Improving Trust Worthiness of data integrity verification in C-MAC Algorithm

C. Lochan yadav[1] Mr. J.S. Ananda Kumar[2]
[1]Student [2]Assistant Professor
[1,2]Department of Computer Applications
[1,2]KMM Institute of PG Studies, Tirupathi, India

Abstract— Many users store their data in the cloud storage and benefit from high quality applications and services from a common group of composition computing resources like networks, servers, storage, applications, and services, by these users can bypass the load of local data storage and protection. However, the fact that users no longer have physical control of the large size of data makes data trustworthiness protection in Cloud computing a challenging task, especially for users with constrained computing resources. Cloud computing is used by many software industries nowadays, since security is not provided in cloud, many organizations adopt their unique security structure. To improve the performance by making use of CMAC algorithm. The cloud must not command on user the responsibility to verify his/ her stored data. Taking this into consideration and relieve client form the overhead of data trustworthiness verification, we introduce an entity called the Third Party Auditor (TPA), which acts on behalf of client for data trustworthiness checking and send an alert to notify the status of the stored data. We further become the TPA to finish various assessing errands all the while and competently. In this paper, we are tying down the customer data and giving security by using CMAC Algorithm. We further become the TPA to finish various assessing errands all the while and competently. Wide-extent of security and execution examination exhibits the proposed plans is provably secure and significantly profitable.

Key words: Data Integrity, C MAC Algorithm, Trust Worthiness, Sensor Trustworthiness

I. INTRODUCTION

Cloud Computing is has gained popularity in modern years. Cloud facilitates the storage of various sets of data. Cloud is highly scalable when it comes to large data and can provide infinite computing resources on demand. Clients can use cloud services without any installation and the data uploaded on cloud is usable from any arrival of the world, all it needs to be accessed is a computer with active internet connection on it. The users can subscribe high quality services of data and software which resides entirely on the remote servers and enjoy the provision of on-demand provision of services. A new algorithm for producing message authenticating code MAC. The first utilization of the CMAC is presented in this paper. Through put has been the main design target. With the quick development of the cloud computing, cloud storage as a new generation of computing infrastructure has received more and more attention. At the same time, more and more cloud storage services spring up which can provide users with low cost but large data storage space. Although cloud storage can provide suitable storage and fast access to data at any time and etc., the paradigm of outsourced data service also introduces new security challenges. No matter how high degree of reliable measures cloud service providers would take, data loss or duplicity could happen in any storage infrastructure due to natural corruption and malicious corruption. Sometimes, in order to save storage space, the harmful storage service provider may delete the data that has not been accessed or accessed less, but claim that these data are completely stored on the remote servers. These misgivings have prompted the data owners to worry whether the outsourced data are intact or corrupted on the remote servers since they are poor of the direct control of these data[1]. Data integrity verification has been proposed to check the integrity of owners' remote stored data. These existing verification algorithms based on homomorphism technology can absolutely identify the corrupted data (i.e., each block or file) in the verification. Currently, many commercial cloud storage services, such as Google Drive and Drop box, utilize reduplication technique at the file/chunk level to store one copy of the same data hosted by different data owners. Liu et al. proposed one-tag checker to analyze the data integrity on encrypted cloud reduplication storage. Threats, passiveness and possibility for cloud computing[3] are describe, and then, we have designed a cloud computing security development lifecycle model to achieve protection and enable the user to take advantage of this technology as much as possible of security and face the risks that may be unprotected to data. A data integrity checking algorithm; which eliminates the third party auditing, is related to secure dynamic and static data from unauthorized observation, change, or intrusion.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the

main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MACs.

