International Journal of Network Security & Its Applications (IJNSA) Vol. 10, No.6, November 2018

SECURE THIRD PARTY AUDITOR (TPA) FOR ENSURING DATA INTEGRITY IN FOG COMPUTING

KashifMunir and Lawan A. Mohammed

University of Hafr Al Batin, KSA

ABSTRACT:

Fog computing is an extended version of Cloud computing. It minimizes the latency by incorporating Fog servers as intermediates between Cloud Server and users. It also provides services similar to Cloud like Storage, Computation and resources utilization and security.Fog systems are capable of processing large amounts of data locally, operate on-premise, are fully portable, and can be installed on the heterogeneous hardware. These features make the Fog platform highly suitable for time and location-sensitive applications. For example, the Internet of Things (IoT) devices isrequired to quickly process a large amount of data. The Significance of enterprise data and increased access rates from low-resource terminal devices demands for reliable and low- cost authentication protocols. Lots of researchers have proposed authentication protocols with varied efficiencies.As a part of our contribution, we propose a protocol to ensure data integrity which is best suited for fog computing environment.

KEYWORDS:

Protocol, Authentication,Fog Computing, Security Threats, IoT

1. INTRODUCTION

Due to the significant physical distance between cloud service provider's Data Centers and End User (EU)[2], cloud computing suffers from substantial end-to-end delay, traffic congestion, processing of huge amount of data, and communication cost. Although few companies like Apple are moving towards more environmental friendly 100 percent renewable DCs with the wind, solar, and geothermal energy, the carbon emission from DCs due to the round-the-clock operation will dominate on global carbon footprint. Fog computing emerges as an alternative to traditional cloud computing to support geographically distributed, latency sensitive, and Quality-of-Service (QoS)-aware Internet of Things (IoT) applications. Fog computing was first initiated by Cisco to extend the cloud computing to the edge of a network as per [5]. Fog computing is a highly virtualized platform [4] that provides computing, storage, and networking services between EU and DC of the traditional cloud computing. According to [3], Fog computing has thefollowing

- Low latency and location awareness
- Supports geographic distribution
- End device mobility
- The Capacity of processing a high number of nodes
- Wireless access
- Real-time applications
- Heterogeneity

The OpenFog Consortium [6], a consortium of high-tech giant companies and academic institutions across the world, aims to standardize and promote fog computing in various fields.

Many technology enablers for fog computing in various fieldsare discussed by [10].Some of the examples are EU experience by GE, TOYOTA, BMW, etc., network equipment like switches, gateway by Cisco, Huawei, Ericsson, etc. The current research trends reflect the tremendous potential of fog computing towards sustainable development in the global IoT market.

The term *Fog Computing* was introduced by Cisco Systems as a new model to bridge the gap between cloud computing and the Internet of Things (IoT) devices (SABU, n.d.). It is a decentralized computing and it is a highly virtualized platform which provides computation, storage, and networking services between IoT devices and traditional cloud servers. Thus, the cloud-based services can be extended closer to the IoT devices [9].

According to [11], fog computing extends a substantial amount of data storage, computing, communication, and networking of cloud computing near to the end devices. Due to close integration with the front-end intelligence enabled end devices, fog computing enhances the overall system efficiency, after that improving the performance of critical cyber-physical systems. An important key difference is that cloud computing tries to optimize resource in a global view, whereas fog computing organizes and manages the local virtual cluster.

According to [3], fog computing is considered as an extension of the cloud computing to the edge of the network, which is a highly virtualized platform of the resource pool that provides computation, storage, and networking services to nearby end users. As per [25], fog computing as "*a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties. These tasks can beused for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so.*" Although those definitions are still debatable before, fog computing is no longer a buzzword.

2. RELATED WORK

In order to propose a Protocol for data integrity in fog computing, we have reviewed some existing schemes. In this section, we briefly summarize some few ones among those that address fog computing.

