International Journal of Cyber-Security and Digital Forensics (IJCSDF) 5(3): 132-141
The Society of Digital Information and Wireless Communications, 2016 (ISSN: 2305-0012)

Enhancing the Security of Electronic Medical Records Using Forward Secure Secret Key Encryption Scheme

Kosmas Kapis and Edwin Kambo
College of Information and Communication Technologies,
University of Dar es Salaam, P.O. Box 33335, Dar es Salaam,
Tanzania.
kapis@udsm.ac.tz, kkapis@gmail.com, edwinstevek@gmail.com

ABSTRACT

Electronic Medical Records (EMRs) pose a number of security challenges. Conventional cryptographic solutions have been used to protect the patients EMRs but with one major weakness; in an event where there is an exposure of a secret key, the secrecy of past and future encrypted data is void. This work concentrated on integrating the concept of forward secrecy in the protection of EMRs.
In order to come up with the right integration, an experimental simulation has been carried out to evaluate the performance of forward secure and non-forward secure secret key encryption schemes.
The study shows that only few cryptographic schemes that have implemented forward secrecy are used in the protection of EMRs, and very few of them use the forward secure secret key encryption scheme (FSSKES). The experiment also shows that the cost of running forward secure symmetric encryption is closely related to that of running non-forward secure symmetric encryption scheme (NFSSKES).

KEYWORDS

Forward Secure, Cryptography, Symmetric, Asymmetric, EMRs

1 INTRODUCTION

Medical practitioners face a number of challenges from different sources such as governments, religious organizations, cultural beliefs, and even from privacy supporters as far as the Hippocratic Oath is concerned. The Hippocratic Oath was created to be the building block of the recent philosophies and moral thinking. It was created with the intention to protect privacy of patients. It directs the medical practitioners not to disclose what they may hear or see while performing treatment or even when not performing treatment [1].
The growth of technology feeds the flow of information which proves to be a problematic. To respond to this, the Institute of Medicine of the National Academy of Sciences, which is a nonprofit organization published a report in 1991 explaining the importance of protecting the medical data. In 1997, the report was revised identifying the advancement made in computer based patient records were a number of recommendations were made such the issue of security [2]. Again in 2000, a new report was published with the focus of protecting privacy of data in health service researches. Recommendations were made advising health care organizations to handle well sensitive data by having good policies, procedures as well as other forms.

The concept of forward security has been developed and few studies have implemented in the EMRs protection. This paper addresses four questions in relation to the EMRs protection. The first question had to do with finding the weaknesses of the existing cryptographic solutions used in the

protection of EMRs. The second question has to do with understanding how forward secure secret key encryption scheme is different with other forward secure encryption schemes. The third question looks at devising a way to implement a forward secure secret key encryption scheme, and the last question is about finding performance comparison between the forward secure secret key encryption scheme and non-forward secure secret key encryption scheme.

Forward Security

Several studies that have been carried out in the field of forward secrecy have come with a number of challenges. These challenges are pointed out in [3]. The concern of challenges focuses on protection of secret keys and it is futher stated that currently, globally patients demand protects of their EMR. One of the challenges is that of exponetial growth of healthcare services dependency on the systems that are digital and whose security systems are becoming more important every day. These systems are developed to be connective but this connectivity is also a problem, attacker can have access to the systems easily from any part of the world. Thus, physical separation is no longer enough to ensure security of the systems, and therefore these systems must depend on other forms of guranteeing their security. All these forms depend depend on keeping or maintaining the secret keys. This means that, the security of the system, depends on the condition that the secret key does not to fall in the hands of the adversaries. This condition is difficult to satisfy, due to a number of reasons. One being, the secret keys are continually used by the people, so they are prone to human errors, like forgetting to keep them hidden, and thus causing unintended keys compromise and any other hacking techniques available to adversaries. The solution is to have a way to protect past encrypted messages even when the key is lost or exposed to an adversary. This concept is thecalled forward security.

