International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Multi Consumers Deduplicatable Effective Evidence of Storage in Cloud

Mujeeb Ur Rehaman k[1], Dr.Prakash[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 - *Growing reputation of cloud computing, an increasing number of data owners are inspired to outsource their records to Cloud servers for extremely good convenience and reduced value in data management where, Dynamic evidence of storage could be a useful primitive that permits a consumer to see the integrity of outsourced document and to with comfortably replace the documents in a cloud server. Although many of them have express numerous dynamic pos schemes in single consumer environments, the analysis is depend in multi-user environments has no longer been sufficiently illustrated. A practical multi-consumer cloud storage system permits the safe Client-side cross-user Deduplication technique that allows a user to skip the importing approach and procure the possession of the files immediately, once subsequent user of an equal file have uploaded them to the cloud server. No other present system will permit to help this system. This paper mainly focuses on the strategies of verifiable data storage and secure records deduplication. We've a functionality to introduce the thought of deduplicatable Dynamic proof of storage associated promoter an economical construction called as deypos, to realize dynamic proof of storage and at comfort cross-user duplication, at the same time. Through analyzing the challenges of structure diversity and private tag Generation, we've a functionality to exploit a unique tool called as homomorphic Authenticated tree (hat). We are enough Capable to prove the protection of our production subsequently according to theoretical evaluation and experimental consequences.*

***Key Words*: Cloud Storage, Deduplication, and Effective Evidence of Storage.**

1. INTRODUCTION

Cloud computing and storage solutions offer customers and companies with diverse skills to keep and process their records in either privately owned, or in third-party centers, cloud computing is based on sharing of resources to obtain coherence and economic system of scale. Cloud is a practice of usage public remote servers hosted at the internet to save, manage, and evolve data, instead of a regional server or a personal laptop and usually price for cloud computing offerings based totally on usage. Storage outsourcing is becoming increasingly attractive to both enterprise and academia because of the benefits of low cost, excessive accessibility, and easy sharing. Many companies, such as amazon, Google, and Microsoft, provide their own cloud storage services, in which users can add, their files to the servers, get access to them from different devices, and share them with the others [4].

Cloud infrastructures may be roughly categorized as both private and public. In a private cloud, the infrastructure is controlled and owned by consumer and located on-premise (Customers area of manage). Specifically, because of this get right of entry to customer data is below its Manage and is best granted to parties it trusts and in a public cloud the infrastructure is owned and Controlled through a cloud provider company and is placed on premise (i.e., within the service issuer's place Of manage). Because of this patron statistics is outside it's manage and could potentially be granted To untrusted events [1]. Regardless of the diverse blessings of cloud services, outsourcing sensitive facts to far off servers brings privacy worries. The cloud service companies that preserve the information for users and may also access user's sensitive information without authorization. A general technique to defend the records confidentiality is to encrypt the data before outsourcing [2].

Information integrity turns into crucial homes while a person outsources its files to cloud storage. Users should convince that the documents hold inside the server need to not be tampered. There are many historical strategies for safeguarding information integrity, like message authentication codes (macs) and digital signatures need customers to transfer all the files from the cloud server for verification that ensures a big communique price. These techniques are not appearing to be appropriate for cloud storage offerings wherever customers could take a look at the integrity typically, like each hour. Therefore researchers brought evidence of storage (pos) for checking the integrity at the same time as not downloading documents from the cloud server. Customers might want many dynamic operations, like change, insertion, and deletion, to replace their files also maintaining the Storage of pos. dynamic pos is created for such dynamic operations.

In addition with pos, dynamic pos employ structures looks like the Merkle tree. As a result each time dynamic operations are useless users regenerate tags for the up to date blocks totally as opposed to creating for all blocks to increased Understand the subsequent contents. Our

tendency to gift extra info concerning pos and dynamic pos. the schemes used in our challenge, each block of a document is established a tag which hired for substantiating the integrity of that block. When a champion confirms the integrity of a record, it selects a few block indexes of the report, and sends the files to the cloud server. On consisting the undertaking beside their tags the cloud server returns corresponding tags generated.