In [17], it was have proposed a PDP (Provable Data Possession) model for verifying if an untrusted server stores a client'sdata. The data owner processes the data file to generate some metadata to store it locally. The file is then sent to be stored in the server, and the owner may delete the local copy of the file. Clients may encrypt the file earlier to upload it to the storage. After that, the client asks the server to reduce a proof of the stored file, which it returns to the client. For security issues, the client verifies the "yes" or "no" answer from the server to make sure from his behavior. The main drawback of this approach is that they have encrypted the whole file F and then permutation is done which incurs a lot of processing overhead. In [8], "Ensuring the data integrity in cloud data storage", proposed a remote data integrity checking protocol based on HLAs and RSA signature with the support of public verifiability. The support of public verifiability makes the protocol very flexiblesince the user can commission the data possession to check the TPA. The main drawback of this approach is that user has to depend onthe third party. There should be trust between TPA and user. Sentinel-based Approach by

SravanThe approach developed by these authors overcomes the drawback of processing overhead due to encryption and permutation in RSA-based approach. They have developed the scheme which does not involve the encryption of the whole data. The scheme encrypts only a few bits of data per data block thus reducing the computational overhead on clients. The verifierOnly needs to store a single cryptographic key. The verifier before storing the file at the archive preprocesses the file and appends some metadata to the fileand stores at the archive. At the time of verification, the verifier uses this metadata to verify the integrity of the data. The approach is divided into two phases Setup Phase and Verification phase. The advantage of this approach is that this algorithm reduces computational overhead because this scheme is not encrypting the whole file. The number of bits which have been encrypted depends on the client [4].

The main problem associated with this approach is that checking the integrity of each bit incurs a lot of network overhead. To check the integrity of each bit causesa large number of challenge request from the client to cloud archive which is time- consuming too.

In a Fog platform, both security and performance factors are considered in conjunction, and mechanisms such as the encryption methodologies known as *fully homomorphic* by [12] and *Fan-Vercauteren somewhat homomorphic* by *[13]* can be used to secure the data. These schemes consist of a hybrid of symmetric and public-key encryption algorithms, as well as other variants of attribute-based encryption. As homomorphic encryption permits normal operations without decrypting the data, the reduction in key distribution will maintain the privacy of data. Other research work provides a similar framework to secure smart grids, regardless of Fog computing, called the Efficient and Privacy- Preserving Aggregation (EPPA) scheme by [14]. The system performs data aggregation based on the homomorphic Paillier cryptosystem. As the homomorphic ability of encryption makes it possible for local network gateways to perform an operation on cipher-text without decryption, it reduces the authentication cost (in terms of processing power) while maintaining the secrecy of data.

Security in fog computing is one of the main concerns, and without proper security and privacy preserving mechanisms, fog computing will not be adopteddespite its usefulness [23]. Table 1 has been compared different cryptography algorithms which have been used to protect the user's data on fog computing. Sincewe have been reviewed different papers and we didn't conduct an experiment. In this section, we will establish a pattern in reviewed work and we will compare all used techniques and cryptography algorithms. Cryptography algorithms havebeen compared in general and according to the reviewed papers, without comparing algorithm characteristics such as speed, key length, etc.

Table 1: Existing Fog Computing Cryptography Algorithms Comparison[1].

Cryptography Algorithm	**Strengths**	**Limitation**
DES	Easy to implement [24]	DES algorithm has been cracked, it is not secure anymore (Van De Zande, 2001). It also showed slow performance on software [24].
3DES	Easy and efficient [24].	3DES algorithm has been cracked, it is not secure anymore. It also showed slow performance on software [24].

AES	It shows the best results for high processing devices. It has been Showed the best results among all symmetric key algorithms in term of security, flexibility, memory usage, and performance [24].	AES is not considered s best encryption algorithm for encrypting low processing devices [24].
RSA	RSA is secure cryptography algorithm. It is faster encryption than ElGamal [24].	The same security level can be obtained by ECC with less key size [24]. Also, key generation step is complex.
ECC	It has been provided good security with smaller key sizes than RSA, it's useful in many fog applications and it's more suitable for fog environments [21].	It is much less explored and known [21].
BLOWFISH.	Appropriate in applications where the key is not changed frequently and it's more efficient in softwares[24].	BLOWFISH could be time consuming in some implementations regarding its complicated encryption functions. Also, once the source code have been obtained, it will be easy to hacked (Kurumanale et al., 2017).
RC5.	RC5 is simple, easy to implement, fast, and requires low memory [27].	It can be broke if the used key is too short [27].
ElGamal.	ElGamal showed faster decryption than RSA [24].	It's slow and needs more time in digital signature .
ECDH	ECDH algorithm showed significant performance with its lightweight, robustness, and low power consumption [26].	