Forward Security can be applied to the cryptography branches. According to [4] these branches are, one, symmetric algorithms: the main characteristic of this algorithm is the sharing of the secret key, two parties will share the same key in performing encryption and decryption process. Second branch is what is known as the asymmetric (or public-key) algorithms. In this cryptography, a part has two keys, a private key and a public key. These algorithm are normally used in digital signatures as well as in establishing of key and in traditional encryption of data.

This paper focuses on using the symmetric encryption scheme in implementation of the forward secure secret key encryption scheme. The summary of the forward secure encryption scheme is given in Table 1.

Table 1. Summary of Forward Secure Encryption Schemes

S/N	AUTHOR	TITLE	CYRPTOGRAPHY AREA
1	Back, A. (1996).	Non Interactive Forward Secrecy. Cypherpunks Mailing List	Asymmetric
2	Song	Practical Forward Secure Group Signatures	Asymmetric: group signatures
3	Bellare and Miner (1999)	Forward Secure Digital Signature Scheme	Asymmetric: signature schemes
4	Itkis and Reyzin (2001)	Forward Secure Signatures with Optimal Signing and Verifying	Asymmetric: signature schemes
5	Kozlov and Reyzin	Forward Secure	Asymmetric: Signature schemes

	(2003)	Signatures with Fast Key Update	
6	Abdalla, Miner, and Nampempre (2001)	Forward-Secure Threshold Signature Schemes	Asymmetric: signature schemes
7	Ma and Tsudik (2007)	Forward Secure Sequential Aggregate Authentication	Asymmetric: authentication and signatures schemes
8	Van Le, Burmester, and De Medeiros (2007)	Universally Composable And Forward-Secure RFID Authentication and Authenticated Key Exchange	Symmetric and Protocols
9	Huang, Adhikarla, Boneh, and Jackson (2014)	An Experimental Study Of Transport Layer Security (TLS) Forward Secrecy Deployments.	Symmetric and Asymetric
10	Yang lu	Efficient Forward-Secure Public Key Encryption Scheme Without Random Oracles	Asymmetric
11	Itkis (2004)	Forward Security Adaptive Cryptography: Time Evolution	Symmetric and Asymmetric
12	Bellare and Yee (2003)	Forward Security In Private Key Cryptography	Symmetric

2 APPROACH

In an attempt to provide answers to the set questions in the preceding section, a number of literature reviews were visited to answer question one and two, and then java simulations programs were developed form which the experiments were performed, results provided the answer to question four.

2.1 Implementation of FSSKES

The implementation of the forward secure secret key encryption scheme, was based on the algorithm that was proposed in [5]. The study proposed an algorithm that follows the concepts of forward security in symmetric settings. The research aimed at generating different encryption key for every medical record entry.

In order to describe the coopted algorithm, some notations and abbreviations were introduced and used in the study: t = 1,2, …,T, denotes an arbitrary period of time, where T is the total number of keys/periods; *PRF* represents a pseudorandom function; K_{MK} denotes the master key; K_t denotes the encryption key generated by the *PRF* from the master key K_{MK} at time t; e is an encryption function, M is the electronic medical record(*EMR*) and Ct is a cipher of the *EMR* at time t.

Using the given description, the following is the description of how the algorithm works. A *PRF* is used and from it we get a K_{MK}. Then at each period of time t and encryption key $K_t = PRF(K_{MK,t})$ is computed. K_{MK} is kept secret at all times. A block cipher e, for example advanced encryption standard (*AES*) can be used for encryption. Figure 1 shows the process of generating symmetric encryption keys and *EMR* encryption as described in this paper.

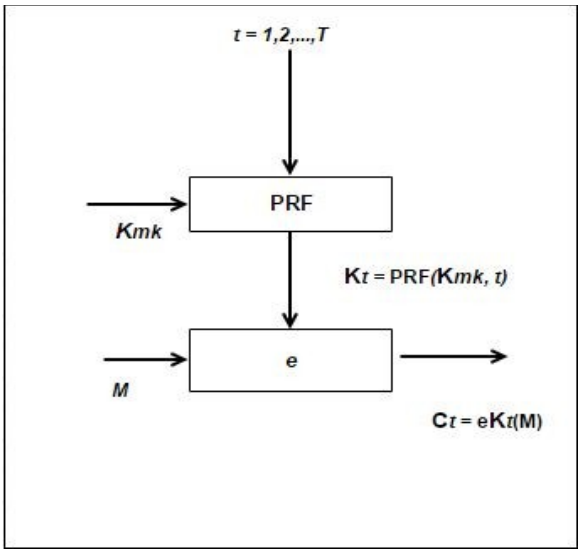

Figure 1.Generation of symmetric encryption keys and EPR/EMR encryption; (Source: [5])

