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 securtiy, 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 $U = \{U1, U2, \ldots, Un\}$ 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

θ: 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 $PK_i = g^{SKi} \bmod p$.

SK_S - the powerful node S's secret key in Z_q^*.

PK_s - the powerful node S's public key such that $PK_s = g^{SKs} \bmod p$.

$H()$ - a one-way hash function H with arbitrary length input and a fixed length output[6], i.e. $\{0,1\}^* \rightarrow \{0,1\}^k$, where k is the length of output.
$Sign(SK_i, m)$ - the signing algorithm based on ElGamal[7] or DSA[8] schemes under U_i's secret key SK_i and the signed message m.
GK: established session key that the participants shared with the server.
K: The shared secret value.
σ_i : denotes the signature algorithm for participating clients of message y_i, $\sigma_i = Sign(SK_i, y_i)$, for $i \in n$
σ_s: denotes the signature algorithm for server of message $c||K_i||PK_s$, $\sigma_s = Sign(SKs, c||K_i||PK_s)$ for $i \in n$

B. The Bresson et al.'s group key exchange protocol

The Bresson et al's group key exchange protocol consists of three algorithms: the setup algorithm GKE.Setup, the remove algorithm GKE.Remove, and the join algorithm GKE.Join. The main GKE.Setup algorithm allows a set of mobile users and a wireless gateway (also called server) to agree on a session key. The other algorithms of the protocol aim to efficiently handle dynamic membership changes of clients in the wireless region.

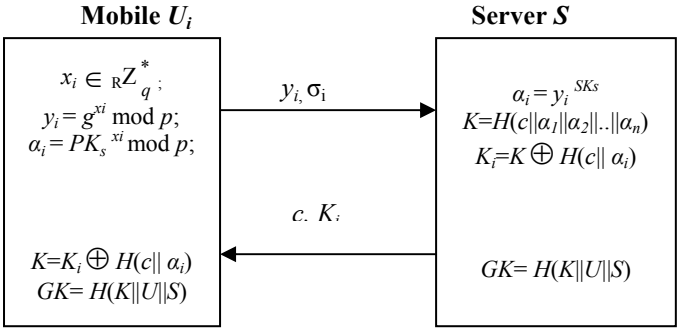

Figure 1. Bresson et al.'s group key exchange protocol

The GKE.Setup algorithm

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 *Ui* chooses a random $x_i \in {}_R Z_q^*$ and computes

$$y_i = g^{xi}$$
$$\alpha_i = PK_s{}^{xi}$$

Client U_i then signs y_i to obtain signature σ_i and sends (y_i, σ_i) to the server *S*.

Step 2: For all $i \in n$, The server *S* verifies the signature σ_i , and computes

$$\alpha_i = y_i{}^{SKs}$$

S then initializes the counter *c* to 0, and computes the shared secret value

$$K=H(C||\alpha_1||\alpha_2||..||\alpha_n)$$

and $K_i=K \oplus H(C||\alpha_i)$

The server *S* sends to each clients Ui the values (c, K_i).
Upon receiving *c* and K_i, client U_i recovers the shared secret value *K* as

$$K=K_i \oplus H(c||\alpha_i)$$

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

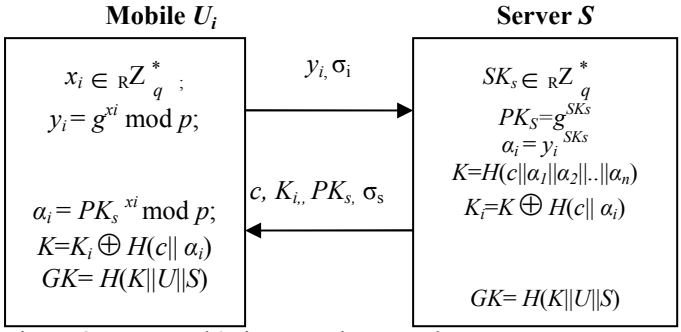

Figure 2. Nam et al.'s improved protocol

The changes made were as follows:

1. Mobile clients will not know public key of Server *S* until step 2; the computation of $\alpha_i = PK_s{}^{xi} \bmod p$ is shifted from step 1 to step 2 on the client side.
2. Server *S* generate its signing private/public keys *(SKs, PKs)* after step 1, and sign the message $c||K_i||PK_s$ to obtain signature σ_s, and broadcasts $(c, (K_i)_{i \in n,} PK_{s,} \sigma_s)$ to the clients.

