LEASE: Lattice and ECC-based Authentication and Integrity Verification Scheme in E-Healthcare

Amit Dua*, Rajat Chaudhary†, Gagangeet Singh Aujla‡, Anish Jindal§, Neeraj Kumar¶, and Joel J.P.C. Rodrigues‖
* Department of Computer Science and Information Systems, BITS Pilani, Pilani (Rajasthan), India
(e-mail: mail.amitdua@gmail.com).
†§¶ Computer Science & Engineering Department, Thapar Institute of Engineering & Technology, Patiala (Punjab), India
(e-mail: rajatlibran@gmail.com, anishjindal90@gmail.com and neeraj.kumar@thapar.edu).
‡ Computer Science & Engineering Department, Chandigarh University, Gharuan (Punjab), India
(e-mail: gagi_aujla82@yahoo.com).
‖ National Institute of Telecommunications (Inatel), Brazil; Instituto de Telecomunicações, Portugal; ITMO University,
Russia; University of Fortaleza (UNIFOR), Ceará, Brazil (email: joeljr@ieee.org).

***Abstract*—Security has become one of major concern especially in critical applications like e-healthcare. To cater to the security needs in e-healthcare, this paper proposes a novel scheme which prevents data from unauthorized fabrication and preserves the integrity of data. The proposed scheme also removes overhead of integrity validation from user's end as this work is assigned to a trusted third party, i.e., a proxy server. For this purpose, the patient's data given by user is sent to proxy server along with user's signature where it is broken down in the form of blocks. A 'tag' is then generated for each block using lightweight elliptic curve cryptography (ECC). This block-tag pair is then uploaded on the data server which is used for integrity checking. Whenever a patient's data access request is raised, the block of data is retrieved using tag value and integrity is then verified. In addition to it, a lightweight lattice-based authentication scheme is proposed in the paper to authenticate the users. The request is served only when the user is deemed authentic and there is no modification in the original data sent by the user. The effectiveness of the proposed authentication scheme has been proven by performing its analysis in terms of computation time and communication cost. Moreover, the superiority of the proposed data integrity scheme has been validated by comparing it with the traditional discrete logarithmic scheme.**

Keywords*—Elliptic curve cryptography (ECC), lattice-based authentication, secure e-healthcare.

I. Introduction

E-healthcare is one of the latest areas which attracts a lot of attention from the research community across the globe. It uses the concept of telemedicine where a patient may be provided with a diagnosis remotely from the doctor by using information and communication technologies (ICT). Using e-healthare services, patients can be provided with tele-consultation, tele-expertise, tele-care, and tele-monitoring on-the-fly [1]. Many authors have proposed different solutions which are being used in wide variety of e-healthcare services. For instance, Cheng *et al.* [2] proposed a clinical decision support system to help doctors during critical situations in real-time. In other studies, the authors have used the communication capabilities in vehicles to provide the patients' information to the cloud [3], [4]. However, none of these studies have incorporated data security in their techniques which can be exploited by malicious entity to misuse the network resources.

The main issue that is associated in e-healthcare domain is the security and privacy of patients' data. There are many types of attacks like server availability (denial of service attack), identity protection and access control, service provider impersonation attack, no provision for revocation, etc., which pose a bigger threat to the integrity of data in such applications [5], [6] To cater to one or more of these attacks, many researchers have proposed different schemes where data security is managed in e-healthcare scenario. For example, Khan *et al.* [7] used the ECG and EEG data of the patients to generate a 'key' which was then used for securely communicating with the data service provider. However, this technique is difficult to implement in real-time because such data is not readily available for every patient. In another study, Tong *et al.* [8] used private clouds to ensure privacy in mobile healthcare applications. Authors integrated pseudorandom numbers with key management to ensure unlinkability, attribute based encryption for access control, and secure indexing for privacy preservation. Although, the proposed scheme provided reasonable security, the authors used pseudorandom numbers and computations which are computationally more complex and may put additional burden on the system where resources are less. Hence, a lightweight security mechanisms are required to manage data security goals for e-healthcare without incurring additional cost to the service providers. Many authors have presented an idea that a proxy can be used on behalf of the client system to communicate the data so as to remove the unwanted burden on the network service provider [9], [10].

