Security Analysis and Performance Evaluation of the Linkable Access Protocol of the Electronic Patient Records

Rima Addas
School of Computer Science
University of Manchester
Manchester, UK
addasr@cs.man.ac.uk

Ning Zhang
School of Computer Science
University of Manchester
Manchester, UK
nzhang@cs.man.ac.uk

***Abstract*—Information security and privacy in the e-health domain is an issue of growing concern. The adoption of electronic patient records, increased regulation, provider collaboration and the increased need for a faster information exchange between patients, providers and payers, all point to the need for a better information security. Therefore, the aim of this paper is to provide secure access to electronic patient records without compromising performance. To achieve this, we have designed a secure protocol called the Linkable Access protocol. In this paper, (1) we formally verify and analyse the Linkble Access protocol against security properties (e.g., confidentiality) using the Casper/FDR2 verification tool. In addition, (2) we build a prototype using the Java technology to demonstrate the performance of the Linkable Access protocol. By doing this, we prove that the Linkable Access protocol maintains a good balance between security and performance.**

Keywords-e-Health; electronic patient records; privacy; security; performance.

I. INTRODUCTION

Privacy is considered as a key governing principle of the patient-physician relationship. Patients are required to share information with their physicians to enable correct diagnosis and treatment, and to elude adverse drug interactions. Nevertheless, patients may refuse to disclose necessary information in cases of sensitive medical problems such as psychiatric behaviour and Human Immunodeficiency Virus (HIV), as their disclosure may lead to social stigma and discrimination [1]. Over time, Electronic Patient Records (EPRs) accumulate important personal information, including identification, history of medical diagnosis, treatments, medication history, dietary habits, sexual preference, genetic information, psychological profiles, employment history, income and physicians subjective assessments of personality and mental state [2].

EPRs offer a wide range of purposes apart from diagnosis and treatment provision. For example, information could be used to improve efficiency within the healthcare system, drive public policy development and administration, and in the conduct of medical research [3]. EPRs can also be shared with payer organisations (e.g., private insurance) to justify payment of services rendered. Health Service Providers (HSPs) also make use of records to manage their operations and improve service quality.

While the above mentioned technology can help improve overall quality of health care delivery, the benefits from this technology must be balanced with the privacy and security concerns of the patient.

In real-life situations, there are various scenarios, where authorized users have legitimate reasons to access patients' EPRs (which could be stored in a single or in multiple locations). Based on the principle of least privilege, users should only be granted with access rights that are just sufficient for them to carry out the tasks assigned to them.

The minimum level of access privilege is to only allow users to access de-identified records. De-identification means that patients' identifiable information is removed from the records [4]. There are three de-identification methods, anonymization, depersonalization and pseudonymization. Anonymization is the process of hiding (or removing) a patient's identification data and only make other information (i.e., de-identified information) available for access [5]. Depersonalization is a process similar to anonymization, but it comprises the removal of as much identification information as necessary to protect patient identity [6]. Pseudonymization, on the other hand, is the process of adding an identifier (called a pseudonym) into a patient's de-identified record [7].

Yet, in practice, there are times when, for legitimate reasons, multiple de-identified records of the same patient may need to be linked (e.g., when we need to study the history of a patient's medical condition) or an anonymised record needs to be re-identified at a later date. In such cases, a patient's pseudonym should be mapped or reversed to the patient's identity and two or more pseudonyms of the same patient should be linkable and these should be done in a controlled manner.

There are two types of pseudonyms, namely, irreversible and reversible pseudonyms [8]. Irreversible pseudonyms are pseudonyms that cannot be reversed back to the patient's real identity. Reversible pseudonyms are pseudonyms that can be reversed back to the original identity (i.e., a patient can be re-identified from his/her reversible pseudonyms).

ISBN: 978-1-61208-327-8

Most pseudonym generation methods used in supporting privacy preserving EPR access [9][10][11] , focus on preserving patient anonymity. They use irreversible pseudonyms to index de-identified records. This type of pseudonyms only supports anonymous data access. Though the pseudonym generation methods in [8][12], have considered the linkability requirement, they do not support a secondary use of patient information. That is, they do not allow linking of multiple pseudonyms of the single patient without revealing the patient's identity. A notable method that has addressed this limitation is LIPA [13]. Yet, LIPA supports this linkablity requirement, but assuming that patient records managed by different HSPs are stored in a single repository. The solution does not support distributed data access. To the authors' best knowledge, the works that are most related to our work are Deng's method [14] and the PIPE method [15]. Both methods aim to securely integrate primary and secondary usage of distributed medical data without compromising the patient's identity privacy. We described an alternative method [16] with the aim of reducing access delays. In other words, our method proved to be more efficient than Deng's and PIPE methods.