This algorithm worked as the base for the development of the application that simulated the running of a FSSKES that were carried out in the study. Also a different simulated application that did not use the concept of forward security, namely, NFSSKES was developed. The two simulated applications were used to perform the experiment that aimed at analyzing the performance of FSSKES Vs NFSSKES to establish, the grounds to use FSSKES in terms of performance.

2.1.1 The Experiment

The experiment that was carried out, aimed at computing the performance of none forward secure secret key encryption scheme (NFSSKES) and that of forward secure secret key encryption scheme (FSSKES), and later comparing the results. The main components of the experiment were; the message generation part, the key generation part, the encryption part, the decryption part and the time measurement part.

2.1.2 Message Generation

The plaintext that was encrypted came from the contents of a book. The file generated was saved with .txt extension. The original first file, formed the first packet size. To generate the second packet size, and other packets sizes, a new file was created by copying and pasting the original content of the file, to the end of the first file, which is the same as duplicating the first file, then triplicating it, all the way to the tenths packet. The packet sizes were, 611,616 bytes; 1,223,236 bytes; 1,834,856 bytes; 2,446,476 bytes; 3,058,096 bytes; 3,669,716 bytes; 4,281,336 bytes; 4,892,956 bytes; 5,504,576 bytes; and 6,116,196 bytes for the last packet. The same packets were used in both experiments for NFSSKES and FSSKES.

2.1.3 Key Generation

In generating secret keys for symmetric cryptography, it is required that the same key to be used for encryption and decryption process. A java method was developed that used hash function and random number generators to generate the secret keys. The setKey method that was used to generate the secret key, received a string of text to be converted to key bytes using the getByte method. The byte keys are digested by an instance of MessageDigest algorithm, the output is passed as an input to SecretKeySpec together with AES encryption algorithm to create a secretKey.
In the NFSSKES case, the key generation for this experiment, the method was called only once, and the generated key was used to encrypt and decrypt ten different packet sizes each with a 10 iterations. In each iteration three measurements were taken and recorded. Whereas in the FSSKES case, the method was called 10 times, that is ten different keys were generated. A master key, was a random string of text, from which the first secret key was generated. The second

key was generated using the first key as the seed. This is the required setting for any forward secure encryption scheme.
For each key generated, encryption and decryption of a single packet size was done and time measurement recorded. Again for every packet the process was repeated 10 times from which an average measurement was taken to represent the accurate time measurement.

2.1.4 Encryption Process

The encryption process used the secret keys that were generated by the Advanced Encryption Standard (AES) algorithm. The plaintext generated was encrypted and the output sent to the cipher text file that was prepared. Figure 2. and Figure 3. show the flowchart for the two experiment settings.
For both experiments, the same java encryption method was used to encrypt a message file, the only difference is on the ways the keys were used. In none forward secure secret key encryption scheme settings, only one same key was used to encrypt all ten packets with different sizes. Again for each packet the three measurements were taken and recorded. In forward secure secret key encryption scheme settings, ten different keys were used to encrypt the packets, each packet was encrypted using a different key. Also in this setting, time measurements were taken for each packet and recorded.