A new algorithm for producing message authenticating code (MAC) was recently proposed by NIST. The MAC protects both a message's integrity - by ensuring that a different MAC will be produced if the message has changed - as well as its authenticity - because only someone who knows the secret key could be able to generate a valid MAC. The proposed process incorporates a FIPS approved and secure block cipher algorithm which was standardized by NIST in May, 2005. The first implementation of the CMAC is presented in this paper. Throughput has been the main design target. The proposed implementation goes one step further introducing an optimized ciphering core to achieve competitive throughput for CMAC, compared to alternative MA

II. LITERATURE SURVEY

A. An Instruction for Handling Arbitrary-Length Messages with the CBC MAC

This paper reports some simple variants of CBC MAC. Unlike the basic CBC MAC, our algorithms correctly and efficiently handle messages of any bit length. In addition to our schemes, we introduce new techniques to demonstrate them secure. Our proofs are much simpler than prior work. We begin with some background.

B. Recommendation of block chipper modes of operation

This document specifies[2] five confidentiality modes of operation for symmetric key block cipher algorithms, such as the algorithm specified in FIPS Pub. 197: Advanced Encryption Standard (AES). The modes may be used in conjunction with any symmetric key block cipher algorithm that is accepted by a Federal Information Processing Standard

(FIPS). The five modes; the Electronic Codebook (ECB), Counter (CTR) modes, Cipher Feedback (CFB), Output Feedback (OFB), and Cipher Block Chaining (CBC) Counter (CTR) modes, can provide data quiet. Two FIPS publications already approve quiet modes of operation for two particular block cipher algorithms. FIPS Pub. 81 specify the ECB, CBC, CFB, and OFB modes of the Data Encryption Standard (DES). FIPS Pub. 46-3 approves the seven modes that are specified in ANSI X9.52. Four of these modes are equivalent to the ECB, CBC, CFB, and OFB modes with the Triple DES algorithm (TDEA) as the underlying block cipher; the other three modes in ANSI X9.52 are variants of the CBC, CFB, and OFB modes of Triple DES that use[4] interleaving or pipelining. Thus, there are three new elements in this guidance: 1) the addition of the four quiet modes in FIPS Pub 81 for use with any FIPS-approved block cipher; 2) the version of the requirements for these modes; and 3) the specification of an additional confidentiality mode, the CTR mode, for use with any FIPS-approved block cipher.

C. Cipher-based message authentication code (CMAC)

Cipher-based message authentication codes (or CMACs) are a tool for calculating message authentication codes using a block cipher coupled with a secret key. You can use a CMAC to verify both the integrity and authenticity of a message.

III. PROPOSED ALGORITHM

A. C-MAC Algorithm

The CMAC algorithm incorporates the usages of symmetric blocks chipper like AES or TDEA. In the implementation scheme in this paper existing implementations of these two block chipper algorithms were used as well as similar implementations there were developed by the authors. Officially there are two MAC algorithms (OMAC1 and OMAC2) which are both essentially the same except for a small tweak[3]. OMAC1 is identical to CMAC, which became an NIST recommendation in May 2005. It is free for all uses: it is not protected by any patents. In cryptography, CMAC (Cipher based Message Authentication Code) is a block cipher-based message verification code algorithm. It may be used to provide support of the authenticity and, hence, the integrity of binary data. This mode of operation fixes security insufficiency of CBC-MAC (CBC-MAC is secure only for fixed-length messages).[5] The core of the CMAC algorithm is a inequality of CBC-MAC that Black and Rogaway proposed and examine under the name XCBC and submitted to NIST. The XCBC algorithm efficiently addresses the security insufficiency of CBC-MAC, but requires three keys. Iwata and Kurosawa proposed a development of XCBC and named the resulting algorithm One-Key CBC-MAC (OMAC) in their papers. They later submitted OMAC1, a clarification of OMAC, and additional security analysis. The OMAC algorithm decreases the amount of key material required for XCBC.

