On the Security Verification of a Short Message Service Protocol

Anca Delia Jurcut*, Madhusanka Liyanage[†], Jinyong Chen[‡], Cornelia Gyorodi[§], Jingsha He[¶]
*[‡]School of Computer Science, University College Dublin, Ireland
[†]Centre for Wireless Communications (CWC), University of Oulu, Finland
[§]Department of Computers and Information Technology, University of Oradea, Romania
[¶]College of Computer Science and Technology, Beijing University of Technology, China
Email: *anca.jurcut@ucd.ie, [†]madhusanka.liyanage@oulu.fi, [‡]jinyong.chen.ul@gmail.com,
[§]cgyorodi@uoradea.ro, [¶]jhe@bjut.edu.cn

Abstract—**Short Message Service (SMS) is a text messaging service component of smart phones, web, or mobile communication systems which requires a high level of security to provide user authentication and data confidentiality. To provide such security features, a high security communication protocol for SMS, called Message Security Communication Protocol (MSCP) was proposed. In this paper, MSCP is formally analyzed using an automated logic-based verification tool with attack detection capabilities. The performed formal verification reveals that the proposed protocol is susceptible to parallel session and denial-of-service (DoS) attacks. The reasoning why these attacks are possible is detailed and an amended protocol is proposed to counter the identified attacks. Formal verification of the amended protocol provides confidence regarding the correctness and effectiveness of the proposed modifications.**

I. Introduction

Short Message Service (SMS) is playing a major role in present-day mobile networks. SMS services are widely used in many online services such as mobile banking, marketing, sales delivery and many more [1], [2]. Since SMS can deliver lots of vital information, SMS systems are becoming vulnerable to more and more attacks, such as deception, eavesdropping, messages tampering, spoofing and forgery [3].

In the technical specifications for SMS [4], the confidentiality and integrity mechanisms are only specified as optional security measures that can be made available, but they are not mandatory requirements for SMS system implementation. Hence, without these SMS security options, the SMS messages transmitted on a network are only protected by the communication network itself such as GSM network, shown to be prone to many errors [5]. Therefore, it is required to design a security mechanism that can provide user authentication, data confidentiality and integrity. To offer such security features, Wu and Tan proposed a high security SMS communication protocol called Message Security Communication Protocol (MSCP) [6].

Many cryptographic security protocols such as MSCP [6] are widely used/proposed in secure data exchange over both mobile and infrastructure networks. The design of correctness-provable security protocols is highly complex and prone to errors. The main difficulty in the development of security protocols is to identify the vast possibilities of an adversary to gain information [7].

In such cases, informal and intuitive techniques are often used to analyze security protocols, resulting in insecure protocols to be widely used in public networks. On the other hand, formal verification techniques have been proven to be able to identify previously unknown flaws in security protocols through the means of protocol verification, providing confidence in the correctness of the protocols. In particular, the use of the automated logic-based technique with attack detection capability described in [7]–[12] has been shown to be an effective approach in detecting flaws in the design of security protocols.

This paper is concern with logic-based formal verification and its use in the design of security protocols. We formally analyze MSCP by using an automated logic-based verification tool, which reveals that the protocol is susceptible to a denial-of-service (DoS) attack and a parallel session attack. We analyze the issues in MSCP and propose an amended version of the protocol that counters the identified weaknesses. Then, a formal verification of the amended protocol is provided to verify the correctness and effectiveness of the proposed modifications.

The rest of paper is organized as follows: Section II presents the logic-based verification process. Section III introduces Message Security Communication Protocol (MSCP). Section IV contains the formal verification of MSCP. Section V presents the amended protocol and Section VI presents its formal verification. Finally, Section VII concludes this paper.

II. Logic-based Verification of Protocols

Logic-based verification techniques use a logic theory to reason about a variety of properties of cryptographic protocols, such as authentication, message meaning, message recognition, data confidentiality, privacy and non-repudiation. Logics enable the generation of concise proofs of protocol goals. If all the goals of a protocol are proven to be true, the correctness of the protocol is established.

