ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/320260429

Securing Private Keys in Electronic Health Records Using Session-Based Hierarchical Key Encryption

Article *in* Journal of Applied Security Research · October 2017

DOI: 10.1080/19361610.2017.1354272

CITATIONS
5

READS
146

3 authors, including:

Adebayo Omotosho
Universität Potsdam
39 PUBLICATIONS **83** CITATIONS

SEE PROFILE

Justice Emuoyibofarhe
Ladoke Akintola University of Technology
50 PUBLICATIONS **127** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Smart and Secure Tele-Clinic diagnostic Systems View project

Project Mobile traffic Management View project

All content following this page was uploaded by Adebayo Omotosho on 09 January 2019.

The user has requested enhancement of the downloaded file.

Routledge
Taylor & Francis Group

Journal of Applied Security Research

ISSN: 1936-1610 (Print) 1936-1629 (Online) Journal homepage: http://tandfonline.com/loi/wasr20

Securing Private Keys in Electronic Health Records Using Session-Based Hierarchical Key Encryption

Adebayo Omotosho, Justice Emuoyibofarhe & Alice Oke

To cite this article: Adebayo Omotosho, Justice Emuoyibofarhe & Alice Oke (2017) Securing Private Keys in Electronic Health Records Using Session-Based Hierarchical Key Encryption, Journal of Applied Security Research, 12:4, 463-477

To link to this article: http://dx.doi.org/10.1080/19361610.2017.1354272

Published online: 06 Oct 2017.

Submit your article to this journal

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
http://tandfonline.com/action/journalInformation?journalCode=wasr20

Download by: [Dr Adebayo Omotosho] **Date:** 06 October 2017, At: 14:16

JOURNAL OF APPLIED SECURITY RESEARCH
2017, VOL. 12, NO. 4, 463–477
https://doi.org/10.1080/19361610.2017.1354272

Routledge
Taylor & Francis Group

Check for updates

Securing Private Keys in Electronic Health Records Using Session-Based Hierarchical Key Encryption

Adebayo Omotosho[a], Justice Emuoyibofarhe[b], and Alice Oke[b]

[a]Department of Computer Science, Landmark University, Omu-Aran, Kwara State, Nigeria; [b]Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

ABSTRACT

Patients want the assurance that the confidentiality of their records accessed through Electronic Health Records (EHR) are safe. With increasing implementation of EHR for health care, privacy concern remains a barrier that limits patients' favorable judgment of this technology. Sensitive records can be compromised and this represents problems in EHRs, which are considered to be more efficient, less error prone, and of higher availability compared to traditional paper health records. In this article, a session-based hierarchical key encryption system was developed that allows patient to have full control over certain nodes of their health records. Health records were organized in a hierarchical structure with records further broken down into subcategories. Cryptography was used to encrypt the health records in their different subcategories. Patients' generate a root keys using Blum Blum Shub Algorithm for pseudorandom number generator from which the session-based subkeys were derived, and only authorize users can access these records within a designated period marked as session. The system development demonstrates one way patients' privacy and security can improve using session-based hierarchical key encryption system for EHR.

KEYWORDS

Health records; EHR; private keys; security, privacy

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

Introduction

Security is a major concern in today's computerized solutions and cryptography has been recognized as one of the most popular technologies to solve the security problems in several applications. Cryptography is simply the art of secret writing and securing information from eavesdroppers. Encryption can be applied in verifying the authentication of a user accessing a system and securing communication in network applications areas. Generally, the strength of most key-based cryptographic systems is dependent on the security of the cipher keys, a security system without strong management procedures and processes has no security (Martin, 2006).

Key management is the administration of tasks involved with protecting, storing, backing up, and organizing encryption keys. Cryptography keys falls into two

CONTACT Adebayo Omotosho bayotosho@gmail.com Department of Computer Science, Landmark University, P.M.B. 1001, Omu-Aran, Kwara State, Nigeria.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/wasr.

© 2017 Taylor & Francis Group, LLC

categories: private key and public key cryptography. A private key (or a symmetric key) is used to perform encryption or decryption using symmetric cryptographic algorithms, and the same key is required for encryption and decryption. A public key is used in asymmetric algorithms where the key used to encrypt a message is not the same as the key used to decrypt it. Each user has a pair of a public encryption key and a private decryption key. Cryptographic key management encompasses the entire lifecycle of a cryptographic key. Keying material and life cycle includes: user registration, system and user initialization, keying material installation, key establishment, key registration, operational use, storage of keying material, key update, key recovery, key deregistration and destruction, and key revocation (Barker, Barker, Burr, Polk, & Smid, 2007).

Electronic Health Records (EHR) are online versions of health record paper charts. An EHR may include medical history, notes, and other information about an individual. EHR allow providers to use information more effectively to improve the quality and efficiency of care, but EHR will not change the privacy protections or security safeguards that apply to patients' health information. Privacy concern is arguably a major barrier that restricts appreciation of EHR systems which are considered to be more efficient, less error prone, and of higher availability compared to traditional paper record systems (Sun, Zhu, Zhang, & Fang, 2011). Privacy and security are important in EHR, however security can only be improved upon but cannot be totally guaranteed because the authentication of data can be compromised by attackers. Personal health information of hundreds of thousands of people has been compromised because of security lapses at hospitals, insurance companies, and government agencies. EHR are open to possible abuses and threats and it has been reported that large amounts of sensitive healthcare information held in data centers is vulnerable to loss, leakage, or theft (Benaloh, Chase, Horvitz, & Lauter, 2009).

Existing methods used in EHR to protect patient's privacy are inadequate given that: (a) the use of smart card approach requires the presence of the patient to have access to the health record, and the smartcard can either be stolen or misplaced, and also be compromised (Jeng & Wang, 2006); (b) approaches that grant full access control to the patient makes it difficult in times of emergency (Hupperich, Löhr, Sadeghi, & Winandy, 2012); (c) recent approaches relied heavily on cryptography technologies whose popular flaw is the security of the cipher keys (Peltier, 2016; Premarathne et al., 2016; Smart, 2016). This study is aimed at the use of cryptography using hierarchical session-based keys to improve patients' privacy in EHR.