3. DATA SECURITY RISKS

Security is one of the main attributes of information technology. By access to fog services and data through the Internet, the importance of security is increased because data is a bigger amount of security risk. Within various fog service models, data responsibility is shared by more groups. Data is the most important for the organization which owns them and uses the fog services for storage. That is the reason why the organization should be aware of all risks which exist for data stored in fog environment.

A. BASIC ATTRIBUTES OF DATA SECURITY

As defined by [15], one of the basic data security concepts is the trio CIA shown in Figure 1.*Confidentiality, integrity* and *availability*, also known as the CIA triad, is a model designed to guide policies for information security within an organization. The model is also sometimes referred to as the AIC triad (availability, integrity and confidentiality) to avoid confusion with the Central Intelligence Agency. The elements of the triad are considered the three most crucial components of security.

Confidentiality
Integrity
Availability

Figure 1: The Security Requirements Triad - CIA

As long as confidentiality is concerned it represents information protection from revealing it to unauthorized persons. A basic way how to ensure confidentiality is setting access authorization to data and introducing policies which define access right to data. The key element of confidentiality protection is cryptography. It enables that only the persons who know the key can read the information. Saved and also transmitted data can be encrypted.

Data integrity represents protection against changes by unauthorized persons. Data has value only if it is safe. Data which has been manipulated does not have any value and may cause financial waste. For example data manipulation in which information about financial accounts is stored. Similarly, cryptography plays an important role in ensuring data integrity. Frequently used methods of data integrity protection contain information about data changes and hash checksums by which data integrity is verified.

Availability concept is to make sure that the services of an organization are available for all legitimate users. This is to ensure that the system is free from DoS or DDoS attacks (denial of service or distributed denial of services). The motive behind DoS attacks is to bring down the respective services and therefore to defeat availability. Though, availability can also be defeated through some other natural causes such as disasters (flood, fire, or earthquake etc). Generally, the aim of availability is to develop systems which are fault tolerant.

B. DATA ENCRYPTION

One of the keys to data protection in the Fog computingis accounting for the possible states in which your data may occur, and what controls are available for that state. For the purpose of Microsoft Azure [16], data security and encryption best practices the recommendations will be around the following data's states:

- At-rest: This includes all information storage objects, containers, and types that exist statically on physical media, be it magnetic or optical disk.

- In-Transit: When data is being transferred between components, locations or programs, such as over the network, across a service bus (from on-premises to cloudand vice-versa, including hybrid connections such as ExpressRoute), or during an input/output process, it is thought of as being inmotion.

Some of the recommended best practices for data security and encryption are as follows:

- Enforce multi-factor authentication
- Use role- based access control (RBAC)
- Encrypt virtual machines
- Use hardware security models
- Manage with Secure Workstations
- Enable SQL data encryption
- Protect data in transit
- Enforce file level data encryption

C. Encryption methods used in securing data

Depending on the application domain, existing encryption algorithms can be used in securing data in fog computing. Advanced Encryption Standard (AES) is asymmetrical block encryption. It is using 128- bit blocks and key of 128, 192 and 256 bits. The algorithm AES consists of bit substitution, permutation, arithmetic operation through a finite field and operation XOR with a key. Another example is the Triple Data Encryption Standard (TDES or 3DES) which is asymmetrical block encryption which is based on encryption algorithm Data Encryption Standard (DES). It is applied three times in order to increase resistance against breaking. TDES is slower than AES and that is why it is replaced by AES nowadays. Data is divided into 64- bit blocks (56 randomly generated bits for encryption algorithm and 8 bits for detection).

RSA is an alternative for asymmetric encryption. It is suitable for encryption and signing. In the case of a key long enough, it is considered to be safe. RSA is built on an assumption that spreading a large number on the product of prime numbers (factorization) is a computationally demanding task. From numbers $n=p*q$ it is computationally impossible to find out p and qin real time. Nowadays, there is no known algorithm of factorization which would work in polynomial time against the size of binary numbers n. Multiplication of two big numbers is computationally not a demanding task.

D. Data integrity check

Data integrity check in fog computing was necessary to realize by the use ofa data audit model by an audit page. The audit page is a confidential entity in which the data owner and the service provider have confidence. It can be directly the owner of data or data integrity check provider.

