On the security analysis of authenticated group key exchange protocols for low-power mobile devices

Yue Li
Department of Electronic and Computer Engineering
University of Limerick,
Limerick, Ireland.
Yue.Li@ul.ie

Thomas Newe
Department of Electronic and Computer Engineering
University of Limerick,
Limerick, Ireland.

Abstract

Secure communications are paramount in today’s wireless network system, where highly sensitive information is delivered through mobile applications. Cryptographic protocols are used to provide security services, such as confidentiality, authentication and non-repudiation. The design of secure group key exchange protocols is one of many important security issues in wireless networks. Recently, Bresson et al. [1] proposed a mutual authentication and group key exchange protocol suitable for a mobile wireless network which consists of many resource-constrained mobile nodes and a powerful server. Nam et al. in [2] identified some attacks on Bresson et al.’s protocol and proposed an improved version which is supposed to fix the security flaws, but this modified protocol also has some security flaws which are identified by a formal verification of the protocol in this paper. In this paper, the Bresson et al.’s and Nam et al.’s modified group key exchange protocols are discussed. A formal verification of these protocols using Coffey-Saidha-Newe(CSN) modal logic is given to detect protocol weakness. The active attacks are presented to demonstrate the security flaws detected by the formal verification.

Keywords: network security, group key exchange, formal method, modal logic, wireless communication.

I. INTRODUCTION

With the rapid development of mobile applications, such as wireless internet services, mobile access service and mobile e-commerce, it is clear that secure communication is essential and important for their full adoption. However, most security technologies currently deployed in wired networks are not fully applicable to wireless networks involved in resource-limited mobile nodes because of their low processing capability and limited power supply which are inherent in the mobility nature.

It is necessary that the cost of security-related operations should be minimized for mobile devices, where the required security services are not compromised. This requirement makes the design of security protocols well suited for wireless mobile networks more difficult, because most cryptographic algorithms require many expensive computations. Protocols for group key exchange are essential in building secure multicast channels for mobile applications where a large number of users are likely to be involved. Recently, Bresson et al.[1] proposed an authenticated group key exchange protocol suitable for asymmetric wireless network that consists of many resource-limited mobile nodes and a powerful node with less restriction. The design goal of the protocol is to achieve mutual authentication and forward secrecy while minimizing the computational burden on low power mobile clients. In paper [2], Nam et al. identified some possible active attacks on Bresson et al.’s protocol and proposed an improved version to fix those security flaws.

In this paper the Bresson et al.’s and Nam et al.’s modified group key exchange protocols are discussed. The Coffey-Saidha-Newe(CSN) logic is then presented and a formal analysis of both security protocols is given. This analysis clearly shows the problems that exist in the Bresson et al.’s group key exchange protocol and how Nam et al’s modified protocol fails to achieve its security goals. An active attack is presented against Nam et al’s modified protocol to demonstrate the weakness detected by the formal verification analysis.

II. REVIEW OF TWO GROUP KEY EXCHANGE PROTOCOLS

A. Notations and Terms

Let \( U = \{U_1, U_2, ..., U_n\} \) be the initial set of low-power nodes that want to generate a group key with powerful node S. Each client as well as the base station holds a pair of secret/public keys at the initialization phase before the protocol starts. The following system parameters and notations are used to describe the protocols in this section:

- \( g \) and \( p \) are publicly known large primes
- \( \theta \): denotes the set of all potential clients,
- \( c \): denotes counter, i.e. for GKE.Setup, the counter value is initialized to zero.
- \( SK_i \): a low-power node \( U_i \)'s secret key in \( Z_q^* \).
- \( PK_i \): a low-power node \( U_i \)'s public key such that \( PK_i = g^{SK_i} \mod p \).
- \( SK_S \): the powerful node S’s secret key in \( Z_q^* \).
- \( PK_S \): the powerful node S’s public key such that \( PK_S = g^{SK_S} \mod p \).
The algorithm executes in two rounds. In the first round, S collects contributions from individual clients and then, in the second round, it sends the group keying material to the clients. The actual protocol proceeds as follows:

Step 1: Each clients \( U_i \) chooses a random \( x_i \in \mathbb{Z}_{q^*} \) and computes
\[
    y_i = g^{x_i} \mod p;
\]
\#
\[
    a_i = PK_i^{x_i} \mod p;
\]
Client \( U_i \) then signs \( y_i \) to obtain signature \( \sigma_i \) and sends \( (y_i, \sigma_i) \) to the server \( S \).