Key management cycle

On key generation, personalized cryptographic keys from face biometric were generated with a two stage technique in Teoh, Ngo, and Goh (2004). In the first stage, transformation of biometric input was discretized to generate a set of bit representation and a set of tokenized pseudo random number, coined as FaceHash. In the second stage, FaceHash was then securely reduced to a single cryptographic

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

key via Shamir secret sharing. Using hierarchy (Access Table ACCESS) and Diffie–Hellman based key generation scheme, Zych, Petkovic, and Jonker (2008) presented a key management scheme for cryptographic enforcement of access control where each user stores a single key and is capable of correctly calculating the suitable keys needed to access requested data. This technique does not require encryption of the same data several times with the keys of different users or groups of users. As the system was designed mainly for the purpose of access control, space needed for storing public parameters was significantly reduced. Omotosho and Emuoyibofarhe (2015) presented a method for private key generation, sharing, and storage from image features through Gray Level Co-occurrence Matrix. In order to achieve storage and generation at the same time, cipher keys were derived, rather than hidden in users' images, from a second order statistics for keys computation and recomputation. The study was able to generate reproducible 120bits key length suitable for symmetric encryption. This method was further improved and deployed in Omotosho, Emuoyibofarhe, and Meinel (2017) to include key expiration through the addition of time features embedded into the keys in an EHR application. This approach implemented with fuzzy vault and fuzzy commitment bio-cryptography was able to ensure patients control of their EHR privacy.

On key derivation, Lin, Huang, Lai, and Lee (2009) used a Shared Key derivation Protocol (SKP) that comprises, key derivation function, hash function (SHA-1) and Pseudo Random Number Generator (PRNG). A group key management protocol based on shared key derivation methods was developed to reduce the communication and computation elevation of centralized secure group communication systems. With shared key derivation, the server does not have to encrypt and transmit new keys to members who have enough information to derive the keys by themselves. The performance of rekeying operations including single join, single leave, and batch update improved. Similarly, Lee, Ho, and Lee (2011) described a new key management scheme to facilitate control of EHR by providing two functionalities, using master key approach and threshold secret sharing. First, a patient can authorize more than one healthcare institute within a designated time period to access his or her protected health information. Second, a patient can revoke authorization and add new authorized institutes at any time as necessary. This implementation was reported to be time and cost efficient

With respect to key distribution, in Zhu, Bao, Deng, Kankanhalli, and Wang (2005), the challenges in security, efficiency, flexibility, and adaptively in key management in large Mobile Ad Hoc Networks (MANET) was solved by using two approaches. At the architecture level, a new key management scheme was proposed to distribute cryptographic keys and provide certification services. At the algorithm level, two algorithms which are based on threshold cryptography and Feldman's VSS scheme, were used independently for the key management scheme. The utilization of autonomous key management using Shamir Secret sharing ensures that a stronger protection to the Global Secret Key can be provided. Jeng and Wang (2006) presented a work on key generation, key derivation, and key selection that used Elliptic Curve Cryptosystem (ECC) to solve the hierarchical access control

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

problem. An efficient key management and derivation scheme based on the ECC was presented where each class in the hierarchy is allowed to select its own secret key and adding or deleting classes was solved without the necessity of regenerating keys for all the users in the hierarchy.

Key generation, key update, and key distribution were described in Pour, Kumakawa, Kato, and Itoh (2007) using Logical Key Hierarchy. The authors proposed an improved protocol for rekeying in logical hierarchy model for secure multicast and increasing the efficiency of key distribution in leave operation. The protocol was able to reduce rekeying overhead, the number of encryptions and transmissions as well as the multicast message size. Using Microsoft Active Directory for storing keys, Acar, Belenkiy, Ellison, and Nguyen (2010) carried out a study on key distribution, key update, and key storage. A key lifecycle management that can be used for cryptographic data protection was developed. The software client accesses keys and other metadata stored in a distributed repository, and the system hides all key management tasks from the user. The user specifies a key management policy and the system enforces this policy. Xiao and colleagues (2007) reported on key establishment, key predistribution, key discovery, and key revocation in sensor network. This work discussed private key schemes (single network wide key, pairwise key establishment, trusted base station, and authentication), public key schemes (Rivest-Shamir-Adleman [RSA] and ECC), key predistribution schemes, and hierarchical key management. The study provided an overview of these techniques, each of which offers different advantages and disadvantages. It was observed that no key distribution technique is faultless to all the scenarios where sensor networks are used and, hence, techniques employed must rely on the requirements of target applications and resources of each individual sensor network.

Mukherjee, Gupta, and Agarawal (2008) presented key generation and key establishment research that uses threshold cryptography and Node-Group-Key (NGK) mapping. Their work introduces key generation mechanism employing a central trusted entity, only during initialization using a group key algorithm and the concept of NGK mapping. This was done with regards to the existing solutions to group security in MANETs depending on multicast Core Based Tree for key distribution which is a problem for dynamic and sparse groups with changing neighborhoods. Using this approach, keys were established between group members with absolutely no prior communication. Khan, Pastrone, Lavagno, and Spirito (2012) proposed a key establishment that employs ECC. An improved, secure mutual authentication and key establishment protocol based on ECC, through which different classes of nodes with very different capabilities can authenticate each other and establish a secret key for secure communication was generated. Ziauddin and Dailey (2010), focused on key generation and key recovery using fuzzy commitment approach to biometric key management. Their method created a scheme for secret key generation and recovery based on iris verification, rather than storing keys or iris templates, it stores recovery information on a smart card carried by the user. This approach uses error-correcting codes to overcome the noisiness inherent in biometric readings in order to achieve correct key generation. Wang and Ma (2012) used smart card-based authentication scheme for key agreement. A smart

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

card-based authentication scheme with key agreement suitable for global computing environments was proposed. The main advantages of this approach are that a security-sensitive verification table is not required in the server, the password can be chosen and changed freely by the clients and cannot be derived by the privileged administrator of the server. Also, the client and the server can establish a common session key. This methodology can protect the client secret key even if the smart card is damaged. On the subject of key exchange and key update, Guo, Xu, Li, Yao, and Mu (2013) using multi-identity key management protocol, proposed an efficient and dynamic identity-based authenticated key management protocol to optimize key management for a user with multiple identities, this scheme allows a user with some basic identities to compute a new private key when some new identities are involved.

