Analysing the Security of a Non-repudiation Communication Protocol with Mandatory Proof of Receipt

TOM COFFEY, PUNEET SAIDHA, PETER BURROWS
Data Communication Security Laboratory
University of Limerick
Ireland
tom.coffey@ul.ie, peter.burrows@ul.ie

Abstract: **-** Non-repudiation is a security service concerned with preventing a denial by one of the principals involved in a communication of having participated in all or part of the communication. Principals exchange evidence information, which proves their participation in the communication. The evidence is digitally signed, making it non-forgeable and undeniable.
In this paper, a non-repudiation communication protocol with mandatory proof of receipt is formally analysed. The verification technique used in the analysis applies a modal logic of knowledge and belief to prove the correctness of the protocol. This involves a process of deductive reasoning, where the desired protocol goals are deduced by applying a set of axioms, theorems and inference rules to the assumptions and message exchanges of the protocol. If the protocol is shown to fulfil its goals it can be deemed to be correct and secure.

Keywords**: -** Cryptographic Protocol Verification, Non-Repudiation, Logics, Public-Key Cryptography, Security Protocols, Proof of Origin, Proof of Receipt, Logical Postulates, Deductive Reasoning.

1. Introduction

Non-repudiation allows an exchange of data between two principals in such a manner that the principals cannot subsequently deny their participation in the exchange. This involves the generation of a *proof of origin* (POO) to prove that the originator sent data and a *proof of receipt* (POR) to prove that the recipient received the data. The proofs of origin and receipt constitute non-repudiation evidence information. In [1,2,3], such non-repudiation schemes are shown. This paper concentrates on the scheme shown in [1] which achieves non-repudiation with mandatory *proof of origin* and mandatory *proof of receipt*. The digitally signed evidence exchanged in the non-repudiation protocol is binding. In [4], a modal verification logic is shown, which combines the logics of knowledge and belief. The logic provides axioms, which express the low-level properties of public-key protocols. These properties can be combined using inference rules. Users can deduce their own theorems, thus allowing for increased flexibility in the analysis of public key cryptographic protocols.

2. Non-Repudiation Scheme

The non-repudiation scheme [1] exchanges a proof of origin and a proof of receipt between the originator and recipient. Public-key digital signatures are used to ensure that the proofs are undeniable and non-forgeable. The scheme uses a trusted third party called the Non-Repudiation Server (NRS), in relation to which, the following points can be noted:

i. The NRS first obtains the evidence information (proofs of origin and receipt) in such a way that the recipient cannot learn the *proof of origin* and the originator cannot learn the *proof or receipt.*
ii. The NRS will only distribute the evidence when it is in possession of both the *proof of origin* and the *proof of receipt.*
iii. Once the NRS has received all required pieces of proof information, it will not renege in its duty to distribute that information

The proofs of origin and receipt are not exchanged until both principals have submitted their signed evidence to the NRS. Therefore, if a principal does not provide its evidence to the NRS, then neither principal can benefit from the message transfer. The scheme makes use of a generic time-stamping authority (TSA) to provide reliable and trustworthy timestamps. Digitally signed evidence is

submitted to the TSA, which appends its identifier and a time-stamp and then signs the resulting message. The time-stamps are used to verify that the private keys used for digital signature generation are timely.

2.1 Protocol Notation

In the protocol specification given below, the following notation is employed:
k_x, k_x^{-1} : public and private keys of entity x
k{M} : encryption/decryption of message M using the key k
n1_NRS, n2_NRS : random nonces generated by the NRS
S→R : {M} : means that the entity S sends message M to entity R
type1 : field which indicates that this structure is intended as *proof of origin*
type2 : field which indicates that this structure is intended as *proof of receipt*
A,B : the distinguishing identifiers of the originator and recipient
TSA : the distinguishing identifier of the time-stamping authority
msg : the message to be exchanged
h(M) : a one-way hashing function applied to msg M
[M]dsgX : the digital signature of entity X appended to the message M
ts1,ts2 : timestamps
{M1,M2} : the concatenation of M1 and M2 without any change to their contents.
The structures *Proof of Origin* (POO) and *Proof of Receipt* (POR), two pieces of evidence information exchanged during a non-repudiable exchange are as follows :
POO = [([type1, A, B, msg]dsgA), TSA, ts1]dsgTSA
POR= [([type2, B, A, h(POO)]dsgB), TSA, ts2]dsgTSA
The Non-Repudiation protocol assumes that all entities are aware of, or are capable of, obtaining each other's valid public keys and that all private keys used by the system are not compromised.