Step 2: For all \( i \in \mathbb{N} \), The server \( S \) verifies the signature \( \sigma_i \), and computes
\[
    a_i = y_i^{SK_S}
\]
then initializes the counter \( c \) to 0, and computes the shared secret value
\[
    K = H(C || a_i, a_2, \ldots, a_n)
\]
and
\[
    K_r = K \oplus H(c || a_i)
\]
The server \( S \) sends to each clients \( U_i \) the values \( (c, K_r) \).

Upon receiving \( c \) and \( K_r \), client \( U_i \) recovers the shared secret value \( K \) as
\[
    K = K_r \oplus H(c || a_i)
\]
Finally, both the server and the clients compute the same session key as:
\[
    GK = H(K || U | S)
\]

### C. The Nam et al.’s group key exchange protocol

Nam et al. demonstrate the insecurity of the Bresson et al.’s protocol by presenting some active attacks against implicit key authentication, forward secrecy, and known key security. The authors applied a replay attacks to demonstrate the security flaws in implicit key authentication of the protocol.

To overcome the attacks and to provide the implicit key authentication, Nam et al. improved the protocol.

**Initialization phase:**

During the initialization phase, each potential participant (including both the server and the clients) generates the signing private/public keys \( (SK, PK) \) by running the key generation algorithm of a signature scheme.

**Modified setup algorithm**

![Diagram](image-url)
III. THE CSN LOGIC LANGUAGE

The CSN [4] logic provides a means of verifying hybrid cryptographic protocols. The logic can analyse the evolution of both knowledge and belief during a protocol execution and is therefore useful in addressing issues of both security and trust. The inference rules provided are the standard inferences required for natural deduction and the axioms of the logic are sufficiently low-level to express the fundamental properties of hybrid cryptographic protocols, such as the ability of a principal to encrypt/decrypt based on knowledge of a cryptographic key. The logic is capable of analysing a wide variety of hybrid cryptographic protocols because the constructs of the logic are general purpose and therefore provide the user with increased flexibility allowing him to develop his own theories.

The underlying assumptions of the logic can also be stated as: The communication environment is hostile but reliable; the cryptosystems used are ideal. That is, the encryption and decryption functions are completely non-invertible without knowledge of the appropriate cryptographic key and are invertible with knowledge of the appropriate cryptographic key; Key’s used by the system are considered valid if they have not exceeded their validity period and only known by the rightful owner(s).

A. The CSN Logic Language

- a,b,c,..., general propositional variables
- \( \Phi \), an arbitrary statement
- \( \Sigma \) and \( \Psi \), arbitrary entities
- \( i \) and \( j \), individual entities
- \( \text{ENT} \), the set of all possible entities
- \( k \), a cryptographic key. In particular, \( k_\Sigma \) is the public key of entity \( \Sigma \) and \( k_\Sigma^{-1} \) is the corresponding private key of entity \( \Sigma \)
- \( t, t', t'' \)... represents moments in time.
- \( t_1, t_2, t_3 \ldots \) represents time after each step of a protocol. For example, \( t_1 \) represents time after step 1 of a protocol has completed
- \( e(x,k_\Sigma) \), encryption function, encryption of \( x \) using key \( k_\Sigma \)
- \( d(x,k_\Sigma^{-1}) \), decryption function, decryption of \( x \) using key \( k_\Sigma^{-1} \)
- \( k_\Sigma, k_\Psi \), Shared secret key for entities \( \Sigma \) and \( \Psi \).
- \( KS_{\Sigma,\Psi} \), Set of good shared keys for entities \( \Sigma \) and \( \Psi \).
- \( SS_{\Sigma,\Psi} \), Shared secret for entities \( \Sigma \) and \( \Psi \) (secret can be fresh).
- \( SS_{\Sigma,\Psi} \), Set of good shared secrets for entities \( \Sigma \) and \( \Psi \).
- \( E(x, k_{\Sigma,\Psi}) \), Encryption of plaintext message \( x \) using the shared secret key of entities \( \Sigma \) and \( \Psi \).
- \( D(x, k_{\Sigma,\Psi}) \), Decryption of ciphertext message \( x \) using the shared secret key of entities \( \Sigma \) and \( \Psi \).
- \( K \), propositional knowledge operator (true or false evaluation) of Hintikka. \( K_{\Sigma,\Phi} \) means \( \Sigma \) knows statement \( \Phi \) at time \( t \).
- \( L \), knowledge predicate (assigns an object a property). \( L_{\Sigma,\Phi} \) means \( \Sigma \) knows and can reproduce object \( x \) at time \( t \).
- \( B \), belief operator. \( B_{\Sigma,\Phi} \) means \( \Sigma \) believes at time \( t \) that statement \( \Phi \) is true.
- \( C \), ‘contains’ operator. \( C(x,y) \) means that the object \( x \) contains the object \( y \). The object \( y \) may be clear text or cipher text in \( x \).
- \( S \), emission operator. \( S(\Sigma, t, x) \) means \( \Sigma \) sends message \( x \) at time \( t \).
- \( R \), reception operator. \( R(\Sigma, t, x) \) means \( \Sigma \) receives message \( x \) at time \( t \).
- \( A \), authentication Operator. \( A(\Sigma, t, \Psi) \) means that \( \Sigma \) authenticates \( \Psi \) at time \( t \).