One approach to ensure data integrity is Provable Data Possession (PDP) as described in Ateniese et al., (2007). This method enables a client to verify that the server possesses his data, without having to retrieve the entire data. The model implements a challenge/response protocol between client and server, in which the server has to deliver a so-called Proof of Possession. Instead of a deterministic proof, this proof is probabilistic, as in this protocol, it is not possible to verify every bit of data. This would require having to download the entire data set, which is undesirable for large amounts of data. Instead, random samples of blocks are verified. The goal of PDP is therefore, to detect misbehavior of a server when it does not possess the complete file anymore. In contrast to some other methods, PDP requires the client to store a small and constant amount of metadata locally, to minimize network communication. This enables verifying large data sets, distributed over many storage devices. The authors propose two schemes that are supposedly more secure than previous solutions, have a constant server overhead, independent of the data size, and have a performance bounded by disk I/O and not by cryptographic computation. The size of the challenge and response are both 1 Kilobit. The method works with homomorphically verifiable tags, that enable possession verification without having to possess the files locally. For each file block, such a tag is computed. The homomorphic property enables multiple tags to be combined into one value. The file blocks, together with their tags, are stored.

Whenever the user wants to check whether the server is performing the right behavior, he can decide to send a challenge to the server. The challenge consists of the references to a random subset of file blocks. The server has to construct a proof of possession out of the blocks that are queried, and their tags, to convince the client that the server possesses the blocks, without having to send them. Note here that the server should not be able to construct a proof without possessing the queried blocks. Figure 2 clearly shows the steps.

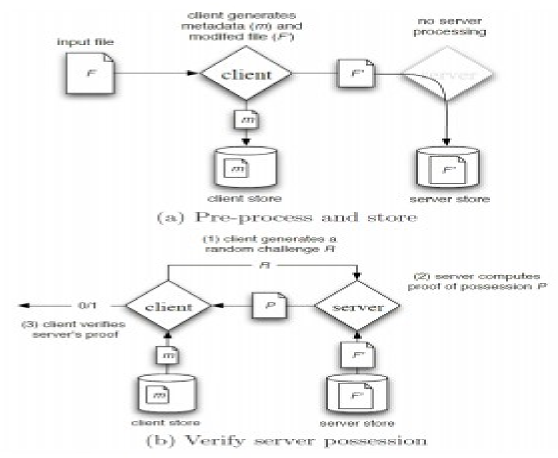
client generates metadata (m) and modified file (F')
input file
F
client
F'
no server processing
server
m
client store
server store
(a) Pre-process and store
(1) client generates a random challenge R
R
(2) server computes proof of possession P
0/1
client
P
server
(3) client verifies server's proof
m
F'
client store
server store
(b) Verify server possession

Figure 2: PDP Protocol (Ateniese et al., (2007))

Step (a) shows the pre-processing phase, in which the client uses the file to create the metadata, and stores this metadata locally. The client may modify the file, by appending other metadataor encrypting the file. Next, the modified file is sent to the server, which stores it without further processing. In (b), the verification step is shown. The client sends a challenge to the server. The server uses the stored file to create a proof and sends it back to the client. The client uses the stored metadata to verify this proof.

E. DATA INTEGRITY CHECK MODEL BY THIRD- PARTY AUDITOR

A mathematical model by [19] has been chosen as a mathematical model for data audit stored onthe cloud which is in fact, an edited model of [28]. The same model can be applied in fog computing as well. Operation descriptions of mathematical model and formulas have been adopted from the source [20]. The model of public audit used four algorithms: *KeyGen*– algorithm to generate the key. *SiSSen*– algorithm used by the user to generate verification metadata. Metadata contained entries needed to verify the integrity of stored data. *GenProof*– algorithm launched by cloud application server to generate metadata needed to verify the integrity of stored data in the cloud. *VerifyProof* – algorithm launched byan audit server to check metadata.

4. Cryptographic Parameters of Mathematical Model

$F = (m_1, \ldots, m_n) \in \mathbb{Z}p$ user´s data which should be stored in the server storage where n is the number of blocks and p is a high prime number.