In detail, to facilitate the minimum access right management, we have proposed a new method called 3LI2Pv2 method to support controlled access to EPRs with three levels of identity privacy reservations [16]. In this method, we have identified three distinctive user groups, each with a defined level of access. The first group of users (called L3 users) are only given rights to access anonymised data. They are not allowed to identify the patient (i.e., the identities of the owners of the data) nor link multiple EPR objects of the same patient. The second group of users (L2 users) are allowed to access and link multiple objects of the same patient, but are not allowed to link the objects to their owner's (i.e., the patient's) identity. In other words, users in this group are allowed to access the multiple objects of the single patient without being able to identify the patient. Finally, the third group of users (L1 users) are allowed to access patients' records as well as identify the owners of the records. In other words, we have three levels of patient identity privacy protection.
* *Level-1 (L1)- Linkable access:* At this level, multiple data objects of the same patient can be linked, and this set of objects can be linked to the patient's identity. L1 access should be limited to L1 users, i.e., users with linkable access privilege.
* *Level-2 (L2)- Linkable anonymous access:* At this level, multiple data objects of the same patient can be linked, but this set of objects cannot be linked to the patient's identity. L2 access should be limited to L1/L2 users, i.e., users with linkable anonymous access privilege.
* *Level-3 (L3)- Anonymous access:* At this level, multiple data objects of the same patient cannot be linked, nor the patient's identity be exposed. L3 access should be limited to L1/L2/L3 users, i.e., users with anonymous access privilege.

The 3LI2Pv2 method made use of cryptographic techniques to achieve its goals. We have informally analysed the 3LI2Pv2 method against some security requirements, and the result was positive. For future work, we suggested to include the design of the access protocol for the three levels. Therefore, in this paper, we introduce a secure and robust protocol for the Level-1 (Linkable access), called the Linkable Access (LA) protocol. This type of access protocol provides the highest level of access in terms of revealing sensitive patient information, and only user holding the right type of credentials can perform this type of access.

Generally, security protocols have been designed and verified using informal techniques. As a result, it is now well known that many security protocols, which were previously proposed have found to be vulnerable afterwards. For example, the Needham-Schroeder public key protocol [17] succeeded in the informal analysis, but failed in formal verification [18]. To address this problem, formal methods have been widely used to specify security protocols and verify security properties, such as confidentiality, authentication and non-repudiation, to guarantee correctness [19].

In this paper, the Casper/FDR2 verification tool [20][21], is used to verify the LA protocol. Casper/FDR2 has proven to be successful for modelling and verifying several security protocols; it has been used to verify authentication, secrecy, and other security properties [22][23]. Accordingly, we consider it also appropriate for the verification of the LA protocol. The Casper/FDR2 model checker is used to verify the security properties of the protocols. If the protocols do not satisfy the specified security properties, then the FDR2 checker shows a counterexample which represents the reason against vulnerability.

After completing the formal verification of the protocol using Casper/FDR2, we implement the protocol using the Java technology [24] to test it against performance. Java is selected because it supports a set of standard security primitives. Examples of these primitives include the hash functions SHA-256 [25] and MD-5 [26], the symmetric cryptographic algorithms AES [27] and 3DES [28] and the asymmetric cryptographic algorithms RSA [29] and DSA [30].

This paper is organized as follows; In Section 2, we introduce possible security threats. In Section 3, we describe, model and verify the LA protocol. Also, we set the goals that the LA protocol should meet. After that, we show the result of the verification. In Section 4, we present the implementation and performance analysis of the LA protocol. In Section 5, we conclude the paper and discuss future work.

II. Possible Security Threats

Access to EPRs is subject to different kinds of security threats. We will not consider here threats of environmental origin (e.g., fire, etc.) or accidental ones (e.g., user errors, software malfunction, etc.). The deliberate threats that we will consider are categorized into three groups.

A. Confidentiality threats.
B. Integrity threats.
C. Authentication threats (including non-repudiation).

A. Confidentiality Threats

In this type of threat, an attacker may gain access to private information. The attack consists in eavesdropping the communication links, without interfering with the transmissions, or in inspecting data stored in the system. Man in the middle attack, replay attack, credential forgery/theft and impersonation are examples of this type of threat.

B. Integrity Threats

Here, an attacker may modify the information exchanged within an e-health service. The attack consists in interfering with the transmissions, so that the recipient receives data, which are different from those sent by the originator. Data tampering is an example of this type of threat.

C. Authentication Threats

In this kind of threat, an attacker may counterfeit false data and deceive the recipient into believing that they come from a different originator (which the recipient takes as the authentic originator). The attack consists in forging the part of the data where the originator is identified (usually in the identity credentials). Spoofing is an example of this type of attack. Repudiation is also a variant of this type of attacks that consists in denying authorship or the contents of data previously sent.

III. Formal Verification of The LA Protocol

In this section, firstly, we describe and model the LA security protocol with Casper/FDR2 verification tool. Secondly, we identify essential security requirements that the LA protocol should meet. Finally, we discuss the verification result of the protocol and analyse its security requirements.