The index correctness and record integrity is test by means of the champion. The direct bonding of tags is performed by cryptanalytic tags. there's a path to have an effect on the latter is that the foremost difference among pos and dynamic pos the indication of correctness and block integrity inside the PoSs scheme by means of the block index that is encoded into its tag . dynamic pos is not able to cypher the block indexes into tags, since the dynamic operations may want to changed many indexes of non-up to date blocks that incurs reserve computation and conversation fee. For example, a report consists of a thousand blocks, and an alternative block which is inserted at the back of the second one block of the document. then, 998 block indexes of the previous first document which might be changed that implies the consumer is able to generate and send 999 tags for this update.

For the sake of unraveling these demanding situations many shape are introduced in dynamic PoSs, result indicate that the tags are established to the structure rather than the block indexes. The reason at the back of the dynamic pos stays to be stepped forward in an extremely multi-person ecosystem is the need of cross-user England country duplication on the patron-facet. This means that customers will able to bypass the uploading method and accumulate the possession of documents now as long due to the fact the uploaded documents present already inside the cloud server. This technique will reduce back space for storing for the cloud server and store transmission information measure for customers. There are no dynamic pos that could offer relaxed pass-person England nation duplication.

2. RELATED WORK

The concept of evidence of storage changed into introduced by using Atenieseet al. [5], and juels and kaliski [17], respectively. The principle concept of pos is to randomly pick out some records blocks as the challenge then the cloud server returns the challenged records blocks and their tags as the reaction because the data blocks and the tags can be mixed thru homomorphic functions, the conversation charges are decreased. The subsequent works [11], prolonged the studies of pos, however the ones works did not take dynamic operations into consideration. Erway et al. [8] and later works [13], [14] focused on the dynamic statistics among them, the scheme in [14] is the maximum efficient answer in practice but, the scheme is stateful, which calls for customers to keep some state records of their very own files regionally for this reason, it isn't appropriate for a multiuser surroundings. Halevi et al. [15] delivered the concept of evidence of possession that is an answer of cross-user deduplication on the customer-side. It calls for that the user can generate the Merkle tree without the help from the cloud server, which is a big project in dynamic pos. pietro and sorniotti [30] proposed every other evidence of ownership scheme which improves the efficiency. Xu et al. [31] proposed a customer-aspect deduplication scheme for encrypted records, however the scheme employs a deterministic proof set of rules which suggests that every record has a deterministic quick proof. Accordingly, all and sundry who obtains this proof can bypass the verification without possessing the record domestically. Other deduplication schemes for Encrypted data [32], [33], [34] were proposed for reinforcing the safety and performance. Notice that, all current strategies for pass-consumer deduplication at the consumer-facet were designed for static files. Once the documents are up to date, the cloud server has to regenerate the whole authenticated systems for those documents, which reasons heavy computation value at the server-side.

Zheng and xu [35] proposed an answer called proof of storage with deduplication, which is the primary try to design a pos scheme with deduplication. Du et al. [36] brought proofs of ownership and retrievability, which can be much like [35] but extra efficient in phrases of computation fee. Be aware that neither [35] nor [36] can aid dynamic operations. because of the trouble of shape diversity and personal tag generation, [35] and [36] cannot be prolonged to dynamic pos. Wang et al. [37], [38], and Yuan and yu [39] considered evidence of storage for multi-user updates, but those schemes awareness at the problem of sharing files in a group. Deduplication in those eventualities is to deduplicate documents among different organizations. Sadly, these schemes cannot aid deduplication because of shape range and personal tag era. On this paper, we recollect an extra general situation that every user has its own files separately. For this reason, we cognizance on a deduplicatable dynamic pos scheme in multiuser Environments. The primary techniques used in pos and dynamic pos schemes are homomorphic message authentication codes [40] and homomorphic signatures [41], [42]. With the assist of homomorphism, the messages and macs/signatures in those schemes may be compressed into a single message and an unmarried mac/signature. Consequently, the communique fee can be dramatically decreased those techniques had been utilized in pos [7], [14], [18] and comfy network coding [43], [44], [45]. A brief survey of homomorphic macs and signatures can be referred in [46].

