Security Improvement and Privacy Preservation in E-Health

Pushkar Kishore
CSE Dept.
NIT Rourkela,
Rourkela, India.
monumit46@gmail.com

Swadhin Kumar Barisal
CSE Dept, NIT Rourkela, India.
Siksha O Anusandhan Deemed to be
University, Bhubaneswar, India.
swadhinbarisal@gmail.com

Kulamala Vinod Kumar
CSE Dept.
NIT Rourkela
Rourkela, India
kvinod2208@gmail.com

Durga Prasad Mohapatra
CSE Dept.
NIT Rourkela
Rourkela, India.
durga@nitrkl.ac.in

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500873

***Abstract*—In today's world, all things are connected and influencing the existing applications. The E-Health domain extensively adopts IoT and presents new healthcare services and medical facilities. However, the major hurdle is improving security and preserving the patients' privacy. Many security/privacy-preserving models and protocols are proposed but prone to adversarial attacks. Our objective is to improve their security and ensure lightweight complexity for the low-powered and limited memory device. In this paper, we improve security using four critical steps. The effectivity of random number generation is improved, which defines the current security level in cryptography. A new technique is proposed utilising timestamp for handling a replay attack. We ensure strong forward security using the Elliptic Curve Discrete Logarithm Problem (ECDLP), making it challenging for an adversary to decode the security parameters. Finally, it is ensured that the hash function's bits maintain the entropy of the key involved in the security model. Thus, the proposed model preserves privacy as well as improves the security of the E-Health model.**

***Index Terms*—E-Health, random number, replay attack, strong forward security, entropy, privacy, security**

I. Introduction

The World Health Organization (WHO) describes E-Health as a cost-effective and safe way of improving health care services, health monitoring, medical education and research on novel diseases. The data available and maintained on the patient side is Patient Health Record (PHR). PHR contains the patient's medical record received from hospitals and accessible to any authorised agency. Reviews related to local health services are analysed using sentic computing [1], which exploit AI and semantic web techniques. Tweets related to outbreaks are processed using word vectors, and higher accuracy is achieved compared to CBOW [2]. To curb suicide rates, relation network [3] is applied along with an attention mechanism to prioritise more critical relational features. The methodologies discussed above needs secured transmission of data from E-Health devices to these models. The Electronic Health Record (EHR) has tools for managing health-related information. The data collected from the PHR is used for research purpose and offering therapeutic services. The EHR can organise, interpret and respond to the data. These days, mobile health is emerging, which provides health service on low-power portable devices like PDAs, mobile, health-band and other wireless devices. Mobile health services enable people to access their medical data at any point. For minor health issues, patients can contact the doctor and take consultations from home. This reduces medical expenses, and doctors can attend to many patients daily. Even doctors can watch the patient's health on e-health device without meeting them physically [4]. Some reasons which make E-Health trending are:

1) It can be used with low-power and cheap devices.
2) It is secured and globally accepted.
3) Medical facilities can be provided 24x7.
4) Patient data can be used for research purpose and suggesting proper medicines.

E-health devices play a vital role in total health revenue growth, reaching $10 trillion. Besides health data, these devices transmit other private data like the user's lifestyle and activities. So, the critical challenge is to improve the user's security and privacy. To assure security and privacy, cryptography is prominently used. However, there are some flaws, which an adversary can easily exploit.

In cryptography, the random number fulfils the CIA (Confidentiality, Integrity and Authentication). An encryption algorithm's key generation is secure whenever the random number generator has enough entropy [5]. A vulnerability in the Debian OpenSSL library reduced the set of potential outputs generated by the Cryptographically Secure Pseudorandom Number Generator (CSPRNGs). This bug had affected a lot of private keys. There is no means of proving the randomness of the RNG, but it can be tested intuitively. When an RNG passes all known polynomial statistical test, then it is fit for cryptography [6]. However, it is limited to a few statistical tests. The timestamp is generally used to avoid replay attack, but monitoring the attack threshold will be ineffective in some case. When the internet connectivity is slow at the user's end but fast at the adversary's end, the adversary's message will be considered but the original user's, discarded. For the same internet connection speed, both messages may lie within the threshold, creating confusion in finding the real user. Whenever the security parameter is leaked or stolen by an adversary, a spoofing attack can happen by analysing the encryption-decryption algorithm. It is observed that sometimes key bits are correlated after applying the one-way hash function and easily predictable.