A. The LA Protocol Description

The purpose of the LA protocol is to link multiple objects (under a single or multiple HSPs management) of the same patient, and to link these objects to the patient's real identity (e.g., NHS number). This type of access should be limited to users with the highest access privileges (i.e., L1 users such as general practices, GPs). In real-life scenarios, this protocol can be applied to a GP who wishes to proceed with a patient's treatment and needs to have access to the patient's real identity from his de-identified records. The GP will need to get this patient's data from the attribute authority (aa). This authority can retrieve the patient's real identity on behalf of the HSP. Assuming in A3 below, this authority is trusted by HSPs and clients (e.g., GPs). Assuming in A4, all the patient's records have been de-identified. In order to get the data, the GP needs to prove to aa that he has been granted the right credentials to perform such type of access. In other words, the GP needs to, firstly, show his identity credential to ensure that he is the person he claims to be. Secondly, he needs to show his access credential, which confirms that he is allowed to perform this type of access and learn the patient's real identity.

Until now, no research has been carried out to analyse the vulnerability of the LA protocol using a model checking tool. Table I shows the basic notation of the LA protocol. Fig 1 shows the message sequences of the LA protocol.

In the LA protocol, the communication channel is based on the Secure Socket Layer (SSL) protocol [31] to provide security for data transmission. SSL protocol uses a combination of public key and symmetric key ciphers to establish a secure communication channel between a server and a client. For protocol analysis using Casper/FDR2, we assume the following.

A1. The underlying cryptographic algorithms used in SSL's public key and symmetric key ciphers are secure.

A2. All parties unconditionally trust the certification authority and public keys signed by it. The certification authority certifies the public key for clients.

A3. All parties unconditionally trust the attribute authority who issues the attribute certificates for clients.

A4. Patients' records have already been de-identified. That is their identity or NHS number has been replaced with a pseudonym.

TABLE I
THE LA PROTOCOL NOTATION AND DESCRIPTION

Notation	Description
a	An identifier of an initiator/client
ca	An identifier of a certification authority
aa	An identifier of a attribute authority
nx	A random nonce of x
PKx	A public key of x
SKx	A secret Key of x
ts	A timestamp (an expiration time)
h	A hash function
msg	A message of data request
certa	A PK-certificate of client a generated by ca
attr-certa	An attribute certificate of client a generated by aa
veri1	An integrity verification of certa
ps3l1	An L3 pseudonym Type-I
sigaa	A signature of aa
integ1, integ2	Used in attr-certa integrity verification

In the LA protocol, *ca* is the certification authority who issues public-key (PK) certificates, and *aa* is the attribute authority who issues attribute certificates to legitimate users. *a* is the client or the initiator of the request.

The PK-certificate includes two parts, {a, Pk(a), 11, ts} and {h(a, Pk(a), 11, ts}{SK(ca)}. The first part, contains information about the client, such as, identity *a*, public key of *a* PK(a), group membership *l1* and timestamp *ts*. The second part, is the signature of the *ca*. Issuer *ca* signs subject *a*, public key of *a*, *PK(a)*, a group membership *l1* and timestamp *ts* using its own private key *SK(ca)*, which is only known to the *ca*. Since the certificate is encrypted with the private key of *ca*, any other user cannot spoof it. This provides confidence of the certificate's information to a participant. The certificate can only be decrypted by the public key of *ca*, which is known to legitimate users. To sum up, The design of PK-certificate ensures that no one can forge or modify a valid PK-certificate. It is important to mention that in this protocol description scenario, we have also included issuing the PK-certificate and attribute certificate to the client. In real-life scenarios, certificates are issued once (unless expired and need renewal) and usually at an earlier stage before submitting a request to access patient data.

The following describes the message sequence of the LA protocol depicted in Fig 1.

Message 1. ca → a : certa
Message 2. aa → a : attr-certa
Message 3. a → aa : {na,msg}{PK(aa)}
Message 4. a → aa : certa
[aa computes dectyptable (certa, PK(ca)) & veri2==h(veri1)]
Message 5. aa → a : enc1
[a computes dectyptable (enc1, SK(a))]
Message 6. a → aa : enc2
[aa computes dectyptable (enc2, SK(aa)) & ga==l1]
Message 7. a → aa : attr-certa
[aa computes dectyptable (sigaa, PK(aa)) & decrypt(ps3l1, SK(aa))==ps1 & ts==now|| ts+1==now & integ1==h(integ2) & decrypt(ps1,SK(aa))==pid]
Message 8. aa → a : $\{\underbrace{a,pid,na,ts}_{int2}\}\{PK(a)\}, \{\underbrace{h(a,pid,na,ts}_{int1}\}\}\{SK(aa)\}$
[a computes decryptable (int1, PK(aa)) & h(int2)==int1 & ts==now || ts+1==now]