To the excellent of our knowledge, this is the primary work to introduce a primitive known as deduplicatable dynamic proof of storage which solves the shape range and personal tag generation challenges in contrast to the prevailing authenticated structures, inclusive

of bypass list [8] and Merkle tree [14], we design a Novel authenticated structure known as homomorphic authenticated tree (hat), to reduce the conversation price in each the evidence of Storage section and the deduplication phase with similar computation value observe that hat can aid integrity verification, dynamic operations, and move-user deduplication with correct consistency. We suggest and implement the first green construction of deduplicatable dynamic pos called deypos, which supports limitless quantity of verification and update operations the safety of this creation is proved inside the random oracle model, and the overall performance is analyzed theoretically and experimentally.

3. RECENT METHODS

To enforce an efficient deduplicatable dynamic pos scheme, we layout a singular authenticated structure referred to as homomorphic authenticated tree. A hat is a binary tree wherein each leaf node corresponds to a data block. Even though hat does now not have any dilemma on the wide variety of records blocks, for the sake of description simplicity, we anticipate that the quantity of facts blocks n is equal to the number of leaf nodes in a full binary tree.
Hence, for a document f = (m_1; m2; m_3; m_4) where in mi represents the ith block of the file, we will construct a tree each node in hat consists of a 4-tuple v_i (i; l_i; v_i; t_i). i is the particular index of the node.

The index of the foundation node is 1, and the indexes will increase from top to bottom and from left to right. L_i denotes the quantity of leaf nodes that may be reached from the ith node. V_i is the model variety of the ith node. t_i represents the tag of the ith node. when a hat is initialized, the version variety of each leaf is 1, and the version number of every non-leaf node is the sum of that of its youngsters. For the ith node, mi denotes the mixture of the blocks similar to its leaves.

The tag ti is computed from f(m_i), where in f denotes a tag era characteristic. we require that for any node v_i and its youngsters v_{2i} and v_{2i+1}, f(m_i) = f(m_{2i} @ m_{2i+1})= f(m_{2i}) @f(m_{2i+1}) holds, in which @ denotes the aggregate of m_{2i} and m_{2i+1},that's why we name it a "homomorphic" tree.

4. PROPOSED WORK

Our machine version considers two types of entities: the cloud server and customers, as proven in fig. for every document, unique user is the consumer who uploaded the document to the cloud server, while next person is the person who proved the possession of the file but did no longer virtually upload the document to the cloud server.

4.1 Challenges in cloud Storage

Most troubles start from the reality that the person loses manipulate of his or her facts, because it's miles stored on a laptop belonging to a person else (the cloud company).this happens when the owner of the far off servers is a person or company other than the consumer; as their hobbies may additionally point in one of a kind guidelines (as an instance, the consumer might also want that his or her facts is saved personal, but the proprietor of the remote servers might also need to take benefit of it for his or her very own business).in most of the prevailing dynamic pos a tag used for integrity verification is generated by way of the secret key of the uploader. Therefore, other proprietors who have the ownership of the report but have not uploaded it because of the cross-user deduplication at the client-aspect cannot generate a brand new tag when they update the record in this situation, the dynamic pos might fail.

Introduced the idea of evidence of ownership which is a solution of cross-consumer deduplication at the purchaser-side. it requires that the person can generate the Merkle tree without the assist from the cloud server, which is a big assignment in dynamic pos. client-side deduplication scheme for encrypted statistics, however the scheme employs a deterministic evidence algorithm which shows that every report has a deterministic brief evidence. as a consequence, anyone who obtains this evidence can bypass the verification without owning the report locally. All present techniques for move-user deduplication on the purchaser-aspect have been designed for static files. as soon as the documents are up to date, the cloud server has to regenerate the whole authenticated systems for these documents, which causes heavy computation fee on the server-side. Because of the problem of shape variety and private tag era, present gadget can't be prolonged to dynamic pos. Alas, these schemes cannot aid deduplication because of shape variety and private tag era conquer these kinds of we proposed this multi consumer deduplicatable dynamic proof of garage in cloud.