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

Contributions: The feasibility of a random generator is proved using a biologically inspired technique that reflects CSPRNGs' quality. Our approach is statistical and used in addition to other statistical techniques. The contributions are summarised as:

- A statistical test is conducted using Numenta-Hierarchical Temporal Memory (N-HTM) to select the highest entropy random generator.
- The elliptic curve discrete logarithm is used to challenge adversary for retrieving the stolen ones' security parameters.
- Timestamp based technique is proposed for handling a replay attack on the model.
- A model relying on hash function is designed to identify between adversary and legitimate user during a replay attack.
- The minimal key bits, strong enough to maintain the hash function's entropy, is selected.

Outline: The rest of this paper is organised as follows: The next section emphasises some related work. Section III presents the proposed model; Section IV manifests the experimental results; Section V includes comparing state-of-the-art approaches. Section VI discusses the threats to our model's validity, and Section VII concludes the work along with future work.

II. Related Work

This section discusses the state-of-the-art works concerning this area.

A. E-Health

Guo et al. [7] proposed a model for detecting brain tumours and glioblastoma multiforme disease patterns. Al-Ayyoub et al. [8] implemented a five-times performance-enhanced hybrid Fuzzy C-Means algorithm utilised for extracting volume object from medical DICOM. These two works increased the quality of medical services or accelerated the diagnosis process. Ghoneim et al. [9] proposed a forgery system for medical images. They drew the image's noise map for applying a regression filter and fed the output to ELM (Extreme Learning Machine) and SVM classifiers. Dorgham et al. [10] secured data transfer to the cloud using a combination of symmetric encryption and asymmetric algorithms. Zhang et al. [11] designed PASH, a privacy-aware s-health access control system with partially hidden access policies. The values were hidden in encrypted s-health records, but attribute names were revealed. They addressed the security challenges, but privacy handling was feeble due to unsuitable security and privacy framework. Sadki et al. [12] compared works done in E-Health privacy and noticed that data privacy was not in reasonable control of the user. The privacy protection solutions were proposed, like access control, anonymous methods and encryption. Some authentication protocols used to secure authenticated access to Medical Information System have specific properties like unlinkability, privacy, untraceability, confidentiality, integrity, and availability. Some attacks can exploit key exchange mechanisms like denial of service, password guessing, impersonation, identity theft, and insider attack. Aslam et al. [13] reviewed these authentication protocols and highlighted their pros, cons and computational cost. Jiang et al. [14] designed a 3-factor business plan for smart health. Their scheme protected the user identity's privacy and provided mutual authentication using Burrows–Abadi–Needham logic. Irshad et al. [15] found a severe flaw in the mutual authentication scheme proposed by Jiang et al. [14], which may allow an adversary to launch a denial-of-service (DoS) attack. Liu et al. [16] used smart card and password to allow the only legal medical team to access patient data. They applied a secure cryptosystem for data transmission. Li et al. [17] presented a secure authentication and data encryption scheme for the IoT-based medical care system and prevented replay or password data disclosure attack. However, they did not use any standard tool for formal verification of their security approach. Beheshti-Atashgah et al. [18] designed a new framework for ensuring security and privacy in E-Health. The patient's identity and his/her record were secured, and privacy was maintained. Their authentication scheme was lightweight and satisfied all the security features. But they were prone to attacks like side-channel, replay or spoofing due to some issues in their framework.