Some important deductions made from this review are as follows:

1. The techniques that utilizes smart-card approach have similar limitations, for example, there is a possibility of the user misplacing the smart-card or it being stolen or damaged.
2. Attribute-based encryption is commonly used to authorize access to sections in EHRs, and hierarchical architecture to distribute keys in the hierarchy.
3. Most of the methods that were proposed for the EHR gives full control to the patient, allowing them to determine who to grant authorize access and this can affect the clinician workflow if encryption is total.
4. The most common-used technique to implement EHRs is attribute-based encryption.

Methodology

Proposed architecture of the hierarchical key encryption system (HKES) for health record

As represented in Figure 1 in the proposed system, the encrypted EHR is organized into hierarchical structure. The patient record is further broken down

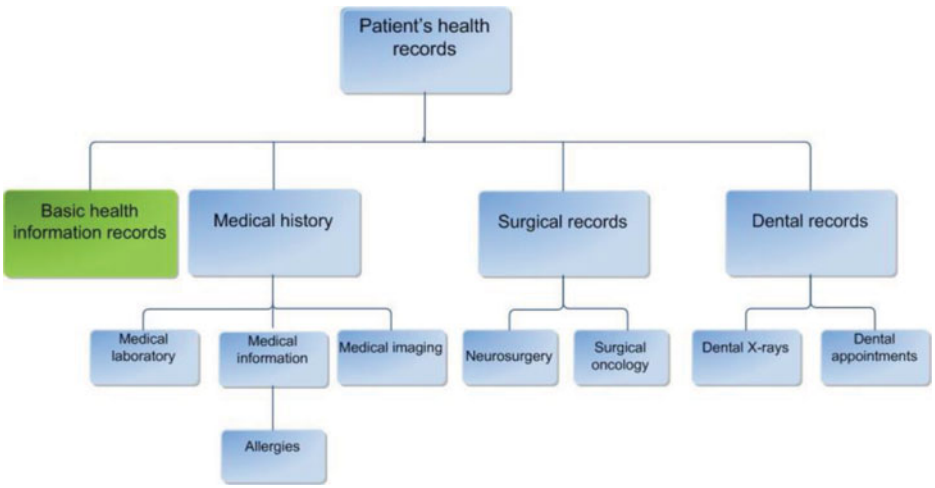

Figure 1. Hierarchical key encryption system for EHR.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

into a high-level categories and subcategories. A record partitioned according to a single hierarchy does not necessarily imply that a structure cannot be easily extended in a case where there are several different hierarchies which can be used to organize the patient's record. This proposed system allows a patient to generate his or her own decryption key which will be used to generate subkeys to authorize access to only certain parts of his or her record, and for a period of time, thereby, establishing a common session key. A key design consideration for this system is to allow the patient to delegate access to any subset of their records categories to any requester such as doctor, dentist, psychiatrists, employer, or spouse. This is because a patient may not want to share his or her entire record with a certain requester for personal reasons to avoid unwanted discrimination and embarrassment. First, the patient will generate and store his or her own secret key, which will be called the root key. Second, the patient can use this root key to generate subkeys for various categories or subcategories. To authorize rights to access a particular record on the hierarchy, the patient would generate and attach a private subkey to their record categories, these session-based keys would be established and acknowledged to authorize access to the record for the stipulated period.

Considering the sample hierarchy displayed in Figure 1, the basic health information record or bio-data will always be unencrypted and the contents of such "public" category were described in Omotosho and colleagues (2017). The patient might decide to grant her dentist access to both the dental records category and the medical information category. This would allow the dentist to read all data concerning dental appointment, dental x-rays, and allergies. However, the dentist would have no way of decrypting any of the information in the patient's surgical records, or medical imaging data. Dental records is an ancestor of dental x-rays because dental x-rays are contained in dental records.

In deriving the system architecture in Figure 2, we assumed that the patient records are stored as a group of records or entries and each record contains the name of the record, the name of the categories in that record, and the record tag which is used to identify the record and the encrypted record itself. The record tag will be encrypted along with the record so as not to reveal any information about the record.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

Building the HKES

In the proposed system, we say that *cat* represents the category on the hierarchy, *cat* $(x_l, \ldots, x_e)$ specifies a category on the hierarchy where (x_1) is the top level ancestor of the category, (x_1, x_2) specifies the next ancestor down the hierarchy and so on. sk_{root} specifies the decryption root key for the patient.

HKES symmetric algorithm (adapted from Benaloh et al., 2009)

This describes the construction for a hierarchical set of categories, which only allows a private key encryption.