A cryptographic protocol must be formalized firstly before it can be verified using a logic-based technique: the protocol steps, assumptions and goals have to be expressed formally using the language of the logic. Then, a process of deductive reasoning is applied, whereby the desired protocol goals are deduced by applying a set of axioms and inference rules to the assumptions as well as to the message exchanges involved in the protocol.

The initial protocol assumptions reflect the initial knowledge, beliefs and possessions of participating principals at the beginning of each session. The desired protocol goals are a set of knowledge, beliefs and possession of protocol participants at the end of each session. Each principal can learn new knowledge and increase its possessions as a result of receiving new messages. The logic postulates enable the derivation of the new knowledge and possessions from current assumptions and receiving messages.

The objective of the logical analysis is to verify whether the desired goals of the protocol can be derived from the initial assumptions and protocol steps. It is imperative that the protocol goals to be correctly formulated. Successful verification of the goals of the protocol can be considered secure within the scope of the logic. Failure to achieve the goals is generally caused by missing some hypotheses in the initial protocol assumptions or the presence of some weaknesses in the protocol. If a weakness is uncovered, the protocol should be provided with a systemic solution to adjust the insecure features.

Many of the existing formal logics theories are applied manually to prove the correctness of security protocols. However, manual completion of the proofs is difficult and error prone [11]. Logic-based techniques require a high level of skill to use, relying on the ability and experience of a user to generate the formal proof of the protocol.

Automation of the verification process minimizes the chance of faulty proofs and simplifies the verification process for the protocol verifier. In addition, logics have the advantages of being decidable and efficiently computable and thus can completely be automated. We use CDVT/AD verification tool [8], [9], [13] in this research, which is an automated system that implements a modal logic of knowledge and an attack detection theory. This tool can analyze the evolution of both knowledge and belief during a protocol execution and therefore is useful in addressing issues of both security and trust. Additionally, the verification tool has the capability of detecting protocol design weaknesses that can be exploited for launching freshness and interleaving session attacks. The attack detection mechanism in CDVT/AD incorporates rules to address the following five main types of issues: (1) message freshness, (2) message symmetry, (3) handshake construction, (4) signed statements and (5) certificates.

III. Message Security Communication Protocol (MSCP)

In 2009, Wu and Tan proposed MSCP for SMS [6] to counter impersonation attacks from illegal intruders. The authentication session of the protocol is described in Figure 1.

1. A → B : {IDa,Na}KbPub
2. B → A : {Na,Nb}KaPub
3. A → B : {Nb}KbPub
4. A → B : {{Ks}KaPriv}KbPub

Fig. 1: Wu and Tan Mobile Communication Protocol for SMS

Here, principal A is the initiator who wants to establish a session with responder principal B. In step 1, A sends a message that includes its identity IDa and a temporary random number (i.e. nonce) Na. The message is encrypted using principal B's public key KbPub. Principal B responds in step 2 by returning nonce Na along with a new nonce generated by B, Nb. This message is encrypted using A's public key KaPub. After receiving the message in step 2, A should be assured that he is talking to B since only B is able to decrypt the message in step 1 to retrieve Na. In step 3, A returns the nonce Nb to B encrypted with B's public key. Finally, in step 4, A generates a session key Ks and sends the encrypted session key to B. B can decrypt and retrieve the session key. As a result, both A and B have the session key Ks. This protocol can be considered as two logically disjoint protocols. While messages 1, 2 and 3 are concerned with mutual authentication between principal A and B, message 4 is a session key update mechanism.

IV. Formal Verification of MSCP

The CDVT/AD verification tool [8], [9] is used to establish the correctness of the authentication session of MSCP. Further, any vulnerability in the design of the verified protocol that can be exploited by freshness and interleaving session attacks will be highlighted by the verification tool.Prior to verification, the protocol must be formally expressed using the language of the verification tool before verification starts. As stated in Section II, a formalized protocol should consist of three components:

- Initial assumptions: conditions that hold before the protocol starts.
- Protocol steps: the messages exchanged between the principals.
- Protocol goals: conditions that are expected to hold if the protocol terminates successfully.