$fkey(\cdot)$ – pseudorandom function (PRF), defined as:

$$\{0, 1\}^{*} \times key \rightarrow \mathbb{Z}p.$$

$\pi key(\cdot)$ – pseudorandom permutation (PRP), defined as:

$$\{0, 1\}^{\log2(n)} \times key \rightarrow \{0, 1\}^{\log2(n)}$$

$MACkey(\cdot)$ – message authentication code (MAC) function, defined as:

$$\{0, 1\}^{*} \times key \rightarrow \{0, 1\}^{1}.$$

$H(\cdot)$, $h(\cdot)$ – map-to-point hash functions, defined as: $\{0, 1\}^{*} \rightarrow G$, where G is some group.

Let G_1, G_2 and G_T to be the multiplicative cyclic group of line p and e: $G_1 \times G_2 \rightarrow G_T$ was a bilinear map. Let g be the generator G_2.$H(\cdot)$ which was a secure map-to-point hash function: $\{0,1\}^{*} \rightarrow G_1$ which mapped the integer (chains) uniformly into G_1. Hash function $h(\cdot)$:$G_1 \rightarrow \mathbb{Z}p$ mapped the element groups G_1 into $\mathbb{Z}p$ [14].

4.1 Mathematical description of TPA model operation

In the process of settings, the user launched the algorithm *KeyGen* and generated the public and private parameters. Randomly selected $x \leftarrow \mathbb{Z}p$, random element $u \leftarrow G_1$ and calculated $u \leftarrow g^x$and $w \leftarrow u^x$. The private parameter was $sk=(x)$ and the public were $vk = (v,q,g,u)$. On data $F = (m_1, \ldots, m_n)$the user launched the algorithm *SigGen* by which he calculated the sign σ_i for each data block mi: $\sigma_i \leftarrow (H(i) \cdot u^{mi})^x \in G_1$, for $i=(1, \ldots, n)$. Then the summary of all signs was $\Phi = \{\sigma_i\}_{1 \leq I \leq n}fori=(1, \ldots, n)$. Consequently, the user sent $\{F, \Phi\}$ on a server and deleted the local copy.

During the audit, the message "*chal*" has been created. It specified the blocks′ position which wasneeded to be checked during the audit phase. The third audit side randomly selected a subset s c elemental $I=\{S_1, \ldots, S_c\}$ sets $[1, n]where\ S_q=\pi k_{prp}(q)$ where q=(1, … ,c) and k_{prp} was the third permutation key for each audit chosen by audit side. For each element $i \in I$, TAP has chosen randomly the value υ_i. Then it has sent the message $chal=\{(i, \upsilon_i)\} i \in I$ to the server.

After receiving the the message "*chal*", the server launched the algorithm *GenProff* which generated an answer as a proof of correctness of data storage. The Server selected a random element $r \leftarrow \mathbb{Z}p$ through $r = fk_{prf}(chal)$ where k_{prf} was a randomly chosen key by the server by the use of the pseudo-random function for every unit. Then it calculated $R= (w)^r = (u^x)^r \in G_1$. Linear combinations of chosen blocks specified in the message "*chal*" was $\mu' = \Sigma v_i m_i$, for i∈I. For μ' and

r server calculated $\mu = \mu' + rh(R) \in \mathbb{Z}p$. Then server calculated aggregated signature $\sigma = \Pi_{i\in I}\sigma_i^{vi} \in G_1$, for $i \in$ I. Finally, the server has sent the answer { σ, μ, R} to third audit side.

After receiving the answer from the server, TAS launched the algorithm *VerifyProof* by which it verified the accuracy of the answer by calculating in the verifying equation:

$$e(\sigma(R^{h(R)}),g) \overset{?}{=} e\ \Pi_{i=si}\ H(i)^{vi}\ .\ u^{\mu},\ v)$$

Random mask *R* did not have any effect on accuracy verification. Simultaneously, the third side did not have any access to data of customer *F,* privacy has been maintained, Cong et al., (2010).

