EDIVP: An efficient data integrity verification protocol using elliptic curve cryptography for vehicular clouds

S. Milton Ganesh and R.S. Arun Kumar
University College of Engineering Tindivanam, Anna University,

Abstract - An elliptic curve cryptography based data integrity verification scheme for the vehicular cloud storage networks is proposed in this research work. With the invention of IoT, cloud storage, the vehicular Adhoc network (VANET) is becoming smart enough such that the vehicles collect the on-road information like traffic density, road conditions and continuously transfer them to the vehicular cloud storage for providing a better service and for the efficient traffic management. Very few data integrity verification works for VANETs have been proposed by authors. Hence, we have proposed a novel data integrity verification procedure for vehicles in this research work. The design of this protocol is based on the security strength and the properties of the pairing operations and the strength of elliptic curve discrete logarithm problem. The security analysis for this work is the protocol's completeness and its soundness during attacks. This protocol when implemented will certainly aid the VANET for safer and secure data storage in the vehicular cloud storage servers.

Index Terms - cloud storage, data integrity verification, pairing based cryptography, security

I. Introduction

The invention of cloud servers in the early twentieth century paved the way for permanent data movement to cloud storage from all the data processing elements like personal computers, data sharing applications, business websites and among many other applications. [1,2]. The VANETs are increasingly become smart with the advent of IoT sensors. The smart vehicles like cars, trucks are provided with on-board units (OBUs) [3] to collect information about the traffic conditions like vehicles density in the path, delay in the expected timing to reach the destination, road conditions, vehicle safety and driver's health conditions [4].

The OBU in a VANET can collect the on-road data and send continuously in order to store them in the cloud for providing better road information to the incoming vehicles in that road [5]. But, unexpectedly, the cloud servers in a data center may lose this sensitive information from the vehicle's OBUs and pretend that the data is completely intact in its storage servers [5,6]. An incident of this kind can be cited due to Amazon customers' data loss during power outage in Amazon's data centers [7]. It took more than one year for the customers to learn the data loss which led to the close of the businesses of many clients [8].

Thus, ensuring the safety of the data uploaded to the cloud storage is a very important concern [9,10]. Hence, in this research work, a vehicle uploads the data to the vehicular cloud server for which the service has already been requested. Once the data are stored, the vehicle asks a third party auditor to do the data verification task on behalf of this vehicle to ensure data integrity and to identify the vehicular cloud server if it hides the data loss and pretends to hold them intact.

II. Literature Survey

Use the full justify option for your paragraphs, and use two-columns for all text. The vehicular Adhoc networks are a topic of growing importance in the recent years due to sophisticated devices and the need for quick transport for providing better service to the businesses [11]. The idea of verifying the data has been in practice more nearly two decades [12]. One of the oldest works is from Deswarte et al.[13] whose work in 2003 was succeded by Filho et al's work [14] in 2006 to reduce the server side computational complexity. One of the famous works in the recent past are from Zhu et al. [15] in 2013. Zhu and others try to provide a novel data verification protocol in which new data blocks can be uploaded to the cloud server after the initial data upload. The existing blocks can be altered as well. Another notable contribution is from Yu et al. in 2017 [16]. Yu and others make use of identity based cryptography to preserve the user identity from the TPA and the cloud server during the data verification scenario.

A much more recent work by Zhou et al. in 2023 [17] for vehicular cloud computing provides a practical data audit scheme. The authors use a bloom filter to capture the data blocks which are the reason for the data verification failure. This method would be very handy to identify the corrupted blocks and address the issue quickly. Another prominent work by Gai et al. in 2023 [18] preserves the user privacy using identity-based cryptography and supports

the of data audit from one vehicle to another vehicle. The proposed work does not support transfer of audit facility from one vehicle to another vehicle nor does it provide support for capturing the block numbers during data loss. But, it provides a novel method to verify the data intactness with a minimal computational cost.
Section 3 provides the architecture and the design of the proposed protocol. Section 4 provides the security analysis and the section 4 is about the results and discussion. Section 5 concludes this research work.

