Formal Security Analysis and Performance Evaluation of the Linkable Anonymous Access Protocol

Rima Addas, Ning Zhang

Formal Security Analysis and Performance Evaluation of The Linkable Anonymous Access Protocol

Rima Addas and Ning Zhang

School of Computer Science
University of Manchester
Manchester, UK

Abstract. The introduction of e-Health applications has not only brought benefits, but also raised serious concerns regarding security and privacy of health data. The increasing demands of accessing health data, highlighted critical questions and challenges concerning the confidentiality of electronic patient records and the efficiency of accessing these records. Therefore, the aim of this paper is to provide secure and efficient access to electronic patient records. In this paper, we propose a novel protocol called the Linkable Anonymous Access protocol (LAA). We formally verify and analyse the protocol against security properties such as secrecy and authentication using the Casper/FDR2 verification tool. In addition, we have implemented the protocol using the Java technology to evaluate its performance. Our formal security analysis and performance evaluation proved that the LAA protocol supports secure access to electronic patient records without compromising performance.

1 Introduction

Information security and privacy in the e-health domain are issues of growing concern. The adoption of electronic patient records, increased regulation, provider consolidation and the increasing need for information exchange between patients, providers and payers, all lead towards the need for a better information security and privacy [1].

The exponential growth of the Internet and electronic health brings not only benefits, but also risks. Numerous attacks pose a real challenge to different aspects of security techniques. Among these security techniques, security protocols play a significant role. They use cryptographic primitives as building blocks to achieve security goals such as authentication, confidentiality and integrity [2].

In [3], we have proposed a new method called 3LI2Pv2 method to support controlled access to electronic patient records (EPRs) with three levels of identity privacy reservations. In the method, we have identified 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 relied of cryptographic primitives to meet its goals. We have informally analysed the 3LI2Pv2 method against some security properties, 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-2 (Linkable anonymous access).

By and large, security protocols have been modelled and verified using informal verification tools. As a result, it is now very common that security protocols, which were previously proposed have found to be vulnerable later on. For example, the Needham-Schroeder public key protocol [4] succeeded in the informal analysis, but failed in the formal verification [5]. 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 ensure correctness [6].

In this paper, a formal method, Casper/FDR2 verification tool [7] [8], is used to model and verify the LAA 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 [9] [10]. Accordingly, we consider it also suitable for the verification of the LAA protocol. The Casper/FDR2 model checker is used to verify the security properties of the protocols. If the protocols do not fulfil the specified security properties, then the FDR2 checker shows a counterexample which represents the cause against vulnerability. After completing the formal verification of the protocol using Casper/FDR2, we implement the protocol using the Java technology [11] to test it against performance. JAVA is selected because it supports a set of standard security primitives such as the hash function SHA-256 [12], the symmetric cryptographic algorithm AES [13] and the asymmetric cryptographic algorithm RSA [14].

This paper is organized as follows; In Section 2, we introduce common security threats. In section 3, we describe, model and verify the LAA protocol. In addition, we set the security goals that the LAA protocol should fulfil. Then, we present the result of the verification. In Section 4, we present the implementation and performance evaluation of the LAA protocol. In Section 5, we conclude the paper and discuss future work.

2 Common Security Threats

Access to electronic patient records is subject to diverse types of security threats and attacks. In this paper, we will exclude threats of environmental origin such as fire or accidental ones such as user errors or software malfunction. The threats that we will consider are, confidentiality threats, integrity threats and authentication threats.

2.1 Confidentiality Threats

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

2.2 Integrity Threats

In this type of threat, an intruder may alter the information exchanged between entities. The attack consists in interfering with the transmissions, so that the recipient receives data, which are different from those sent by the originator. An example of this type of threat is data tampering.

2.3 Authentication Threats

In this type of threat, an intruder may prepare 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). An example of this type of threat is spoofing. Repudiation is also a variant of this type of threats that consists in denying authorship or the contents of data previously sent.

3 Formal Verification and Security Analysis of the Linkable Anonymous Access (LAA) Protocol

In this section, we first describe and model the LAA security protocol with Casper. Then, we introduce some important security requirements that the LAA protocol should fulfil. Finally, we verify the protocol using the FDR2 model checker, discuss the verification result of the protocol and analyse its security requirements.