However, security of the patients' data and maintaining the proper data access mechanism are still the main concerns for any healthcare service provider. Moreover, the security measures should not put heavy burden on the network resources with respect to bandwidth consumption and processing time. Thus, there is a requirement of a lightweight security solution which can easily manage the security goals like data integrity verification and authentication of data without putting additional burden on network resources. So, keeping this in mind, a scheme is proposed in this paper which deals with the issues of authentication and integrity verification to enhance the overall data security for providing e-healthcare services. The proposed scheme accomplishes these security goals in an efficient way in terms of the computation power and communication time. The system also deals with forming a lightweight security system which is suitable for mobile devices and at the same time

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

provides the level of security which is comparable to more computationally expensive systems.

A. *Motivation*

Maintaining security of data in e-healthcare is of prime importance as any modification made in patient's sensitive data may lead to wrong medical treatments or in the worst case, loss of patient's life. Thus, as far as the security aspects of e-healthcare data are concerned, a system is needed that can protect itself from such breaches and is able to check whether a breach has occurred or not. The system should provide lightweight solution which ensures data confidentiality, integrity, and authentication; and at the same time, the time complexity of such solution should be low, so that it is practically feasible.

B. *Contribution*

The major contributions made by the paper related to the security of data for providing e-healthcare services are as follows.

1) A ECC-based digital signature generation scheme for data integrity verification of the patients' data while providing the e-healthcare services.
2) A lattice-based post quantum cryptography using ring learning with error to authenticate the legitimate user entities whenever a new request is raised.
3) The efficacy of the proposed scheme has been validated with respect to the variation in latency by comparing with existing public key cryptographic schemes.

C. *Organization*

The rest of the paper is organized as follows. The description of the system model is presented in Section II. The details of entity authentication and data integrity scheme along with analysis and security proof is presented in Section III. The simulation results and comparison with traditional schemes are discussed in Section IV. Finally, the paper is concluded in Section V.

II. SYSTEM MODEL

The proposed scheme maintains the security goals of data integrity and authentication in e-healthcare services. The system model for the same has been presented in Fig. 1. This model consists of the following entities:

1. **User:** Any patient or entity seeking e-healthcare service(s) is referred to as an user. The user needs to have a communication medium to send and receive the data from the service provider.
2. **Service provider:** Whenever a new request is raised for data accessing or storing, the service provider authenticates the user entity which raises this request.
3. **Proxy server:** It is a third party entity which is responsible for generating the tags and checking the integrity of user's data. From the user's side, it generates the tags which are used by the user to communicate their data to the service provider. From the service provider's viewpoint, it is used to check the integrity of data received by the service provider.
4. **Data server:** This entity is responsible for keeping the log of data along with the tags that helps in verifying the integrity of data.

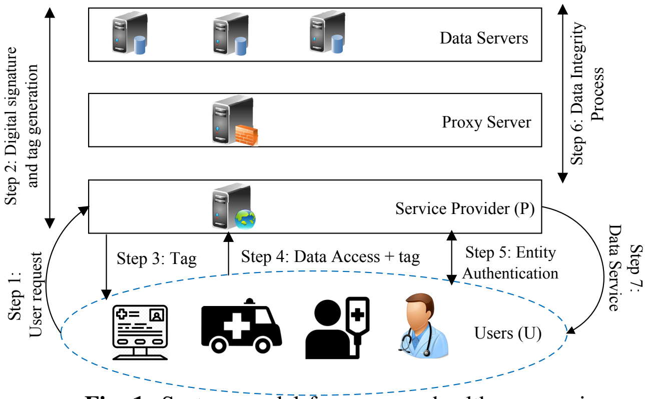
Data Servers
Proxy Server
Service Provider (P)
Users (U)
Step 2: Digital signature and tag generation
Step 6: Data Integrity Process
Step 1: User request
Step 3: Tag
Step 4: Data Access + tag
Step 5: Entity Authentication
Step 7: Data Service

Fig. 1: System model for secure e-healthcare services.