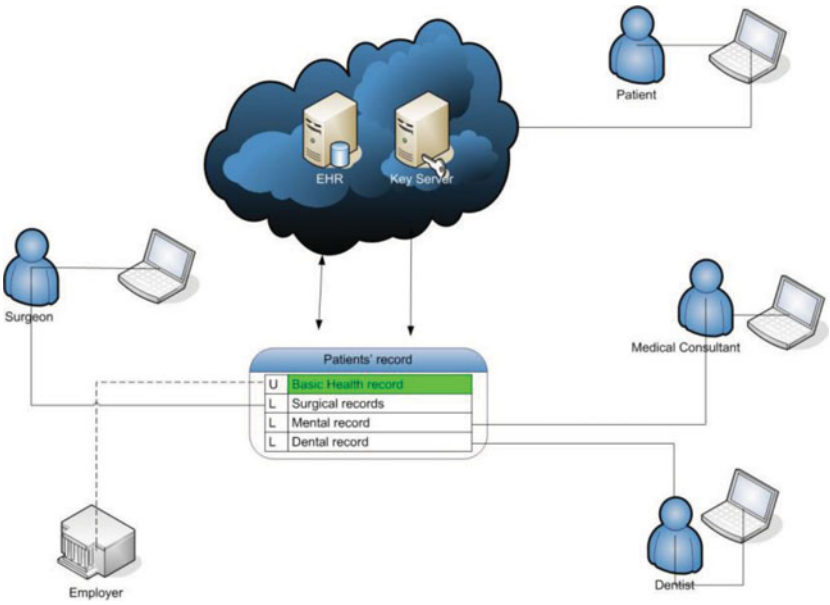

Figure 2. Hierarchical key encryption system for EHR conceptual model.

i. The key generation algorithm KeyGen(1^k) → sk_{root} which takes as input the security parameter k and generates a root decryption key for the patient $sk_{root.}$
ii. They key derivation algorithm F(KeyDer(sk (x_1, . . ., $x_{e\text{-}1}$, t);(x_1, . . ., x_e)) → sk (x_1, . . ., x_e) takes as input the name of a category specified as a hierarchical list (x_1, . . ., x_e) and the decryption key sk(x_1, . . ., $x_{e\text{-}1}$) for the patient category cat(x_1, . . ., x_e). or (sk_{root} for $e = 1$). It outputs a decryption key sk(x_1, . . ., x_e) for category cat(x_1, . . ., x_e).
iii. The encryption algorithm Enc(sk(x_1, . . ., x_e); (x_1, . . ., x_e); m) → c takes as input a public key, a message m, a category name specified as (x_1, . . ., x_e), and the corresponding decryption key sk (x_1, . . ., x_e). It outputs an encryption of m for category cat(x_1, . . ., x_e).
iv. A decryption algorithm Dec(sk (x_1, . . ., x_e); (x_1, . . ., x_e); c) m takes as input the name of a category ((x_1, . . ., x_e)), a corresponding decryption key sk (x_1, . . ., x_e) and a ciphertext c. It outputs decrypted message m if the ciphertext was formed correctly for category cat(x_1, . . ., x_e).

Root key generation method

For the purpose of demonstrating the HKES, the Blum Blum Shub (BBS) algorithm for Pseudo Random Number Generator (PRNG) is used for generating the root key. BBS chooses two odd primes p and q, and compute $n = p$ *and* q (Ankur & Divyanjaliy, 2013; Sidorenko & Schoenmakers, 2005) Then square modulo n of seed is computed and the resulting number is considered to be the first number generated. The seed is replaced with generated number in subsequent iterations and a square modulo n is computed again, generating a number per iteration. To gather bit sequence, least significant bit of generated number is extracted per iteration and is added to the generated binary sequence. Hence, BBS needs only one square (or multiplication)

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

per bit generated. Let the seed be s, $s \in Z_{n,}$ then first number is

$$X_1 \equiv s^2 \ mod \ n \quad \text{and the bit } b_1 \equiv X_1 \ mod \ 2$$

For i^{th} iteration

$$X_i \equiv X_{i-1}^2 mod \ n \quad \text{and } b_i \equiv X_i mod \ 2$$

This way the algorithm needs to compute only one square operation to generate a bit, which is much less than any of the other cryptographically secure algorithm. Following is the pseudo code of the algorithm:

BLUM_BLUM_SHUB (SEED):

i. $X_0 = SEED$
ii. Choose two odd prime p and q
iii. $n \leftarrow p \ . \ q$
ii. $l \leftarrow$ length of sequence
v. for $i \leftarrow 1$ to l
vi. do $x_i \equiv x_{i-1}^2 \ mod \ n$
vii. do $b_i \equiv x_i \ mod \ 2$

return B = $\langle b_1 b_2 b_{3....} b_i \rangle$

Vulnerabilities of BBS. The security of BBS is related to the hardness of factorization which makes it a quite effective PRNG but BBS is not the most secure PRNG and it has two problems: (a) BBS can be very slow and (b) the existence of the proof of security for BBS can also be misleading and usually misinterpreted.

The standard proof of security states that if you choose a modulus N that is large enough, then no attacker will be able to efficiently break BBS. For BBS to be provably secure, it needs a very big modulus, larger than those regularly used with RSA. Such that, if using BBS with a 2048-bit modulus, there is no proof of security and using BBS this way renders the security warranty null and void; the proof is no longer applicable. However, if using BBS with a much larger modulus, the resulting effect is that BBS becomes ridiculously slow. In other words, BBS is a very good theoretical construct, but of negligible relevance to practical crypto-engineering (Chatterjee, Menezes, & Sarkar, 2011; Sidorenko & Schoenmakers, 2005)

Implementation and discussions

Figure 3 shows the homepage of the patient. The design of this EHR allows patients to have full control to their health record except some "public" basic health information record. Patients, therefore, can determine who can view or access the other parts of their record. The side tabs allow the patient to navigate and browse easily through the system. For a patient to book an appointment with a doctor, the patient has to request for a doctor on their page as shown in Figure 4 and selects from the list of available doctors provided by their registered hospital. After a successful establishment of a session, the patient then selects the record categories that the requesting doctor will be allowed to view, depending on the medical conditions diagnosed for as shown in Figure 5. The Y and N symbol means YES and NO. Selecting Y means