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 theorems.

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
- **Φ**, an arbitrary statement
- **Σ** and **Ψ**, arbitrary entities
- **i** and **j**, individual entities
- **ENT**, the set of all possible entities
- **k**, a cryptographic key. In particular, $\mathbf{k_\Sigma}$ is the public key of entity **Σ** and $\mathbf{k_\Sigma^{-1}}$ is the corresponding private key of entity **Σ**
- **t, t', t''...** represents moments in time.
- **t1, t2, t3…** represents time after each step of a protocol. For example, t1 represents time after step 1 of a protocol has completed
- **e(x,$\mathbf{k_\Sigma}$),** encryption function, encryption of x using key $\mathbf{k_\Sigma}$
- **d(x,$\mathbf{k_\Sigma^{-1}}$),** decryption function, decryption of x using key $\mathbf{k_\Sigma^{-1}}$
- $\mathbf{ks_{(\Sigma,\Psi)}}$ Shared secret key for entities **Σ** and **Ψ**.
- $\mathbf{KS_{\{\Sigma,\Psi\}}}$ Set of good shared keys for entities **Σ** and **Ψ**.
- $\mathbf{ss_{(\Sigma,\Psi)}}$ Shared secret for entities **Σ** and **Ψ** (secret can be fresh).
- $\mathbf{SS_{\{\Sigma,\Psi\}}}$ Set of good shared secrets for entities **Σ** and **Ψ**.
- **E(x, $\mathbf{ks_{(\Sigma,\Psi)}}$),** Encryption of plaintext message **x** using the shared secret key of entities **Σ** and **Ψ**.
- **D(x, $\mathbf{ks_{(\Sigma,\Psi)}}$),** Decryption of ciphertext message **x** using the shared secret key of entities **Σ** and **Ψ**.
- ***K***, propositional knowledge operator (true or false evaluation) of Hintikka. $\boldsymbol{K_{\Sigma,t}\Phi}$ means **Σ** knows statement **Φ** at time **t**.
- ***L***, knowledge predicate (assigns an object a property). $\boldsymbol{L_{\Sigma,t}x}$ means **Σ** knows and can reproduce object **x** at time **t**.
- ***B***, belief operator. $\boldsymbol{B_{\Sigma,t}\Phi}$ means **Σ** believes at time **t** that statement **Φ** 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(Σ,t,x)** means **Σ** sends message **x** at time **t**.
- ***R***, reception operator. **R(Σ,t,x)** means **Σ** receives message **x** at time **t**.
- **A**, authentication Operator. **A(Σ,t,Ψ)** means that **Σ** authenticates **Ψ** at time **t**.

The language includes the classical logical connectives of conjunction (∧), disjunction (∨), complementation (¬) and material implication (→). The symbols ∀ and ∃ denote universal and existential quantification respectively. ∈ indicates membership of a set and / denotes set exclusion. The symbol ⊢ 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 ⊢ p and ⊢ (p→ q) infer ⊢ q

R2: (a) From ⊢ p infer ⊢ $K_{\Sigma,t}$p;
(b) From ⊢ p infer ⊢ $B_{\Sigma,t}$p

R1 is the *Modus Ponens* and states that if schema *p* can be deduced and (*p* → *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 ∧ q) infer p

R4: From p and q infer (p ∧ 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: $\exists t \exists p \exists q (K_{\Sigma,t}p \wedge K_{\Sigma,t}(p \rightarrow q) \rightarrow K_{\Sigma,t}q)$

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

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) $\exists t \exists x \exists i, i \in \{ENT\}(L_{i,t}x \rightarrow \forall t', t' \geq t\ L_{i,t'}x)$;

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

Axioms A3 (a) and A3(b) asserts that knowledge, once gained, cannot be lost.