The language includes the classical logical connectives of conjunction (\&), disjunction (\lor), complementation (\neg) and material implication (\rightarrow). The symbols \( \forall \) and \( \exists \) denote universal and existential quantification respectively. \( \in \) indicates membership of a set and / denotes set exclusion. The symbol \( \vdash \) denotes a logical theorem. The logic does not contain specific temporal operators, but the knowledge, belief and message transfer operators are time-indexed.

B. Inference Rules

The logic incorporates the following rules of inference:

R1: From \( \vdash p \) and \( \vdash (p \rightarrow q) \) infer \( \vdash q \)

R2: (a) From \( \vdash p \) infer \( \vdash K_{\Sigma, p} \);

(b) From \( \vdash p \) infer \( \vdash B_{\Sigma, p} \)

R1 is the Modus Ponens and states that if schema \( p \) can be deduced and \( (p \rightarrow q) \) can be deduced, then \( q \) can also be deduced. R2 consists of the Generalisation rules which state that if \( p \) is a theorem, then knowledge and belief in \( p \) are also theorems.

The logic also includes the following standard propositional rules of natural deduction:

R3: From \( (p \land q) \) infer \( p \)

R4: From \( p \) and \( q \) infer \( (p \land q) \)

C. Axioms

Two types of axioms are used in this logic, logical and non-logical. Logical axioms are general statements made in relation to any system, while non-logical are system specific.

Logical Axioms

The logic includes the following standard modal axioms for knowledge and belief:

A1: \( \exists \forall \exists \forall q(K_{\Sigma, p} \land K_{\Sigma, l} (p \rightarrow q) \rightarrow K_{\Sigma, q}) \)

A2: \( \exists \forall p(K_{\Sigma, p} \rightarrow p) \)

The axiom A1 is application of the Modus Ponens to the knowledge operator. The axiom A2 is called the knowledge axiom and is said to logically characterise knowledge. If something is known, then it is true. This property distinguishes between knowledge and belief.

A3: (a) \( \exists \forall x \exists i, i \in \{\text{ENT}\}(L_{\Sigma, x} \rightarrow \forall t', t \geq t L_{i, t, x}); \)

(b) \( \exists \forall x \exists i, i \in \{\text{ENT}\}(K_{\Sigma, x} \rightarrow \forall t', t \geq t K_{i, t, x}) \)

Axioms A3 (a) and A3(b) asserts that knowledge, once gained, cannot be lost.
A4: \( \exists x \exists y (\exists i,i \in \{\text{ENT}\} L_{i,x} y \land C(y,x) \rightarrow \exists j \in \{\text{ENT}\} L_{j,x} y) \)

If a piece of data is constructed from other pieces of data, then each piece of data involved in the construction must be known to some entity.

**Non-logical Axioms**

The non-logical axioms reflect the underlying assumptions of the logic. These assumptions relate to the emission and reception of messages and to the use of encryption and decryption in these messages.

A5: \( \exists x \exists i \in \{\text{ENT}\} \Sigma_i(x,t) \rightarrow L_{i,x} \land \exists j \in \{\text{ENT}\} \Sigma_j(t') \rightarrow R(i,t',x) \)

The emission axiom A5 states that: if a message is sent at time \( t \), then \( \Sigma \) knows \( x \) at time \( t \) and some entity \( i \) other than \( \Sigma \) will receive \( x \) at time \( t' \) subsequent to \( t \).

A6: \( \exists x \exists (R(\Sigma,t,x) \rightarrow L_{\Sigma,x} \land \exists i \in \{\text{ENT}\} \exists t',t < t \rightarrow S(i,t,x) \) )

The reception axiom A6 states that: if \( \Sigma \) receives a message \( x \) at time \( t \), then \( \Sigma \) knows \( x \) at time \( t \) and some entity \( i \) other than \( \Sigma \) has sent \( x \) at time \( t' \) prior to \( t \).