Fig. 1. The LA protocol description

Message 1: Certificate authority *ca* issues and sends the PK-certificate, *certa*, to client *a* in order to authenticate client *a* and distribute *PK(a)* safely.
Message 2: Attribute Authority *aa* issues and sends the attribute certificate, *attr-certa*, to client *a*. This certificate includes the issuer's name (aa), the client's name (a), an L3 pseudonym (ps3l1), a timestamp (ts) and the issuer's signature on the certificate. The L3 pseudonym (ps3l1), contains another pseudonym, a lower-level one called, ps1, which can be used to recover the patient's real identity.
Message 3: Client *a* sends his/her nonce (na) along with a message of the request encrypted with *aa*'s pubic key.
Message 4: Client *a* sends his PK-certificate (certa) to *aa*. This certificate contains *veri1* and *veri2*. *veri1* contains the plain content of the certificate. *veri2* contains the deciphered *ca*'s signature on the certificate. Using *veri1* and *veri2* allows checking the integrity of the certificate to ensure that the certificate has not been modified during transmission. So first, verifier *aa* validates the *ca's* signature on the certificate and then, it verifies the certificate's integrity using *veri1* and *veri2*.
Message 5: Verifier *aa* sends *enc1* to client *a* which contains the verifier's identity (aa), user's nonce (na) and the verifier's nonce (naa) encrypted with $PK(a)$. Client *a* checks if *enc1* is decryptable by $SK(a)$ and contains the right nonce na. This step is essential to allow client *a* to authenticate verfier *aa*.
Message 6: Now client *a* sends *encr2* to recipient *aa*. Variable *encr2* contains the items a and naa encrypted with $PK(aa)$. Recipient *aa* checks if *enc2* is decryptable by $SK(aa)$ and contains the right nonce naa. This step is essential to allow *aa* to authenticate *a*. Also in this step, *aa* checks a's group membership to ensure that the client belongs to the right group and legitimate for this type of access.
Message 7: After successful authentication, *a* sends to *aa* his *attr-cert* to check his authorisation. Verifier *aa* checks the correctness of the certificate. It completes this by verifying the signature on the certificate and checks a's access credentials. That is to ensure that the certificate contains the right type of L3 pseudonym (ps3l1). After that, it verifies the integrity of the lower-level pseudonym (ps1) to ensure it has not been altered during transmission.
Message 8: After successful authorisation, *aa* forwards *int1* and *int2* to *a*. Variable *int2* contains the requested patient's data (pid), a timestamp (ts), user's nonce (na) and the user's identity (a) all encrypted with the user' pubic key. Variable *int1* contains same items as in *int2* but hashed. Finally, user *a* performs the final checks. (1) Checking the aa' signature on *int1* and verifying the integrity of the data using *int1* and *int2*. (2) Checking the timestamp to ensure data freshness.

B. Modelling the LA protocol Using Casper/FDR2

Based on the LA protocol's notation in Table I, we model the LA protocol in Casper's script, as shown below.

```
#Protocol description
--ca issues and sends PK-certificate to client a
0. ca -> a : {{a,PK(a),{l1}%ga,ts}%veri1,{{h(a,PK(a),
{l1}%ga,ts)} %veri2}{SK(ca)%skca}%certa}{PK(a)}
--a wants to contact aa
1. -> a : aa
--a sends his original request message with a nonce
2a. a -> aa : {msg, na}{PK(aa)}
--a sends his PK-certificate to be verified by aa
2b. a -> aa :{veri1%{a,PK(a),ga%{l1},ts},{certa
%{veri2% {h(a,PK(a),ga%{l1},ts)}}} {SK(ca)}}{PK(aa)}
[decryptable(certa, PK(ca)) and veri2== h(veri1) and
ts==now or ts+1==now]
--Mutual authentication and check user membership
3. aa -> a : {aa, na, naa}{PK(a)} %enc1
[decryptable (enc1, SK(a))]
4. a -> aa :{a, naa}{PK(aa)} %enc2
[decryptable (enc2, SK(aa)) and ga==l1]
--aa issues and sends attribute certificate to a
5a. aa -> a :{aa,a,{{ps1,l1, aa, nonce}%integrity2,
{h(ps1, l1, aa, nonce)}% integrity1}%ps3l1,ts}{PK(a)}
5b. aa -> a : {h(aa,a,ps3l1,ts)} {SK(aa)} %sigaa
[ts==now or ts+1==now]
--a sends to aa his attribute certificate for
authorisation verification
6a. a -> aa :{aa,a, ps3l1 %{integrity2%{ps1,
l1,aa,nonce},integrity1%{h(ps1,l1,aa,nonce)}}, ts}
6b. a -> aa :sigaa%{h(aa,a,ps3l1,ts)}{skaa%SK(aa)}
[decryptable(sigaa,PK(aa)) and integrity1==
h(integrity2) and decrypt(ps3l1, SK(aa))== (ps1,
l1, aa, nonce) and decrypt(ps1, SK(aa))==pid and
ts==now or ts+1==now]
--aa sends the response to a
7. aa -> a : {{a, na, pid, ts} %int2,
{h(a,na,pid,ts)%int1}{SK(aa)}%sigaa2}{PK(a)}
[decryptable(sigaa2,PK(aa)) and int1== h(int2) and
ts==now or ts+1==now]
```

C. LA Protocol Goals

In this section, we identify the LA protocol security goals or properties.