3.1 The LAA Protocol Description

The purpose of the LAA protocol is to link multiple data objects of the same patient managed by a heath service provider (HSP), but this set of objects cannot be linked to the patient's real identity (e.g. NHS number). This type of access should be granted to users with higher privileges (L1 and L2 users) such as general practices (GPs) and specialist who need such information to proceed with the patient's treatment. Until now, no research has been conducted to analyse the vulnerability of the LAA protocol using a formal verification tool. Table 1 shows the basic notation of the LAA protocol. Figure 1 shows message sequences of the LAA protocol.

Table 1. The LAA Protocol Notation and Meaning

Notation	**Meaning**
a	An identifier of an initiator/client
b	An identifier of a responder/server
ca	An identifier of a certification authority
nx	A random nonce of x
cr	A challenge response
PKx	A public key of x
SKx	A secret Key of x
ts	A time Stamp (an expiration time)
h	A hash function
msg	A message of data request
certa	A PK-certificate of *a* generated by *ca*
attr-certa	An attribute certificate of *a* generated by *ca*
veri1	An integrity verification of *certa*
ps3intral2	An L3 pseudonym Type-III
sigb	A signature of *b*
integrity1, integrity2	Used in attr-certa integrity verification

The communication channel in the LAA protocol, is based on the Secure Socket Layer (SSL) protocol [15] to ensure security for data transmission. For protocol analysis using Casper/FDR2, we assume the following.
- The underlying cryptographic algorithms used in SSL's public key and symmetric key ciphers are secure.
- All parties unconditionally trust the certification authority. The certification authority certifies the public key for clients.
- All parties unconditionally trust the attribute authority who issues the attribute certificates for clients.
- Patients' records have already been de-identified.

In the LAA protocol, *ca* is the certification authority who issues public-key (PK) certificates to legitimate users. Server, *b* is the health service provider (HSP) who provides patient data to the requesting client *a*, the initiator.

The PK-certificate includes two parts, {a, Pk(a), l2, ts} and {h(a, Pk(a), l2, ts}{SK(ca)}. The first part, contains information about the client, such as, identity a, public key of a PK(a), group membership *l2* and timestamp *ts*. The second part, is the signature of the *ca*. *ca* signs subject *a*, public key of *a*, *PK(a)*, a group membership *l2* and timestamp *ts* using its own private key *SK(ca)*, which is only known to the *ca*. Since it is encrypted with the private key of *ca*, any other user cannot spoof it. It provides confidence of certificate's

information to a participant. The certificate can only be decrypted by the public key of *ca*, which is known to client *a* and HSP *b*. The following describes the message sequence of the LAA protocol depicted in Figure 1.

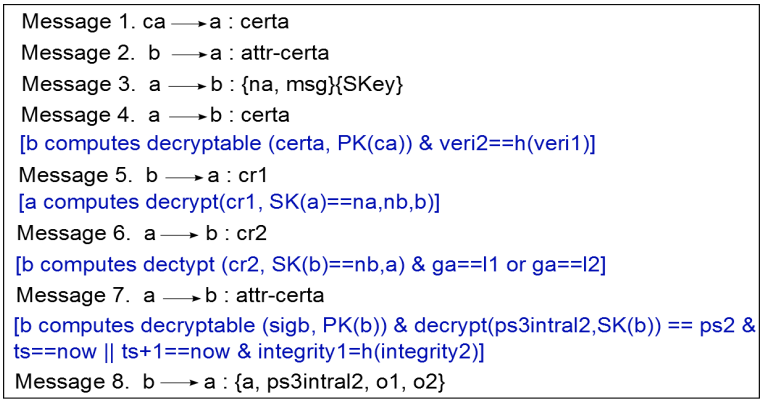

Fig. 1. The LAA protocol description

Message 1: *ca* issues and sends the PK-certificate, *certa*, to client *a* in order to authenticate client *a* and distribute *PK(a)* safely.

Message 2: *b* issues and sends the attribute certificate, *attr-certa*, to *a*. This certificate includes the issuer name (b), the client name (a), an L3 pseudonym (ps3intral2), a timestamp (ts) and the issuer's signature on the certificate (sigb). The L3 pseudonym (ps3intral2), contains another pseudonym (ps2), a recovery token (w), issuer name (b) and the request (All) which indicates all objects of a patient managed by this HSP *b*.