A. Initial Assumptions

Initial assumptions are statements that define what each principal possesses and knows at the beginning of a protocol run. Following specifies the list of the initial assumptions:

A1: A possess at[0] KaPriv;
A2: A possess at[0] KaPub;
A3: A possess at[0] Kab;
A4: A know at [0] NOT(Zero possess at[0] Kab);
A5: A possess at[0] KbPub;
A6: A know at[0] B possess at[0] KbPriv;
A7: A possess at[0] Na;
A8: A know at[0] NOT(Zero possess at[0] Na);
A9: B possess at[0] KbPriv;
A10: B possess at[0] KbPub;
A11: B possess at[0] KaPub;
A12: B know at[0] A possess at[0] KaPriv;

A13: B possess at[0] Nb;
A14: B know at[0] NOT(Zero possess at[0] Nb);
A15: B know at [0] NOT(Zero possess at[0] Kab);

Statements A1-8 define the initial assumptions for principal A before a protocol run with principal B, i.e. at time t0. Assumptions A1 and A2 express the fact that before the start of the protocol run, A possesses his private and public keys. A3 specifies that the fresh session key Kab is possessed by A and A4 indicates that A knows that he is the only principal that possesses this session key before the start of the protocol run. A5 expresses that A possesses the public key of B and A6 indicates that A knows that only B possesses his own private key before the start of the protocol run. Assumption A7 specifies that A possesses the nonce Na and assumption A8 states that A knows that no other principal possesses this nonce at the time. Statements A9-A15 define the initial assumptions of B's possessions and knowledge before the start of the protocol run. A9 and A10 state that B possesses his public and private keys. A11 expresses that B possesses A's public key and A12 indicates that B knows that A is the only principal that possesses his own private key. A13 expresses the fact that B possesses the nonce Nb and A14 indicates that B knows that the nonce is fresh for the current run of the protocol. Statement A15 specifies that the fresh session key Kab is not possessed by any principal before the start of the protocol run.

B. Protocol Steps

The steps of MSCP are formalized as follows:

S1: B receive at[1] {A, Na}KbPub;
S2: A receive at[2] {Na, Nb}KaPub;
S3: B receive at[3] {Nb}KbPub;
S4: B receive at[4] {{Kab}KaPriv}KbPub;

C. Protocol Goals

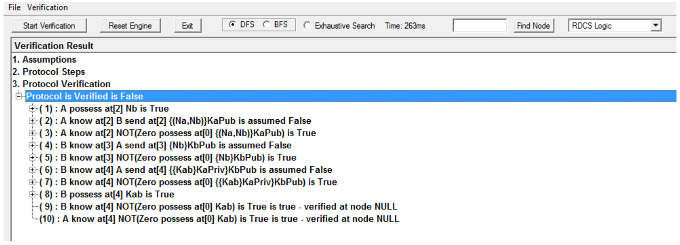
The objectives of MSCP are to achieve the mutual authentication of principals A and B and to distribute a secret session key Kab. These goals are formalized as follows:

G1: A possess at[2] Nb;
G2: A know at[2] (B send at[2] {Na, Nb}KaPub);
G3: A know at[2] NOT(Zero possess at[0] {Na, Nb}KaPub);
G4: B know at[3] (A send at[3] {Nb}KbPub);
G5: B know at[3] NOT(Zero possess at[0] {Nb}KbPub); // true freshness
G6: B know at[4] (A send at[4] {{Kab}KaPriv}KbPub); // false any data encrypted with KbPub
G7: B know at[4] NOT(Zero possess at[0] {{Kab}KaPriv}KbPub);
G8: B possess at [4] Kab;
G9: B know at[4] NOT(Zero possess at[0] Kab);
G10: A know at[4] NOT(Zero possess at[0] Kab);

Goals G1-G3 relate to authentication of B to A. G1 states that A possesses the nonce Nb at step 2. G2 states that A knows at step 2 that B is the source of message component {Na, Nb}KaPub, which is the reply to As nonce challenge. G3 states that A knows that this message is fresh, i.e. it has been created by B for the current protocol run. Goals G4-G7 are the corresponding goals regarding authentication of A to B. G4 states that B knows that A is indeed the source of message component {Nb}KbPub, i.e. the reply to B's nonce challenge. G5 states that B knows that this message component has been created during the current protocol run. G6 states that B knows at step 4 that A is the source of message component {{Kab}KaPriv}KbPub and G7 expresses that B knows that this message component has been created during the current protocol run. Finally, goals G8-G10 are the corresponding key establishment goals for both participating principals after step 4 completes.