B. Architecture of C MAC

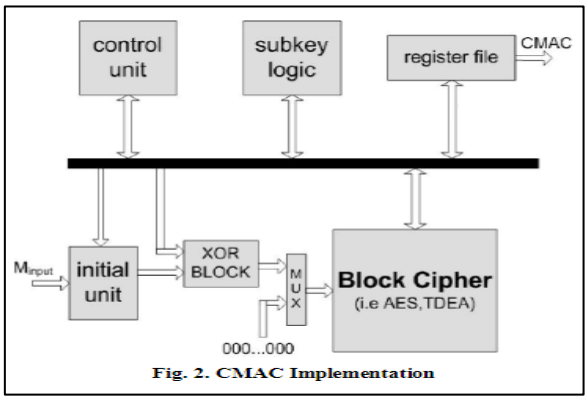

Fig. 1: CMAC Implementation

IV. RESULT & ANALYSIS

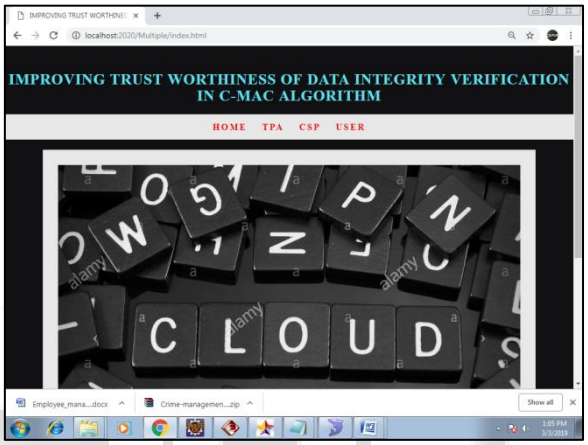

Fig. 2: Home page

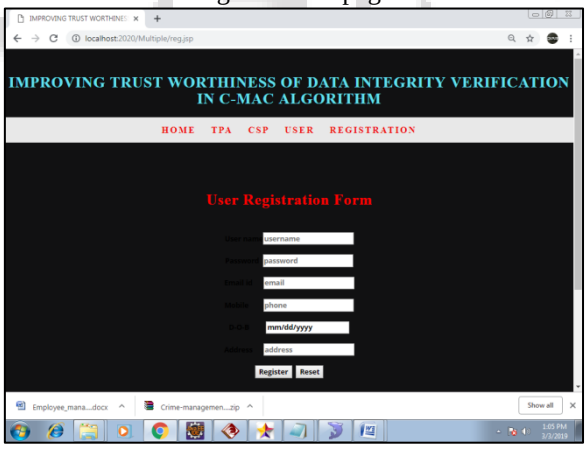

Fig. 3: By using this module any User can register In the home page

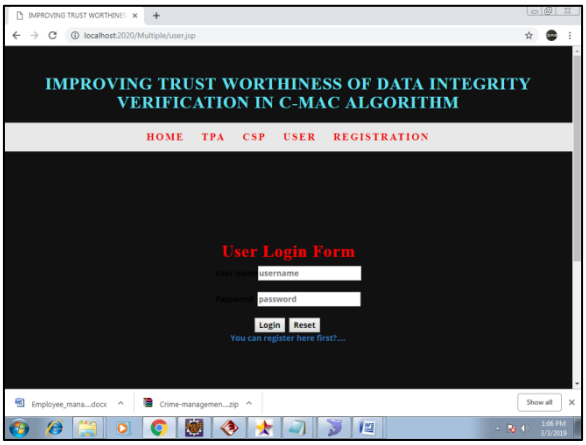

Fig. 4: User registration is success then go to login page

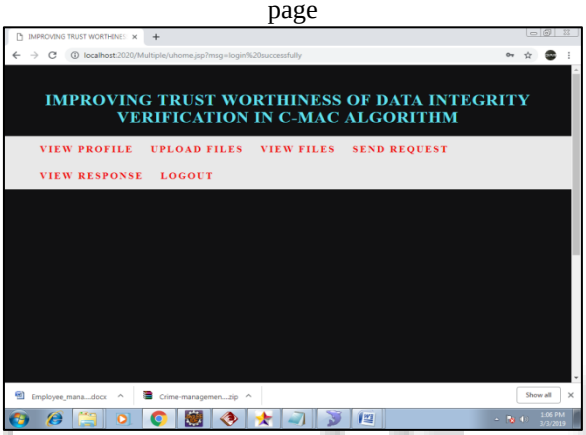

Fig. 5: after login is success then go to user home page

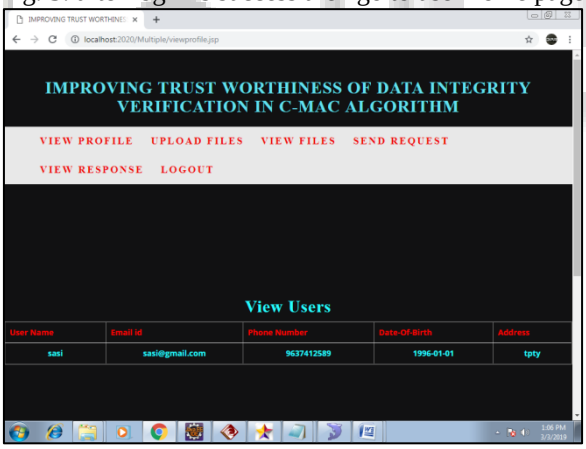

Fig 6: we can view user details

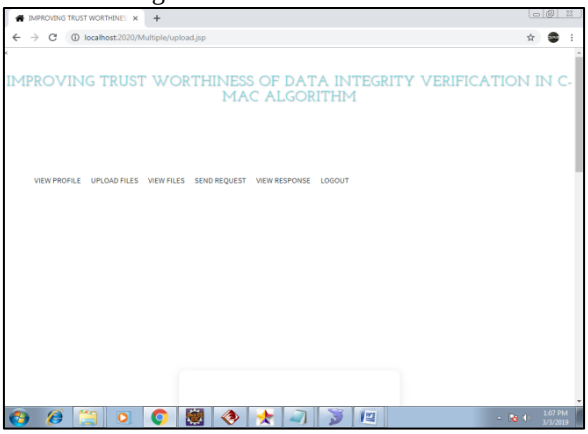

Fig. 7: we can upload the user details

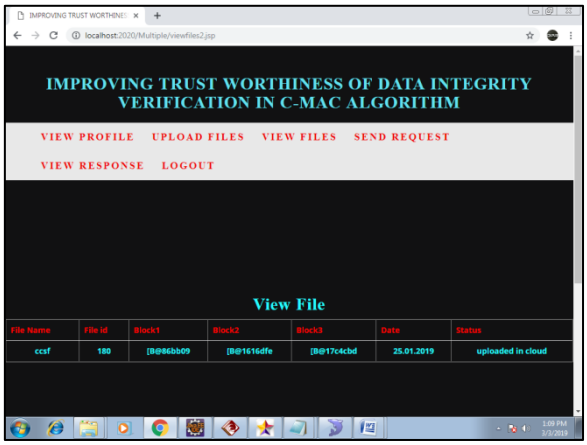

Fig. 8: we can view the files

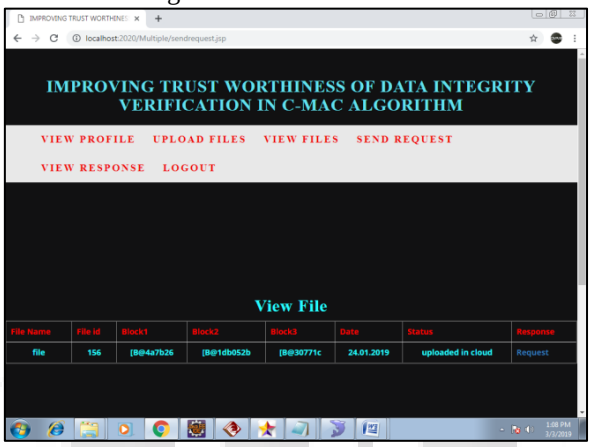

Fig. 9: view the files and we can send request to TPA

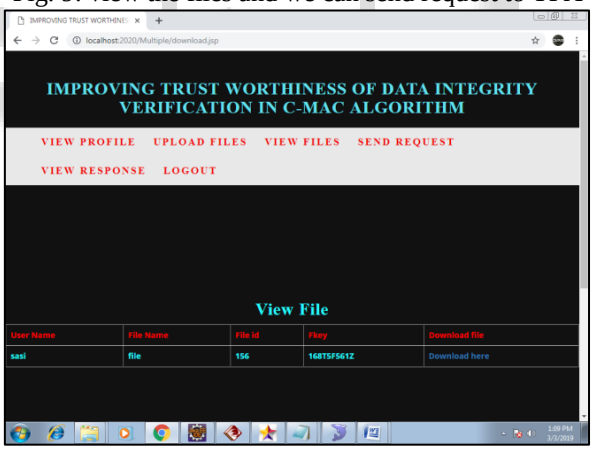

Fig. 10: Verify the request and then view response

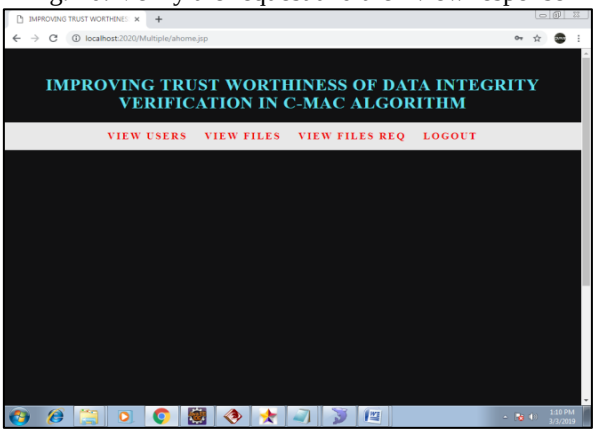

Fig. 11: Enter the TPA home