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

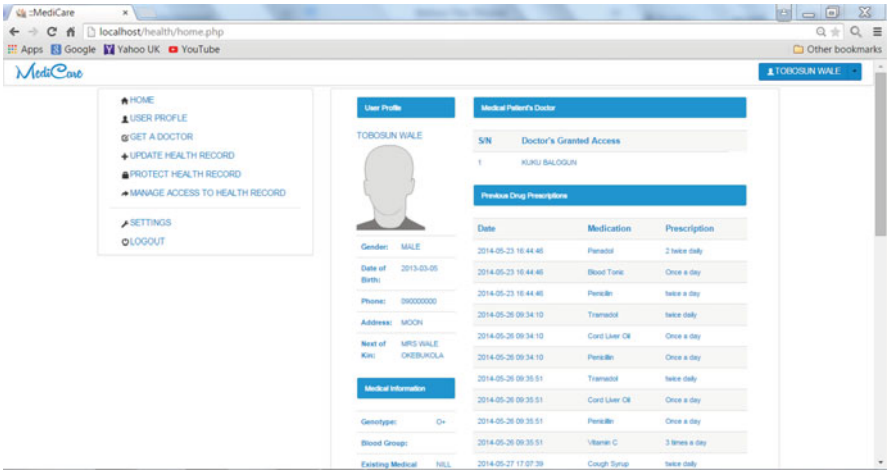

Figure 3. Patient's homepage.

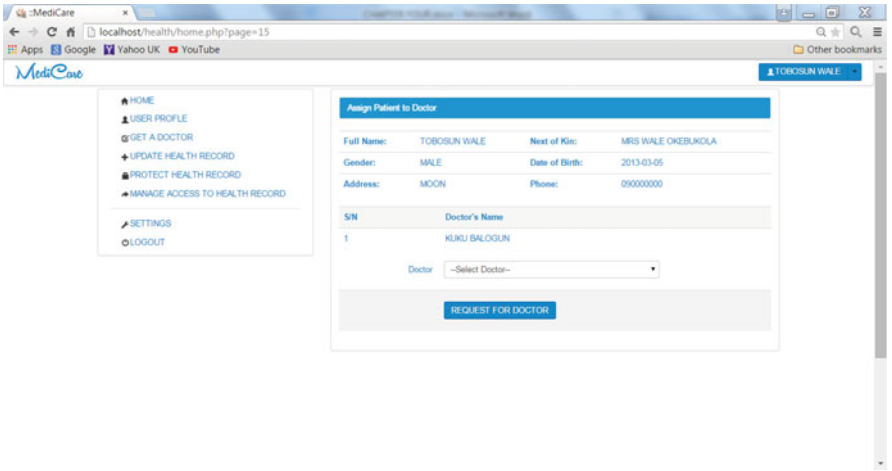

Figure 4. Patient request for doctor.

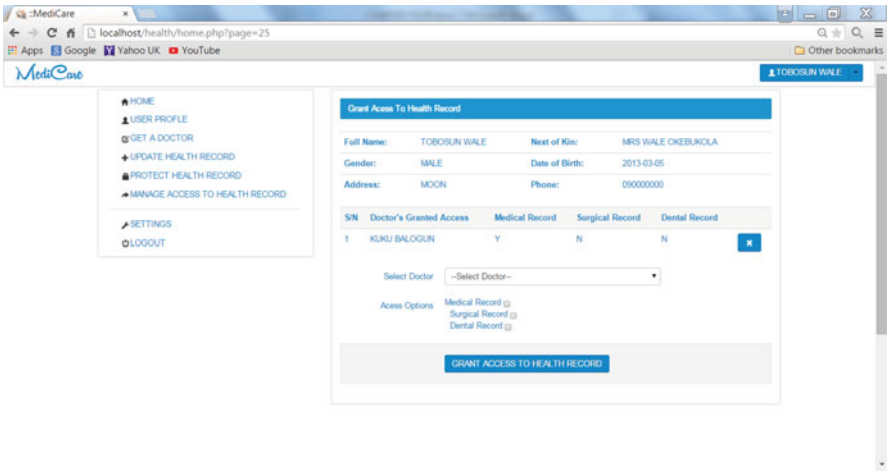

Figure 5. Patient grants access to record.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

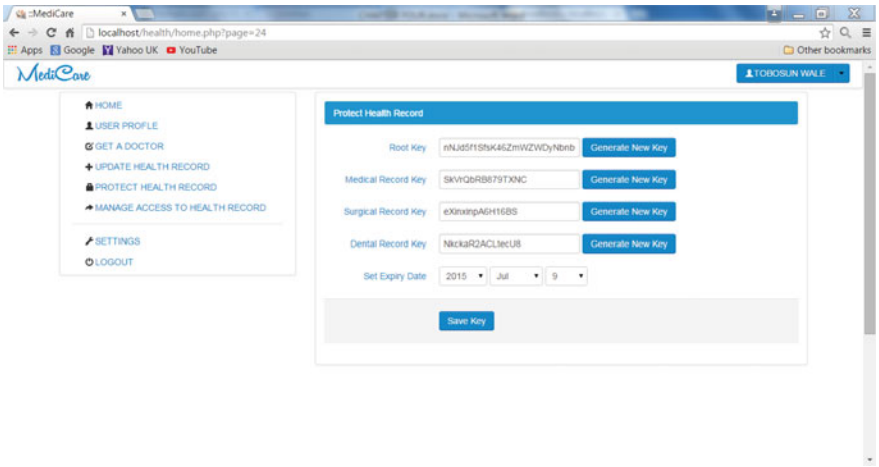

Figure 6. Key generation phase.

that a record can be viewed and that the current doctor has the decryption key to the selected record categories. While N means the record is still encrypted and restricted and the doctor will only be able to view records after the patient has provided their decryption keys. This is due to the fact that almost all of the categories of a patient's EHR is encrypted by default and nothing confidential can be accessed until a decryption key is explicitly initiated on any of the categories except the "public" bio-data.

After the patient has granted access to specific leaves or categories of their EHR as requested by the doctor, he or she generates the decryption key as seen in Figure 6 from the root key where other subkeys are derived. Using similar platforms, the decryption key of the selected record will then be available to the doctor to enable him or her access to the specified record for the current session before the key expires. Once the key expires, it is becomes useless and the doctor will not be able to access the record beyond this period until another key is derived.