4.2 System Model

There are 5 stages in a deduplicatable dynamic pos device: pre-process, add, deduplication, update, and proof of storage. Within the pre-process segment, customers intend to upload their nearby documents. The cloud server decides whether or not these documents should be uploaded. If the upload manner is granted, move into the add section; otherwise, go into the deduplication section.

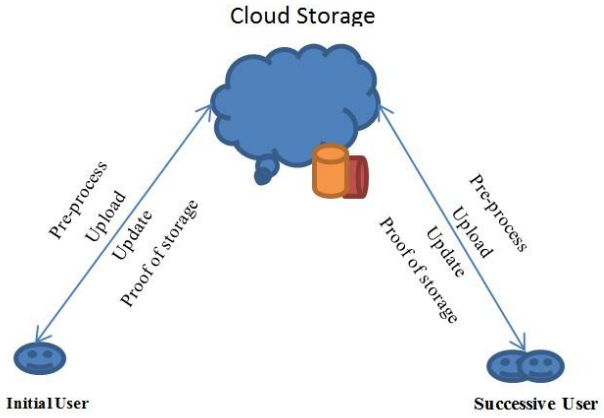

Figure 1: System Diagram

In the add phase, the documents to be uploaded do no longer exist inside the cloud server the unique customers encodes the local files and upload them to the cloud server.
Within the deduplication phase, the files to be uploaded exist already within the cloud server the subsequent users own the files domestically and the cloud server shops the authenticated systems of the documents. Next customers want to persuade the cloud server that they own the documents without uploading them to the cloud server.

Word that, these 3 phases (pre-process, add, and deduplication) are executed only once inside the existence cycle of a record from the attitude of users. This is, these three levels appear best whilst customers intend to add documents if those stages terminate generally, i.e., customers end uploading within the upload section, or they pass the verification within the deduplication section, we say that the users have the ownerships of the documents in the update segment, users may adjust, insert, or delete some blocks of the documents. Then, they update the corresponding components of the encoded files and the authenticated structures in the cloud server, even the original documents have been not uploaded by themselves.

Notice that, customers can update the files only if they have the ownerships of the files, which means that that the customers ought to add the documents inside the add phase or bypass the verification inside the deduplication segment.

For every update, the cloud server has to reserve the authentic file and the authenticated shape if there exist different Owners, and record the up to date part of the report and the authenticated structure this allows users to replace a file Concurrently in our version, when you consider that each replace is simplest "connected" to the unique record and authenticated shape.

In the proof of garage section, users handiest own a small consistent length metadata domestically and they need to test whether or not the files are faithfully saved in the cloud server without downloading them the files might not be uploaded by means of these users, but they bypass the deduplication section and prove that they've the ownerships of the files.

Observe that, the update section and the evidence of storage segment can be done a couple of instances in the life cycle of a report. Once the possession is established, the customers can arbitrarily enter the replace section and the proof of storage segment without keeping the original documents locally.

5. RESULTS

The Evaluated outcome displayed how data can be outsourced to the cloud form consumer side and cloud can manage record deduplication when the number of consumer are importing the unique records to cloud at some time. Then cloud import the record to the cloud of first consumer and it skip the importing records of other consumer which is already in cloud imported by first consumer and cloud just shows you are imported to cloud to other consumer by holding first record in cloud to both consumers for their uses.