III. The Proposed EDIVP protocol

The proposed Efficient Data Integrity Verification Protocol (EDIVP) architecture is shown in the Figure 1 and it has three components such as data owner which is the vehicle (with on-board unit) in this research work. The cloud storage server present in a distant data center into which the vehicle is going to upload data during its transit in the VANET. The third party auditor is a computer or a software-as-a-service which is capable of doing the data integrity verification when requested by the vehicle.

A vehicle is the data owner and it collects the data through its OBU accumulates the data until it forms a fixed file size. Then, the vehicle splits the file into n blocks and uploads it to the vehicle cloud storage. At a later time, the data owner requests the TPA to do the audit for which the TPA sends a challenge for a very small percentage of blocks. The cloud server returns a response which is verified by the TPA and the status is sent to the data owner as well.

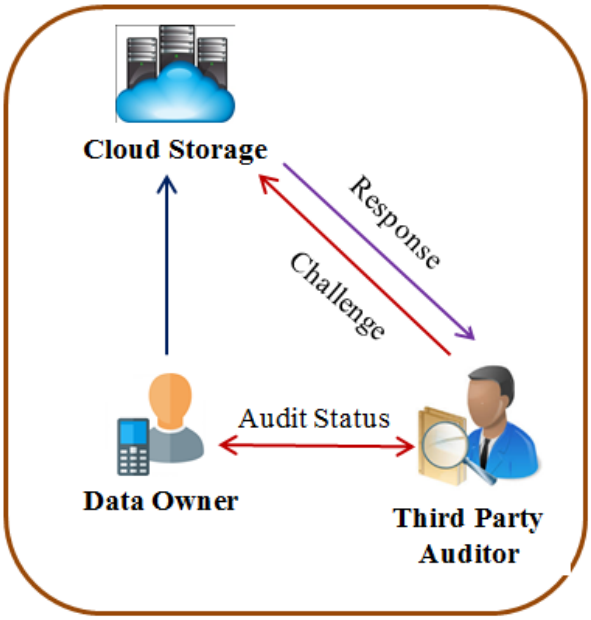
FIGURE I
ARCHITECTURE OF THE DATA INTEGRITY VERIFICATION PROTOCOL

- **Initialization**: Use the style "First Paragraph" for the first paragraph in each section, and "Text" for succeeding paragraphs in the section. The vehicle initializes the system with two multiplicative cyclic groups G and G_T of prime order p. Let g be a generator in G, is a random number in Z_q^* and $Y = g^x$. Let e(,.,) be the pairing operation such that e$(G * G) \rightarrow G_T$. Now, the vehicle publicly announces the parameters $P_{pub} = G, G_T, q, Y, e(,.,)$ and keeps x as its secret key.
- **Authenticator generation**: The vehicle divides the file F into n blocks as $\{m_i\}_{i=1}^{n}$. For the block m_i, it computes the block signature $S_i = Y.g^{\frac{m_i.H(m_i \| i)}{x}}$. Similarly for each of the block $m_1, m_2, \dots, m_n$, the signature is computed. Finally, the vehicle uploads the blocks $\{m_i\}_{i=1}^{n}$ and the corresponding signatures $\{S_i\}_{i=1}^{n}$ to the vehicle cloud server VCS.
- **Challenge**: This is the first step of the data audit process. In this phase, the vehicle requests the TPA to do the audit in its behalf by giving the File id F_{id} and the number of blocks n. Now, the TPA randomly chooses a small subset of randomly selected s block numbers represented by $i_1, i_2, \dots, i_c$ from n. Then, the TPA also selects each of $v_1, v_2, \dots, v_c$ from Z_q^* and sends chal $= \{i, v_i\}_{i=1}^{c}$ to the VCS as the challenge.
  - **Response**: This VCS receives the challenge chal $= \{i, v_i\}_{i=1}^{c}$. Now, for each of the block in $i_1, i_2, \dots, i_c$ it retrieves $S_{i_1}, S_{i_2}, \dots, S_{i_c}$ and computes the aggregated signature as $\sigma = \prod_{i=1}^{c} S_i^{v_i}$. Also, it retrieves $m_{i_1}, m_{i_2}, \dots, m_{i_c}$ and computes $\lambda = \sum_{i=1}^{c} (m_i . v_i)$. At last, VCS sends (σ, λ) as the proof of the perfect possession of the data blocks and their signatures.
  - **Data Integrity verification**: This The TPA receives the proof (σ, λ) and verifies it as

$$e(\sigma, Y) = e\left(\prod_{i=1}^{c} Y^{v_i}, Y\right) . e\left(\prod_{i=1}^{c} g^{\lambda}, g^{H(m_i \| i)}\right)$$

