parameter settings of pbc library. Parameter a.param provides a symmetric pairing with the wildest speed among all default parameters. The implementation time overheads of the protocol are showed in the following two parts. In the first portion, we arrange a file of a constant size of 1 MB, and observe the effect of the number of challenged blocks in terms of the time cost. In our setting, the size of a data block is bounded by the group order p, i.e., 160 bits. Hence, we have 50,000 blocks in total. This infers that the timing results for Setup, Citation and TagGen steps are constant for this part. We can see that both the Setup algorithm and the Citation algorithm are extremely fast. The Setup algorithm picks some random values and calculate a modular exponentiation in S1, which costs 4:8ms, and the Citation algorithm needs to perform one modular exponentiation in S1 for generating the private key of a cloud user, which cost 0:1ms.

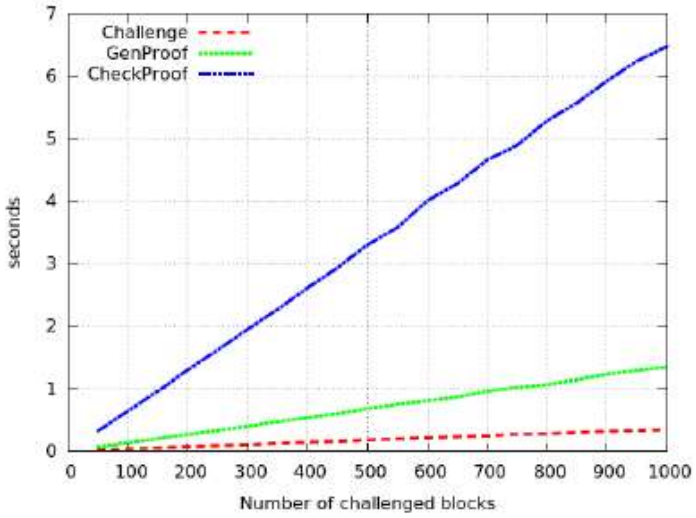

Fig. 2. Increasing number of challenges for fixed size of file.

The time cost of off-line computing-process of generating tags for1 MB file is 241:9 seconds while the on-online time cost is20:3 seconds. This is an satisfactory result compared with the previous preprocessing result in [5]. Note that we can adopt the technique of provable outsourcing computation [44]–[47]which permits a data owner to offload the computation of some function, to other possibly untrusted servers such as a cloud, though maintaining verifiable results, to improve the efficiency of tag generation if needed. The implementation shows that producing tags is more expensive than other parts but providentially, computing tags for a file is a onetime job, as compared to interesting the outsourced data, which will bed one recurrently. Meanwhile the cloud users can do the off-line work entirely parallelizable in advance, we pay only attention to the on-line cost. We can see that the efficiency of TagGen of our protocol is comparable to that of the existing well-known schemes, say [5]. To produce tags for a 2 MB file, it costs almost 42 seconds. As such, one shall be able to do in advance, the time cost of generating tags for any size of files.

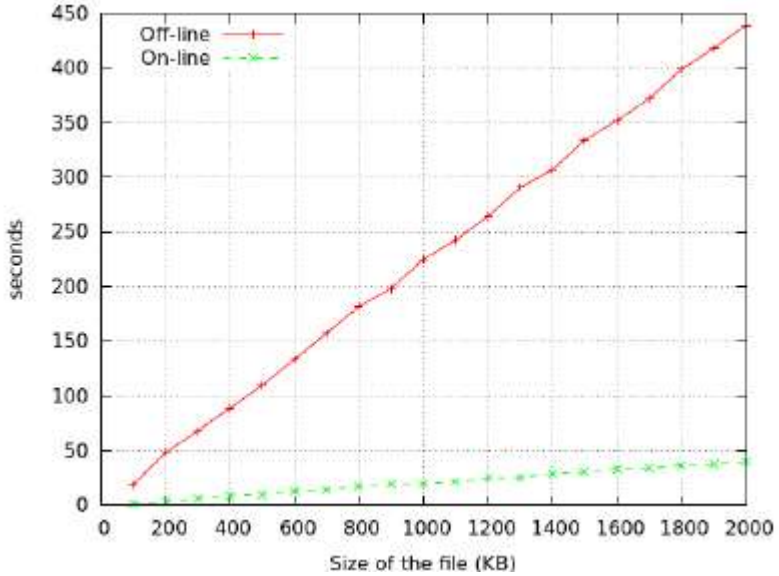

Fig. 3. Tag generation time for increased size of files.