D. Results of MSCP verification

The results of the automated verification are illustrated in Figure 2. The results show that not all the goals of authentication of B to A (i.e. goal G2) and of authentication of A to B (i.e. goal G4) are successfully verified. Additionally, not all the goals concerning key establishment for B (i.e. goal G6) are satisfied.



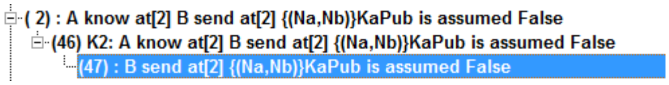
Time: 263ms

Verification Result
1. Assumptions
2. Protocol Steps
3. Protocol Verification

```
⊟·Protocol is Verified is False
   ⊞·( 1) : A possess at[2] Nb is True
   ⊞·( 2) : A know at[2] B send at[2] {(Na,Nb)}KaPub is assumed False
   ⊞·( 3) : A know at[2] NOT(Zero possess at[0] {(Na,Nb)}KaPub) is True
   ⊞·( 4) : B know at[3] A send at[3] {Nb}KbPub is assumed False
   ⊞·( 5) : B know at[3] NOT(Zero possess at[0] {Nb}KbPub) is True
   ⊞·( 6) : B know at[4] A send at[4] {{Kab}KaPriv}KbPub is assumed False
   ⊞·( 7) : B know at[4] NOT(Zero possess at[0] {{Kab}KaPriv}KbPub) is True
   ⊞·( 8) : B possess at[4] Kab is True
    ·( 9) : B know at[4] NOT(Zero possess at[0] Kab) is True is true - verified at node NULL
    ·(10) : A know at[4] NOT(Zero possess at[0] Kab) is True is true - verified at node NULL
```

Fig. 2: Results of the Automated Verification

The verification tool allows to investigate the causes of failed goals by browsing the detail of the verification process. Figure 3 is an example that shows the details of the failed verification of goal G2, where it can be seen that A's inability to authenticate the source of the message component is the cause of the failure.



```
⊟·( 2) : A know at[2] B send at[2] {(Na,Nb)}KaPub is assumed False
   ⊟·(46) K2: A know at[2] B send at[2] {(Na,Nb)}KaPub is assumed False
      ·(47) : B send at[2] {(Na,Nb)}KaPub is assumed False
```

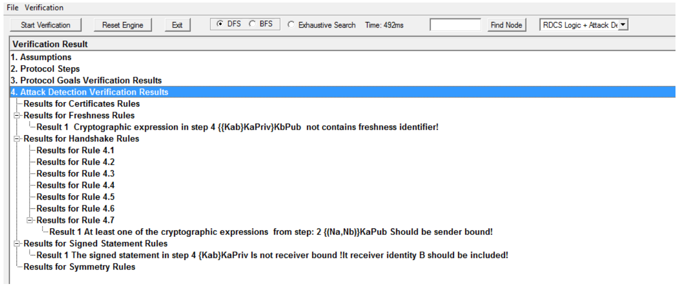
Fig. 3: Details of the Failed Verification of Goal G2

Investigation of the failed protocol goals in this fashion reveals that the protocol suffers from several weaknesses illustrated as follows:

- A's inability to establish that B is the source of message component {Na, Nb}KaPub in step 2 (goal G2) prevents the authentication of B to A.
- B's inability to establish that A is the source of message component {Nb}KbPub in step 3 (goal G4) prevents the authentication of A to B.
- B's inability to establish that A is the source of message component {{Kab}KaPriv}KbPub in step 4 (goal G6) prevents B from accepting the session key.

The conclusion is that neither authentication of A to B nor that of B to A is achieved by MSCP. In addition, three design weaknesses of MSCP concerning identifying freshness and interleaving session vulnerabilities are revealed.