The proof for this equation can be understood as

$$e(\sigma, Y) = e\left(\prod_{i=1}^{c} S_i^{v_i}, g^{x}\right)$$



$$= e\left(\prod_{i=1}^{c}\left(Y.g^{\frac{m_i.H(m_i\|i)}{x}}\right)^{v_i}, g^x\right)$$

$$= e\left(\prod_{i=1}^{c} Y^{v_i}.g^{\frac{m_i.H(m_i\|i)}{x}.v_i}, g^x\right)$$

$$= e\left(\prod_{i=1}^{c} Y^{v_i}, g^x\right).e\left(\prod_{i=1}^{c} g^{\frac{m_i.H(m_i\|i)}{x}.v_i}, g^x\right)$$

$$= e\left(\prod_{i=1}^{c} Y^{v_i}, Y\right).e\left(\prod_{i=1}^{c} g^{m_i.v_i}, g^{\frac{H(m_i\|i)}{x}.x}\right)$$

$$= e\left(\prod_{i=1}^{c} Y^{v_i}, Y\right).e\left(\prod_{i=1}^{c} g^{\lambda}, g^{H(m_i\|i)}\right)$$

Now, after verification, the TPA will give the status of the audit as 0 or 1 to the vehicle where 0 represents failure and 1 represents success.

IV. Security analysis

- **Completeness**: The proposed protocol is complete as the proof for the mathematical equation $e(\sigma, Y) = e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{\lambda}, g^{H(m_i\|i)}\right)$ is a valid one. Since this entire auditing work is based on the mathematical proof based verification procedures, this proof of data possession ascertains the completeness of this data integrity verification protocol.

- **Sadness:** It refers to the fact that, the VCS should not be able to pass the integrity verification if it loses either the data blocks $m_{i_1}, m_{i_2}, \dots, m_{i_c}$ or its signatures $S_{i_1}, S_{i_2}, \dots, S_{i_c}$. Let us assume that, during to power outage as happened in the Amazon data centers in 2011, some of the blocks of $i_1, i_2, \dots, i_c$ are lost or corrupted. But, to keep the reputation, the VCS wants to hide this fact and tries to pass audit by the TPA. Hence, during an audit, it creates the response manipulated equation $e(\sigma, Y) = e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{\lambda'}, g^{H(m_i\|i)}\right)$, the L.H.S. $e(\sigma, Y)$ will contain the expected and valid value. But,+ the R.H.S. $e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{\lambda'}, g^{H(m_i\|i)}\right)$ certainly will not produce the expected value as λ' in this equation is based on ${g^{m_i}}'.v_i$ as $e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{{m_i}'.v_i}, g^{\frac{H(m_i\|i)}{x}.x}\right)$. Hence, L.H.S. $\neq$ R.H.S. which concludes that the VCS has not passed the verification procedure. Moreover, let us assume that, the signatures are corrupted in the data center. In this case, $\sigma' = \prod_{i=1}^{c} {S_i}'^{v_i}$. Now, during the audit scenario, in

the equation $e(\sigma', Y) = e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{\lambda}, g^{H(m_i \| i)}\right)$, L.H.S. will evaluate to an unexpected and ***V. Results***

- The invalid element in G_T. But, R.H.S. will generate a valid value. Nevertheless, the L.H.S. ≠ R.H.S. Therefore, the VCS cannot pass the integrity verification if it loses the data blocks or signatures or both.

computations involved in this research work are due to time taken hashing operation, and the operations in G_1 such as pairing operation, point exponentiation operation, point addition operation. This implementation is realized through simulation using pbc library pbc-0.5.13 installed in Core i3 processor with 2.4GHz speed, 4 GB of primary memory and 500 GB of Hard Disk Drive. Windows 7 operating system was used with Cygwin software tool for running the pbc library on top of it. The time for one pairing operation is 1.6ms, one hash operation for SHA-1 is 2.7ms, point exponentiation in G_1 is 0.7ms, the addition of two points in G_1 is 0.2ms and point multiplication is 0.6ms.