A4: $\exists t \exists x \exists y(\exists i,i \in \{ENT\} L_{i,t} y \wedge C(y,x) \rightarrow \exists j,j \in \{ENT\} L_{j,t} x)$

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: $\exists t \exists x(S(\Sigma,t,x) \rightarrow L_{\Sigma,t} x \wedge \exists i,i \in \{ENT/\Sigma\} \exists t',t'>t\ R(i,t',x))$

The emission axiom A5 states that: if Σ sends a message *x* at time *t*, then Σ knows *x* at time *t* and some entity *i* other than Σ will receive *x* at time *t'* subsequent to *t*.

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

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

A7: **(a)** $\exists t \exists x \exists i,i \in \{ENT\}(L_{i,t} x \wedge L_{i,t} k_{\Sigma} \rightarrow L_{i,t}(e(x,k_{\Sigma})))$

(b) $\exists t \exists x \exists i,i \in \{ENT\}(L_{i,t} x \wedge L_{i,t} k_{\Sigma}^{-1} \rightarrow L_{i,t}(d(x,k_{\Sigma}^{-1})))$

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)** $\exists t \exists x \exists i,i \in \{ENT\}(\neg L_{i,t} k_{\Sigma} \wedge \forall t',t'<t \ \neg L_{i,t'}(e(x,k_{\Sigma})) \wedge \neg (\exists y(R(i,t,y) \wedge C(y,e(x,k_{\Sigma})))) \rightarrow \neg L_{i,t}(e(x,k_{\Sigma})))$

(b) $\exists t \exists x \exists i,i \in \{ENT\}(\neg L_{i,t} k_{\Sigma}^{-1} \wedge \forall t',t'<t \ \neg L_{i,t'}(d(x,k_{\Sigma}^{-1})) \wedge \neg(\exists y(R(i,t,y) \wedge C(y,d(x,k_{\Sigma}^{-1})))) \rightarrow \neg L_{i,t}(d(x,k_{\Sigma}^{-1})))$

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_{Σ}) and also does not receive e(x, k_{Σ}) at *t* in a message, then the entity cannot know e(x, k_{Σ}) at time *t*. Axiom A8b makes a similar statement for the decryption of a message *x* without knowledge of the decryption key.

A9: $\forall t(\forall i,i \in \{ENT\} L_{i,t} k_i^{-1} \wedge \forall j,j \in \{ENT/i\} \neg L_{j,t} k_i^{-1})$

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

A10: $\exists t \exists x(\exists i,\ i \in \{ENT\} L_{i,t} d(x,k_{\Sigma}^{-1}) \rightarrow L_{\Sigma,t} x)$

Axiom A10 states that if an entity knows and can reproduce d(x, k_{Σ}^{-1}) and k_{Σ} at time *t* then it knows and can reproduce *x*, this implies that this entity knows at time *t* that Σ knows and can reproduce *x* prior to *t*.

A11:

(a) $\exists t \exists x \exists i,i \in \{ENT\}\ (L_{i,t} x \wedge L_{i,t} ks_{(\Sigma,\Psi)} \rightarrow L_{i,t}(E(x, ks_{(\Sigma,\Psi)})))$

(b) $\exists t \exists x \exists i,i \in \{ENT\}\ (L_{i,t} y \wedge C(y,E(x, ks_{(\Sigma,\Psi)})) \wedge L_{i,t} ks_{(\Sigma,\Psi)} \rightarrow L_{i,t}(D(x, ks_{(\Sigma,\Psi)})))$

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

A12:

(a) $\exists t \exists x \exists i,i \in \{ENT\}(\neg L_{i,t}\ ks_{(\Sigma,\Psi)} \wedge \forall t',t' < t, \neg L_{i,t'}(E(x, ks_{(\Sigma,\Psi)})) \wedge \neg(\exists y(R(i,t,y) \wedge C(y,E(x, ks_{(\Sigma,\Psi)})))) \rightarrow \neg L_{i,t}(E(x, ks_{(\Sigma,\Psi)})))$

(b) $\exists t \exists x \exists i,i \in \{ENT\}(\neg L_{i,t}\ ks_{(\Sigma,\Psi)} \wedge \forall t',t' < t, \neg\ L_{i,t'}(D(x, ks_{(\Sigma,\Psi)})) \wedge \neg(\exists y(R(i,t,y) \wedge C(y,D(x, ks_{(\Sigma,\Psi)})))) \rightarrow \neg L_{i,t}(D(x, ks_{(\Sigma,\Psi)})))$

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

A13: $\forall t((\forall i,i \in \{ENT/\Sigma,\Psi\} \neg L_{i,t} ks_{(\Sigma,\Psi)} \wedge \exists j,j \in \{\Sigma,\Psi\} L_{j,t} ks_{(\Sigma,\Psi)}) \rightarrow ks_{(\Sigma,\Psi)} \in \{KS_{\{\Sigma,\Psi\}}\}))$

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: $\forall t((\forall i,i \in \{ENT/\Sigma,\Psi\} \neg L_{i,t} ss_{(\Sigma,\Psi)} \wedge \exists j,j \in \{\Sigma,\Psi\} L_{j,t} ss_{(\Sigma,\Psi)}) \rightarrow ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\}))$

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

A15:

(a) $\exists x \exists t(A(\Sigma,t,\Psi) \rightarrow (L_{\Sigma,t} ss_{(\Sigma,\Psi)} \wedge ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\} \wedge R(\Sigma,t,x) \wedge C(x,ss_{(\Sigma,\Psi)}) \wedge \forall t',t'<t \neg S(\Sigma,t',x)) \rightarrow K_{\Sigma,t}(S(\Psi,t',x))))$

(b) $\exists x \exists t(A(\Sigma,t,\Psi) \rightarrow (L_{\Sigma,t} k_{\Psi} \wedge L_{\Sigma,t} x \wedge R(\Sigma,t,y) \wedge C(y,e(x,k_{\Psi}^{-1}))) \rightarrow (\forall t',\ t' < t,\ K_{\Sigma,t}(S(\Psi,t',y))))$

A15 (a) states: If Σ knows a secret $ss_{(\Sigma,\Psi)}$ that it shares with Ψ (the secret can be fresh), and this secret is a good secret, and Σ receives a message containing $ss_{(\Sigma,\Psi)}$ at *t* that it did not send, then Σ knows that Ψ sent this message prior to *t*.

A15 (b) states: If Σ knows the public key of Ψ (k_{Ψ}) and message x, and if Σ receives a message y containing e(x, k_{Ψ}^{-1}) then Σ knows that Ψ 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 knows it will obtain value y_i, where $y_i = g^{xi} \bmod p$ 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

Goal 1 : $K_{S,\,t1}(\exists\, t,\ t<t1,\ S\ (U_i,\ t,\ X) \wedge C(X,\ (y_i,\ e(y_i,\ K_U^{-1}))),$ for all $i \in n$

Goal 2 : $K_{Ui,\,t2}(\exists\, t,\ t1<t<t2,\ S(S,\ t,\ X) \wedge C(X,\ (c,\ K_i))),$ for all $i \in n$