The steps involved in the proposed system model for verifying the integrity of user's data and authentication of user are shown in Fig. 1 and described as follows.
Step 1: Initially, the user which requires the e-healthcare service sends the request to the service provider.
Step 2: The service provider initiates the data interaction with the proxy server and the data server. The proxy server computes a tag value and generates a digital signature which is stored in the form of blocks along with their tag values at the data server.
Step 3: The service provider returns this tag value to the user which is used to access or store the data.
Step 4: Whenever the user has to access the data, the user sends the data request to the service provider along with the tag value.
Step 5: The service provider authenticates the user entity using lattice cryptography and forwards its request to the proxy server, if the entity is authenticated.
Step 6: The proxy server stores the tag and sends the request to the data server for accessing the block associated with the received tag. The proxy server retrieves the tag value from the block-tag pair stored at the data server and compares it with the user tag value. If both these values are matched, then the data integrity is validated.
Step 7: Finally, the data service is provided to the user by the service provider. The detailed procedures for data integrity checking and entity authentication are discussed in the following section.

III. PROPOSED SCHEME

The working of the proposed scheme is divided into two phases. First phase is the secure entity authentication using lattice cryptography; the next phase is to check for the data integrity.

A. *Secure Authentication Key Exchange (SAKE)*

A lightweight secure authentication key exchange (SAKE) mechanism is shown in Fig. 2 to exchange keys between the user (U) and the service provider (P) in non-trusted channels. The underlying principle behind the SAKE is for entity authentication is the use of post quantum cryptography scheme known as lattice-based ring-learning with errors. The term *lattice* is defined as a set of vectors in a two-dimensional plane. The basis of a lattice is generated by the combination of set of vectors and their coefficients to form a polynomial equation. As this vector multiplication creates a polynomial of higher degree, so a cyclotomic ring $R_n = Z_n[x]/(x^p + 1)$ using modulo $(x^p + 1)$ is used to reduce the polynomial degree to the at-most degree of (p) [11]. The reason to apply this cryptography mechanism in the proposed scheme is that the lattice cryptography is resilient against quantum attacks. Moreover, it uses the operations of modern algebra such as group, ring, and field which are difficult to break [12]. Using the modern algebraic properties and Gaussian sampling techniques, the lattice cryptography is successful in generating complex public keys. After computing the public keys by the participating entities, a common shared session key is computed which is verified by the participating entities. If the session key used for communicating the request is same, then the entity is deemed authentic and service provider serves the request of the user; otherwise, the service provider denies to provide the data services.

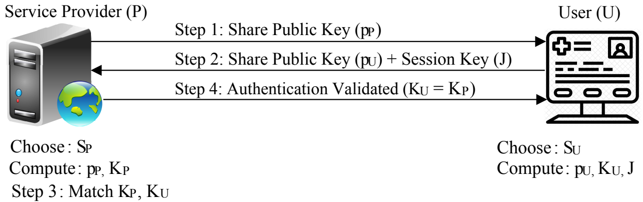
Service Provider (P)
User (U)
Step 1: Share Public Key (pP)
Step 2: Share Public Key (pU) + Session Key (J)
Step 4: Authentication Validated (KU = KP)
Choose : SP
Compute: pP, KP
Step 3 : Match KP, KU
Choose : SU
Compute: pU, KU, J

Fig. 2: Entity authentication process.

The steps followed in the SAKE protocol between (U) and (P) using lattice based on ring learning with errors are shown in Fig. 3 and are explained as below:

Step 1: Initially, the entity P assumes the list of input parameters as: (m, n, p, e_r, χ). These parameters are described as follows. m is the message size in bits, n is the prime number used for modulo, p is the degree of the polynomial such that $(p = m/2)$, e_r is the error rate such that $e_r < 1$, and χ is the discrete Gaussian function. The $\chi \xleftarrow{\$} D_{Z^p, e^r n}$ is used for sampling using arithmetic operations rather than exponential or logarithm functions. This creates a complex secret key vector which is difficult to break for an attacker [13]. Using the sampled secret key vector $s_P \xleftarrow{\$} \chi$, error vector $e_P \xleftarrow{\$} \chi$, and a uniform random symmetric matrix $r \leftarrow Z_n^{p \times p}$, entity P computes it's public key vector p_P and sends it to the entity U.