6. CONCLUSIONS

Outsourcing data to remote servers has turn into a developing trend for many businesses to ease the weight of local records storage and safety. In this we've got taken into consideration of defined comprehensive necessities in multi-consumer cloud storage structures and added the model of deduplicatable dynamic pos, We used a device called hat which is a good authenticated structure by the use of hat, we provide deduplicatable dynamic pos scheme referred to as deypos. It can pop out with overall performance evaluation of deypos, the performance is measured with reaction time and storage block. At the end up we are able to have the ability to implement a realistic multi-user cloud storage machine that makes use of cross user deduplication approach, which lets in a person to skip the importing technique and acquire the possession of the files straight away, while different owners of the same files have uploaded them to the cloud server. The analysis display that our deypos implementation is good, especially when the document size and the range of the challenged blocks are massive.

REFERENCES

[1] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proc. 14th Int. Conf. Financial Cryptography Data Security, 2010, pp. 136–149.
[2] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 340–352, Feb. 2016.

[3] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," IEEE Commun. Surveys Tuts., vol. 15, no. 2, pp. 843–859, Jul. 2013.
[4] C. A. Ardagna, R. Asal, E. Damiani, and Q. H. Vu, "From security to assurance in the cloud: A survey," ACM Comput. Surv., vol. 48, no. 1, pp. 2:1–2:50, 2015.
[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Security, 2007, pp. 598–609.
[6] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Security Privacy Commun. Netow., 2008, pp. 1–10.
[7] G. Ateniese, S. Kamara, and J. Katz, "Proofs of storage from homomorphic identification protocols," in Proc. 15th Int. Conf. Theory Appl. Cryptology Inf. Security: Adv. Cryptol., 2009, pp. 319–333.
[8] C. Erway, A. K€upc€u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Security, 2009, pp. 213–222.
[9] R. Tamassia, "Authenticated data structures," in Proc. 11th Annu. Eur. Symp. Algorithm, 2003, pp. 2–5.
[10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. 14th Eur. Conf. Res. Comput. Security, 2009, pp. 355–370.
[11] F. Armknecht, J.-M. Bohli, G. O. Karame, Z. Liu, and C. A. Reuter, "Outsourced proofs of retrievability," in Proc. 21st ACM Conf. Comput. Commun. Security, 2014, pp. 831–843.
[12] H. Shachamand B. Waters,"Compact proofs of retrievability," J. Cryptol., vol. 26, no. 3, pp. 442–483, 2013.
[13] Z. Mo, Y. Zhou, and S. Chen, "A dynamic proof of retrievability(PoR) scheme with O(logn) complexity," in Proc. IEEE Int. Conf. Commun., 2012, pp. 912–916.
[14] E. Shi, E. Stefanov, and C. Papamanthou, "Practical dynamic proofs of retrievability," in Proc. 20th ACM Conf. Comput. Commun. Security, 2013, pp. 325–336.
[15] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in Proc. 18th ACM Conf. Comput. Commun. Security, 2011, pp. 491–500.
[16] J. Douceur, A. Adya, W. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in Proc. 22nd Int. Conf. Distrib. Comput. Syst., 2002, pp. 617–624.
[17] A. Juels and B. S. Kaliski, Jr, "PORs: Proofs of retrievability for large files," in Proc. 16th ACM Conf. Comput. Commun. Security, 2007, pp. 584–597.
[18] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. 15th Int. Conf. Theory Appl. Cryptol. Inf. Security: Adv. Cryptol., 2008, pp. 90–107. Fig. 15. Verification time of 4 kB block in the proof of storage phase, when the number of challenged blocks are 30 and 120, respectively.
[19] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. 6th Theory Cryptography Conf. Theory Cryptography, 2009, pp. 109–127.
[20] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," in Proc. 14th ACM Conf. Comput. Commun. Security, 2009, pp. 187–198.
[21] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in Proc. INFOCOM, 2010, pp. 1–9.
[22] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Security, vol. 14, no. 1, pp. 1–34, 2011.
[23] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Feb. 2012.
[24] J. Xu and E.-C. Chang, "Towards efficient proofs of retrievability," in Proc. 7th ACM Symp. Inf., Comput. Commun. Security, 2012, pp. 79–80.
[25] J. Chen, Y. Peng, R. Du, Q. Yuan, and M. Zheng, "Regeneratingcodes- based efficient remote data checking and repairing in cloud storage," in Proc. TrustCom, 2015, pp. 143–150.
[26] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," in Proc. 28th Annu. Comput. Security Appl. Conf., 2012, pp. 229–238.
[27] D. Cash, A. K€upc¸€u, and D. Wichs, "Dynamic proofs of retrievability via oblivious RAM," in Proc. EUROCRYPT, 2013, pp. 279–295.
[28] M. Azraoui, K. Elkhiyaoui, R. Molva, and M. € Onen, "StealthGuard: Proofs of retrievability with hidden watchdogs," in Proc. 19th Eur. Symp. Res. Comput. Security, 2014, pp. 239–256.
[29] Z. Ren, L. Wang, Q. Wang, and M. Xu, "Dynamic proofs of retrievability for coded cloud storage systems," IEEE Trans. Serv. Comput., 2015, Doi: 10.1109/TSC.2015.2481880.
[30] R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proc. ASIACCS, 2012, pp. 81–82.
[31] J. Xu, E.-C. Chang, and J. Zhou, "Weak leakage-resilient clientside deduplication of encrypted data in cloud storage," in Proc. 8th ACM Symp. Inf., Comput. Commun. Security, 2013, pp. 195–206.
[32] S. Keelveedhi, M. Bellare, and T. Ristenpart, "DupLESS: Serveraided encryption for deduplicated storage," in Proc. 22nd USENIX Conf. Security, 2013, pp. 179–194.
[33] J. Li, et al., "Secure deduplication with efficient and reliable convergent key management," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 1615–1625, Nov. 2014.
[34] J. Li, Y. K. Li, X. Chen, P. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 5, pp. 1206–1216, May 2015.