B. Neuro-Cryptography

Silver et al. [19] designed AlphaGo, which defeated the world champion in the game, GO. They used reinforcement learning, where the algorithm was aware of the possible moves and combined them independently and improved to get rewarded often. Kaiser et al. [20] proposed MultiModel, which could learn simultaneously different tasks from multiple domains. Shanmuganathan et al. [21] discussed Artificial Neural Network (ANN) architectures, where the first layer received the input, and the last layer returned the final result. Ahmad et al. [22] proposed a technique named N-HTM, a biologically inspired technique useful for finding anomalies and their score in sequential data. It is a predictor suitable for real-time applications and finds anomalies in any data stream. It is preferred against ANN and deep learning techniques since it can detect subtle temporal anomalies, adapt to statistical data, maintain accuracy in the presence of noisy data and generates few false positives. Savicky et al. [23] used reinforcement learning to identify dependencies between random numbers generated using some Pseudorandom Number Generators (PRNGs) in MATLAB, e.g. Mersenne Twister (MT). Fan et al. [24] used ANN for measuring randomization in the digits of π along with MT. The results indicated that randomness was absent in both of them. Fischer [25] defined a way of testing PRNGs as well as CSPRNGs. To obtain a proper technique for testing RNGs, they compared three testers. The major drawback was training promising models for detecting randomness, which is ineffective in real-time.

III. Proposed Model

First, we discuss the E-Health security framework and conclude by emphasizing the solutions proposed for eliminating

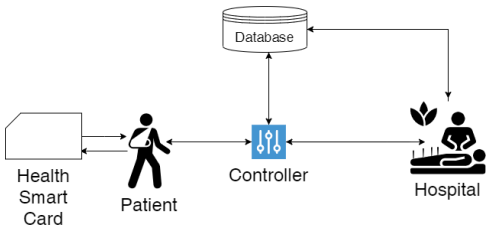

Fig. 1. Proposed security framework

challenges. Figure 1 has three entities: Patient, Controller, Hospital and two objects (Health Smart Card and Database). The four phases associated with this framework are:

A. Phase 1: Alias ID

An alias ID is used by the patient, $ID_i^{'}$, and doctor, $ID_j^{'}$, to stealth their original ID.

B. Phase 2: Initialization

Controller generates an additive group G with prime order $q \geq 3$ on an Elliptic Curve $E(F_p)$. P is the generator selected by controller, x_{ct} is the secret key, ID_{ct} is the identity and h_1, h_2 are two one-way hash functions.

C. Phase 3: Registration

Patient, P_i picks a random number r_0, password PW_i and alias identity $ID_i^{'}$. MP_i is computed using Equation 1.

$$MP_i = h_1(r_0||PW_i) \tag{1}$$

The controller is sent the message $[MP_i, ID_i^{'}]$. The controller evaluates e_i using Equation 2.

$$e_i = h_1(ID_{ct}||x_{ct}||ID_i^{'}) \oplus MP_i \tag{2}$$

Then the controller sends the message $[e_i, p, q, P]$ to the patient. The patient stores d_i evaluated using Equation 3.

$$d_i = h_1(ID_i^{'}||PW_i) \oplus r_0 \tag{3}$$

The doctor D_j sends the alias ID $ID_j^{'}$ to controller. The controller evaluates c_j using Equation 4.

$$c_j = h_1(ID_j^{'}||x_{ct}) \tag{4}$$

The patient and doctor computes his original ID using Equation 5 and 6.

$$B_i = e_i \oplus MP_i \oplus ID_i \tag{5}$$

$$B_j = c_j \oplus ID_j^{'} \oplus ID_j \tag{6}$$

The B_i and B_j is sent to the controller, and the controller receives back the original identity. As privacy is safeguarded; thus, the doctor and patient can use their original ID for ongoing phases.