VI. CONCLUSION

In this paper, we propose a new primitive called identity-based remote data integrity checking for secure cloud storage. We formalized the security model of two significant properties of this primeval, namely, soundness and perfect data privacy. We provided a new construction of this primitive and showed that it achieves accuracy and perfect data privacy. The implementation confirmed that the planned protocol is efficient and practical.

VII. ACKNOWLEDGE

I would like to express my special thanks of gratitude to C.Md. Gulzar Associate Professor, Department of Computer Science and Engineering, Dr.K.V.Subba Reddy Institute of technology. Who gave me the golden opportunity to do this wonderful project on the topic which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to him.

VIII. REFERENCES

[1]P. Mell, T. Grance, Draft NIST working definition of cloud computing, Reference on June. 3rd, 2009. http://csrc. nist.gov/ groups/SNC/cloudcomputing/index.html.
[2]Cloud Security Alliance. Top threats to cloud computing. http://www.cloudsecurityalliance.org, 2010.
[3]M. Blum, W. Evans, P.Gemmell, S. Kannan, M. Naor, Checking the correctness of memories. Proc. of the 32nd Anual Symposium on Foundations fo Vomputers, SFCS 1991, pp. 90–99, 1991.
[4]G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, D. X. Song, Provable data possession at untrusted stores. ACM Conference on Computer and Communications Security, 598-609, 2007.
[5]G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. N. J. Peterson, and D. Song, Remote data checking using provable data possession. ACM Trans. Inf. Syst. Secur., 14, 1–34, 2011.
[6]A.Juels, and B. S. K. Jr. Pors, proofs of retrievability for large files.Proc. of CCS 2007, 584-597, 2007.
[7]H. Shacham, and B. Waters, Compact proofs of retrievability. Proc. Of Cryptology-ASIACRYPT 2008, LNCS 5350, pp. 90-107, 2008.
[8]G. Ateniese, S. Kamara, J. Katz, Proofs of storage from homomorphic identification protocols. Proc. of ASIACRYPT 2009, 319-333, 2009.
[9]A. F. Barsoum, M. A. Hasan, Provable multicopy dynamic data possession in cloud computing systems, IEEE Trans. on Information Forensics and Security, 10(3): 485–497, 2015.
[10]J. Yu, K. Ren, C.Wang, V. Varadharajan, Enabling cloud storage auditing with key-exposure resistance, IEEE Trans. on Information Forensics and Security, 10(6): 1167–1179, 2015.
[11]J. Liu, K. Huang, H. Rong, H. M. Wang, Privacy-preserving public auditing for regenerating-code-based cloud storage, IEEE Trans. On Information Forensics and Security, 10(7): 1513–1528, 2015.
[12]Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, Enabling public verifiability and data dynamics for storage security in cloud computing. Proc. of ESORICS2009, LNCS 5789, 355–370, 2009.
[13]C. Wang, Q. Wang, K. Ren, and W. Lou, Privacy-preserving public auditing for data storage security in cloud computing. Proc of IEEE INFOCOM 2010, 525–533, 2010.
[14]C. Wang, K. Ren, W. Lou, and J. Li, Toward publicly auditable secure cloud data storage services. IEEE Network, 24, 19-24, 2010.
[15]Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, Enabling public audibility and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst., 22, 847-859, 2011.
[16]C. Wang, S. S.Chow, Q. Wang, K. Ren, and W. Lou, Privacy-preserving public auditing for secure cloud storage. IEEE Trans. on Computers, 62, 362-375, 2013.
[17]K. Yang, and X. Jia. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans. on Parallel and Distributed Systems, 24(9): 1717-1726, 2013.
[18]Y. Zhu, G. J. Ahn, H. Hu, S. S. Yau, H. G. An, C. J. Hu, Dynamic audit services for outsourced storages in Clouds. IEEE Trans. Services Computing, 6(2): 227-238, 2013.
[19]Y. Zhu, H. Hu, G. J. Ahn, S. S. Yau, Efficient audit service outsourcing for data integrity in clouds. Journal of Systems and Software, 85(5): 1083-1095, 2012.
[20]H. Wang, Y. Zhang, On the knowledge soundness of a cooperative provable data possession scheme in multicloud storage, IEEE Trans. on Parallel and Distributed System, 25(1): 264–267, 2014.
[21]J. Wang, X. Chen, X. Huang, I. You, Y. Xiang, Verifiable auditing for outsourced database in cloud computing, IEEE Transactions on Computers, 64(11),3293-33032015.
[22]Y. Yu, Y. Li, J. Ni, G. Yang, Y. Mu, W. Susilo, Comments on Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification, IEEE Trans. Information Forensics and Security, 11(3): 658-659, 2016.
[23]G. Ateniese, R. D. Pietro, L. V. Mancini, G. Tsudik, Scalable and efficient provable data possession, Proc. of SecureComm 2008, Article No. 9 ,doi:10.1145/1460877.1460889.
[24]C. Wang, Q. Wang, K. Ren, W. Lou. Ensuring data storage secruty in cloud computing, Proc. of IWQoS 2009, 1-9, 2009.
[25]C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia, Dynamic provable data possession, Proc. of CCS 2009, 213-222, 2009.
[26]C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, J. Chen, MuRDPA: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud. IEEE Trans. On Computers, 64(9): 2609–2622, 2015.
[27]C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, Privacy-preserving public auditing for secure cloud storage, IEEE Trans. on Computers, 62, 362–375, 2013.
[28]Y. Yu, M H Au, Y. Mu, S. Tang, J. Ren, W. Susilo, and L. Dong, Enhanced privacy of a remote data integrity checking protocol for secure cloud storage, International Journal of Information Security, 14(4): 307– 318, 2015.
[29]G. Ateniese, A. Faonio, S. Kamara, Leakage-resilient identification schemes from zero-knowledge proofs of storage. IMA Int. Conf. 2015, 311-328, 2015.
[30]D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, Proc. of CRYPTO 2001, LNCS 2139, 213–229, 2001.
[31]J. N. Zhao, C. X. Xu, F. G. Li, and W. Z. Zhang, Identity-Based Public Verification with Privacy-Preserving for Data Storage Security in Cloud Computing, IEICE Transactions, 96-A(12), 2709-2716, 2013.
[32]G. Gentry, Z. Ramzan, Identity-Based aggregate signature, Proc. Of Public Key Cryptography 2006, LNCS 3958, 257-271, 2006.
[33]H. Wang, Identity-based distributed provable data possession in multicloud storage, IEEE Trans. on Service Computing, 8(2), 328–340, 2015.