4.2 Propose Protocol for TPA Model

The proposed protocol is similar to some previously proposed schemes in the context of remote data integrity checking such as [19]. The protocol consists of four algorithms; *KeyGen* is run by the client and sets up the scheme, *SigGen* is run by the client and generates verification metadata used for auditing, *GenProof* is run by the server and generates the proof of correctness of data, and finally *VerifyProof* which is run by the TPA and verifies the server's proof. The client first runs *KeyGen* and *SigGen*. He sends the file to the server and the metadata to the TPA. Then he deletes his local data. In the auditing phase, the TPA creates a challenge and sends it to the server. The server runs *GenProof*, sends the proof to the TPA, and the TPA runs *VerifyProof* to verify

The protocol consists of the following algorithms:

- *KeyGen*(1^k) → *(pk, sk)*. Run by the client. Creates a public and private key pair.
- *SigGen(sk, F)* → *(Φ, sig*$_{sk}$*(H(R)))*. Run by client. Creates a set of signatures $\Phi = \{\sigma i\}$ for each block mi in file *F*, and a signature of the root *R* (signed with the private key) of the Merkle hash tree.
- *GenProof(F, Φ, chal)* → *(P)*. Run by the server. Constructs a proof out of file *F*, the signature set, and the challenge *chal* (created by the TPA).
- *VerifyProof(pk, chal, P)* → *{T RUE, F ALSE}*. Run by TPA. Verifies the proof, using the public key and the corresponding challenge. Returns true or false.
- *ExecUpdate(F, Φ, update)* → *(F' , Φ' ,* P_{update}*)*. Run by the server. Performs the update on the file, and returns the new file and signatures set, as well as a proof for the update.
- *VerifyUpdate(pk, update,* P_{update}*)* → *{(T RUE, sig*$_{sk}$*(H(R'))), F ALSE}*. Run by the client. Verifies whether the updated proof is correct, meaning that the update has been performed correctly. If correct, returns a new root signature

5. Securing TCP/IP Communications

There are two main subgroups of cipher suites recommended for TCP/IP transport layer security communication. RSA and ECDHE. RSA-based cipher suites use RSA as the key-exchange algorithm, while the ECDHE-based ones use an algorithm that makes use of Ephemeral Diffie–Hellman based on Elliptic Curves. The Ephemeral part means that the key is regenerated after each session, providing Perfect Forward Secrecy (PFS) in contrast to the variants based on RSA.

Figure 3 below illustrates the messages exchanged during the handshake when using the cipher suite with an ECDHE key-exchange algorithm. Refers to private keys, while public keys are the ones defined as the procedure takes place as follows:

- The client sends a ClientHello specifying its supported cipher suites.
- The server responds with a ServerHello with the cipher suite selected. This is the cipher suite that is going to be used during the whole TLS session.
- The server sends its certificate in a Certificate message. Along with it, the public key (Qcert) of the aforementioned certificate is sent.
- The server generates a key pair (*ds* ,*Qs*) needed for the ECDHE algorithm and sends the public key to the client, encrypted with the private key of the certificate (*dcert* (*Qs*). This corresponds to the ServerKeyExchange message.
- Once the client receives the ServerKeyExchange, it uses the certificate's public key received in the Certificate message to check the authenticity of the ECDHE public key by verifying the RSA signature (*Qcert* (*dcert* (*Qs*))), thus obtaining the ECDHE public key (*Qs*) of the server.
- Finally, the client generates its own ECDHE key pair (*dc* ,*Qc*) and sends the public key to the server.
- At this point, both theserver and client can obtain the Session Secret by performing an operation (ECC dot product) with one's own private key and the other party's publickey.