A7: (a) \( \exists x \exists i \in \{\text{ENT}\} (L_{i,x} \land L_{i,k} \rightarrow L_{i,y}(e(x,k))) \quad (b) \exists x \exists i \in \{\text{ENT}\} (L_{i,x} \rightarrow L_{i,y}(d(x,k))) \)

Axioms A7(a) and A7(b) refer to the ability of an entity to encrypt or decrypt a message when it has knowledge of a public or private cryptographic key.

A8: (a) \( \exists x \exists i \in \{\text{ENT}\} (\neg L_{i,x} \land \forall t', t < t \rightarrow \neg L_{i,y}(e(x,k))) \land \neg (\exists y (R(i,t,y) \land C(y,e(x,k)))) \rightarrow L_{i,y}(e(x,k)) \)

(b) \( \exists x \exists i \in \{\text{ENT}\} (\neg L_{i,k} \land \forall t', t < t \rightarrow \neg L_{i,y}(d(x,k))) \land \neg (\exists y (R(i,t,y) \land C(y,d(x,k)))) \rightarrow L_{i,y}(d(x,k)) \)

Axioms A8 (a) and A8(b) refer to the impossibility of encrypting or decrypting a message without knowledge of the correct key. Axiom A8(a) states that if an entity does not know \( k \) at \( t \) and does not know, prior to \( t \), the encryption \( e(x,k) \) and also does not receive \( e(x,k) \) at \( t \) in a message, then the entity cannot know \( e(x,k) \) at \( t \). Axiom A8b makes a similar statement for the decryption of a message \( x \) without knowledge of the decryption key.

A9: \( \forall t \forall i \in \{\text{ENT}\} L_{i,k} \land \forall j \in \{\text{ENT}\} \neg L_{i,j} \land k \)

The secrecy axiom A9 states that the private keys used by the system are known only to their rightful owners.

A10: \( \exists x \exists i \in \{\text{ENT}\} L_{i,x}(d(x,k)) \rightarrow L_{i,x} \)

Axiom A10 states that if an entity knows and can reproduce \( d(x,k) \) and \( k \) at time \( t \) then it knows and can reproduce \( x \), this implies that this entity knows at time \( t \) that \( \Sigma \) knows and can reproduce \( x \) prior to \( t \).

A11: (a) \( \exists x \exists i \in \{\text{ENT}\} (L_{i,x} \land L_{i,k} \rightarrow L_{i,y}(E(x,k))) \)

(b) \( \exists x \exists i \in \{\text{ENT}\} (L_{i,x} \land C(y,E(x,k))) \land L_{i,k} \rightarrow L_{i,y}(D(x,k))) \)

Axiom 11 refers to the ability an entity has to encrypt or decrypt a message in a symmetric way when it has knowledge of a secret key.

A12:

(a) \( \exists x \exists i \in \{\text{ENT}\} (\neg L_{i,x} \land \forall t', t < t \rightarrow \neg L_{i,y}(E(x,k))) \land \neg (\exists y (R(i,t,y) \land C(y,E(x,k)))) \rightarrow L_{i,y}(E(x,k))) \)

(b) \( \exists x \exists i \in \{\text{ENT}\} (\neg L_{i,x} \land \forall t', t < t \rightarrow \neg L_{i,y}(D(x,k))) \land \neg (\exists y (R(i,t,y) \land C(y,D(x,k)))) \rightarrow L_{i,y}(D(x,k))) \)

Axiom 12 refers to the inability of an entity to encrypt or decrypt data without knowledge of the appropriate shared secret key.

A13: \( \forall t \forall i \in \{\text{ENT}\} \neg L_{i,x} \land L_{i,k} \rightarrow L_{i,y}(e(x,k)) \)

Axiom 13 states that only the rightful owners of a shared secret key know that key and this implies that this key is a good key.

A14: \( \forall t \forall i \in \{\text{ENT}\} \neg L_{i,x} \land L_{i,k} \rightarrow L_{i,y}(s(x,k)) \)

Axiom 14 states that only the rightful owners of a shared secret key that secret and this implies that this is a good secret.

A15: (a) \( \exists x \exists i \in \{\text{ENT}\} (L_{i,x} \land \forall t', t < t \rightarrow \neg L_{i,y}(e(x,k))) \land \neg (\exists y (R(i,t,y) \land C(y,E(x,k)))) \rightarrow L_{i,y}(E(x,k))) \)

(b) \( \exists x \exists i \in \{\text{ENT}\} (L_{i,x} \land \forall t', t < t \rightarrow \neg L_{i,y}(D(x,k))) \land \neg (\exists y (R(i,t,y) \land C(y,D(x,k)))) \rightarrow L_{i,y}(D(x,k))) \)

A15 (a) states: If \( \Sigma \) knows a secret \( s(x,k) \) that it shares with \( \Psi \) (the secret can be fresh), and this secret is a good secret, and \( \Sigma \) receives a message containing \( s(x,k) \) at \( t \) that it did not send, then \( \Sigma \) knows that \( \Psi \) sent this message prior to \( t \).