(P1) Data Confidentiality: Confidentiality is a vital requirement that provides secrecy and privacy in e-health applications. It offers protection against attacks such as forgery and spoofing. To support data confidentiality, the communication channel between entities should be secured typically via encryption. An unauthorised party should not be able to learn anything about any communication between two entities by observing or even tampering the communication lines. That is, one cannot infer the contents of the message, sender and receiver, the message length, the time they were sent, and not even the fact that a message was sent in the first place.

(P2) Integrity Protection: A strong integrity protection mechanism should be deployed to protect against data tampering. The LA protocol should detect any unauthorised alteration to data being transmitted between the authorised entities.

(P3) Ensuring Accountability: The protocol should obtain an undeniable response from entities participating in the protocol. That is, to ensure that the originator of a communication cannot deny it later.

(P4) Mutual Authentication: Or two-way authentication, refers to both entities of the protocol should authenticate each other to permit the exchange of information there-between.

(P5) Certificate Manipulation Protection: It should be guaranteed that the certificates (i.e., PK-certificates) used in the protocol are valid and have not been corrupted or modified during transmission.

(P6) Credential Forgery Protection: It should be assured that users' credentials are not stolen or forged. This is because it can lead to the elevation of privileges attack. This attack occurs when a user with limited privileges assumes the identity of a user with higher privileges to gain access to patient confidential data.

(P7) Data Freshness: There should be a proof that nonces, generated during protocols, are fresh and the integrity of the session key is preserved. Both entities should also have undeniable proof that the other party is in possession of a valid session key. Any previous compromised key should be easily detected, and the protocol run should terminate.

(P8) Linkability: A user with L1 access credentials, i.e, highest access privileges, should be able to link de-identified or anonymous objects to the patient's real identity.

D. Verification Result and Security Analysis of The LA Protocol

The verification result using the Casper/FDR2 model checking tool confirms that the LA protocol has fulfilled all the properties identified in Section III-C. The result of the verification is shown in Fig 2.

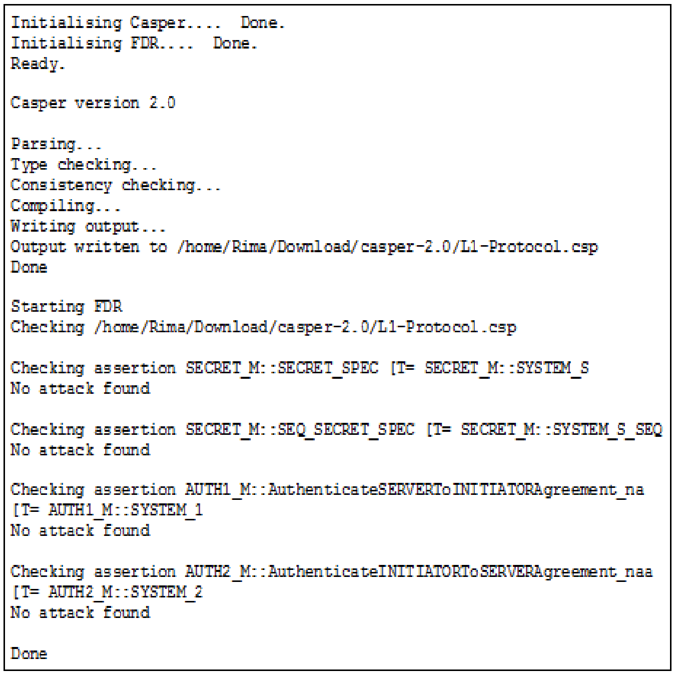
(P1) Data Confidentiality: was achieved by deploying cryptographic techniques (symmetric cryptoystem, asymmetric cryptoystem, and hash functions).

(P2) Integrity Protection: was met by incorporating digital signatures and hash functions, which can detect any data alteration during transmission.

(P3) Ensuring Accountability: was fulfilled by using digital signatures of both entities, the sender and receiver.

(P4) Mutual Authentication: was accomplished by integrating the challenge response protocol.