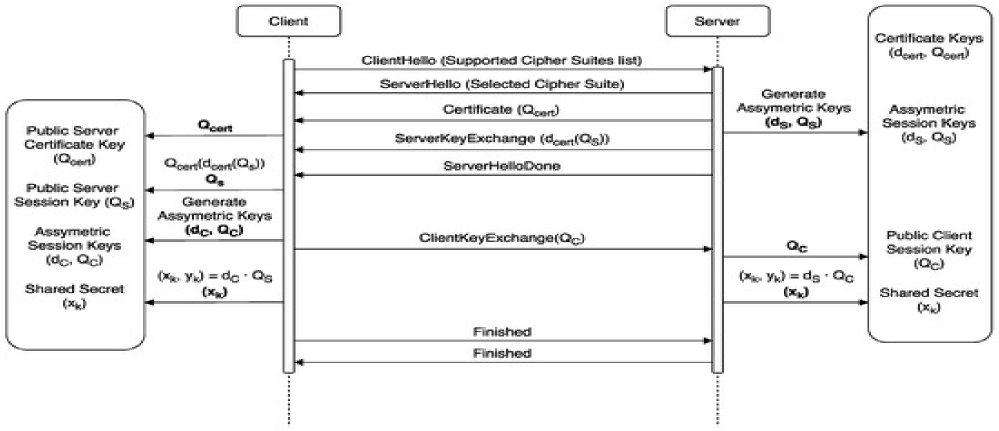
Client
Server
Certificate Keys (d_{cert}, Q_{cert})
ClientHello (Supported Cipher Suites list)
ServerHello (Selected Cipher Suite)
Certificate (Q_{cert})
ServerKeyExchange $(d_{cert}(Q_S))$
ServerHelloDone
Generate Assymetric Keys (d_S, Q_S)
Assymetric Session Keys (d_S, Q_S)
Public Server Certificate Key (Q_{cert})
Q_{cert}
$Q_{cert}(d_{cert}(Q_S))$ Q_S
Public Server Session Key (Q_S)
Generate Assymetric Keys (d_C, Q_C)
Assymetric Session Keys (d_C, Q_C)
ClientKeyExchange(Q_C)
Q_C
Public Client Session Key (Q_C)
$(x_k, y_k) = d_C \cdot Q_S$ (x_k)
$(x_k, y_k) = d_S \cdot Q_C$ (x_k)
Shared Secret (x_k)
Shared Secret (x_k)
Finished
Finished

Figure 3. Transport Layer Security (TLS) Handshake procedure [1]

6. CONCLUSION

Proposed data integrity model gives a proof of the integrity of the data which the user wishes to store in the storage servers with bare minimum costs and efforts. This scheme is developed to reduce the computational and storage overhead of the client. This approach gives better accuracy as compared to sentinel-based approach. The accuracy totally depends on the client.

In the context of computation integrity, most algorithms still require computationally expensive pre-computation steps, requiring one function to be executed many times before it becomes effective. This is not always a realistic scenario. Often, a client may only want to execute a function once by the server. Future work should make the verifiable computation for these types of functions more efficient.

In future work, there will be a growing body of work dealing with fog computing security issues. It would be desirable to propose new mechanisms and cryptography algorithms which is light, fast and strong enough to secure fog computing data. Also, integrating one or more good techniques together such as multilevel algorithm on the enhanced user behavior and decoy technology may result in a secure and effective mechanism. Combining one or more techniques need more detailed experiments which focus on the algorithms performance such as robust and secure.

REFERENCES

[1] K. Munir, Lawan A. Mohammad, (2018). Secure Data Integrity Protocol for Fog Computing Environment, Advancing Consumer-Centric Fog Computing Architectures, Ed.KashifMunir, IGI Global Publishing. In Press.

[2] Data Center Companies (2016). Retrieved November, 9, 2017, fromhttps://www.datacenters.com/directory/companies.

[3] Bonomi, F., Milito, R., Zhu, J., &Addepalli, S.(2012). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing. pp 13-16

[4] Sravan Kumar R, AshutoshSaxena,."Data Integrity Proofs in Cloud Storage" 978-1-4244-8953-4/11/$26.00 c 2011 IEEE.

[5] Cisco (2015). Cisco delivers vision of fog computing to accelerate value from billions of connected devices. Retrieved December 02, 2017, from https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1334100

[6] OpenFog Consortium (2015).Retrieved December 02, 2017, fromhttps://www.openfogconsortium.org

[7] Ateniese G., et al.(2007), "Provable data possession at untrusted stores," in Proceedings of CCS'07, New York, USA, pp. 598- 603.

[8] WenjunLuo; GuojingBai "Ensuring the data integrity in cloud data storage" CloudComputing and Intelligence Systems (CCIS), 2011 IEEE International Conference

[9] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," Proc. 2015 Work. Mob. Big Data - Mobidata '15, pp. 37–42, 2015.

[10] Chiang, M., Zhang, T.(2016). Fog and IoT: An overview of research opportunities, IEEE Internet of Things Journal, pp. 1–11.

[11] Bader, A., Ghazzai, H., Kadri, A., &Alouini, M.-S. (2016). Front-end intelligence for large-scale application-oriented Internet-of-things. IEEE Access, vol. 4, pp. 3257–3272, June 2016.

[12] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices In: STOC, vol. 9, 169–178.. ACM.

[13] Bos, JW.,Castryck, W., Iliashenko, I., &Vercauteren, F. (2017). Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. International Conference on Cryptology in Africa, 184–201.. Springer.