D. Phase 4: Login and Authentication

The patient calculates r_1 and MP_i using Equation 7 and 8 respectively.

$$r_1 = d_i \oplus h_1(ID_i^{'}||PW_i) \tag{7}$$

$$MP_i = h_1(r_1||PW_i) \tag{8}$$

Then, the patient selects a number $\alpha \epsilon [1, q-1]$, r_2, r_3, doctor d_j and Timestamp T_i. The patient evaluates the parameters listed in Equation 9-11.

$$B_1 = e_i \oplus MP_i \oplus r_3 \tag{9}$$

$$B_2 = \alpha P \tag{10}$$

$$B_4 = h_1(r_3||ID_i||ID_j) \oplus ID_i \tag{11}$$

A message $M_1 = [B1, B2, B4, T_i, ID_j, ID_i^{'}]$ is created and sent to the controller. The controller checks whether $T_{current} - T_i \geq \Delta$. When adversary has better hardware and speediest network connectivity, then two similar messages are received within the threshold and request is sent to sender for sending back $h_1(T_i||e_i)$. Using this, we can find the adversary trying to get into the system. When an actual patient is identified with timestamp within threshold limit, the controller checks few more parameters like r_3 and ID_j using Equation 12 and 13.

$$r_3 = B1 \oplus h_1(ID_{ct}||x_{ct}||ID_i^{'}) \tag{12}$$

$$ID_i = B4 \oplus h_1(r_3||ID_i||ID_j) \tag{13}$$

If parameter checks are successful, controller selects a random number $\lambda \epsilon [1, q-1]$ and a Timestamp T_g. Then, a few more parameters are evaluated using Equation 14- 16 .

$$C_0 = \lambda P \tag{14}$$

$$c_j = h_1(ID_j^{'}) \tag{15}$$

$$D_1 = h_1(ID_i||ID_j||c_jC_0||B2||T_g) \tag{16}$$

Then the message $M_2 = [ID_i||ID_j, D_1, B_2, C_0, T_g]$ is sent to the doctor. The doctor checks the accuracy of D_1 and timestamp threshold of T_g in similar way as done for patient. If check results are fine, doctor selects a random number $\beta \epsilon [1, q-1]$ and evaluate additional parameters using Equation 17-20.

$$C_1 = \beta P \tag{17}$$

$$C_2 = \beta B_2 \tag{18}$$

$$sk_d = h_2(B_2||C_1||C_2) \tag{19}$$

$$C_4 = h_1(c_jC_0||ID_i||ID_j) \tag{20}$$

Then a message $M_3 = [C_4, C_1]$ is sent to the controller. The controller checks the correctness of the C_4 and evaluates D_5 using Equation 21.

$$D_5 = h_1(ID_i)||ID_j)||r_3||) \tag{21}$$

Controller sends message $M_4 = [D_5, C_1]$ to patient. Patient checks the correctness of D_5 and sets up session key as shown in Equation 22.

$$sk_p = h_2(B_2||C_1||r_3 \oplus MP_i||\alpha C_1) \tag{22}$$

Finally, the correctness of C_4 is verified and d_i and e_i are updated using Equation 23-24.

$$d_i^{new} = h_1(ID_i^{'}||PW_i) \oplus r_2 \tag{23}$$

$$e_i^{new} = h_1(ID_{ct}||x_{ct}||ID_i^{'}) \oplus h(r_2||PW_i) \tag{24}$$

After these phases are accomplished, the patient and doctor can communicate via session key or use ECDLP. We use ECDLP due to its tough behaviour while breaking. There are some operations which are discussed below.