(P5) Certificate Manipulation Protection: this property has been abided by including a timestamp in the certificate, which can detect any sniffing and manipulation by the intruder.



```
Initialising Casper....  Done.
Initialising FDR....  Done.
Ready.

Casper version 2.0

Parsing...
Type checking...
Consistency checking...
Compiling...
Writing output...
Output written to /home/Rima/Download/casper-2.0/L1-Protocol.csp
Done

Starting FDR
Checking /home/Rima/Download/casper-2.0/L1-Protocol.csp

Checking assertion SECRET_M::SECRET_SPEC [T= SECRET_M::SYSTEM_S
No attack found

Checking assertion SECRET_M::SEQ_SECRET_SPEC [T= SECRET_M::SYSTEM_S_SEQ
No attack found

Checking assertion AUTH1_M::AuthenticateSERVERToINITIATORAgreement_na
[T= AUTH1_M::SYSTEM_1
No attack found

Checking assertion AUTH2_M::AuthenticateINITIATORToSERVERAgreement_naa
[T= AUTH2_M::SYSTEM_2
No attack found

Done
```

Fig. 2. Verification result of the LA protocol using Casper/FDR2

(P6) Credential Forgery Protection: was met by including the legitimate credential holder identity in both types of certificates, the PK-certificate and the attribute certificate. So by checking that both certificates contain the same credential holder identity, we can ensure that both credentials have not been forged.

(P7) Data Freshness: was achieved by including a freshly random nonce with the transmitted data.

(P8) Linkability: was fulfilled by integrating the L3 pseudonym-Type1 in the L1 user's access credential. This pseudonym allows linkable access to patient data as it contains a lower-level pseudonym that can recover the patient's real identity, using the right secret key.

IV. Implementation and Performance Evaluation

In this section, we describe the implementation and performance evaluation of the LA security protocol. To achieve this, we have built a prototype using the Java 2 platform (standard edition), as it is suitable for e-health applications. It offers implementations for several cryptographic primitives and key management services needed for our solution.

Performance is measured by two metrics, minimising access delay and minimising server computation time. An access delay is defined as the time elapsed from submitting an access request to the time when the response to the access request is received. A server computation time is the time needed for the server to complete the necessary operations, verifications and checks from receiving the request to the time when the response to the request is sent. Both metrics should be kept as low as possible.

To know the access delay and server computational time incurred by the LA protocol, we have measured the time taken to execute (run) the protocol based upon the prototype under two scenarios.

- In the first scenario (L3 Scenario), we run the protocol without applying an extra security layer to the protocol. This scenario is based on the principle of least privilege. This scenario is called the Level-3 access or the anonymous access scenario, which has been described in the introduction section.
- In the second scenario (L1 Scenario), we run the protocol with applying our additional security mechanism. This scenario is called the Level-1 access or the linkable access.

The measurements are taken for 10 execution rounds for each scenario, and the averages are calculated. The results are shown in Fig 3 and Fig 4.

A. Implementation Platform

To prototype the LA protocol, the following hardware and software have been used. We have used a desktop computer running Windows 8 with a 2.30 GHz Intel Core i3 and 8GB of RAM. The timing results from the LA protocol execution presented here are based on this computer specification. The software used to implement the LA protocol is JAVA 2 Platform, Standard Edition (J2SE).

B. Performance Evaluation Parameters and Target

The performance evaluation parameters we rely on are as follows.

- The patient's records are distributed in different databases which are managed by different HSP (e.g., hospitals). That is we run the simulation on a distributed manner and test its performance.
- Running the simulation where the database size of each HSP increases, patient wise and record wise. We first, run the simulation with the parameter 10 objects by 1000 patients and then we increase the object's size by ten and the patients' number by 1000.
- A single patient data request.

As we gave an real-life example in Section III-A, we show in the following section two things. (1) The time needed for the GP to obtain the patient's data. We call this access delay. (2) The time needed for each hospital to verify and complete the GP's request. We call this server computation time.

The target of the performance evaluation is to show that the LA protocol (Level-1 Scenario) offers a higher security than the protocol under the least access privilege scenario (Level-3 Scenario) and with a linear increase in performance. In other words, the LA protocol aims to balance between security and performance without adding a massive amount of overhead into the solution.

C. Performance Evaluation Result and Analysis

It can be seen from Fig 3 that the time (Access delay) taken to execute the LA protocol (L1 Scenario) is 1200 milliseconds in its peak, which is approximately 90% more than the time taken in the normal case or L3 Scenario, which is 101 milliseconds shown in Fig 4. The server computation time in L1 Scenario is 1150 milliseconds, which is approximately 91% more than that in L3 Scenario, which is 100 milliseconds.