Step 2: The entity U performs the similar task by choosing a secret vector s_U, error vector e_U, and random matrix r to compute it's public key p_U. Now, a new error vector $e_U^{'} \xleftarrow{\$} \chi$ is sampled again to compute the session key K_U. The K_U is calculated on the basis of $(p_P, s_U, e_U^{'})$ and a randomized function f_{rand} which is used to avoid larger interval states. The functions modular rounding $\lfloor I \rceil_2$ and cross rounding $< I >_2$ are applied to the result using mod 2 in order to drop the less significant bits [11]. The result of cross rounding J is derived as a shared session key which is sent to the entity P.

Step 3: The entity P uses the reconciliation function rec [11], [14] to compute the similar value. It then matches that whether $K_U = K_P$. If these keys are matched successfully, then the requesting entity U is considered as authentic, otherwise, the request is invalidated. Finally, an acknowledgment of the success or failure is sent to the entity U.

Step 4: Once the authenticity of U is proved, the service provider serves its request and provides the requested data service.

B. Data Integrity Scheme

Once the entity is authenticated, the proposed scheme verifies the integrity of the data before providing data to the user. For this purpose, the digital signatures are computed for the users; the steps for which are shown in Fig. 4. These steps are carried out at different entities which are summarized as follows.

i. **User:** In the first step, the user (U) (either patient or doctor) login to his/her account and initiates the request to the service provider (P) for the secure data storage.

ii. **Service Provider:** The service provider forwards the user request to the proxy server which generates its ditigal signature. This signature consists of a fixed ‘id’ and a master public key information is sent to the proxy server.

ii. **Proxy Server:** The proxy server computes the block-tag pair as shown in algorithm 1 using ECC. The list of symbols used in algorithm 1 is explained in Table I.

iii. **Data server:** The data server stores each encrypted block-tag pair or the signature of the message digest. The block-tag pair is stored in the form of $\{(f_1, t_1), ..., (f_n, t_n)\}$ on the data server till the data access request is processed by the user.

TABLE I: Symbols used in the algorithms.

Symbol	**Meaning**
Curve	The representation of Elliptic curve field and equation
G	It generates Elliptic curve with a huge prime order ‘n’
n	Order of G
m	Message to be encrypted
e	Digest of the message
Ln	Leftmost bits of e, where Ln is the bit length of the group order ‘n’.
K	A random integer in the range [1, ..., n-1].
Q	Public-key of the curve point Q.
z	leftmost n bits of e
d	Private key of the user.

In algorithm 1, the proxy server first breaks the data into a fixed size chunks to create the blocks. In the case where the file size is smaller than the fixed block size, then extra bits (known as padding bits) are appended to the data bits. Now, a message digest of 256 bits is computed for every block using one-way hash function, i.e., SHA-2. The next step is to compute the curve point and based on these points, the block-tag pair or the digital signature for the message digest is calculated as (f, t). Finally, (f, t) is sent to the data server.

Theorem 1: In algorithm 1, the parameters will always produce the same result irrespective of choosing the value of