Also, the patient has the privilege of updating their records as shown Figure 7 and can determine for themselves how their records are viewed, the person that

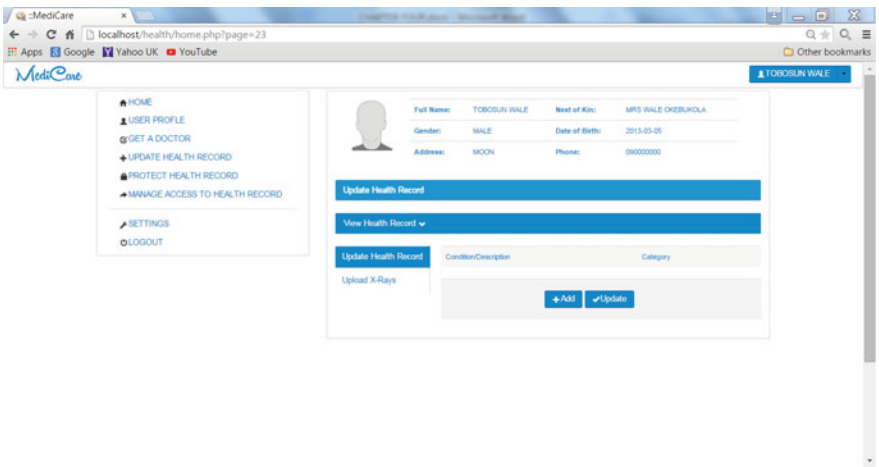

Figure 7. Update health record.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

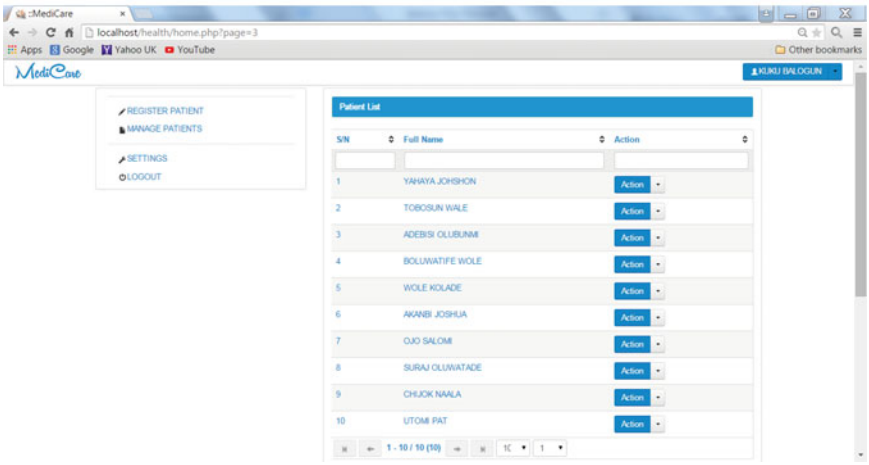

Figure 8. Doctor's page.

should be allowed to view their records, and the duration the person have to view the records.

The doctor's homepage provides a list of current and past patients as shown in Figure 8. Each record has an action button where there is an option to open the patient's record or attend to the patient. If the doctor needs to open or view the health record of a patient, a decryption key of that record must be available as depicted in Figure 9 before access can be granted to view the record. If the key is not provided, the system will trigger an access denial message as seen in Figure 10. If the decryption key is accepted by the system, the doctor can then have access to the records categories he or she has been granted as demonstrated in Figure 11. When the decryption key expires, the doctor no longer have access to the health record as shown in Figure 12 and the system will trigger a key renewal message which will pop up asking the doctor to request new keys. A successful authorization will allow

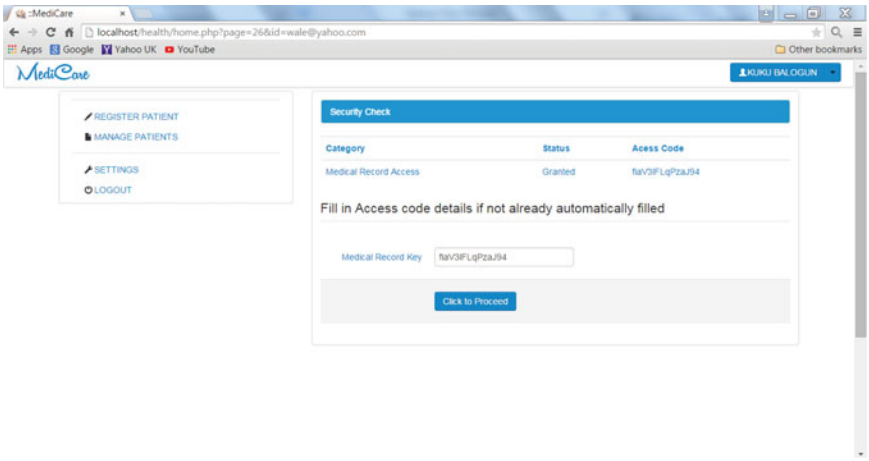

Figure 9. Security check.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

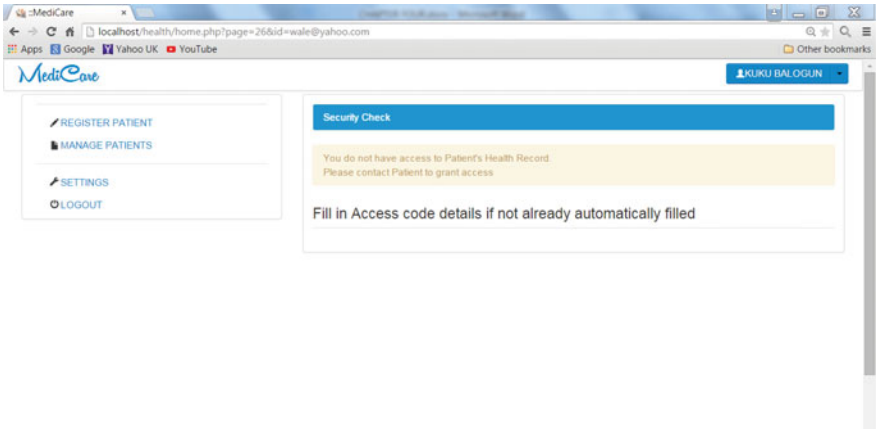

Figure 10. Security check with error message.