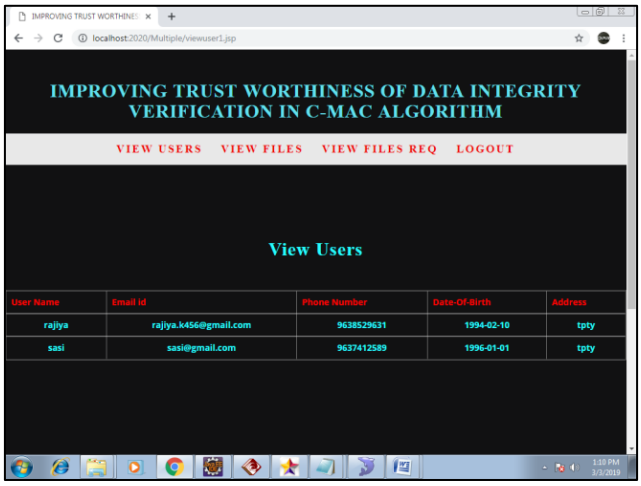

Fig 12: All user details are viewed

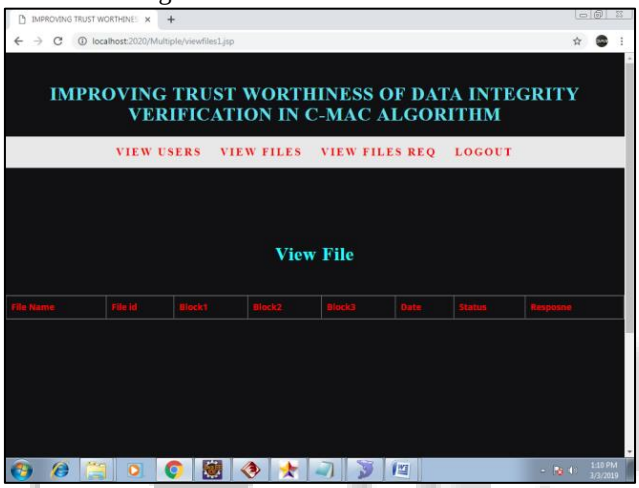

Fig 13: upload all view files are viewed View file request

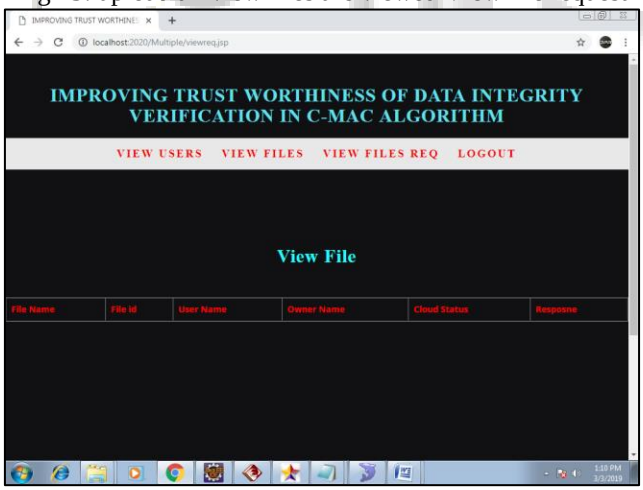

Fig 14: after complete view file upload then send request

V. Conclusion and Future Scope

Although the cloud deduplication storage can reduce storage cost, the reliability of files will decrease while traditional verification schemes are applied into the integrity verification of version files. The reason is that these schemes do not distinguish version files and treat them the same as common files. These common files can utilize the traditional verification schemes based on the cloud deduplication storage to decrease the verification[6] cost. However, these version files have the same data among each others so that the traditional verification schemes cannot still be used directly. In this paper, we proposed an efficient integrity verification algorithm