Message 3: Client *a* sends his/her nonce (na) and a message containing the requested data encrypted with the shared symmetric key.

Message 4: *a* sends his/her PK-certificate (certa) to *b*. This certificates contains *veri1* and *veri2*. The variable *veri1* contains the plain content of the certificate. The variable *veri2* contains the deciphered *b*'s signature on the certificate. Using *veri1* and *veri2* allows checking the integrity of the certificate. HSP *b* then validates the *ca's* public on the certificate and verifies the certificate's integrity.

Message 5: HSP *b* sends to client *a* the challenge response *cr1*, which contains the random nonces (*na*, *nb*) and his/her identity *b*, encrypted with a's public key. Client *a* checks if *cr1* is decryptable by SK(a) and contains the right nonce *na*. This step is essential to allow client *a* to authenticate verifier *b*.

Message 6: Client *a* sends to *b* the challenge response *cr2*, which contains *nb* and *a*. Recipient *b* checks if *cr2* is decryptable by SK(b) and contains the right nonce *nb*. This step is essential to allow *b* to authenticate *a*. Also, in this step, *b* checks a's group membership to ensure that s/he belongs to the right group and legitimate for this type of access.

Message 7: After successful authentication, *a* sends to *b* his attr-cert to check his authorisation. HSP *b* then 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 (ps3intral2). After that, it verifies the integrity of the lower-level pseudonym (ps2) to ensure that it has not been altered during transmission.

Message 8: Finally, after successful authorisation, *b* forwards to *a* the requested patient's data objects indexed with the right pseudonym and encrypted with the shared secret key.

3.2 Modelling the LAA Protocol Using Casper

Based on the LAA protocol's notation in Table 1, we model the LAA protocol in Casper script as shown in Figure 2.

```
#Protocol description
--ca issued and sends PK-certificate to client a
0. -> a : {{a,PK(a),{12}%ga,ts}%veri1,{{h(a,PK(a),{12}%ga,ts)}%veri2}{SK(ca)%skca}
%certa}{PK(a)}
--a wants to contact b
1. -> a : b
--a sends his original request message with a nonce
2a. a -> b : {msg, na}{PK(b)}
--a sends his PK-certificate to be verified by b
2b. a -> b :{veri1%{a,PK(a),ga%{12},ts},{certa%{veri2% {h(a,PK(a),ga%{12},ts)}}}
{SK(ca)}} {SKey}
[decryptable(certa, PK(ca)) and veri2== h(veri1) and ts==now or ts+1==now]
--Mutual authentication and check user membership
3. b -> a : {{b, nb, na}{PK(a)} %cr1}{Skey}
[decryptable(cr1, SK(a))]
4. a -> b :{{a, nb}{PK(b)} %cr2}{Skey}
[decryptable(cr2, SK(b)) and ga==12 or ga==11]
-- b issues and sends an attribute certificate to a
5. b -> a :{{b,a,{ps2,w,b,ALL, nonce}%integrity2, {h(ps2, w, b, ALL, nonce)}%
integrity1}{PK(b)}% ps3intral2,ts, {h(b,a,ps3intral2,ts)} {SK(b)} %sigb} {SKey}
[ts==now or ts+1==now]
--a sends to b his attribute certificate for authorisation verification
6. a -> b :{{b,a, ps3intral2 %{integrity2%{ps2, w,b,ALL,nonce},integrity1%
{h(ps2,w,b,ALL,nonce)}}, ts}{PK(b)},sigb%{h(b,a,ps3intral2,ts)}{skb%SK(b)}} {SKey}
[decryptable(sigb, PK(b)) and integrity1== h(integrity2) and decrypt(ps3intral2,
SK(b))== (ps2, w, b, ALL) and ts==now or ts+1==now]
--b sends the response to client a
7. b -> a : {a, na, ps3intral2, o1, o2, ts}{SKey}
[ts==now or ts+1==now]
```

Fig. 2. The LAA protocol modelling using Casper

3.3 LAA Protocol Goals

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

(P1) Data Confidentiality: Confidentiality is a vital requirement that provides secrecy and privacy in e-health applications. An unauthorised party or an intruder should not be able to learn anything about any communication between two entities by observing or even tampering the communication lines.