A15 (b) states: If \( \Sigma \) knows the public key of \( \Psi (k_P) \) and message \( x \), and if \( \Sigma \) receives a message \( y \) containing \( e(x,k) \) then \( \Sigma \) knows that \( \Psi \) sent message \( y \) prior to \( t \).

**IV. FORMAL ANALYSIS OF BRESSON ET AL.’S PROTOCOL**

Bresson et al.’s key exchange protocol was discussed in section 2. In this section the CSN logic [4] is applied to the protocol to check whether any security weakness or flaws exist in its specifications.

**A. Goals of the protocol**

The goals of the key exchange protocol are defined as follows:

Goal 1 states that the Server \( S \) will obtain value \( y_1 \) where \( y_1 = g^{x_1} \mod p \) and a signed message from \( U_i \) containing value \( y_i \).

Goal 2 states that the low power node \( U_i \) will obtain a message from \( S \) containing the counter \( c \) and shared secret value \( K_i \).

Figure 3. Goals of Bresson et al.’s protocol
B. Initial assumptions

1: \( \forall i, \forall t, i \in \{\text{ENT}1\} (L_{i,r}(K_{i}) \land L_{i,s}(K_{i})) \)
2: \( B_{i,r} (\forall i, i \in \{\text{ENT}/U\}, \forall t, t0 \leq t \leq t1, \neg L_{i,s}(y_{i})) \)
3: \( K_{i,r} (\forall i, i \in \{\text{ENT}/S\}, t1 < t < t2, \neg L_{i,s}(K_{i} \land \neg L_{i,s}(K_{i} \land \neg L_{i,s}(G_{K}))) \)
4: \( L_{i,r}(K_{i}^{U_{i}}) \land K_{i,s}(\forall t, \forall i, i \in \{\text{ENT}/U\}, \neg L_{i,s}(K_{i}^{U_{i}})) \)
5: \( L_{i,s}(K_{i}^{U_{i}}) \land K_{i,s}(\forall t, \forall i, i \in \{\text{ENT}/S\}, \neg L_{i,s}(K_{i}^{U_{i}})) \)

Figure 4. Initial assumptions of Bresson’s protocol

Assumption 1 states that the public keys of \( U_{i} \) and \( S \) are known to all entities.
Assumption 2 refers to the fact that \( y_{i} \) is generated entirely by node \( U_{i} \), there is no timestamp or nonce utilized to establish the freshness of \( y_{i} \), therefore \( S \) only believes in the freshness of \( x \), as it has no knowledge of it.
Assumption 3 states that Server \( S \) generates the fresh group key \( K_{s} \), shared secret \( K_{e} \) and session key \( G_{K} \) and as such it knows that no entity has knowledge of \( K_{s} \), \( K_{e} \) and \( G_{K} \) before step 2 of the protocol.
Assumption 4 states that the private key of \( U_{i} \) \((K_{i}^{U_{i}})\) is known only to \( U_{i} \).
Assumption 5 states that the private key of server \( S \) is known only to \( S \).

C. Analysis of the message exchanges

The following messages are exchanged during the operation of Bresson et al’s key exchange protocol.

Step 1: \( U_{i} \rightarrow S, y_{i}, \text{Sign}(SK_{i}, y_{i}) \)

Step 2: \( S \rightarrow U_{i}: C, K_{i}^{U_{i}} \)

Rewriting each step in the language of the CSN logic we get:

Step 1:

\( K_{i,s} (R(S,t1,X) \land C(X, (y_{i}e(y_{i}, K_{i}^{U_{i}})))) \)

This states that \( S \) knows at time \( t1 \) that it will receive a message \( X \) from a participant node \( U_{i} \) where \( i \in \{1,2,\ldots,n\} \), and this message contains \( y_{i} \) and a signature of \( y_{i} \) using the secret key of \( U_{i} \), where \( y_{i} = g^{x_{i}} \mod p \) and \( x_{i} \) is a random value selected from \( Z_{q}^{*} \).