E. Random Number Generation

There are popular PRNGs like rand(), MT, arc4random() and CSPRNGs like SHA1PRNG and /dev/urandom. We examine all of them using N-HTM. There are three reasons for using N-HTM, which are discussed below:

1) Entropy is dependent upon non-prediction of futuristic element and absence of pattern. It implies that a predictor can be used for analysing these random number generators. The predictions which are not even close to accurate ones are termed anomalies. So, the percentage of anomalies in the data stream defines entropy. A higher amount of anomalies is preferable for security purpose. So, analysers like Long Short Term Memory (LSTM), Gated Recurrent Units (GRU) and N-HTM are considered.
2) LSTM and GRU model need to train the data, but unsuitable for real-time. We consistently monitor the pool of random numbers and enhance the pool size if the entropy decreases in real-time. N-HTM solves this issue by working in real-time, even without training.
3) The volume of data required for learning is enormous for LSTM and GRU, but some instances suffice for N-HTM to find anomalies with few false positives.

F. Used Hash function

Communication via a network channel is used for sending and receiving data. The security of the channel is a questionable point to ponder. In the past, algorithms like SHA-1, MD5, SHA-2, etc., were broken. So, it cannot be guaranteed that they can never be broken. Thus, two techniques are proposed for enhancing the quality of the hash function. Whenever the channel is secure (Private Network), the number of strong bits are trimmed in the hash output. Suppose we need a key with an attack resistance of 160 bits, but it needs to be used with a cipher that uses 192-bit keys. Then, Bits 1-160 will be $A1 = hash - function(Data||0x00)$, i.e., we concatenate a single octet of value 0x00 to the right side of the data. For the Bits 161-192, it is $select32bits(hash - function(A1||0x01))$. For insecure (Public Network), the number of strong bits are increased in the hashed output. Then, bits 1-160 will be $A2 = hash - function(0x00||Data)$, i.e., we concatenate a single octet of value 0x00 to the left side of the data. For the Bits 161-192, it is $select32bits(hash - function(0x01||Data))$. Iteration over the data is done to increase the strength; otherwise, entropy diminishes upon iterating over the same bits obtained at 1-160.

TABLE I
TEST RESULT OF DIFFERENT PRNG

Sl. No.	PRNG	$\bar{\mathbf{H}}$	$\bar{\mathbf{var}}$	p-value	Entropy presence
1	rand()	0.479	0.027	0	0.959
2	MT	0.478	**0.02**	0	0.91
3	arc4random	**0.482**	0.3	0.001	**0.961**
4	SHA1PRNG	**0.48**	**0.023**	0.001	**0.964**
5	/dev/urandom	**0.48**	**0.025**	**0.032**	**0.961**

IV. EXPERIMENTAL RESULTS

The analysis of the random generator technique's test results are summarized in Table I. The performance metrics used in Table I are $\bar{\mathbf{H}}$, $\bar{\mathbf{var}}$, p-value and Entropy presence. The $\bar{\mathbf{H}}$ is the mean of all losses during testing. The range of acceptable value is [0.48,0.482] since a higher value indicates that the pattern is not found in the stream of random numbers. The $\bar{\mathbf{var}}$ is the mean-variance of the predicted numbers. The lower value is preferable, and the acceptable range is [0.02,0.025]. The p-value is evaluated using the Kolmogorov-Smirnov (KS) test. This test indicates the amount of inconsistency in the sample. The higher value is preferred, and the acceptable range is [0.03,0.032]. Lastly, the entropy indicates the proportion of anomalies (non-predicted random data) in the total data stream. Whenever it is higher, the random generator is efficient and adjudged suitable for cryptography. Its acceptable range is [0.96,0.964]. So, it is concluded from Table I, /dev/urandom is most prefered for cryptography while rand() is the worst one. The generators' security coverage is highlighted in the order: (/dev/urandom, SHA1PRNG, acr4random, MT and rand()).
Now, the proposed security framework's resistance is analysed in the presence of numerous threats and attacks.

1) **Resistance to insider attack:** During the registration phase, P_i shares MP_i with controller. If the adversary tries to break the hash function, r_0 is needed. However, the secrecy of the random number is maintained by the patient; thus, PW_i is safe.
2) **Resistance to offline password guessing attack:** Whenever M_1 is eavesdropped by an adversary, then communication message between patient and controller is open. However, the absence of PW_i and the random number in the message prevents offline password guessing.
3) **Resistance to user forgery attack:** Adversary has not got x_{ct} and hash functions clear the messages received within the threshold. So, forging a message is almost impossible.
4) **Resistance to controller forgery attack:** Adversary cannot evaluate r_3 or other security parameters as x_{ct} is unknown.