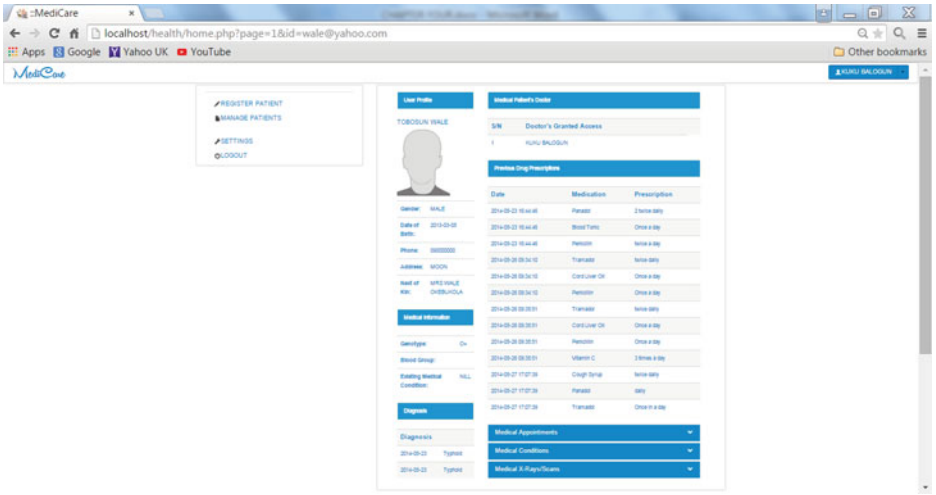

Figure 11. Patient's health record.

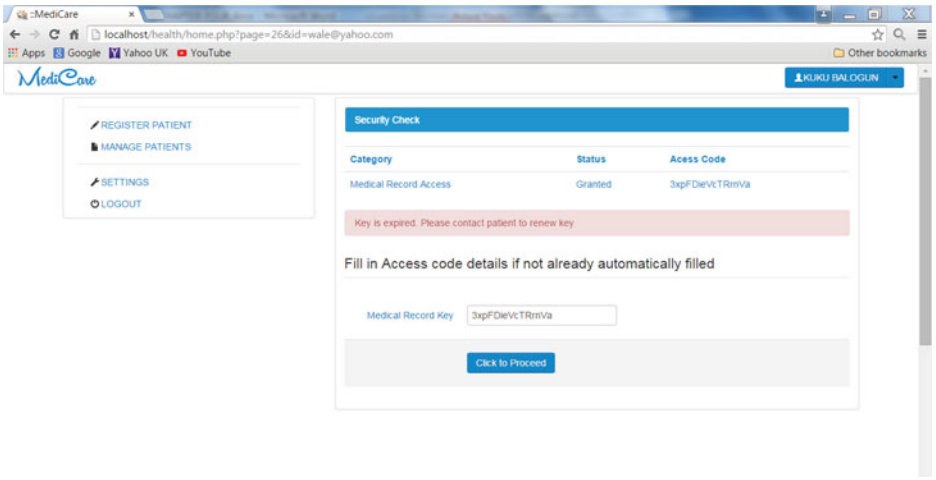

Figure 12. Key expiration.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

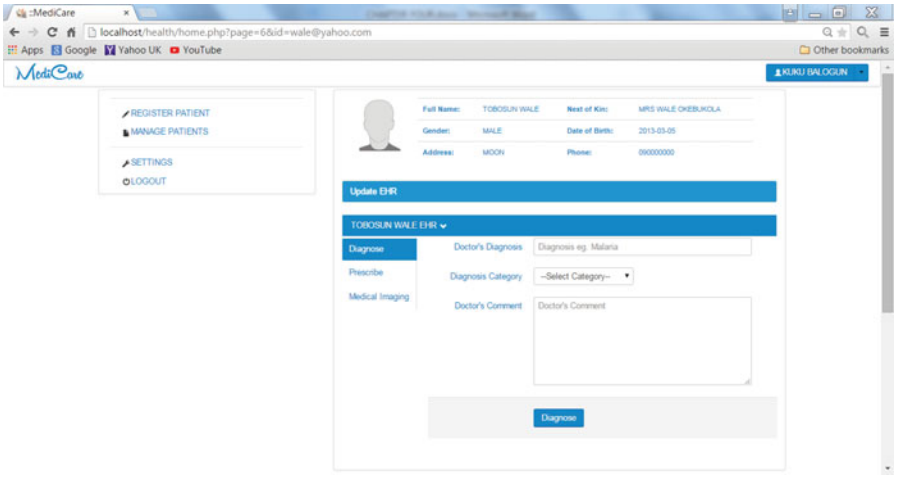

Figure 13. Update EHR.

a doctor to update patients' EHR with the diagnosis, prescription, and comments about the patient medical condition in the form provided in Figure 13.

Implication and conclusion

Paper-based methods of health record management has several challenges such as: limitations of storage difficulty, patient record being misplaced, among others. These shortcomings led to the evolution of the electronic health system in which the confidentiality of personal health records of the patient has serious issue. A session based hierarchical key encryption system has been developed in this article to protect patients' privacy and confidentiality by not exposing records or information to every requester in a hospital and third parties. The proposed system splits patients' records in encrypted hierarchical fashion that allows patients' to have almost full but not total control over their record. Patients can then generate the decryption keys for record leaves that needed to be accessed by the doctor or authorized user for a session. This approach could improve patient's trust and appreciation of EHR because, patients can determine for themselves by whom and when their EHR record categories can be accessed. When a cipher key expires, the requester will no longer be able to access until the key is renewed by the patient. Also, physician can carry out their routine without a breakdown in their work flow once they have the right permissions. During emergency, patients can be identified and given basic care that will not breach their privacy because encryption is not total. It is recommended that future work should include more categories of health records and other forms of access policies.