Figure 4 shows that three of the attack detection rules (one freshness rule, one handshake rule and one signed statement rule) are triggered. The result obtained with respect to the freshness rules is that the cryptographic expression in step 4 {{Kab}KaPriv}KbPub is not freshness-protected. This implies that {{Kab}KaPriv}KbPub does not contain anything that would allow principal B to recognize it as being fresh (i.e., a nonce previously generated by B in the same protocol run). The results derived for the handshake rules also lead to revealing a weakness in the protocol where it should contain a sender identifier to prove the source of the message component. The results for the signed statement rules state that the receiver identity needs to be included to distinguish the random value, encrypted using the public key of principal B.



```
Verification Result
1. Assumptions
2. Protocol Steps
3. Protocol Goals Verification Results
4. Attack Detection Verification Results
  Results for Certificates Rules
  Results for Freshness Rules
    Result 1  Cryptographic expression in step 4 {{Kab}KaPriv}KbPub  not contains freshness identifier!
  Results for Handshake Rules
    Results for Rule 4.1
    Results for Rule 4.2
    Results for Rule 4.3
    Results for Rule 4.4
    Results for Rule 4.5
    Results for Rule 4.6
    Results for Rule 4.7
      Result 1 At least one of the cryptographic expressions  from step: 2 {(Na,Nb)}KaPub Should be sender bound!
  Results for Signed Statement Rules
    Result 1 The signed statement in step 4 {Kab}KaPriv Is not receiver bound !It receiver identity B should be included!
  Results for Symmetry Rules
```

Fig. 4: Verification Results of Attack Detection

E. A DoS attack on MSCP

The dependence on successful message reception is a weakness in the protocol that would make the protocol susceptible to DoS conditions. The security of the communication could be compromised by an attacker through intentional message jamming and the jamming of a mobile user can be easily achieved through using low-grade technology [14]–[16]. If the last message is not successfully delivered to principal B, then the attacker could send a random value to B, encrypted with the public key of B. The scenario of the attack is illustrated in Figure 5.

1. A→B : {IDa, Na}KbPub
2. B→A : {Na, Nb}KaPub
3. A→B : {Nb}KbPub
4. A→X|B : {{Ks}KaPriv}KbPub
5. I(A)→B : {data}KbPub
6. I(A)←{data}KaPub→B
7. A|X←Ks→X|B

Fig. 5: Denial of Server Attack on the Wu and Tan Mobile Communication Protocol

In step 1, A initiates and establishes a session with B. B responds with the message {Na,Nb}KaPub in step 2. The attacker can intercept the messages during the session even if he would not be able to decrypt the messages. The attacker could suppress the messages in step 4 and then send any value encrypted using the public key of B in step 5. Finally, B decrypts this message using his own private key KbPriv and encrypts it using A's public key KaPub to get the session key {data}KaPub. Now the attacker has successfully impersonated A to set up a session with B. As a result, the mobile user is not able to use the correct session key Ks to set up a session and, consequently, will be denied access to B. Furthermore, all the user's subsequent authentication requests will be altered by the attacker through repeating the steps described above.

F. A parallel session attack on MSCP

The revealed weaknesses in the protocol can also be exploited by an intruder in a parallel session attack. Suppose intruder I is a user of the mobile communication network. Thus, I is able to set up normal sessions with other users, and vice versa. Besides, as an intruder, I can intercept any messages transmitted over the network and can also generate new messages. However, the intruder is not capable of guessing the content from the encrypted messages unless the messages are encrypted with his own public key. The intruder can generate random values and can also replay encrypted messages that he intercepted in previous sessions of the protocol.