$SERVICE\,PROVIDER\,(P)$	$USERS\,(U)$
Input $\{m, n, p, e_r, \chi\}$	
$\chi \overset{\$}{\leftarrow} D_{Z^p, e^r n}$	
$(e_r < 1)$	
$R_n \leftarrow Z_n[x]/(x^p + 1)$	
Choose Secret Vector	
$s_P \overset{\$}{\leftarrow} \chi,\ e_P \overset{\$}{\leftarrow} \chi$	**Input** $\{n, p, e_r, \chi\}$
$r \leftarrow Z_n^{p \times p}$	**Choose Secret Vector**
	$s_U \overset{\$}{\leftarrow} \chi,\ e_U \overset{\$}{\leftarrow} \chi, r \leftarrow Z_n^{p \times p}$
Choose Public Vector	**Choose Public Vector**
$p_P = (r \times s_P + e_P)(mod\ n)$	
$\xrightarrow[\textbf{(public key)}]{\langle p_P \rangle}$	$p_U = (r \times s_U + e_U)(mod\ n)$
	Compute Session Key
	$I = f_{rand}(p_P \times s_U + e_U^{'})(mod\ n)$
Compute Session Key	$k_U = \lfloor I \rceil_2, J =< I >_2$
	$\xleftarrow[\textbf{(public, session key)}]{\langle p_U, J \rangle}$
$K_P = f_{rec}(2 \times p_U \times s_P + J)(mod\ n)$	
$K_P = K_U$ (key matched)	
$\xrightarrow[\textbf{(authenticated validated)}]{\langle ACK \rangle}$	$\xleftarrow{\langle request\ for\ data\ service \rangle}$

Fig. 3: Secure authentication key exchange (SAKE) mechanism.

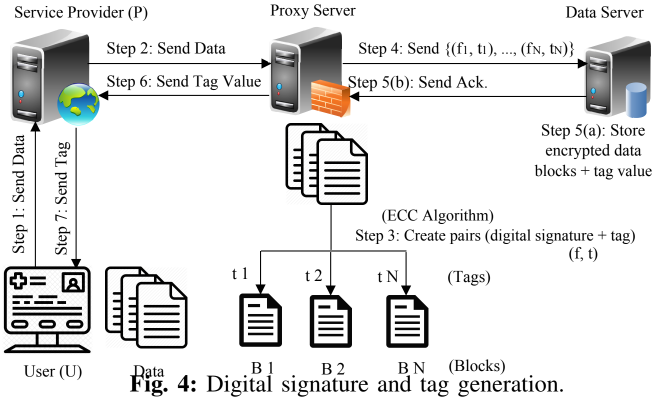

Fig. 4: Digital signature and tag generation.

Algorithm 1 Digital signature computation at user side.

Input: File to be uploaded
Output: Block-Tag pair

```
First, the proxy server divides the file into fixed size chunks (40, 80, 120,...) according to the parameter set by the server. If the file is smaller than the block size than a uniform padding is used.
For every block $B$, a digest $'e'$ is calculated through hashing. $e = Hash(B)$ using SHA-2.
The curve point $(x1, y1) = K \times G$ is computed.
$f = x1(mod\ n)$ is computed.
$t = k1(z + f^{*}d)(mod\ n)$ is computed.
$(f, t)$ is the computed digital signature for the digest.
```

n or k. An assumption has been made that 'K' is not static but dynamic as there have been successful attempts in breaking the security when 'K' is static. The proof below verifies the strength of the security of the algorithm.

Proof:

1) V is the curve point taken of the elliptic curve.

$$V = u1 \times G + u2 \times Q. \tag{1}$$

2) We know that, public key as $Q = d \times G$.

$$V = u1 \times G + u2 \times (d \times G) \tag{2}$$

3) Distribution on addition is valid,

$$V = (u1 + u2 \times d) \times G. \tag{3}$$

4) Elaborating $u1$ and $u2$.

$$V = (z \times f1 + t \times f1 \times d) \times G. \tag{4}$$

$$V = (z + t \times d)f1 \times G. \tag{5}$$

$$V = (z + f \times d)(z + f \times d)1(k1)1 \times G \tag{6}$$

5) We know that inverting an already inverted element will give us the original element, and the product of its inverse and the element is the identity. Thus,

$$V = K \times G. \tag{7}$$

The data integrity plays a vital role in the security aspect to check whether the data has been modified or not by an attacker while the data was stored on the data server entity. Fig 5 shows the data integrity verification scheme to be followed by the communication entities, i.e., user, service provider, proxy server, and the data server. The steps that are being followed in Fig 5 are explained as below.

Step 1: The user sends the request along with the tag value to the service provider.

Step 2: The service provider formulates the block information based on the tag value and sends this information to the proxy server for the data integrity verification.