2.2 Protocol Interactions

1. **A→TSA : k_{TSA}{[type1, A, B, msg]dsgA}**
2. **TSA→A : k_A{POO}. Where POO= [[type1, A, B, msg]dsgA , TSA, ts1]dsgTSA**
3. **A→NRS : {"N_R_req"}**
4. **NRS→A : k_A{n1_NRS}**
5. **A→NRS : k_{NRS}{[n1_NRS, POO, p_por]dsgA}.Where p_por = {type2, B, A, h(POO)}**
6. **NRS→B : k_B{[n2_NRS,p_por]dsgNRS}**
7. **B→TSA : k_{TSA}{[p_por]dsgB}**
8. **TSA→B : k_B{POR}**
9. **B→NRS : k_{NRS}{[n2_NRS, POR]dsgB}**
10. **NRS→B : k_B{POO}**
11. **NRS→A : k_A{POR}**

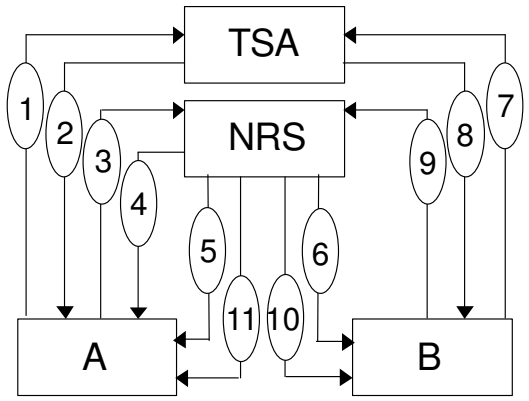

Figure 1 : Protocol Interaction Diagram

3. The Protocol Verification Logic

The modal logic of Coffey-Saidha [4] is used for the analysis. The logic combines the modal logics of knowledge and belief. Axioms are used to model the low level properties of cryptographic communication systems such as: (i) data encryption and decryption, (ii) the transmission and reception of network data and (iii) the inability of a principal to decrypt a message without knowing the appropriate private key. The language of this logic has the following syntax:
a,b,c,..., general propositional variables. **Φ**, an arbitrary statement. **Σ** and **Ψ**, arbitrary entities. **i** and **j**, range over entities. **ENT**, the set of all possible entities. **k**, a cryptographic key. In particular, k_Σ is the public key of entity Σ and k_Σ^{-1} is the corresponding secret key of entity Σ. **t, t', t''...**, times.
e(x,k_Σ), encryption function, encryption of x using key k_Σ.
d(x,k_Σ^{-1}), decryption function, decryption of x using key k_Σ^{-1}.

$\mathbf{d}(h(x),k_{\Sigma}^{-1})$, the digital signature of entity Σ for x.
K, knowledge operator of Hintikka [5]. $K_{\Sigma,t}\Phi$ means Σ knows statement Φ at time t.
L, knowledge predicate. $L_{\Sigma,t}x$ means Σ knows and can reproduce object x at time t.
B, belief operator. $B_{\Sigma,t}\Phi$ means Σ believes at time t that statement Φ is true.
C, 'contains' operator. $C(x,y)$ means the object x contains the object y, y may be cleartext or ciphertext in x.
S, emission operator. $S(\Sigma,t,x)$ means Σ sends message x at time t.
R, reception operator. $R(\Sigma,t,x)$ means Σ receives message x at time t.
The classical logical connectives are also used: ∧ (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.

3.1 Inference Rules

Inference rules allow us to deduce new statements (conclusions) from statements that are already known. The inference rules of this logic are as follows:

R1:	From ⊢ p and ⊢ (p → q) infer ⊢ q		**R4:**	From p and q infer (p ∧ q)
R2:	**(a)**	From ⊢ p infer ⊢ $K_{\Sigma,t}$p	**R5:**	From p infer (p ∨ q)
	(b)	From ⊢ p infer ⊢ $B_{\Sigma,t}$p	**R6:**	From ¬(¬p) infer p
R3:	From (p ∧ q) infer p		**R7:**	From (from p infer q) infer (p → q)

Rule R1 is the Modus Ponens which states that if a schema p can be deduced and (p → q) can be deduced, then schema q can be deduced. Rules R2a and R2b refer to the fact that if a schema p is derivable, then knowledge/belief of p is also derivable. The remaining rules R3-R7 are the standard propositional rules for natural deduction. Rule R7 is known as the material implication introduction rule and comprises an inference within an inference. This rule means that, if it is possible to infer the schema q from a set of schemas $\{p, s_1, s_2, s_3, ..., s_n\}$ then it is permitted to infer the schema (p → q) from the set of schemas $\{s_1, s_2, s_3, ..., s_n\}$.

3.2 Axioms

Axioms are statements of truth that apply to a logic theory. The logic makes use of the following axioms:

A1: **(a)** $\exists t\exists p\exists q\exists i,i\in\{ENT\}(K_{i,t}p \wedge K_{i,t}(p \rightarrow q) \rightarrow K_{i,t}q)$
(b) $\exists t\exists p\exists q\exists i,i\in\{ENT\}(B_{i,t}p \wedge B_{i,t}(p \rightarrow q) \rightarrow B_{i,t}q)$

The A1 axioms are the applications of the Modus Ponens to the knowledge and belief operators. Axiom A1(a) states that if an entity knows at time t that p is true, and that (p → q) is true, then the entity knows at time t that q is true. Axiom A1(b) makes a similar statement for the belief operator.