[14] Lu, R., Liang, X., Li, X., Lin, X., &Shen, X. (2012).Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Trans Parallel Distributed Syst 23(9): 1621–1631

[15] TechTarget (2015), Confidentiality, integrity, and availability (CIA triad), Retrieved April 02, 2018, from:https://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA

[16] Azure (2018), Data Security and Encryption Best Practices, Retrieved April 17, 2018, from https://docs.microsoft.com/en-us/azure/security/azure-security-data-encryption-best-practices

[17] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, (2007), "Provable data possession at untrusted stores," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 598–609.

A.Juels and J. Burton S. Kaliski, (2007), "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

[18] Polk, T.; McKay, K.; Chokhani, S. (2005).,Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations; NIST Special Publication 800-52 Revision 1; NIST: Gaithersburg, MD, USA, 2005.
[19] H. Shacham and B. Waters, (2013), "Compact proofs of retrievability," Journal ofCryptology, vol. 26, pp. 442-483, 2013.
[20] W. Cong, W. Qian, R. Kui, and L. Wenjing, (2010)., "Privacy-Preserving PublicAuditing for Data Storage Security in Cloud Computing," in INFOCOM,2010 Proceedings IEEE, pp. 1-9.
[21] M. T. Dong and X. Zhou, “Fog Computing: Comprehensive Approach for Security Data Theft Attack Using Elliptic Curve Cryptography and Decoy Technology,” OALib, vol. 3, pp. 1–14, 2016
[22] R. Sinha, H. K. Srivastava, and S. Gupta, “Performance Based Comparison Study of RSA,” International Journal of Scientific & Engineering Research, vol. 4, no. 4, May 2013.
[23] Lee , K., Kim, D., Ha, D., Rajput, U., & Oh, H. (2015). On security and privacy issues of fog computing supported Internet of Things environment. doi:10.1109/NOF.2015.7333287
[24] N. Mishra, S. Siddiqui, and J. P. Tripathi, “A Compendium Over Cloud Computing Cryptographic Algorithms and Security Issues,” Int. J. Inf. Technol. BharatiVidyapeeth’s Inst. Comput. Appl. Manag., vol. 7, no. 1, pp. 973–5658, 2015.
[25] L. M. Vaquero, L. R. Merino, (2014), “Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing”, ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, pp. 27-32, 2014.
[26] T. K. Goyal, V.Sahula. (2016). Lightweight security algorithm for low power IoT devices. International Conference on Advances in Computing, Communications and Informatics (ICACCI).
[27] D. H. Gawali, V. M. Wadhai (2012)., “RC5 Algorithm: potential cipher solution for security in WBSN” International Journal of Advanced Smart Sensor Network System s (IJASSN), Volume 2, No.3, July 2012.
[28] A.Juels and J. Burton S. Kaliski, (2007), “Pors: Proofs of retrievability for large files,” in Proc. of CCS’07, Alexandria, VA, October 2007, pp. 584–597.

AUTHORS

KashifMunir received his BSc degree in Mathematics and Physics from Islamia University Bahawalpur, Pakistan in 1999. He received his MSc degree in Information Technology from University Sains Malaysia in 2001. He also obtained another MS degree in Software Engineering from University of Malaya, Malaysia in 2005. He completed his PhD in Informatics from Malaysia University of Science and Technology, Malaysia. His research interests are in the areas of Cloud Computing Security, Software Engineering, and Project Management. He has published journal, conference papers and book chapters.

KashifMunir has been in the field of higher education since 2002. After an initial teaching experience with courses in Stamford College, Malaysia for around four years, he later relocated to Saudi Arabia. He worked with King Fahd University of Petroleum and Minerals, KSA from September 2006 till December 2014. He moved into University of Hafr Al-Batin, KSA in January 2015.

KashifMunir is a researcher and published author/editor of 4 books on cloud computing including subjects such as Security in Cloud Computing, Mobile Cloud and Green Enterprises, (https://www.amazon.com/Kashif-Munir/e/B079KP1LFJ).

Lawan A. Mohammad, Holds a PhD degree in computer and communication systems engineering from University Putra Malaysia. Research interest include smartcard security, authentication protocols, wireless and mobile security, biometrics, mathematical programming and e-learning.