Applying Axiom A2:

\( R(S,t1,X) \land C(X, (y_{i}e(y_{i}, K_{i}^{U_{i}}))) \)

Applying Axiom A6 and Inference Rule R2:

\( L_{i,s}(X) \land K_{i,s} (\exists i, i \in \{\text{ENT}/S\}, \exists t, t < t1, S(i,t,X)) \land C(X, (y_{i}e(y_{i}, K_{i}^{U_{i}}))) \)

Applying Inference Rule R3:

\( K_{i,s} (\exists i, i \in \{\text{ENT}/S\}, \exists t, t < t1, S(i,t,X) \land C(X, (y_{i}e(y_{i}, K_{i}^{U_{i}}))) \)

Using Assumption 4 states that only \( U_{i} \) has knowledge of private key \( K_{i}^{U_{i}} \). This gives \( S \) the identity of the entity sending the signature:

\( K_{i,s} (\exists t, t < t1, S(U_{i}, t, X) \land C(X, (y_{i}e(y_{i}, K_{i}^{U_{i}})))) \)

Using assumption 2 which states that \( S \) has no knowledge of the freshness of \( y_{i} \), this assumption allows the above expression to be rewritten as an expression of belief rather than knowledge:

\( B_{i,s}(\exists t, t < t1, S(U_{i}, t, X) \land C(X, (y_{i}e(y_{i}, K_{i}^{U_{i}})))) : \text{Goal 1} \)

Only belief achieved, not knowledge

The goal of the timely arrival of the signature and \( y_{i} \) is not achieved. This is due to the fact that the signature and \( y_{i} \) could be compromised because of the fact that only trust and not knowledge in the freshness of \( y_{i} \) is established by assumption 2. Thereby leaving the scheme opens to some active attacks as presented by Nam et al in [2].

Step 2:

\( K_{i,s} (R(U_{i},t2,X) \land C(X, (c, K_{i}))) \)

This states that \( U_{i} \) knows at time \( t2 \) that it will receive a message which contains the counter \( c \), and shared secret \( K_{i} \).

Applying Axiom A2 to reduce the formula: \( R(U_{i},t2,X) \land C(X, (c, K_{i})) \)

Applying Axiom A6 and Inference Rule R2:

\( L_{U_{i}}(X) \land K_{U_{i}} (\exists i, i \in \{\text{ENT}/U\}, \exists t, t < t2, S(i,t,X)) \land C(X, (c, K_{i})) \)

Applying Inference Rule R3:

\( K_{U_{i}} (\exists i, i \in \{\text{ENT}/U\}, \exists t, t < t2, S(i,t,X)) \land C(X, (c, K_{i})) \)

Using Assumption 3 states that the counter \( c \) and shared secret key \( K_{i} \) are only known to Server \( S \) and no other entity has knowledge of \( c \) and \( K_{i} \) before step 2 of the protocol.

This gives \( U_{i} \) the identity of the entity sending the counter and shared secret key. Using Assumption 3 also establishes the time when \( c \) and \( K_{i} \) are generated:

\( K_{U_{i}} (\exists i, i \in \{\text{ENT}/U\}, t1 < t < t2, S(S,t,X) \land C(X, (c, K_{i}))) \) \quad (Bres1)

Given that \( c \) and \( K_{i} \) are not known there is no way for clients to check whether the values \( c \) and \( K_{i} \) are indeed from the authentic server \( S \) or not. Neither certificate nor authentication algorithm is utilized to authenticate Server \( S \). Another initial assumption may be introduced:

\( B_{U_{i},c} (\exists t, t < t2, S(S,t,X) \land C(X, (c, K_{i}))) \) \quad (Initial assumption 6)

This new assumption allows expression Bres1 to be rewritten as an expression of belief rather than knowledge:

\( B_{U_{i},c} (\exists t, t < t2, S(S,t,X) \land C(X, (c, K_{i}))) \)

Only belief achieved, not knowledge

This means that Goal 2 should be an expression of belief rather than knowledge. This presents a possible weakness that will allow a rogue entity to present itself as Server \( S \).

D. Summary

In summary, the Bresson et al’s key exchange protocol has many weakness associated with it. The Lack of data freshness and sufficient entity authentication are the main weaknesses highlighted here by the formal analysis. Nam et al in [2] discovered the weakness in Goal 1 and applied an attack to impersonate a client in the group and replay the previous message and signature to pass the authentication in Server \( S \). It then was able to gain the information from the server necessary to compute the group session key. This problem has been highlighted in key-exchange protocols in the past due to mutual key agreement not being implemented [3].
weakness identified in goal 2 is new, which is not identified in [2]. This security weakness will allow the adversary to masquerade as Server S and make the participant clients to share the session key given by adversary.