[34]Q. Wu, Y. Mu, W. Susilo, B. Qin, J. Domingo-Ferrer, Asymmetric group key agreement. Proc. of Eurocrypt 2009, LNCS 5479, 153–170, 2009.
[35]L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, Z. Dong, Roundefficient and sender-unrestricted dynamic group key agreement protocol for secure group communications. IEEE Transactions on Information Forensics and Security, 10(11): 2352–2364, 2015.
[36]V. Shoup, Lower bounds for discrete logarithms and related problems, Proc. of Eurocrypt 1997, 256–266, 1997.
[37]Y. Yu, Y. Zhang, Y. Mu, W. Susilo, Provably Secure Identity based Provable Data Possession. Proc. of ProvSec 2015, LNCS 9451, 1–16, 2015.
[38]D. Boneh, B.Lynn, and H. Shacham, Short signatures from the weil pairing. Journal of Cryptology, 17, 297–319, 2004.
[39]D. Chaum, T. P. Pedersen, Wallet databases with observers. Proc. Of Crypto 1992, 89–105, 1993.
[40]J. C. Cha, and J. H. Cheon, An identity-based signature from gap Diffie- Hellman groups, in: Processing of PKC 3003, LNCS 2567, 2003, pp. 18-30.
[41]F. Hess, Efficient identity-based signature schemes based on pairings, in: Processing of SAC 2002, LNCS 2595, 2003, pp. 310-324.
[42]B. Lynn, The Pairing-Based Cryptography Library (0.5.13). online: http://crypto.stanford.edu/pbc/
[43]A. Kate, The Pairing-Based Cryptography (PBC) Library - C++ Wrapper Classes (0.8.0). online: http://crysp.uwaterloo.ca/ software/PBCWrapper/.
[44]R. Gennaro, C. Gentry, B. Parno, Non-interactive verifiable computing: outsourcing computation to untrusted workers. Proc. of CRYPTO 2010, LNCS 6223, 465-482.
[45]X.F. Chen, J. Li, X. Y. Huang, J. F. Ma, W. Lou, New publicly verifiable databases with efficient updates, IEEE Transactions on Dependable and Secure Computing, 12(5): 546-556, 2015.
[46]J. Lai, R. H. Deng, H. Pang, J. Weng, Verifiable computation on outsourced encrypted data. ESORICS (1) 2014: 273-291
[47]X. Liu, R. Choo, R. Deng, R. Lu, J. Weng, Efficient and privacy preserving outsourced calculation of rational numbers, IEEE Trans. On Dependable and Secure Computing, online: http://dx.doi.org/10.1109/TDSC.2016.2536601.