Step 3: The proxy server stores the tag value and block information (say Bt_I) sent by the service provider in the database.

Step 4: The proxy server sends the request for the digital signature along with the tag value to the data server.

Step 5: The data server extracts the signature on the basis of the tag value and forwards it to the proxy server.
Step 6: The proxy server compares the stored signatures with the received signatures and tag pair from the data server. If the tag matching is verified successfully, then it is proved that the data is not modified.
Step 7: The proxy server sends the data integrity verification result along with the data to the service provider.
Step 8: The service provider provides the requested data service to the user.

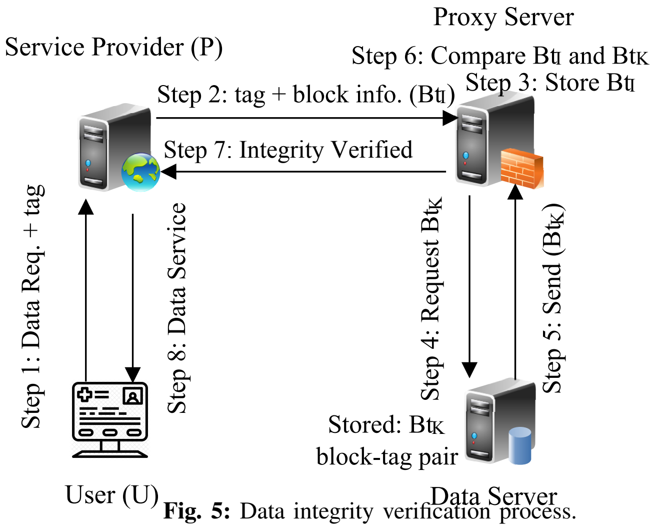

Fig. 5: Data integrity verification process.

Algorithm 2 shows the computation of the verification of the digital signature by the proxy server. Initially, the input received is in the form of block and tag (f, t) pair. Secondly, the message digest is computed using SHA-2 based on the data blocks. Next step is to compute the curve points. Finally, a digital signature is computed and matched with the received signature. The matched digital signature guarantees that the data integrity is achieved.

Algorithm 2 Verification by proxy server.

```
Input: Block-Tag pair
Output: Integrity Validation
1: Check whether f and t are in the range between [1,....., (n-1)].
2: Calculate e = HASH (B), where HASH is SHA-2 algorithm.
3: Compute $w = t(mod\ n)$.
4: Compute $u1 = z \times w(mod\ n)$ and $u2 = r \times w(mod\ n)$.
5: Compute the curve point
       $$(x1, y1) = u1 \times G + u2 \times Q$$
6: Digital signature is right if $f \equiv x1(mod\ n)$.
```

IV. RESULTS AND DISCUSSION

The following section discusses the simulation results obtained for the proposed scheme. In the simulation, the lattice operations are used for authentication while the ECC technique is used for integrity checking of the block.

A. *Performance analysis for authentication*

The performance analysis of the authentication scheme is based on the computation time and communication cost.

1) Computation time: The computation time is analyzed based on the operations performed by both the entities U and P. The list of functions and operations used are as follows: mod, sampling, cross rounding, modular rounding, randomized function, multiplication, addition, and reconciliation. The average time taken by the functions in milliseconds (ms) is T_{mod}- 0.774 ms, $T_{sampling}$- 0.265 ms, $T_{\lfloor\rceil}$ is 0.005 ms, $T_{<>}$ is 0.005 ms, T_{rand} - 0.005 ms, T_{mul} takes 0.12 ms, T_{add} is 1 ms, and T_{rec} takes 0.001 ms.
Entity *P*: The computation time taken in processing the operations are: T_P = 4 T_{mul} + 2 T_{add} + 2 T_{mod} + 1 T_{rec} = $(4 \times 0.12 + 2 \times 1 + 2 \times 0.774 + 1 \times 0.001)$ = 4.029 ms.
Entity *U*: The computation time taken by user U are: $T_U = 3 \times T_{sampling} + 2 \times T_{mul} + 2 \times T_{add} + 2 \times T_{mod} + 1 \times T_{rand} + 1 \times T_{\lfloor\rceil} + 1 \times T_{<>} = (3 \times 0.265 + 2 \times 0.12 + 2 \times 1 + 2 \times 0.774 + 3 \times 0.005)$= 4.598 ms.
Therefore, the total computation time is calculated as: $T_{tot} = T_P + T_U$= (4.029 + 4.598) ms = 8.627 ms.