References

Acar T., Belenkiy, M., Ellison, C., & Nguyen, L. (2010). Key management in distributed systems. *Microsoft Research* (pp. 1–14). Retrieved from http://docplayer.net/11794546-Key-management-in-distributed-systems.html

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

Ankur, R., & Divyanjaliy, S. (2013). An introduction to pseudorandom number generator. *HCTL Open International Journal of Technology Innovations and Research*, *4*, 1–9.

Barker, E., Barker, W., Burr, W., Polk, W., & Smid, M. (2007). NIST special publication 800–57. *NIST Special Publication*, *800*(57), 1–142.

Benaloh, J., Chase, M., Horvitz, E., & Lauter, K. (2009, November). Patient controlled encryption: Ensuring privacy of electronic medical records. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security* (pp. 103–114).

Chatterjee, S., Menezes, A., & Sarkar, P. (2011, August). Another look at tightness. In *International Workshop on Selected Areas in Cryptography* (pp. 293–319). Berlin-Heidelberg, Germany: Springer.

Guo, H., Xu, C., Li, Z., Yao, Y., & Mu, Y. (2013). Efficient and dynamic key management for multiple identities in identity-based systems. *Information Sciences*, *221*, 579–590.

Hupperich, T., Löhr, H., Sadeghi, A. R., & Winandy, M. (2012, January). Flexible patient-controlled security for electronic health records. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium* (pp. 727–732).

Jeng, F. G., & Wang, C. M. (2006). An efficient key-management scheme for hierarchical access control based on elliptic curve cryptosystem. *Journal of Systems and Software*, *79*(8), 1161–1167.

Khan, S. U., Pastrone, C., Lavagno, L., & Spirito, M. A. (2012). An authentication and key establishment scheme for the IP-based wireless sensor networks. *Procedia Computer Science*, *10*, 1039–1045.

Lee, C. D., Ho, K. I. J., & Lee, W. B. (2011). A novel key management solution for reinforcing compliance with HIPAA privacy/security regulations. *IEEE Transactions on Information Technology in Biomedicine*, *15*, 550–556.

Lin, J., Huang, K., Lai, F., & Lee, H. (2009). Secure and efficient group key management with shared key derivation. *Computer Standards & Interfaces*, *31*, 192–208.

Martin, K. (2006). *Cryptographic key management*. Information Security Group, Royal Holloway, University of London, London, UK.

Mukherjee, A., Gupta, A., & Agarawal, D. (2008). Distributed key management for dynamic groups in MANETs (Elsevier). *Pervasive and Mobile Computing*, *4*, 562–578.

Omotosho, A., & Emuoyibofarhe, J. (2015). Private key management scheme using image features. *Journal of Applied Security Research*, *10*(4), 543–557.

Omotosho, A., Emuoyibofarhe, J., & Meinel, C. (2017). Ensuring patients' privacy in a cryptographic-based-electronic health records using bio-cryptography. *International Journal of Electronic Healthcare*, *9*(4), 227–254.

Peltier, T. R. (2016). Information security policies, procedures, and standards: Guidelines for effective information security management. Boca Raton, FL: CRC Press.

Pour, A.N., Kumekawa, K., Kato, T., & Itoh, S. (2007). A hierarchical group key management scheme for secure multicast increasing efficiency of key distribution in leave operation. *Elsevier Computer Networks*, *51*(17), 4727–4743.

Premarathne, U., Abuadbba, A., Alabdulatif, A., Khalil, I., Tari, Z., Zomaya, A., & Buyya, R. (2016). Hybrid cryptographic access control for cloud-based EHR systems. *IEEE Cloud Computing*, *3*(4), 58–64.

Sidorenko, A., & Schoenmakers, B. (2005, December). Concrete security of the blum-blum-shub pseudorandom generator. In *IMA International Conference on Cryptography and Coding* (pp. 355–375). Berlin-Heidelberg, Germany: Springer.

Smart, N. P. (2016). Certificates, key transport and key agreement. In *Cryptography Made Simple* (pp. 369–399). Berlin-Heidelberg, Germany: Springer International Publishing.

Sun, J., Zhu, X., Zhang, C., & Fang, Y. (2011, June). HCPP: Cryptography based secure EHR system for patient privacy and emergency healthcare. In *31st International Conference on Distributed Computing Systems* (ICDCS; pp. 373–382).

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

Teoh, B. J. A, Ngo, C. L. D, & Goh, A. (2004). Personalised cryptographic key generation based on FaceHashing. *Computers and Security*, *23*(7), 606–614.

Wang, D., & Ma, C. (2012). *On the (in)security of some smart-card-based password authentication schemes for WSN*. Retrieved from https://eprint.iacr.org/2012/581.pdf

Xiao, Y., Rayi, V. K, Sun, B., Du, X., Hu, F., & Galloway, G. (2007). A survey of key management schemes in wireless sensor networks. *Computer Communications*, *30*(11/12), 2314–2341.

Ziauddin, S., & Dailey, M. N. (2010). Robust iris verification for key management. *Pattern Recognition Letters*, *31*(9), 926–935.

Zhu, B., Bao, F., Deng, R. H., Kankanhalli, M.S., & Wang, G. (2005). Efficient and robust key management for large mobile ad hoc networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, *48*(4), 657–682.

Zych, A., Petkovic, M., & Jonker, W. (2008). Efficient key management for cryptographically enforced access control. *Computer Standards & Interfaces*, *30*, 410–417.

Downloaded by [Dr Adebayo Omotosho] at 14:16 06 October 2017

View publication stats