Figure 3. Goals of Bresson et al.'s protocol

B. Initial assumptions

1: $\forall$ i, $\forall$ t, i$\in$ {ENT}($L_{i,t}K_{Ui} \wedge L_{i,t}K_S$)
2 : $B_{S,\ t0}$($\forall$ i, i$\in$ {ENT/U_i}, $\forall$ t, t0<t<t1, $\neg L_{i,t1}y_i$)
3 : $K_{s,\ t0}$ ($\forall$ i, i$\in$ {ENT/S}, t1<t<t2, $\neg L_{i,\ t}K \wedge \neg L_{i,\ t}K_i \wedge \neg L_{i,t}GK$)
4 : $L_{Ui,\ t0}\ K_{Ui}^{-1} \wedge K_{Ui,t0}$($\forall$ t, $\forall$ i, i$\in$ {ENT/U_i}, $\neg L_{i,t}K_{Ui}^{-1}$)
5: $L_{S,t0}\ K_S^{-1} \wedge K_{S,t0}$($\forall$ t, $\forall$ i, i$\in$ {ENT/S}, $\neg L_{i,t}K_S^{-1}$)

Figure 4. Initial assumptions of Bresson's protocol

Assumption 1 states that the public keys of U_i and S are know 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_i, and session key GK and as such it knows that no entity has knowledge of K, K_i and GK before step2 of the protocol.
Assumption 4 states that the private key of U_i (K_{Ui}^{-1}) is known only to U_i.
Assumption 5 states that the private key of server S is know 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, $Sign(SKi, y_i)$,

Step 2: $S \rightarrow U_i$: C, K_i;

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

Step 1:
$K_{S,t1}$ (R(S,t1,X) $\wedge$ C(X, (y_i,e(y_i, K_{Ui}^{-1}))))
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,…,n}, and this message contains y_i and a signature of y_i using the secret key of U_i, where $y_i = g^{xi} \bmod p$ and x_i is a random value selected from ${}_RZ_q^*$.

Applying Axiom A2:
R(S,t1,X) $\wedge$ C(X, (y_i,e(y_i, K_{Ui}^{-1})))
Applying Axiom A6 and Inference Rule R2:
$L_{S,t1}$ X $\wedge K_{S,t1}$ ($\exists$ i, i$\in$ {ENT/S}, $\exists$ t, t<t1, S(i,t, X)) $\wedge$C(X, (y_i,e(y_i, K_{Ui}^{-1})))
Applying Inference Rule R3:
$K_{S,t1}$ ($\exists$ i, i$\in$ {ENT/S}, $\exists$ t, t<t1, S(i,t, X)) $\wedge$C(X, (y_i,e(y_i, K_{Ui}^{-1}))
Using Assumption 4 states that only U_i has knowledge of private key K_{Ui}^{-1}. This gives S the identity of the entity sending the signature:
$K_{S,\ t1}$($\exists$ t, t<t1, S (U_i, t, X) $\wedge$ C(X, (y_i, e(y_i, K_U^{-1}))))
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:
$B_{S,\ t1}$($\exists$ t, t<t1, S (U_i, t, X) $\wedge$ C(X, (y_i, e(y_i, K_U^{-1})))) : !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:
$K_{Ui,t2}$ (R(U_i,t2,X) $\wedge$ 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) $\wedge$ C(X, (c, K_i))
Applying Axiom A6 and Inference Rule R2:
$L_{Ui,t2}$X$\wedge K_{Ui,t2}$($\exists$ i, i$\in$ {ENT/U_i}, $\exists$ t, t<t2, S(i,t,X)) $\wedge$$C$(X, (c, K_i))
Applying Inference Rule R3:
$K_{Ui,t2}$($\exists$ i, i$\in$ {ENT/U_i}, $\exists$ t, t<t2, S(i,t,X)) $\wedge$$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:
$K_{Ui,\ t2}$($\exists$ t, t1<t<t2, S(S, t, X) $\wedge$ C(X, (c, K_i))) **(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:
$B_{ui,t0}$($\exists$ t, t<t2, S(S, t,X) $\wedge$ C(X,(c,K_i))) :Initial assumption 6
This new assumption allows expression **Bres1** to be rewritten as an expression of belief rather than knowledge:
$B_{Ui,\ t2}$($\exists$ t, t1<t<t2, S(S, t, X) $\wedge$ C(X, (c, K_i))) :!Goal 2
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]. The