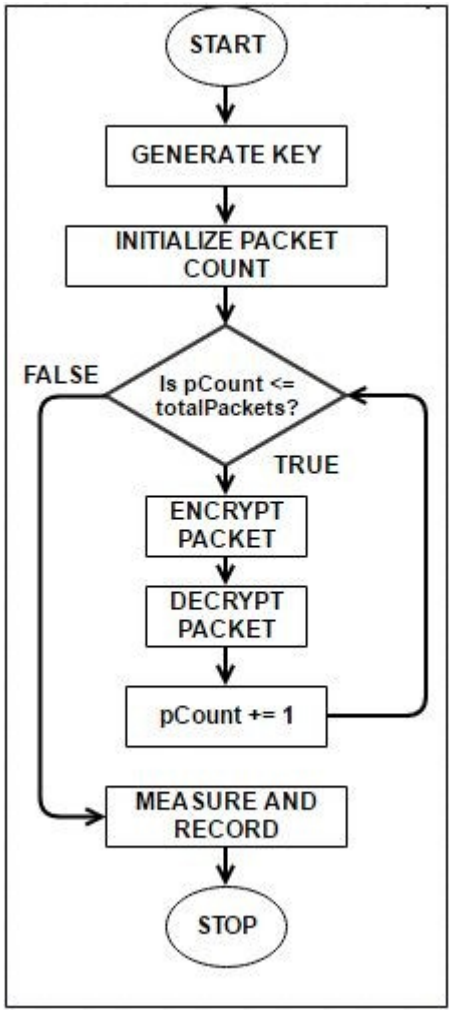

Figure 2. None Forward Secure Secret Key Encryption Scheme Flowchart

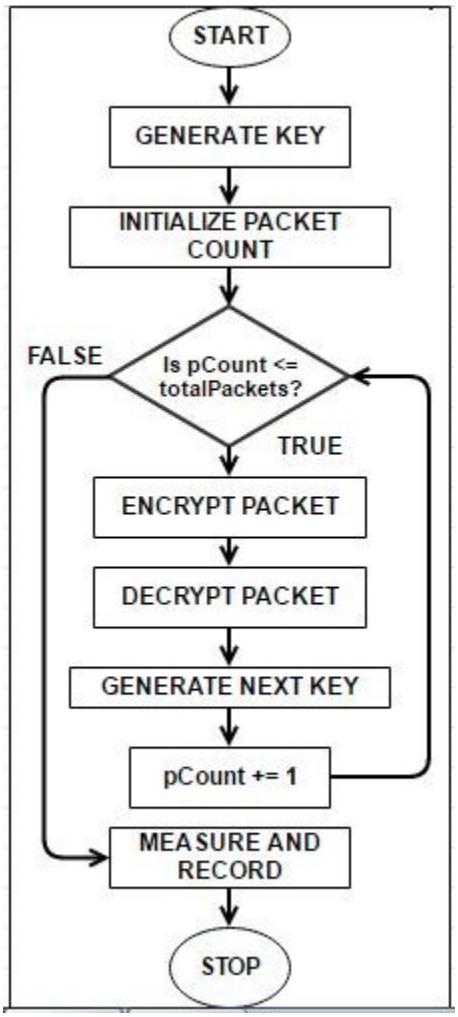

Figure 3. FSSKES Flowchart

2.1.5 Decryption Process

The decryption process used the same secret key that was generated by the method explained, to decrypt the cipher text. In both experiment settings, the same java method was used to decrypt the encrypted file. The only difference is on how the method was used.

In none forward secure secret key settings, the decryption process used only one generated key to decrypt the encrypted file. In forward secure secret key settings, the decryption process used ten different keys to decrypt each encrypted packet file.

2.1.6 Time Measurements Process

In both settings, measurements were taken and recorded. Two java imports were used to find the execution times. These are ManagementFactory and ThreadMXBean both belong to the java language management class with path: java.lang.management. The method, getCurrentThreadCpuTime(),getCurrentThreadUserTime() and getCurrentThreadUserTime( ) were used to get the relevant times.

3 RESULTS

In the experimental setups, 10 different data packets were used in the encryption and decryption process, and for each data packet, 10 iterations were done from which an average variable value was obtain. This was done to improve the accuracy.