TABLE II
COMPUTATIONAL COMPLEXITY OF THE PROPOSED FRAMEWORK

Phase	Step	Computing Cost	Total Computing Cost
Registration	P_i and D_i	2-Hash; 5-XOR	
	Controller	2-Hash; 5-XOR	4-Hash; 10-XOR
Login	P_i	9-Hash; 7-XOR	
and	Controller	7-Hash; 3-XOR	
Authenticate	D_i	3-Hash	19-Hash; 10-XOR

5) **Resistance to replay attack:** Random numbers, timestamps, along with hashed reply for threshold issue, secures our framework against a replay attack.
6) **Resistance to known-key attack:** In the proposed scheme, the α and β are randomly selected at each session. Therefore, the attacker cannot determine the next session keys even if they have one session key.
7) **Patient anonymity:** Patient and doctor's anonymous ID is for registration, while a real ID is used for other phases. Thus, privacy is maintained on the communication channel.
8) **Strong forward security:** The ECDLP makes it tough for an adversary to predict the security parameters.
9) **Resistance to desynchronisation attack:** During presence of inconsistency and adversary, the session is straightaway terminated to avoid desynchronisation attack. We also evaluate our proposed model's computational complexity, presented in Table II. We show the time required for performing the hash function using $HASH$ and exclusive-or operation using XOR. The time required for performing the hash function is represented using $HASH$ and exclusive-or operation using XOR. Table II reveals that our proposed framework has low computational complexity. On the other hand, the time required for XOR can be neglected, and finally, it can be concluded that our proposed framework is lightweight.

V. COMPARISON WITH STATE-OF-THE-ART APPROACHES

Table III provides the state-of-the-art model's analysis result of the random number generator using RNNs. Table III points out that LSTM and BLSTM prefer PRNG selection (4/5 times) while GRU and BGRU suggest only 2 PRNG. Our proposed model is useful in predicting all of them correctly as we consider the entropy presence parameter against the p-value. The p-value is not considered since the two distributions with a small sample size are not entirely different, thus reducing the test's significance. For relying only on p-value, more samples are needed leading to RNN's longer training time. However, our proposed model, N-HTM, does not need training and effectively classify PRNGs using another metric, entropy presence. The amount of data required for training and testing bidirectional RNNs are more than their linear one. This illustrates their inefficiency in testing RNGs. GRU classifies PRNG to unknown state two times while LSTM does this for a single time only. Therefore, LSTM can be considered for testing RNG. If we consider the entropy presence parameter, then N-HTM is efficient than LSTM for classification in real-time.

TABLE III
TEST RESULT OF VARIOUS PRNG USING DIFFERENT TECHNIQUES

PRNG	Model	H	var̄	p-value
rand()	LSTM	0.48	0.08	0
rand()	GRU	0.48	0.08	0
rand()	BLSTM	0.48	0.09	0
rand()	BGRU	0.48	0.08	0
rand()	N-HTM	0.47	0.027	0
MT	LSTM	0.43	0.09	0
MT	GRU	0.45	0.07	0
MT	BLSTM	0.48	0.06	0
MT	BGRU	0.48	**0.05**	0
MT	N-HTM	0.47	**0.02**	0
Arc4random	LSTM	**0.49**	0.07	0.001
Arc4random	GRU	**0.49**	**0.05**	0
Arc4random	BLSTM	**0.49**	0.06	0.001
Arc4random	BGRU	**0.49**	**0.05**	0.006
Arc4random	N-HTM	**0.48**	0.3	0.001
SHA1PRNG	LSTM	**0.49**	**0.05**	0.001
SHA1PRNG	GRU	**0.49**	0.06	0.002
SHA1PRNG	BLSTM	**0.49**	0.06	0.001
SHA1PRNG	BGRU	**0.49**	**0.04**	0.001
SHA1PRNG	N-HTM	**0.48**	**0.023**	0.001
/dev/urandom	LSTM	**0.49**	**0.05**	**0.031**
/dev/urandom	GRU	**0.49**	**0.04**	0.001
/dev/urandom	BLSTM	**0.50**	**0.05**	**0.015**
/dev/urandom	BGRU	**0.49**	**0.04**	0.006
/dev/urandom	N-HTM	**0.48**	**0.025**	**0.032**