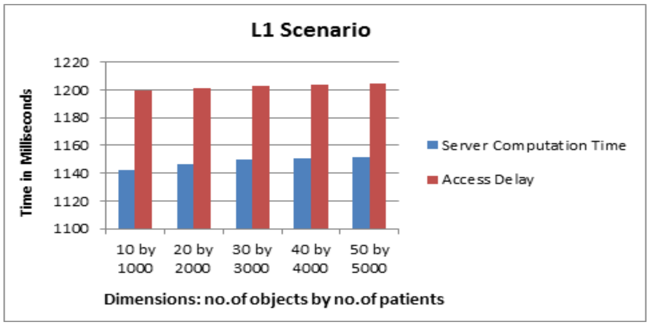


Fig. 3. Performance evaluation result of the LA protocol-L1 Scenario

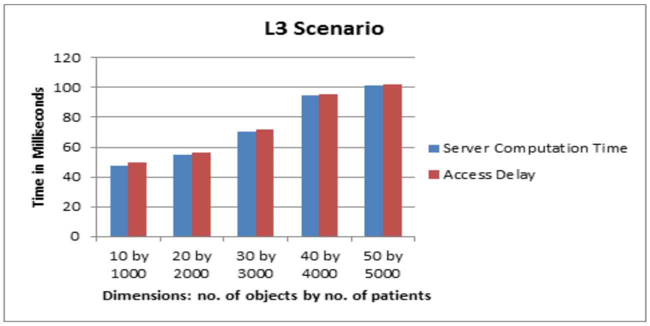


Fig. 4. Performance evaluation result of the L3 Scenario

The extra cost in the L1 Scenario is caused by the following reasons.

- The L1 Scenario contains three additional security layers, which were added on top of the L3 Scenario.
- The extra communications between the client and the verifier.
- The extra computations in signature verifications by both the client and the verifier.
- The extra computation in the attribute certificate verification by the verifier. In the L3 scenario, only PK-certificate verification is necessarily for completing the access request. No attribute certificate verification is involved in the L3 scenario.
- The extra computation in checking the timestamp in the attribute certificate.
- The extra computation in validating the pseudonym (PS3l1) included in the attribute certificate.
- The extra integrity check of the lower-level pseudonym (PS1) included in PS3l1.
- The extra computation in the decryption operation to retrieve or recover the patient's identity.
- The extra computation in signing the requested data or the response before sending it to the client.
- Finally, the extra cost in L1 Scenario between the server computation time and the access delay is due to the

ISBN: 978-1-61208-327-8

distributed patient's objects, which normally increases the waiting time. While in L3 Scenario a patient's objects are not distributed and are managed by a single HSP.

V. Conclusion and Future Work

In this paper, we focused on two major aspects. Firstly, the formal verification and security analysis of the LA protocol using Casper/FDR2 tool verification. Secondly, the formal performance evaluation of the LA protocol by building a prototype using the Java technology.

The result from the verification using Casper/FDR2 tool showed that the LA protocol has fulfilled important security requirements. It supports linkable access to patient data by integrating significant cryptographic techniques. It ensures confidentiality of patient sensitive data. It provides data freshness by relying on timestamps and nonces. It is protected from certificate manipulation and credential forgery. It ensures accountability by deploying digital signatures. Mutual authentication is also provided to obtain unforgeable proof of other participant's authenticity before it engages in the protocol with that participant.

In addition to fulfilling important security requirements, the result from the LA protocol implementation showed that the LA protocol had successfully balanced between security and performance. That is the increase in performance was linear with the increase of security. So our analysis proved that the LA protocol is secure and efficient. It allows a client and a server to exchange some sensitive patient data in a secure manner and within a reasonable amount of time. Our future work is concerned with extending our analysis of the LA protocol to other security protocols and specifically, e-health protocols, taking into account security and performance as major criteria.

Acknowledgement

This work is financially sponsored by the Ministry of Higher Education in Saudi Arabia.

References

[1] P. S. Appelbaum, "Privacy in psychiatric treatment: threats and responses," *FOCUS: The Journal of Lifelong Learning in Psychiatry*, vol. 1, no. 4, pp. 396–406, 2003.

[2] A. Appari, M. E. Johnson, and D. L. Anthony, "Hipaa compliance: An institutional theory perspective." in *AMCIS*, 2009, p. 252.

[3] J. G. Hodge, "Health information privacy and public health," *The Journal of Law, Medicine & Ethics*, vol. 31, no. 4, pp. 663–671, 2003.

[4] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology," Tech. Rep., February 2008.

[5] R. Clayton, "Anonymity and traceability in cyberspace," University of Cambridge, Computer Laboratory, Darwin College, Tech. Rep., 2005.

[6] J. Rogers, C. Puleston, and A. Rector, "The clef chronicle: Patient histories derived from electronic health records," in *ICDEW '06: Proceedings of the 22nd International Conference on Data Engineering Workshops*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 109+.

[7] B. Alhaqbani and C. Fidge, "Privacy-preserving electronic health record linkage using pseudonym identifiers," July 2008, pp. 108–117.