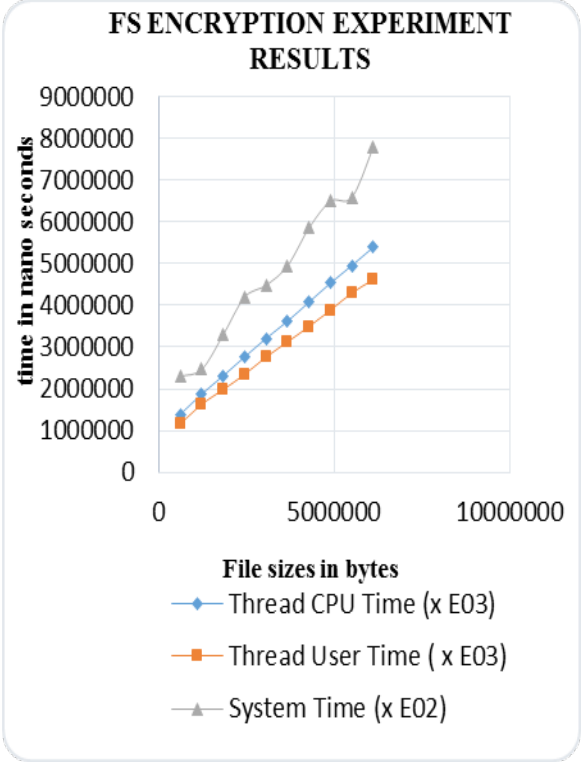

Figure 4. NFS experiment results

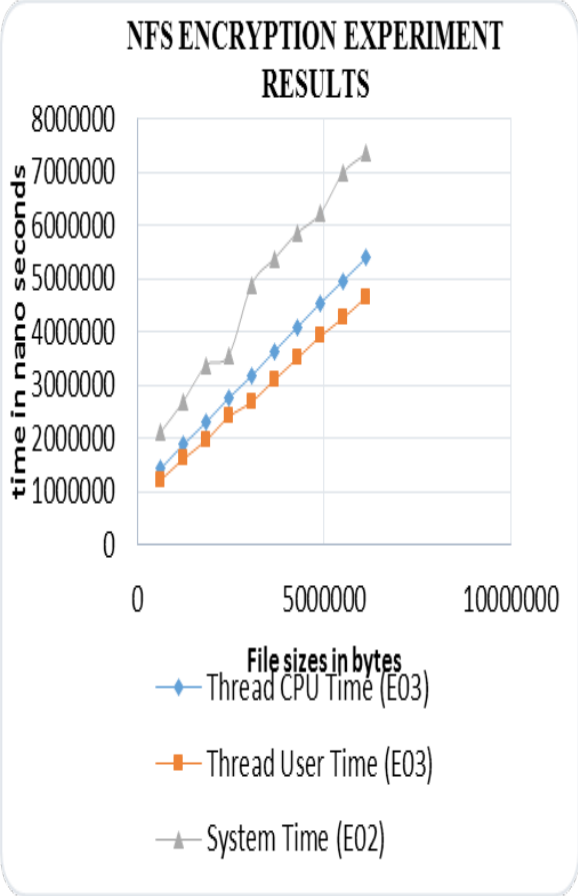

Figure 5. FS experiment results

Figure 4. and Figure 5. show the graphs depicting the results obtained from the two experiment.

3.1 Comparing the Results of the Experiments

In comparing the results of forward secure and none forward secure settings for secret key encryption scheme, the two graphs were joined and the result of their combination resulted to a new graph, as depicted in Figure 6.