The parallel session attack on the protocol allows intruder I to impersonate principal A by starting a fake session with principal B. The attack involves two simultaneous runs of the protocol: in run i, A establishes a valid session with I; in run ii, I impersonates A to establish a fake session with B. The corresponding attack scenario is illustrated in Figure 6.

i.1. A → I : {IDa, Na}KiPub

ii.1. I(A) → B : {IDa, Na}KbPub

ii.2. B → I(A) : {Na, Nb}KaPub

i.2. I → A : {Na, Nb}KaPub

i.3. A → I : {Nb}KiPub

i.4. A → I : {{Ks}KaPriv}KiPub

ii.3. I(A) → B : {Nb}KbPub

ii.4. I(A) → B : {data}KbPub

Fig. 6: Parallel Session Attack on MSCP

In step i.1, A starts session i with user I through sending nonce Na. In step ii.1, I, who is a dishonest user, impersonates A in order to establish a false session with B, by sending nonce Na obtained in the previous message. B then responds in step ii.2 by generating a new nonce Nb and sending Nb along with Na to A. Intruder I intercepts this message but cannot decrypt it since the message is encrypted with A's public key. Intruder I then forwards the message to A in step i.2. After verification of the message, A believes that B is the source of nonce Nb, since i.2 contains the corresponding response to its challenge sent as part of the message i.1. A replies to I's nonce challenge in step i.3 and therefore I can decrypt the message and obtain Nb. Nb is then sent to B as part of the message ii.3. Hence, B believes that A has correctly established a session with him after completing run ii of the protocol. Note that intruder I can now freely generate a session key which B can obtain from message {data}KaPub in step ii.4. Therefore, intruder I can impersonate a remote user A to get access to B without knowing any secret information.

V. Amendment to the Message Security Communication Protocol (MSCP)

As shown in the Sections IV, MSCP protocol [6] is not secure and it has weaknesses that can be exploited by DoS and interleaving session attacks (i.e. replay and parallel session). An amendment is proposed in this section to remove the discovered weaknesses.

To counter potential replay attacks, all encrypted messages transmitted over the network need to be fresh. The cryptographic message {{Ks}KaPriv}KbPub in step 4 should include a component which would allow the recipient B to recognize the freshness of this message. This can be achieved by introducing the nonce Nb, generated by B at step 2 in the protocol, as illustrated in Figure 7. Thus, the cryptographic expression that contains the new generated key Ks can be identified by B as fresh, i.e. as belonging to the current protocol run. Consequently, any attempt by an intruder to replay message of step 4 will fail, as B can identify the replay through the incorrect value of Nb.

In order to fix the other two weaknesses that can be exploited by parallel session attacks, the outcome of the tool recommends the following solutions: (1) to add the identity of the receiver B in the content of the signed expression {Ks}KaPriv, transmitted in step 4 of the protocol (i.e. the cryptographic expression {Ks}KaPriv should be receiver bound) and (2) to add the identity of the sender B in the content of the cryptographic expression {Na, Nb}KaPub, transmitted in step 2 of the protocol (i.e. modify the content of this message in order to be sender bound). Figure 7 outlines our proposed amended version of the Wu and Tan mobile communication protocol with the original notations.

1. A → B : {IDa, Na}KbPub
2. B → A : {Na, Nb, IDb}KaPub
3. A → B : {Nb}KbPub
4. A → B : {{Ks, IDb, Nb}KaPriv}KbPub

Fig. 7: An Amended Version of the Wu and Tan Mobile Communication Protocol for SMS

VI. Verification of the Proposed Amended Version of MSCP

A. Initial Assumptions

The initial assumptions are similar to the original ones for the Wu and Tan mobile communication protocol. The only changes occur in assumptions A9 and A18 where modifications are made to the messages.

A1: A possess at[0] KaPriv;
A2: A possess at[0] KaPub;
A3: A possess at[0] Kab;
A4: A know at [0] NOT(Zero possess at[0] Kab);
A5: A possess at[0] KbPub;
A6: A know at[0] B possess at[0] KbPriv;
A7: A possess at[0] Na;
A8: A know at[0] NOT(Zero possess at[0] Na);
A9: A know at[0] (A receive at[2] {Na, Nb, B}KaPub IMPLY B send at[2] {Na, Nb, B}KaPub);
A10: B possess at[0] KbPriv;
A11: B possess at[0] KbPub;
A12: B possess at[0] KaPub;
A13: B know at[0] A possess at[0] KaPriv;
A14: B know at[0] A possess at[0] KaPub;
A15: B possess at[0] Nb;
A16: B know at[0] NOT(Zero possess at[0] Nb);
A18: B know at[0] (B receive at[3] {Nb}KbPub IMPLY A send at[3] {Nb}KbPub);
A19: B know at [0] NOT(Zero possess at[0] Kab);

B. Amended protocol steps

The formalization needs to be adjusted due to changes made in the steps of the protocol.