[35] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in Proc. 2nd ACM Conf. Data Appl. Security Privacy, 2012, pp. 1–12.
[36] R. Du, L. Deng, J. Chen, K. He, and M. Zheng, "Proofs of ownership and retrievability in cloud storage," in Proc. IEEE 13th Int. Conf. Trust, Security Privacy Comput. Commun., 2014, pp. 328–335.
[37] B. Wang, B. Li, and H. Li, "Public auditing for shared data with efficient user revocation in the cloud," in Proc. INFOCOM, 2013, pp. 2904–2912.
[38] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 43–56, Jan.-Mar. 2014.
[39] J. Yuan and S. Yu, "Efficient public integrity checking for cloud data sharing with multi-user modification," in Proc. INFOCOM, 2014, pp. 2121–2129.
[40] R. Gennaro and D. Wichs, "Fully homomorphic message authenticators," in Proc. 19th Int. Conf. Theory Appl. Cryptol. Inf. Security: Adv. Cryptol., 2013, pp. 301–320.
[41] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in Proc. 30th Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2011, pp. 149–168.
[42] D. Catalano, D. Fiore, and B. Warinschi, "Homomorphic signatures with efficient verification for polynomial functions," in Proc. 34th Annu. Cryptology Conf., 2014, pp. 371–389.
[43] A. Yun, J. H. Cheon, and Y. Kim, "On homomorphic signatures for network coding," IEEE Trans. Comput., vol. 59, no. 9, pp. 1295– 1296, Apr. 2010.
[44] C. Cheng and T. Jiang, "An efficient homomorphic MAC with small key size for authentication in network coding," IEEE Trans. Comput., vol. 62, no. 10, pp. 2096–2100, Jun. 2013.
[45] J. Chen, et al., "Dominating set and network coding-based routing in wireless mesh networks," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 2, pp. 423–433, Dec. 2015.
[46] D. Catalano, "Homomorphic signatures and message authentication codes," in Proc. 9th Int. Conf. Security Cryptography Netw., 2014, pp. 514–519.
[47] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2013, pp. 296–312.
[48] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in Proc. 33rd Annu. Cryptol. Conf., 2013, pp. 374–391.