TABLE IV
COMPUTATIONAL COMPLEXITY OF FRAMEWORK BY BEHESHTI-ATASHGAH ET AL. [18]

Phase	Step	Computing Cost	Total Computing Cost
Registration	P_i	2-Hash; 1-XOR	
	Controller	3-Hash; 2-XOR	5-Hash; 3-XOR
Login	P_i	12-Hash; 12-XOR	
and	Controller	7-Hash; 8-XOR	
Authenticate	D_i	4-Hash; 1-XOR	23-Hash; 21-XOR

The computing cost of the security framework outlined by Beheshti-Atashgah et al. [18] is analysed and described in Table IV. Table IV confirms that their framework needs 3 XOR operations and 5 Hash functions during the registration phase. During the login and authentication phase, their model needs 23 Hash functions and 21 XOR operations. Considering all these operations, the adequate number of Hash functions for our proposed model is 23 compared to 28 in the state-of-the-art work. Thus, our proposed framework is lightweight and resistant to nine widespread attacks.

VI. THREATS TO THE VALIDITY OF OUR MODEL

We prefer random generators having lower complexity to get accurate entropy presence. The Hardware Random

Number Generator (HRNG) is excluded from our testing due to its potency in generating a high-quality random number. The results are generated upon executing PNGs based on Python, C or Java language. The number of samples in the experiment is fixed at 100,000, and generated numbers are within a fixed range. The results will be consistent even after the reduction in the number of samples. Our proposed framework has solutions for security issues in registration, login, and authentication. It will not assure security during a password change or clicking malicious links on the E-health APP or devices for other phases.

VII. Conclusion and Future Work

We propose a security framework that will ensure the patient and doctor's security and privacy in an E-Health system. This framework secures the registration, login and authentication phase and resilient against adversary attacks. The random number generators are tested using N-HTM to determine the data stream's entropy in real-time, thus enhancing the security mechanism. We alert the user to improve the generator's quality whenever entropy drops. For finding the adversary during the replay attack, the real user is asked to send their hashed data. The hashed data judges whether or not the adversary is present. Data is sent using Elliptic Curve Discrete Logarithm Problem (ECDLP), making it challenging for an adversary to crack security parameters. Hash function output security's strength is modified based on channel security, and it is more challenging for an adversary to get back the data from the hashed result.

In the future, we will analyse the hardware random number generator. We will add a password mechanism and design framework for finding Adware, Backdoor or Trojan, which targets the E-health APP. The channel's security is improved, but the APP should be equally secured for a fully secured and privacy based system. Lastly, we will try to reduce the security framework's complexity such that it works faster on light-weight processors.

References

[1] E. Cambria, A. Hussain, T. Durrani, C. Havasi, C. Eckl, and J. Munro, "Sentic computing for patient centered applications," in *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*, 2010, pp. 1279–1282.

[2] A. Khatua, A. Khatua, and E. Cambria, "A tale of two epidemics: Contextual word2vec for classifying twitter streams during outbreaks," *Information Processing & Management*, vol. 56, no. 1, pp. 247–257, 2019.

[3] S. Ji, X. Li, Z. Huang, and E. Cambria, "Suicidal ideation and mental disorder detection with attentive relation networks," *arXiv preprint arXiv:2004.07601*, 2020.