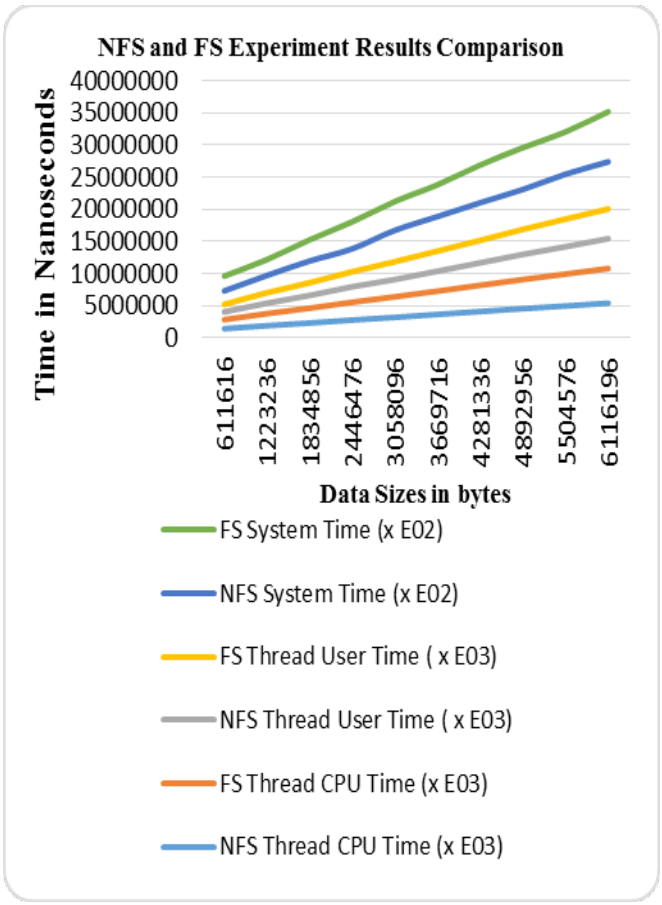

Figure 6. A comparison of FS and NFS secret key encryption schemes

It has been shown by other researchers that execution time is directly proportional to the running cost of the application [6]. Using the same concept, from the figure 6, it was observed that the running cost for NFSSKES is closely relating to the running cost of FSSKES. This fact was confirmed by computing the slopes of the two graphs in Figure 4 and Figure 5. The result showed that FSSKES graphs slopes were, slightly higher than those of NFSSKES graph slopes, this is because in forward security settings, the execution times were slightly higher than those of none forward secure.

Throughput is the rate of processing data per unit time, [6]. Inverting the slopes obtained in Table 2., gives out the throughput. Figure 7 shows the relationship in percentage, the

throughput of FSSKES and NFSSKES using the thread CPU time.

Table 2. Average Slopes of the Two Experimental Settings

SLOPES	Thread CPU Time(EO3) /FILESIZE	Thread User Time(EO3) /FILESIZE	System Time(EO2) /FILESIZE
NFSSKES	0.735	0.638	0.976
FSSKES	0.744	0.641	1.027

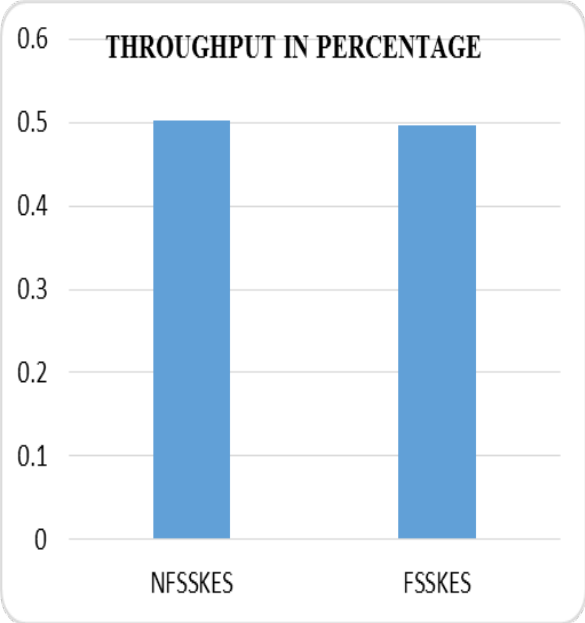

Figure 7. Performance comparison

4 DISCUSSIONS

The results of the experiments have shown that FSSKES performs just as the conventional symmetric encryptions schemes with a slight lower throughput which is an advantages because the FSSKES has an additional ability to protect past encrypted messages. The disadvantage of using FSSKES which has been observed was on the implementation of the property of having a master key that can override all its child key. This property threatens the security of the system because there is a threat of an adversary discovering the master key which can prove to be a disaster.

4.1 Lack of Implementation of Forward Secrecy on Medical Records

It was observed that, from going through a number of studies, there were few studies that have implemented the concept of forward security in protection of medical records. Only two, before this study, and these are by [5] and [7]. More studies are needed so that the security of EMRs can be improved.