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 $y_i = g^{xi} \bmod p$ 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

Goal 1 : $K_{S,\,t1}(\exists\, t, t<t1, S(U_i, t, X) \wedge C(X, (y_i, e(y_{i,} K_U^{-1}))))$, for all $i \in n$

Goal 2 : $K_{Ui,\,t2}(\exists\, t, t1<t<t2, S(S, t, X) \wedge C(X, (c, K_{i,,} PK_{s,} e((c, K_{i,,} PK_s), K_s^{-1}))))$, for all $i \in n$

Figure 5. Goals of the improved protocol

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

1: $\forall$ i, $\forall$ t, $i \in \{ENT\}(L_{i,t}K_{Ui})$
2 : $B_{S,\,t0}(\forall\, i, i \in \{ENT/U_i\}, \forall\, t, t0<t<t1, \neg L_{i,t1}y_i)$
3 : $K_{s,\,t0}(\forall\, i, i \in \{ENT/S\}, t1<t<t2, \neg L_{i,\,t}K \wedge \neg L_{i,\,t}K_i \wedge \neg L_{i,t}GK)$
4 : $L_{Ui,\,t0}K_{Ui}^{-1} \wedge K_{Ui,t0}(\forall\, t, \forall\, i, i \in \{ENT/U_i\}, \neg L_{i,t}K_{Ui}^{-1})$
5: $L_{S,t0}K_S \wedge K_{S,t0}(\forall\, i, i \in \{ENT/S\}, \forall\, t, t<t2, \neg L_{i,t}K_S)$
6:$B_{Ui,t0}(\forall\, i, i \in \{ENT/S\}, \forall\, t, t1<t<t2, \neg L_{i,t}K_S)$

Figure 6. Initial assumptions of the improved 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_i, and session key *GK* and as such it knows that no entity has knowledge of *K*, K_i and *GK* before step2 of the protocol.
Assumption 4 states that the private key of U_i, (K_{Ui}^{-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 *Ui* 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: $U_i \rightarrow S$: $y_{i,}$ $Sign(SKi, y_i)$,

Step 2: $S \rightarrow U_i$: c, K_i, PK_s, $Sign(SKs, c||K_i||PK_s)$

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

Step 1:
$K_{S,t1}(R(S,t1,X) \wedge C(X, (y_i, e(y_i, K_{Ui}^{-1}))))$
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 $y_i = g^{xi} \bmod p$ and x_i is a random value selected from ${}_RZ_q^*$.

Applying Axiom A2:
$R(S,t1,X) \wedge C(X, (y_i, e(y_i, K_{Ui}^{-1})))$
Applying Axiom A6 and Inference Rule R2:
$L_{S,t1}\, X \wedge K_{S,t1}(\exists\, i, i \in \{ENT/S\}, \exists\, t, t<t1, S(i,t, X)) \wedge C(X, (y_i, e(y_i, K_{Ui}^{-1})))$
Applying Inference Rule R3:
$K_{S,t1}(\exists\, i, i \in \{ENT/S\}, \exists\, t, t<t1, S(i,t, X)) \wedge C(X, (y_i, e(y_i, K_{Ui}^{-1}))$
Using Assumption 4 states that only U_i has knowledge of private key K_{Ui}^{-1}. This gives S the identity of the entity sending the signature:
$K_{S,\,t1}(\exists\, t, t<t1, S(U_i, t, X) \wedge C(X, (y_i, e(y_{i,} K_U^{-1}))))$
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:
$B_{S,\,t1}(\exists\, t, t<t1, S(U_i, t, X) \wedge C(X, (y_i, e(y_{i,} K_U^{-1})))) : !Goal\ 1$