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

(P3) Mutual Authentication: Or two-way authentication, refers to both entities of the protocol should authenticate each other to allow secure exchange of data between them.

(P4) Certificate Manipulation Protection: It should be guaranteed that certificates (i.e., PK-certificates) presented in the protocol by entities are valid and have not been corrupted or modified during transmission.

(P5) Credential Forgery Protection: It should be assured that users' credentials are not stolen or forged. This is because it can lead to elevation of privileges attack. That is when a user with limited privileges assumes the identity of a user with higher privileges to gain access to patient confidential data.

(P6) 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.

(P7) Anonymous Linkability: A user with L2 access credentials should be able to link multiple de-identified or anonymous objects of the same patient managed by an HSP but should not be able to link them to the patient's real identity.

3.4 Verification Result and Security Analysis of the LAA Protocol

The result of the verification using Casper/FDR2 tool confirms that the LAA protocol has fulfilled l the security properties identified in Section 3.3. The result of the verification is shown in Figure 3.

```
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/L2intra-
HSPAccessProtocol.csp
Done

Starting FDR
Checking /home/Rima/Download/casper-2.0/L2intra-
HSPAccessProtocol.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::AuthenticateRESPONDERToINITIATORAgreement_na [T= AUTH1
_M::SYSTEM_1
No attack found

Checking assertion AUTH2
_M::AuthenticateINITIATORToRESPONDERAgreement_nb [T= AUTH2
_M::SYSTEM_2
No attack found

Done
```

Fig. 3. Verification result of the LAA protocol using Casper/FDR2

(P1) Data Confidentiality: was fulfilled by deploying cryptographic techniques such as symmetric cryptoystem, asymmetric cryptoystem, and hash functions.
(P2) Integrity Protection: was achieved by using digital signatures and hash functions that can detect any data modification during transmission.
(P3) Mutual Authentication: was met by integrating the challenge response protocol.
(P4) Certificate Manipulation Protection: was abided by including a timestamp in the certificate, which can spot any manipulation or sniffing.
(P5) Credential Forgery Protection: was met by adding the credential holder's 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 detect any forgery.
(P6) Data Freshness: was achieved by including a freshly random nonce with the transmitted data.
(P7) Anonymous Linkability: was fulfilled by integrating the L3 pseudonym-TypeIII in the user's access credential. This allows linkable anonymous access to patient data as it contains a lower-level pseudonym that can be used to link all the patient's objects.

4 Implementation and Performance Evaluation

This section illustrates the implementation and performance evaluation of the LAA security protocol.

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

To extract the access delay and server computational time resulted from the LAA protocol, we have measured the time taken to execute the protocol under two scenarios.
- The first scenario is called the L3 Scenario. In this scenario, we run the protocol without applying any extra security layer to the protocol. This scenario is the Level-3 access, which has been described in the introduction section.
- The second scenario is called the L2 Scenario. In this scenario, we run the protocol with applying our additional security solution.

The measurements are taken for 10 execution rounds for each scenario, and the averages are calculated. The results are depicted in Figure 4.

4.1 Implementation Hardware and Software

To implement the LAA protocol, we have used a desktop computer running Windows 8 with a 2.30 GHz Intel Core i3 and 8GB of RAM. The software used to implement the LAA protocol is JAVA 2 Platform, Standard Edition (J2SE).

4.2 Performance Evaluation Result and Analysis

Figure 4 shows that the time (Access delay) taken to execute the LAA protocol (L2 Scenario) is 111 milliseconds, which is approximately 30% more than the time taken in L3 Scenario, which is 85 milliseconds. The server computation time in L2 Scenario is 107 milliseconds, which is approximately 33% more than that in L3 Scenario, which is 81 milliseconds.