4.2 Affordability in Running Forward Secure Encryption Schemes

The experiments proved that, the cost of running forward secure encryption schemes in symmetrical settings was affordable, and other times it was even cheaper. The research noted that, the only cost that was added, was in the key generation. Other cost were similar to ordinary symmetrical encryption schemes. The added cost for running forward secure secret key encryption scheme in terms of percentage, for thread CPU time, it was approximated to 0.572% and for thread user time, it was 0.264% and for system time, it was 2.539%.
In this part, the answers to the set guiding research questions are going to be provided. The first question focused on finding the weaknesses of the existing cryptographic solutions used in the protection of EMRs. The main weakness that was observed after going through a number of studies, on different techniques that have been used to protect EMRs, was that, most of the cryptographic techniques used, had not addressed the protection of past encrypted messages.

Second on finding the difference between forward secure secret key encryption schemes with other forward secure encryption schemes. In answering this research question, more than 10 studies were reviewed. It was found that most of the studies that related to forward secrecy, had to do with forward secrecy on the asymmetric side of cryptographic encryption schemes. The main differences that existed between forward secure secret key encryption schemes and the other forward encryption schemes, were based on the differences of their underlying cryptography techniques that is symmetric or asymmetric. Different studies have explained their differences, concept. The fast encryption time and decryption time of symmetric encryption scheme led researchers to adopt the algorithm in the process of improving the security of EMRs.

Regarding implementation of a forward secure secret key encryption scheme, a simulation tool was developed and implemented the forward secure secret key encryption scheme using java programming language. The symmetric encryption scheme that was implemented was an AES for encryption and decryption purposes, and an addition of a Blowfish algorithm was used to protect the secret key at a particular period of time. A pseudo random number generator was used to generate the key, as provided by the java package, and a SHA-1 MessageDigest algorithm was used in the generation of the keys.

Lastly, the question on the performance of the forward secure secret key encryption scheme as compared to non-forward secure secret key encryption scheme. Using a simulation experiment, the result showed that there was a very small difference of running cost when comparing the two encryption settings. The non-forward secure secret key encryption scheme had a higher throughput of about 0.57% more than that of forward secure secret key encryption scheme. This matched with other studies by [8], [9] and [10]. The addition of key generation to both cryptographic system just added the complexity to both techniques. It did not affect their basic cryptographic characteristics. The experiment performed in the study, has proved this.

5 REFERENCES

1. Edelstein, L. (1943). The Hippocratic oath, text, translation and interpretation, page 56 ISBN 978-0-8018-0184-6.
2. Dick, R. S., Steen, E. B., & Detmer, D. E. (1997). *The Computer-Based Patient Record: An Essential Technology for Health Care*: National Academies Press page 50-51.
3. Itkis, G. (2004). Forward security, adaptive cryptography: Time evolution page 3.
4. Paar, C., & Pelzl, J. (2010). *Understanding cryptography: a textbook for students and practitioners* (Vol. 1): Springer-Verlag Berlin Heidelberg page 3-4.
5. Kapis, K. (2011). *Security and privacy of electronic patients records.* Thesis, The open university of Tanzania, Tanzania page 75-77.
6. Elminaam, D. S. A., Abdual-Kader, H. M., & Hadhoud, M. M. (2010). Evaluating The Performance of Symmetric Encryption Algorithms. *IJ Network Security, 10*(3), 216-222.
7. Yu, Y.-C., Huang, T.-Y., & Hou, T.-W. (2012). Forward secure digital signature for electronic medical records. *Journal of medical systems, 36*(2), 399-406.
8. Tripathi, R., & Agrawal, S. (2014). Comparative Study of Symmetric and Asymmetric Cryptography Techniques. *International Journal of Advance Foundation and Research in Computer (IJAFRC), 1*(1).
9. Kumar, Y., Munjal, R., & Sharma, H. (2011). Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures. *International Journal of Computer Science and Management Studies, 11*(03).
10. Sasi, S. B., Dixon, D., Wilson, J., & No, P. (2014). A General Comparison of Symmetric and Asymmetric Cryptosystems for WSNs and an Overview of Location Based Encryption Technique for Improving

Security. *IOSR Journal of Engineering, page 3-4.*