S1: B receivefrom A at[1] {A, Na}KbPub;
S2: A receivefrom B at[2] {Na, Nb, B}KaPub;
S3: B receivefrom A at[3] {Nb}KbPub;
S4: B receivefrom A at[4] {{Kab, B, Nb}KaPriv}KbPub;

C. Amended protocol goals

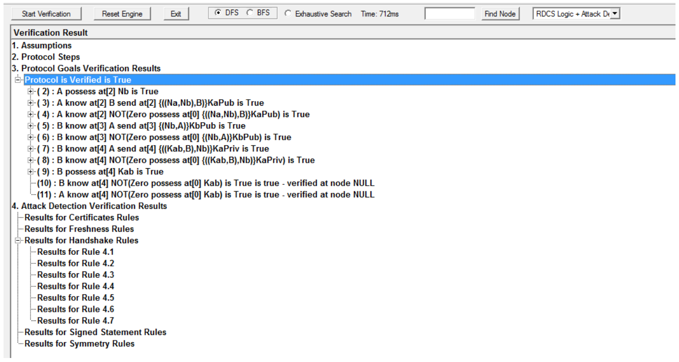
Modifications to the message exchange are also reflected in the corresponding goals.

G1: A possess at[2] Nb;
G2: A know at[2] (B send at[2] {Na, Nb, B}KaPub);
G3: A know at[2] NOT(Zero possess at[0] {Na, Nb, B}KaPub);
G5: B know at[3] (A send at[3] {Nb}KbPub);
G6: B know at[3] NOT(Zero possess at[0] {Nb}KbPub);
G7: B know at[4] (A send at[4] {Kab, B, Nb}KaPriv);
G8: B know at[4] NOT(Zero possess at[0] {Kab, B, Nb}KaPriv);
G9: B possess at [4] Kab;
G11: B know at[4] NOT(Zero possess at[0] Kab);
G12: A know at[4] NOT(Zero possess at[0] Kab);

D. Verification results of the proposed protocol

The results of the automated verification for the amended version of the MSCP protocol are illustrated in Figure 8. As it shows, the outcome for the attack detection verification is null of any message indicating a weakness in the design of the protocol that can be exploited by potential freshness or interleaving session attacks. In addition, all goals are verified successfully, which indicates that the proposed protocol can be considered secure. Further, the total verification time spans

712 ms and memory expense is only 19960 Kbytes during the formal verification of the amended protocol. This outcome provides confidence in the correctness and effectiveness of our proposed solution.



```
● DFS ○ BFS    ○ Exhaustive Search    Time: 712ms    RDCS Logic + Attack De
Verification Result
1. Assumptions
2. Protocol Steps
3. Protocol Goals Verification Results
  Protocol is Verified is True
    ( 2) : A possess at[2] Nb is True
    ( 3) : A know at[2] B send at[2] {((Na,Nb),B)}KaPub is True
    ( 4) : A know at[2] NOT(Zero possess at[0] {((Na,Nb),B)}KaPub) is True
    ( 5) : B know at[3] A send at[3] {(Nb,A)}KbPub is True
    ( 6) : B know at[3] NOT(Zero possess at[0] {(Nb,A)}KbPub) is True
    ( 7) : B know at[4] A send at[4] {((Kab,B),Nb)}KaPriv is True
    ( 8) : B know at[4] NOT(Zero possess at[0] {((Kab,B),Nb)}KaPriv) is True
    ( 9) : B possess at[4] Kab is True
    (10) : B know at[4] NOT(Zero possess at[0] Kab) is True is true - verified at node NULL
    (11) : A know at[4] NOT(Zero possess at[0] Kab) is True is true - verified at node NULL
4. Attack Detection Verification Results
  Results for Certificates Rules
  Results for Freshness Rules
  Results for Handshake Rules
    Results for Rule 4.1
    Results for Rule 4.2
    Results for Rule 4.3
    Results for Rule 4.4
    Results for Rule 4.5
    Results for Rule 4.6
    Results for Rule 4.7
  Results for Signed Statement Rules
  Results for Symmetry Rules
```