[4] F. Zubaydi, A. Saleh, F. Aloul, and A. Sagahyroon, "Security of mobile health (mhealth) systems," in *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, 2015, pp. 1–5.

[5] H. Shulman and M. Waidner, "Method for generating a random number, random number generation circuit and computer program," May 28 2020, uS Patent App. 16/661,026.

[6] P. L'Ecuyer and R. Simard, "Testu01: Ac library for empirical testing of random number generators," *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 4, pp. 1–40, 2007.

[7] P. Guo and P. Bhattacharya, "An innovative model for detecting brain tumors and glioblastoma multiforme disease patterns," *International Journal of Software Science and Computational Intelligence (IJSSCI)*, vol. 9, no. 4, pp. 34–45, 2017.

[8] M. Al-Ayyoub, S. AlZu'bi, Y. Jararweh, M. A. Shehab, and B. B. Gupta, "Accelerating 3d medical volume segmentation using gpus," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4939–4958, 2018.

[9] A. Ghoneim, G. Muhammad, S. U. Amin, and B. Gupta, "Medical image forgery detection for smart healthcare," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 33–37, 2018.

[10] O. Dorgham, B. Al-Rahamneh, A. Almomani, K. F. Khatatneh *et al.*, "Enhancing the security of exchanging and storing dicom medical images on the cloud," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 8, no. 1, pp. 154–172, 2018.

[11] Y. Zhang, R. H. Deng, G. Han, and D. Zheng, "Secure smart health with privacy-aware aggregate authentication and access control in internet of things," *Journal of Network and Computer Applications*, vol. 123, pp. 89–100, 2018.

[12] S. Sadki and H. El Bakkali, "Towards controlled-privacy in e-health: A comparative study," in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*. IEEE, 2014, pp. 674–679.

[13] M. U. Aslam, A. Derhab, K. Saleem, H. Abbas, M. Orgun, W. Iqbal, and B. Aslam, "A survey of authentication schemes in telecare medicine information systems," *Journal of medical systems*, vol. 41, no. 1, p. 14, 2017.

[14] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-health clouds," *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826–3849, 2016.

[15] A. Irshad and S. A. Chaudhry, "Comments on "a privacy preserving three-factor authentication protocol for e-health clouds"," *The Journal of Supercomputing*, vol. 73, no. 4, pp. 1504–1508, 2017.

[16] C.-H. Liu and Y.-F. Chung, "Secure user authentication scheme for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 59, pp. 250–261, 2017.

[17] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, and C.-M. Chen, "An efficient user authentication and user anonymity scheme with provably security for iot-based medical care system," *Sensors*, vol. 17, no. 7, p. 1482, 2017.

[18] M. Beheshti-Atashgah, M. R. Aref, M. Barari, and M. Bayat, "Security and privacy-preserving in e-health: a new framework for patient," *Internet of Things*, p. 100290, 2020.

[19] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[20] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, "One model to learn them all," *arXiv preprint arXiv:1706.05137*, 2017.

[21] S. Shanmuganathan, "Artificial neural network modelling: An introduction," in *Artificial neural network modelling*. Springer, 2016, pp. 1–14.

[22] S. Ahmad, D. George, J. L. Edwards, W. C. Saphir, F. Astier, and R. Marianetti, "Hierarchical temporal memory (htm) system deployed as web service," May 20 2014, uS Patent 8,732,098.

[23] P. Savicky and M. Robnik-Šikonja, "Learning random numbers: A matlab anomaly," *Applied Artificial Intelligence*, vol. 22, no. 3, pp. 254–265, 2008.

[24] F. Fan and G. Wang, "Learning from pseudo-randomness with an artificial neural network–does god play pseudo-dice?" *IEEE Access*, vol. 6, pp. 22 987–22 992, 2018.

[25] T. Fischer, "Testing cryptographically secure pseudo random number generators with artificial neural networks," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 1214–1223.