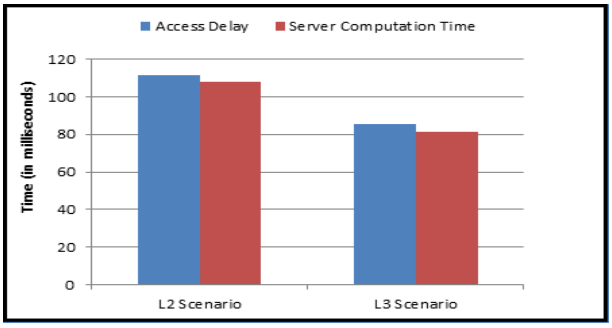

Fig. 4. Performance evaluation result of the LAA protocol

The extra cost in the L2 Scenario is caused by the following reasons.
- 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. - The extra computation in checking the timestamp in the attribute certificate.
- The extra computation in validating the pseudonym, PS3intral2 in the attribute certificate.
- The extra integrity check of the lower-level pseudonym (PS2) included in PS3intral2.

5 Conclusion

The focus of this paper was on two important aspects. The first aspect is designing a robust protocol to facilitate secure access to patient electronic records. The second aspect is providing an efficient access to electronic patient records.

The first aspect was achieved by relying on the formal verification tool, Casper/FDR2. The result of the verification showed that the protocol has fulfilled important security

requirements. We have incorporated SSL protocol, which allowed communication channels to be confidentiality, integrity and authentication protected. Our protocol offers a wide range of significant features. It supports linkable anonymous access to patient data by deploying important cryptographic techniques. It ensures confidentiality of patient sensitive data. It supports data freshness by making use of timestamp and random nonces. It protects from certificate manipulation and credential forgery. Mutual authentication is also supported to obtain unforgeable proof of the participants in the protocol.

The second aspect was achieved by implementing the LAA protocol using the Java technology to evaluate its performance. The result from the protocol implementation showed that we had successfully balanced between security and performance. This is because the increase in performance was linear with the increase of security layer. In other words, the analysis proved that our LAA protocol is secure and efficient. For future work, we aim to extend the analysis of the LAA protocol to other security e-health protocols, considering security and performance as essential criteria.

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

References

1. Alvarez, R.C.: The promise of e-health - a canadian perspective. Ehealth international **1**(1) (September 2002)
2. Pang, C., Hansen, D.: Improved record linkage for encrypted identifying data, Sydney,Austrailia (2006) 164–168
3. Addas, R., Zhang, N.: An enhanced approach to supporting controlled access to eprs with three levels of identity privacy preservations. In Holzinger, A., Simonic, K.M., eds.: Information Quality in e-Health. Volume 7058 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2011) 547–561
4. Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Commun. ACM **21**(12) (December 1978) 993–999
5. Lowe, G.: An attack on the needham-schroeder public-key authentication protocol. Information Processing Letters **56**(3) (1995) 131 – 133
6. Kim, I.G., Choi, J.Y.: 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. Volume 2. (2004) 264–269 Vol.2
7. Lowe, G.: Casper: a compiler for the analysis of security protocols. (June 1997) 18–30
8. LTD., F.S.E.: Failure-divergences refinement fdr2 manual. (2010)
9. Kim, I.G., Kim, H.S., Lee, J.Y., Choi, J.Y.: Analysis and modification of ask mobile security protocol. In: Mobile Commerce and Services, 2005. WMCS '05. The Second IEEE International Workshop on. (2005) 79–83
10. Kim, H.S., Oh, J.H., Choi, J.Y., Kim, J.W.: 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) 152–152
11. Chan, P., Lee, R., Kramer, D.: The Java Class Libraries, Volume 1: Supplement for the Java 2 Platform, Standard Edition, V 1.2. Volume 1. Addison-Wesley Professional (1999)
12. Gilbert, H., Handschuh, H.: Security Analysis of SHA-256 and Sisters Selected Areas in Cryptography. Selected Areas in Cryptography **3006** (2004) 175–193
13. Blömer, J., Seifert, J.P.: Fault based cryptanalysis of the advanced encryption standard (aes). In Wright, R.N., ed.: Financial Cryptography. Volume 2742. Springer Berlin Heidelberg, Berlin, Heidelberg (2003) 162–181
14. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21** (February 1978) 120–126
15. Wagner, D., Schneier, B.: Analysis of the ssl 3.0 protocol. In: In proceedings of the second Unix Workshop on electronic commerce, USENIX Association (1996) 29–40