TABLE I

COMPUTATION COMPLEXITY OF INTEGRITY VERIFICATION OPERATIONS

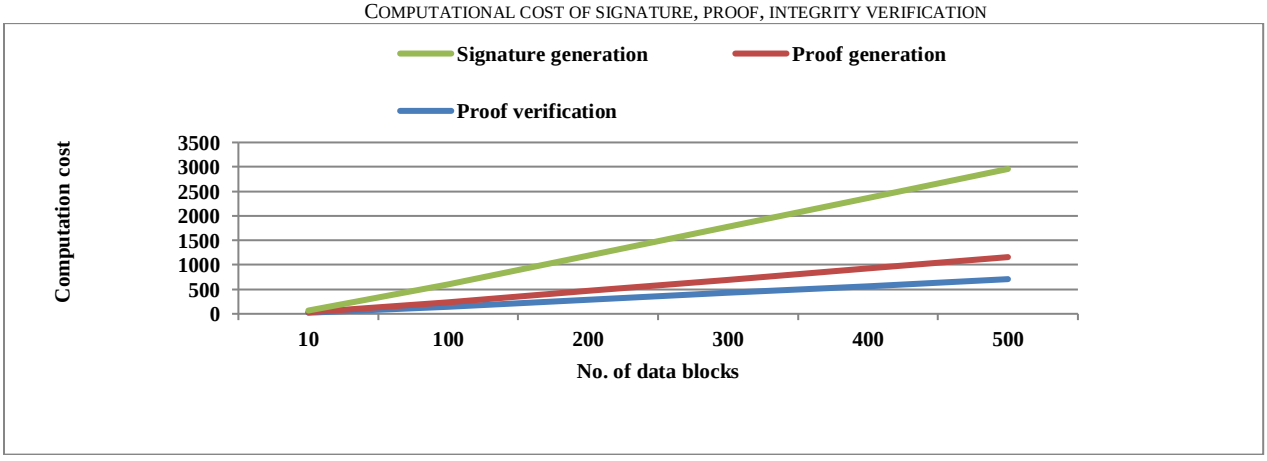
Sl.No.	Operation	Time complexity	1 block	10 blocks	20 blocks
1.	Signature generation	$T_h + T_{pe} + T_{pa}$	3.6ms	35ms	73ms
2.	Proof generation	$T_{pe} + T_{pa}$	0.9ms	8.8ms	17ms
3.	Proof verification	$3T_p + 2cT_{pe} + T_h + T_{pa}$	9.1ms	21.7ms	35.7ms

The time for signature generation is given by $S_i = Y.g^{\frac{m_i.H(m_i \| i)}{x}}$ which consists of 1 hash operation, 1 point exponentiation of g in G_1 and point addition of Y and $g^{\frac{m_i.H(m_i \| i)}{x}}$. The time for proof generation is based on $\sigma = \prod_{i=1}^{c} S_i{}^{v_i}$ which involves c point exponentiations and $c-1$ point additions. Also, proof generation consists of time for computing $\lambda = \sum_{i=1}^{c}(m_i.v_i)$ in which we do only integer additions which are ignored in this research work as these operations are very relatively fast compared to point additions. The time taken during proof verification is for $e(\sigma, Y) = e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{\lambda}, g^{H(m_i \| i)}\right)$ which consists of 3 pairing operations, c point exponentiations for $\prod_{i=1}^{c} Y^{v_i}$, c point exponentiations for $\prod_{i=1}^{c} g^{\lambda}$, one hash operation for $H(m_i \| i)$ and one point addition for $e(\prod_{i=1}^{c} Y^{v_i}, Y).e\left(\prod_{i=1}^{c} g^{\lambda}, g^{H(m_i \| i)}\right)$.

Figure 2 shows the computational cost for the signature generation, the proof generation and the proof verification. The graph shows the comparison with respect to 10, 100, 200, 300, 400 and 500 blocks of file data. The cost of signature generation is relatively very high compared to proof generation and proof verification. Similarly, the cost of proof verification is slightly high compared to proof generation. The computational cost for proof generation is the least one in this research work.