[8] B. S. Elger, J. Iavindrasana, L. Lo Iacono, H. Müller, N. Roduit, P. Summers, and J. Wright, "Strategies for health data exchange for secondary, cross-institutional clinical research." *Computer methods and programs in biomedicine*, vol. 99, no. 3, pp. 230–251, September 2010.

[9] K. Pommerening and M. Reng, "Secondary use of the ehr via pseudonymisation." *Studies in health technology and informatics*, vol. 103, pp. 441–446, 2004.

[10] D. Slamanig and C. Stingl, "Privacy aspects of ehealth," in *ARES '08: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security*. Washington, DC, USA: IEEE Computer Society, March 2008, pp. 1226–1233.

[11] L. Iacono, "Tmulti-centric universal pseudonymisation for secondary use of the ehr," *In: Proc. of HealthGrid:*, pp. 239–247, 2007.

[12] P. Schartner and M. Schaffer, "Unique user-generated digital pseudonyms," in *Computer Network Security*, ser. Lecture Notes in Computer Science, V. Gorodetsky, I. Kotenko, and V. Skormin, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2005, vol. 3685, ch. 15, pp. 194–205.

[13] N. Zhang, A. Rector, I. Buchan, Q. Shi, D. Kalra, J. Rogers, C. Goble, S. Walker, D. Ingram, and P. Singleton, "A linkable identity privacy algorithm for healthgrid," in *From Grid to Healthgrid: Proceedings of Healthgrid 2005*, 2005, pp. 234–245.

[14] M. Deng, D. DeCock, and B. Preneel, "Towards a cross-context identity management framework in e-health," *Online Information Review*, vol. 33, no. 3, pp. 422–442, 2009.

[15] T. Neubauer and J. Heurix, "A methodology for the pseudonymization of medical data," *International Journal of Medical Informatics*, vol. 80, no. 3, pp. 190–204, Mar. 2011.

[16] R. Addas and N. Zhang, "An enhanced approach to supporting controlled access to eprs with three levels of identity privacy preservations," in *Information Quality in e-Health*, ser. Lecture Notes in Computer Science, A. Holzinger and K.-M. Simonic, Eds., vol. 7058. Springer Berlin / Heidelberg, 2011, pp. 547–561.

[17] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978.

[18] G. Lowe, "An attack on the needham-schroeder public-key authentication protocol," *Information Processing Letters*, vol. 56, no. 3, pp. 131 – 133, 1995.

[19] I.-G. Kim and J.-Y. Choi, "Formal verification of pap and eap-md5 protocols in wireless networks: Fdr model checking," in *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on*, vol. 2, 2004, pp. 264–269 Vol.2.

[20] G. Lowe, "Casper: a compiler for the analysis of security protocols," pp. 18–30, Jun. 1997.

[21] F. S. E. LTD., "Failure-divergences refinement fdr2 manual," 2010.

[22] I.-G. Kim, H.-S. Kim, J.-Y. Lee, and J.-Y. Choi, "Analysis and modification of ask mobile security protocol," in *Mobile Commerce and Services, 2005. WMCS '05. The Second IEEE International Workshop on*, 2005, pp. 79–83.

[23] H.-S. Kim, J.-H. Oh, J.-Y. Choi, and J.-W. Kim, "The vulnerabilities analysis and design of the security protocol for rfid system," in *Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on*, 2006, pp. 152–152.

[24] P. Chan, R. Lee, and D. Kramer, *The Java Class Libraries, Volume 1: Supplement for the Java 2 Platform, Standard Edition, V 1.2*. Addison-Wesley Professional, 1999, vol. 1.

[25] H. Gilbert and H. Handschuh, "Security Analysis of SHA-256 and Sisters Selected Areas in Cryptography," *Selected Areas in Cryptography*, vol. 3006, pp. 175–193, 2004.

[26] B. Kaliski and M. Robshaw, "Message authentication with md5," *CryptoBytes (RSA Labs Technical Newsletter)*, vol. 1, no. 1, 1995.

[27] J. Blömer and J.-P. Seifert, "Fault based cryptanalysis of the advanced encryption standard (aes)," in *Financial Cryptography*, R. N. Wright, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, vol. 2742, ch. 12, pp. 162–181.

[28] Y. Fuping, S. Liyuan, and J. Yuanming, "Design and implementation of 3des encryption system based on dsp [j]," *Computer Measurement & Control*, vol. 7, p. 051, 2009.

[29] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, pp. 120–126, February 1978.

[30] L. Harn, M. Mehta, and W.-J. Hsin, "Integrating diffie-hellman key exchange into the digital signature algorithm (dsa)," *Communications Letters, IEEE*, vol. 8, no. 3, pp. 198–200, 2004.

[31] D. Wagner and B. Schneier, "Analysis of the ssl 3.0 protocol," in *In proceedings of the second Unix Workshop on electronic commerce*. USENIX Association, 1996, pp. 29–40.