for remote data storage of version files. This algorithm adopts the combination storage model based on the full storage and incremental storage to reduce the size of file storage and tag storage. Also, it reduces the transmission overhead in the verification due to less verification costs at the same time. The chained key is used to improve security of the storage keys of different version files. The data security in the process of the verification is assured by applying BLS technology. From theoretical and experimental analysis, we can figure out that the algorithm effectively expands the coverage of verified files, improves the efficiency of file verification, and meets the security request at the same time. In the future, we will further make a trade-off between the optimal storage of version files and data integrity protecting to meet the needs of rapid restoration of any version file at any time.

References

[1] *Dworkin, M J (2016).* "Recommendation for block cipher modes of operation" (PDF). doi:10.6028/nist.sp.800-38b
[2] *Black, John; Rogaway, Phillip (2000-08-20). Advances in Cryptology – CRYPTO 2000. Springer, Berlin, Heidelberg. pp. 197–215.* doi:10.1007/3-540-44598-6_12. ISBN 978-3540445982.
[3] *Black, J; Rogaway, P. "A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC" (PDF).*
[4] *Iwata, Tetsu; Kurosawa, Kaoru (2003-02-24). "OMAC: One-Key CBC MAC". Fast Software Encryption. Lecture Notes in Computer Science. 2887. Springer, Berlin, Heidelberg. pp. 129–153.* doi:10.1007/978-3-540-39887-5_11. ISBN 978-3-540-20449-7.
[5] *Iwata, Tetsu; Kurosawa, Kaoru (2003).* "OMAC: One-Key CBC MAC – Addendum" (PDF).
[6] *Iwata, Tetsu; Kurosawa, Kaoru (2003-12-08). "Stronger Security Bounds for OMAC, TMAC, and XCBC". In Johansson, Thomas; Maitra, Subhamoy. Progress in Cryptology – INDOCRYPT 2003. Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp. 402–415.* CiteSeerX 10.1.1.13.8229. doi:10.1007/978-3-540-24582-7_30. ISBN 9783540206095.