V. FORMAL ANALYSIS OF NAM ET AL.’S PROTOCOL

A. Goals of the Nam et al.’s protocol

The goals of the protocol are defined as follows:
Goal 1 states that the Server S knows it will obtain value $y_i$ where $y_i = g^x \mod p$ and a signed message from $U_i$ containing value $y_i$.
Goal 2 states that the low power node $U_i$ will obtain a message from $S$ containing the counter $c$ and shared secret value $K$.

\[
\text{Goal 1 : } K_{S,t} = C(t, t < t_1, S(U_i, t, X) \land C(X, (y_i, e(y_i, K_{U_i}))), \text{ for all } i \in n
\]

\[
\text{Goal 2 : } K_{U,i} = C(t, t < t_2, S(S, t, X) \land C(X, (c, K_{S}, PK_s, e((c, K_{S}, PK_s), K_{U_i}))), \text{ for all } i \in n
\]

Figure 5. Goals of the improved protocol

B. Initial assumptions of the Nam et al.’s protocol

Assumption 1 states that the public key of $U_i$ is known to all entities.
Assumption 2 refers to the fact that $y_i$ is generated entirely by node $U_i$, there is no timestamp or nonce utilized to establish the freshness of $y_i$, therefore $S$ only believes in the freshness of $x$, as it has no knowledge of it.
Assumption 3 states that Server S generates the fresh group key $K$, shared secret $K_s$ and session key $GK$ and as such it knows that no entity has knowledge of $K$, $K_s$ and $GK$ before step 2 of the protocol.
Assumption 4 states that the private key of $U_i$ ($K_{U_i}^{-1}$) is known only to $U_i$.
Assumption 5 states that Server S generates the new public key $K_S$ and as such it knows that no entity has knowledge of $K_S$ before step 2 of the protocol.
Assumption 6 refers to the fact that the public key $K_s$ is generated entirely by Server $S$. Therefore $U_i$ only believes in $K_S$ as it has no knowledge of it.

C. Message exchanges of the Nam et al.’s protocol

The following messages are exchanged during the operation of Nam et al’s key exchange protocol.

Step 1: $U_i \rightarrow S: y_i, \text{Sign}(_{SK_i}(y_i))$

Step 2: $S \rightarrow U_i: c, K_s, \text{Sign}(_{SK_s}(c||K_s||PK_s))$

Rewriting each step in the language of the CSN logic we get:

Step 1:
$K_{S,t} (R(S, t_1, X) \land C(X, (y_i, e(y_i, K_{U_i}))))$

This states that $S$ knows at time $t_1$ that it will receive a message $X$ from a participant node $U_i$ where $i \in \{1, 2, \ldots, n\}$, and this message contains $y_i$ and a signature of $y_i$ using the secret key of $U_i$, where $y_i = g^x \mod p$ and $x_i$ is a random value selected from $g^{Z_s}$.

Applying Axiom A2:
$R(S, t_1, X) \land C(X, (y_i, e(y_i, K_{U_i})))$

Applying Axiom A6 and Inference Rule R2:
$L_{S,t_1} X \land K_{S,t_2} (\exists i, i \in \{\text{ENT}/S\}, \exists t, t < t_1, S(t, t, X) \land C(X, (y_i, e(y_i, K_{U_i})))$

Applying Inference Rule R3:
$K_{S,t_1} (\exists i, i \in \{\text{ENT}/S\}, \exists t, t < t_1, S(t, t, X) \land C(X, (y_i, e(y_i, K_{U_i}))))$

Using Assumption 4 states that only $U_i$ has knowledge of private key $K_{U_i}^{-1}$. This gives $S$ the identity of the entity sending the signature:
$K_{S,t_1} (\exists i, t < t_1, S(U_i, t, X) \land C(X, (y_i, e(y_i, K_{U_i}))))$

Using assumption 2 which states that $S$ has no knowledge of the freshness of $y_i$, this assumption allows the above expression to be rewritten as an expression of belief rather than knowledge:
$B_{S,t_1}(\exists i, t < t_1, S(U_i, t, X) \land C(X, (y_i, e(y_i, K_{U_i})))) : \text{Goal 1}$

Only belief achieved, not knowledge

The goal of the timely arrival of the signature and $y_i$ is not achieved. This is due to the fact that the signature and $y_i$ could be compromised because of the fact that only trust and not knowledge in the freshness of $y_i$ is established by assumption 2.

Step 2:
$K_{U,t_2} (R(U_i, t_2, X) \land C(X, (c, K_s, K_e \land (c, K_s, K_e), K_{U_i}^{-1})))$

This states that $U_i$ knows at time $t_2$ that it will receive a message which contains the counter $c$, and shared secret $K_i$.