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:
$K_{Ui,t2}(R(U_i,t2,X) \wedge C(X, (c, K_{i,,} K_{s,} e((c, K_{i,,} K_s), K_s^{-1}))))$
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: $R(U_i,t2,X) \wedge C(X, (c, K_{i,,} K_{s,} e((c, K_{i,,} K_s), K_s^{-1})))$
Applying Axiom A6 and Inference Rule R2:
$L_{Ui,t2}X \wedge K_{Ui,t2}(\exists\, i, i \in \{ENT/U_i\}, \exists\, t, t<t2, S(i,t,X)) \wedge C(X, (c, K_{i,,} K_{s,} e((c, K_{i,,} K_s), K_s^{-1})))$
Applying Inference Rule R3:
$K_{Ui,t2}(\exists\, i, i \in \{ENT/U_i\}, \exists\, t, t<t2, S(i,t,X)) \wedge C(X, (c, K_{i,,,} K_{s,} e((c, K_{i,,} K_s), K_s^{-1})))$
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:

$K_{Ui,\ t2}(\exists\ t,\ t1<t<t2,\ S(S,\ t,\ X) \wedge C(X,\ (c,\ K_{i,,}\ K_{s,}\ e\ ((c,\ K_{i,,}\ K_s),\ K_s^{-1}\ ))))$

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:

$B_{ui,to}(\exists\ t,\ t<t2,\ S(S,\ t,X) \wedge C(X,\ (c,\ K_{i,,}\ K_{s,}\ e\ ((c,\ K_{i,,}\ K_s),\ K_s^{-1}\ ))))$:!Goal 2

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, σ_i) from *Ui* for all $i \in n$.
2. Adversary masquerades as Server *S* and generate $PKs' = g^{SKs'}$, where $SKs' \in {}_RZ_q^*$; Adversary computes:

   $$\alpha_i' = y_i^{SKs'}$$

   Adversary gives K' a random number and computes:

   $Ki' = K' \oplus H(c \| \alpha_i')$ for all $i \in n$,

   with the assumption that hash function *H()* is exposed to A

   The adversary A signs the message $c||\{Ki'\}_{i\in n}||PKs'$ use the private key SKs' to obtain signature σ_s' and broadcasts $\{c, \{Ki'\}_{i\in n}, PKs', \sigma_s'\}$ to the clients.
3. upon receiving $\{c, \{Ki'\}_{i\in n}, PKs', \sigma_s'\}$, each client Ui verifies the signature σ_s' 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

   $$K' = K' \oplus H(c \| \alpha_i')$$

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

   $$GK = H(K||U||S)$$

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 $\{c, \{K_i\}_{i\in n}, PKs, \sigma_s\}$ 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.' 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

[1] E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval, Mutual authentication and group key agreement for low-power mobile devices," in Proc. of the 5th IFIP-TC6 International Conference on Mobile and Wireless Communications Networks (MWCN'03), 2003, pp. 59

[2] Nam, J., Kim, S. and Won, D. (2005) A weakness in the Bresson-Chevassut-Essiari-Pointcheval's group key agreement scheme for low-power mobile devices. IEEE Commun. Lett., 9, 429–431.

[3] Newe, T., Coffey, T., 2003. "On the Logical Verification of Key Exchange Protocols for Mobile Communications". Proceedings of the 7th WSEAS International Conference on COMMUNICATIONS, Corfu Island, Greece, July 7-10, 2003. ISBN:960-8052-82-3.

[4]Newe, T. and Coffey, T. (2003). Formal Verification Logic for Hybrid Security Protocols. *International Journal of Computer Systems Science and Engineering,* Vol. 18 no 1, pp. 17-25.

[5] Newe, T., Coffey, T., (2003). Security Protocols for 2G and 3G Wireless Communications. ACM ISICT 03, Dublin, Ireland, September 24-26, 2003. pp 348-353.

[6]NIST/NSA FIPS 180-2 (2005) Secure Hash Standard (SHS). NIST/NSA, Gaithersburg, MD, USA

[7] ELGamal, T. (1985) A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory, 31, 469–472.

[8] NIST (1992)The Digital Signature Standard (DSA). Commun. ACM, 35, 36–40.