FIGURE II

COMPUTATIONAL COST OF SIGNATURE, PROOF, INTEGRITY VERIFICATION



VI. conclusion

This data integrity verification work is based on pairing based cryptography which utilizes the strength of the elliptic curve discrete logarithm problem and properties of pairing operations. In this research work, we have carefully designed the protocol during the signature generation, the proof generation and the proof verification process in order to reduce the computational overheads for a moving vehicle in the road. The efficiency in signature generation will certainly help the vehicle to compute the signature will minimal computational overhead and the efficiency in proof verification will enable the TPA to quickly complete the auditing process. Moreover, the proposed protocol is complete and sound which ensures that, it can be used in a vehicle cloud scenario where data upload to the cloud servers is mandatory. In future, this research work can be extended to support the audit requests of multiple users and storage of multiple copies of data blocks in multiple cloud servers.

REFERENCES

[1] M.Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson , A. Rabkin, I. Stocia and M. Zaharia, "A view of cloud computing", Comm., ACM, Vol. 53, no. 4, pp. 50-58, 2010.

[2] S. Kamara and K. Lauter, "Cryptographic Cloud Storage", Proc. Intl. conf. Financial Cryptography and Data Security (FC), pp. 136-149, 2010.

[3] S. Sharms and A. Kaul, "VANETS cloud: Architecture, applications, challenges, and issues,", Arch. Comput. Methods Eng., Vol. 28, pp. 2081-2102, 2021.

[4] M.S. Sheikh, J. Liang, and W. wang, "Security and privacy in the vehicular Adhoc network vehicle cloud computing: A survey", Wireless Communications and Mobile Comput., Vol. 2020, 2020.

[5] S.Das, A.R. Tripath and A.Tripathy, "Vehicular cloud computing: Architecture and its challenges," in Smart Intelligent Computing and Applications, Springer, Vol. 2, pp. 481-493, 2022.

[6] P.Kohli, S. Sharma, and P. Matta, "Security of cloud-based vehicular ad-hoc communication networks, challenges and solutions," in proc. IEEE 6th Int. Conf. Wireless Commun., Signal, Process., Net., pp. 283-287, 2021.

[7] S.M. Ganesh, S.P. Manikandan, "An efficient integrity verification and authentication scheme over the remote data in the public clouds', Security and Communication Networks, Wiley, Vol. 2020.

[8] G. Ateniese et al., " Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., pp. 598-609, 2007.

[9] Huang, K., Xian, M., Fu S., Liu, J.: Securing the cloud storage audit service: defending against frame and collude attacks of third party auditor. IET Communications. 8(12), 2106-2113 (2014)

[10] Ren, Z., Wang, L., Wang, Q.: Dynamic proofs of retrievability for coded cloud storage systems. IEEE Transactions on Services Computing, 11(4), 685-698 (2018)

[11] J. Shen, D. Liu, X. Chen, J. Li, N. Kumar, and P. Vijayakumar, "Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs", IEEE Trans. Veh. Technol., Vol. 69, no. 1, pp. 807-8177, 2020.

[12] Jensen, M., Schwenk, J., Gruschk, N., and Iacono, L.L.: On Technical Security Issues in Cloud Computing. IEEE conference on Cloud Computing, CLOUD'09, 109-116, (2006)

[13] Y. Deswarte, J.-J. Quisquater, and A. Saidane. Remote integrity checking. In Proc. of Conference on Integrity and Internal Control in Information Systems (IICIS'03), November 2003.

[14] D. L. G. Filho and P. S. L. M. Baretto. Demonstrating data possession and uncheatable data transfer. IACR ePrint archive, 2006. Report 2006/150, http://eprint.iacr.org/2006/150.

[15] Y. Zhu, G. -J. Ahn, H. Hu, S. S. Yau, H. G. An and C. -J. Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," in IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 227-238, April-June 2013, doi: 10.1109/TSC.2011.51.

[16] Y. Yu et al., "Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 767-778, April 2017, doi: 10.1109/TIFS.2016.2615853.

AUTHOR INFORMATION

S.Milton Ganesh, Assistant Professor, Department of Computer Science and Engineering, University College of Engineering Tindivanam, Tamil Nadu, India.

R.S.Arun Kumar, Teaching Fellow, Department of Computer Science and Engineering, University College of Engineering Tindivam, Tamil Nadu, India.