Applying Axiom A2 to reduce the formula: $R(U_i, t_2, X) \land C(X, (c, K_i, K_e \land (c, K_s, K_e), K_{U_i}^{-1})))$

Applying Axiom A6 and Inference Rule R2:
$L_{U,t_2} X \land K_{U,t_2} (\exists i, i \in \{\text{ENT}/U\}, \exists t, t < t_2, S(t, t, X) \land C(X, (c, K_i, K_e \land (c, K_s, K_e), K_{U_i}^{-1})))$

Applying Inference Rule R3:
$K_{U,t_2} (\exists i, i \in \{\text{ENT}/U\}, \exists t, t < t_2, S(t, t, X) \land C(X, (c, K_i, K_e \land (c, K_s, K_e), K_{U_i}^{-1})))$

Using Assumption 3 states that the counter $c$ and shared secret key $K_i$ are only known to Server $S$ and no other entity has knowledge of $c$ and $K_i$ before step 2 of the protocol.
This gives \( U_i \) the identity of the entity sending the counter and shared secret key. Using Assumption 3 also establishes the time when \( c \) and \( K_i \) are generated:

\[
K_{i\text{t}}\text{is}\ (\exists t, t'=t+2, \ S(S, t, X) \land C(X, (c, K_i, K_s) \land (c, K_i, K_s', K_s'))) \]

Using assumption 6 which states that \( U_i \) has no knowledge of the new generated public key \( K_s \), this assumption allows the above expression to be rewritten as expression of belief rather than knowledge:

\[
B_{a,b}((\exists t, t'=t+2, \ S(S, t, X) \land C(X, (c, K_i, K_s) \land (c, K_i, K_s'), K_s')))\]

**Only belief achieved, not knowledge**

This presents a possible weakness that will allow a rogue entity to present itself as Server \( S \).

D. **Attacks on the Nam et al. ’s protocol**

The formal verification of Nam et al.’s group key exchange protocol shows there are security flaws are detected, and one of them allows a rogue entity to masquerade as Server \( S \) and share the session key with clients.

To show that the modified protocol is insecure, an impersonating attack is applied as follows:

1. In the first step of the protocol, the adversary \( A \) eavesdrops and records the transmitted messages \( (y_i, \sigma_i) \) from \( U_i \) for all \( i \in n \).
2. Adversary masquerades as Server \( S \) and generate \( PKs = g^{SKs'} \), where \( SKs' \in \mathbb{G}_q^* \); Adversary computes:

\[
\alpha'_i = y_i^{SKs'},
\]

Adversary gives \( K' \) a random number and computes:

\[
Ki' = K' \oplus \text{H}(c) \oplus \alpha'_i
\]

for all \( i \in n \), with the assumption that hash function \( \text{H}(\cdot) \) is exposed to \( A \).

The adversary \( A \) signs the message \( \{c, \{Ki' \}_{c=\sigma_i} \} \) use the private key \( SKs' \) to obtain signature \( \sigma_i' \) and broadcasts \( \{c, \{Ki' \}_{c=\sigma_i} \} \) to the clients.

3. Upon receiving \( \{c, \{Ki' \}_{c=\sigma_i} \} \) by each client \( U_i \), it verifies the signature \( \sigma_i' \) using the public key \( PKs \) given by the adversary, computes:

\[
\alpha'_i = y_i^{SKs'},
\]

and then get the shared secret value produced by adversary \( K' \) as:

\[
K' = K' \oplus \text{H}(c) \oplus \alpha'_i
\]

Finally, both adversary and clients shared the same session key as:

\[
GK = H(K_i || U_i || S)
\]

Consequently, implicit key authentication is not guaranteed in the modified protocol, as the adversary can masquerade as the server. This weakness also make the protocol vulnerable to reply attacks as such the adversary repeat the messages \( \{c, \{Ki' \}_{c=\sigma_i} \} \) in the previous run of the protocol, clients are impossible to detect this attack as they has no knowledge of the public key \( PKs \) before step 2 of the protocol.

VI. CONCLUSIONS

In this paper two group key exchange protocols for low power mobile network were discussed, the Bresson et al.’s and Nam et al.’s protocols. The verification of Bresson et al.’s protocol presented in this paper highlighted a number of weaknesses in the protocol. The analysis of the Nam et al.’s modified protocol shows that the improved protocol doesn’t fix the problem and is still vulnerable to active attacks.

ACKNOWLEDGMENT

The authors wish to thank the following for their financial support:

- SFI Research Frontiers Programme grant number 05/RFP/CMS0071.
- The Embark Initiative, who fund this research through the Irish Research Council for Science, Engineering and Technology (IRCSET) postgraduate Research Scholarship Scheme.

REFERENCES