2) Communication cost: The communication bits processed by entity U and P are the $B_{p_P}, B_{p_U}, B_{K_P}, B_{ACK}$. The public key and the session key is of 1024 bits each, while the ACK is of 32 bits. Thus, the total communication cost in bits is calculated as: $T_{tot} = B_{p_P} + B_{p_U} + B_{K_P} + B_{ACK} = (1024 + 1024 + 1024 + 32)$= 3104 bits.

B. *Performance analysis for integrity verification*

The message digest is created after hashing of the block by SHA-2 hashing technique. For comparison, the RSA (a discrete logarithmic technique) is also implemented to check which scheme is faster. For computing the time delay, the propagation time plus the processing time of both the schemes are computed. The result below shows the major findings.

Fig. 6 depicts the time latency for the two schemes. It can be seen in the figure that the latency for RSA is high and in most of the cases, it is almost 10-folds more expensive than the ECC technique. This observation is very crucial to understand as this shows the benefits of the proposed scheme on mobile devices which have very less computation power.

The other comparison is performed between the time latency when the user did not delegate the authority for integrity checking to the scenario in which user delegated. The associated findings are shown in Fig. 7.

It can be observed from the figure that the private checking is costlier than the delegated one. This can be attributed to the fact that the proxy server is more powerful as it is a dedicated machine. So, the delegated model is simpler and faster than the private model. It is also to be noted that when the authority is delegated to an external server, the user is free to do other important tasks. However, it should be noted that these results are only depicted for the scenario where the proxy server is located near to the user. So, if the user is far from the proxy server, than the propagation time will dominate and the effects of faster processing will diminish.

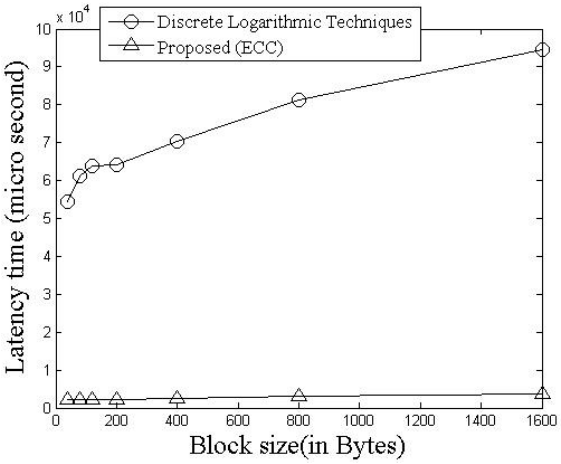

Fig. 6: Variation of latency time in RSA and ECC.

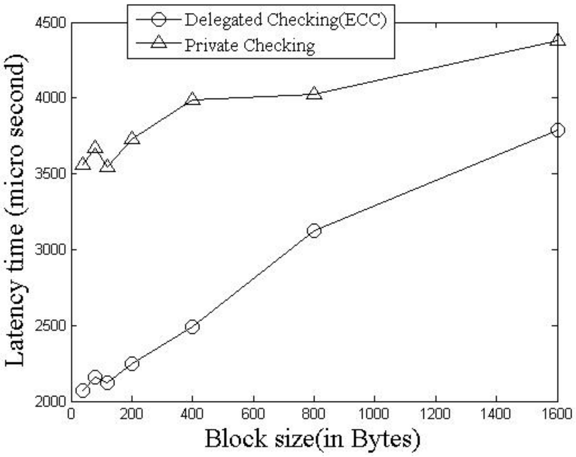

Fig. 7: Variation of latency time in delegated and private checking.