Fig. 8: Verification Results of the Amended Protocol

VII. Conclusions

In this paper, the process of formally verifying a security protocol based on modal logic technique was introduced. An automated logic-based verification tool with the capability of detecting freshness and interleaving session attacks was used to verify the security properties of Message Security Communication Protocol (MSCP), which was proposed as a high security communication protocol for Short Message Service (SMS). The formal verification results revealed several weaknesses in MSCP that can be exploited by potential parallel session and Denial of Service (DoS) attacks. These weaknesses were analyzed and an amended protocol immune to these attacks was proposed. Formal verification of the amended protocol verified all of its security properties and could thus provide confidence in the correctness and effectiveness of the proposed modifications.

Acknowledgement

This work has been performed under the framework of COST Action CA15127 (RECODIS) and CA16226 (SHELDON) projects.

References

[1] J. Mullan, J. Mullan, L. Bradley, L. Bradley, S. Loane, and S. Loane, "Bank adoption of mobile banking: stakeholder perspective," *International Journal of Bank Marketing*, vol. 35, no. 7, pp. 1152–1172, 2017.

[2] W. Chmielarz and M. Zborowski, "Aspects of mobility in e-marketing from the perspective of a customer," in *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on*. IEEE, 2016, pp. 1329–1333.

[3] I. I. Androulidakis, "Sms security issues," in *Mobile Phone Security and Forensics*. Springer, 2016, pp. 71–86.

[4] "Technical Specifications for SMSs," http://www.3gpp.org/ftp/Specs/archive/03_series/03.48/, [Online; accessed 01-February-2018].

[5] S. M. Siddique and M. Amir, "GSM Security Issues and Challenges," *7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD*, pp. 19–20, 2006.

[6] S. Wu and C. Tan, "High security communication protocol for sms," in *Multimedia Information Networking and Security, 2009. MINES'09. International Conference on*, vol. 2. IEEE, 2009, pp. 53–56.

[7] R. Dojen and T. Coffey, "The concept of layered proving trees and its application to the automation of security protocol verification," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 3, pp. 287–311, 2005.

[8] A. Jurcut, T. Coffey, and R. Dojen, "A novel security protocol attack detection logic with unique fault discovery capability for freshness attacks and interleaving session attacks," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[9] A. D. Jurcut, T. Coffey, and R. Dojen, "Establishing and fixing security protocols weaknesses using a logic-based verification tool," *Journal of Communication*, vol. 8, no. 11, pp. 795–806, 2013.

[10] A. Jurcut, T. Coffey, and R. Dojen, "Design guidelines for security protocols to prevent replay & parallel session attacks," *computers & Security*, vol. 45, pp. 255–273, 2014.

[11] T. Coffey and P. Saidha, "Logic for verifying public-key cryptographic protocols," *IEE Proceedings-Computers and Digital Techniques*, vol. 144, no. 1, pp. 28–32, 1997.

[12] T. Coffey, R. Dojen, and T. Flanagan, "Formal verification: an imperative step in the design of security protocols," *Computer Networks*, vol. 43, no. 5, pp. 601–618, 2003.

[13] "Crytpographic-procotol Development and Verification Tools," http://dcsl.ul.ie/cdvt-ad-tool/, [Online; accessed 30-September-2017].

[14] T. Mahoney, P. Kerr, B. Felstead, P. Wells, M. Cunningham, G. Baumgartner, and L. Jeromin, "An investigation of the military applications of commercial personal satellite-communications systems," in *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, vol. 1. IEEE, 1999, pp. 112–116.

[15] E. B. Felstead and R. J. Keightley, "Robustness capabilities of transponded commercial satellite communications," in *Military Communications Conference, 1995. MILCOM'95, Conference Record, IEEE*, vol. 2. IEEE, 1995, pp. 783–787.

[16] J. W. Lee and V. A. Marshall, "Maximum capacity prediction and anti-jam performance analysis for commercial satellite communication systems," in *IEEE Military Communications Conference*. IEEE, 1994, pp. 506–510.