V. CONCLUSION

A scheme for authenticating the users' data requests and checking the message integrity for providing e-healthcare services is proposed in this paper. This scheme uses lightweight lattice operations to authenticate the users so as to put minimum additional burden on the network resources. This scheme also reduces the cost of integrity checking by using ECC which is less costly as compared to other integer factorization and discrete logarithmic techniques. This makes the complete security scheme computationally less expensive which makes it suitable for real-time implementation. Additionally, the overhead of authentication and integrity validation is moved from user to a third party authentication, i.e., a proxy server which further simplifies the system. The results prove that the computation time and communication costs for the authentication process are not much while the overhead in integrity checking is less as compared to the other traditional schemes.

ACKNOWLEDGMENTS

This work was partially supported by national funding from the FCT - Fundação para a Ciência e a Tecnologia through the UID/EEA/50008/2013 Project; by the Government of Russian Federation, Grant 074-U01; by Brazilian National Council for Research and Development (CNPq) via Grant No. $309335/2017-5$; by Finep, with resources from Funttel, Grant No. 01.14.0231.00, under the Centro de Referência em Radiocomunicações - CRR project of the Instituto Nacional de Telecomunicações (Inatel), Brazil.

REFERENCES

[1] O. Hamdi, M. A. Chalouf, D. Ouattara, and F. Krief, "eHealth: Survey on research projects, comparative study of telemonitoring architectures and main issues," *Journal of Network and Computer Applications*, vol. 46, pp. 100-112, 2014.

[2] C.W. Cheng, N. Chanani, J. Venugopalan, K. Maher, and M.D. Wang, "icuARM - An ICU Clinical Decision Support System Using Association Rule Mining," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 1, 2013, DOI: 10.1109/JTEHM.2013.2290113.

[3] D. Lin, X. Wu, F. Labeau, and A. Vasilakos, "Internet of Vehicles for E-Health Applications in View of EMI on Medical Sensors," *Journal of Sensors*, vol. 2015, pp. 1-10, 2015.

[4] N. Kumar, K. Kaur, A. Jindal, and J.J.P.C. Rodrigues, "Providing Healthcare Services On-the-Fly Using Multi-player Cooperation Game Theory in Internet of Vehicles (IoV) Environment," *Digital Communications and Networks*, vol. 1, no. 3, pp. 191-203, 2015.

[5] Z. Tari, X. Yi, U. Premarathne, P. Bertok, and I. Khalil, "Security and Privacy in Cloud Computing: Vision, Trends, and Challenges," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 30-38, 2015.

[6] J. L. Tsai and N. W. Lo, "A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805-815, 2015.

[7] F. A. Khan, A. Ali, H. Abbas, N. A. H. Haldar, "A cloud-based healthcare framework for security and patients' data privacy using wireless body area networks," *Procedia Computer Science*, vol. 34, pp. 511-517, 2014.

[8] Y. Tong, J. Sun, S. S. Chow, and P. Li, "Cloud-Assisted Mobile-Access of Health Data with Privacy and Auditability," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 419-429, 2014.

[9] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in 3^{rd} *ACM conference on Computer and communications security*, 1996, pp. 48-57.

[10] A. Boldyreva, A. Palacio, and B. Warinshchi, "Secure Proxy Signature Schemes for Delegation of Signing Rights," *Journal of Cryptology*, vol. 25, no. 1, pp. 57115, 2012.

[11] V. Lyubashevsky, C. Peikert, and O. Regev. "On ideal lattices and learning with errors over rings." *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, vol. 6110, pp. 1-23, 2010.

[12] R. Chaudhary, A. Jindal, G. S. Aujla, N. Kumar, A. K. Das and N. Saxena, "LSCSH: Lattice-Based Secure Cryptosystem for Smart Healthcare in Smart Cities Environment," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 24-32, 2018.

[13] C. Peikert, "An efficient and parallel Gaussian sampler for lattices." *Annual Cryptology Conference*, Springer, pp. 80-97 2010.

[14] J. Ding, X. Xiang, and L. Xiaodong "A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem." *IACR Cryptology EPrint Archive*, vol. 2012, pp. 688, 2012.