A2: $\exists t\exists p\exists i,i\in\{ENT\}(K_{i,t}p \rightarrow p)$

Axiom A2 is the knowledge axiom and states that if something is known, then it is true.

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

The A3 axioms express the monotonicity of knowledge and belief.

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

Axiom A4 states that if some data is constructed from other data, then each piece of data involved in the construction must exist and be known to some entity.

A5: $\exists t\exists x(\ S(\Sigma,t,x) \rightarrow L_{\Sigma,t}x \wedge \exists i,i\in\{ENT/\Sigma\}\exists t',t'>t\ R(i,t',x)\ )$

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

Axioms A5 and A6 deal with the emission and reception of messages in a hostile communications environment. This assumes that the communication system is reliable so that message loss and transmission errors cannot occur without interference from a hostile party.

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

The A7 axioms refer to ability of a principal to encrypt/decrypt if the encryption/decryption key is known.

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

The A8 axioms express the inability of a principal to encrypt/decrypt without knowledge of the appropriate key. A8a states that if an entity does not know k_Σ at t and does not know, prior to t, the encryption $e(x, k_\Sigma)$ and also does not receive $e(x, k_\Sigma)$ at t in a message, then the entity cannot know $e(x, k_\Sigma)$ at time t. Axioms A8b makes a similar statement for decryption without knowledge of the decryption key.

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

Axiom A9 states that secret keys are known only to their rightful owners.

A10: $\exists t \exists x(\exists i,i \in \{ENT\}(L_{i,t}d(x,k_\Sigma^{-1}) \wedge L_{i,t}k_\Sigma) \rightarrow L_{i,t}x) \rightarrow K_{i,t}(\exists t',t'<t\ L_{\Sigma,t'}x)$

Axiom A10 states that if an entity knows and can reproduce $d(x,k_\Sigma^{-1})$ 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*. This indicates that all entities know that any entity which decrypts/signs data *x* with its secret key must know *x* prior to the decryption/signing.

4. Verification of the Non-Repudiation Protocol

4.1 Introduction

The Non Repudiation Protocol is verified by applying the logic as follows:

- Each protocol goal is formally expressed
- Each step in the protocol is formally expressed in the language of logic
- Starting with the initial protocol assumptions, logical statements for each protocol step are built up using the logical axioms and inference rules.
- These logical statements are compared with the desired protocol goals to see if the goals are achieved.

If the desired goals are not achieved, then this generally points to a missing hypothesis in the protocol assumptions or the presence of some protocol flaw. It is imperative for this process that the protocol goals are correctly formulated. If any of the protocol goals is inaccurate or unaccounted for, then the verification cannot succeed.

4.2 Protocol Goals and Assumptions

The non-repudiation protocol has four main goals and a number of underlying assumptions. The goals can be formally expressed as follows:

Goal 1: $B_{A,t9}(\neg L_{B,t9}msg)$. Originator A requires that the recipient B does not know msg prior to t9, at which time the NRS is in possession of a valid *proof of receipt* for distribution to A. A trusts that the NRS and the TSA will not divulge msg prior to t9.

Goal 2: $B_{B,t9}(\neg L_{A,t9}dsgB)$. Recipient B requires that A does not know the dsgB prior to t9, and does not know and cannot construct POR. B trusts the NRS and the TSA will not release POR prior to t9.

Goal 3: $K_{NRS,t5}(\exists y,\exists t,t0<t<t5,S(A,t,y) \wedge C(y,x))$. Where x = {n1_NRS, POO, p_por, $d(h(n1\_NRS,POO,p\_por), k_A^{-1})$ }, which is the digitally signed portion of message m5. The NRS requires that any non-repudiation evidence information received from principal A is origin-authentic and timely.

Goal 4: $K_{NRS,t9}(\exists w,\exists t',t0<t'<t9,S(B, t', w) \wedge C(w,v))$. Where v ={n2_NRS, POR, $d(h(n2\_NRS,POR), k_B^{-1})$ } which is the digitally signed portion of message m9. The NRS requires that any non-repudiation evidence information received from principal B is origin-authentic and timely.The initial assumptions are now formally expressed as follows:

(1) $L_{A,t0}(k_{NRS})$ (2) $L_{B,t0}(k_{NRS})$ (3) $L_{A,t0}(k_{TSA})$ (4) $L_{B,t0}(k_{TSA})$ (5) $L_{A,t0}(k_B)$ (6) $L_{B,t0}(k_A)$

(7)$L_{NRS,t0}(k_A)$ (8) $L_{NRS,t0}(k_B)$ (9) $L_{TSA,t0}(k_A)$ (10) $L_{TSA,t0}(k_B)$

(11) $K_{A,t0}(\forall i, i \in \{ENT/A\}, \neg L_{i,t0}msg)$ (12) $K_{B,t0}(\forall i, i \in \{ENT/B\}, \neg L_{i,t0}dsgB)$

(13) $B_{A,t0}(\forall i, i\in\{TSA,NRS\},\forall t, t0<t<t9, S(i, t, m) \wedge C(m,msg) \rightarrow m=e(x, k_A) \wedge C(x,msg))$
(14) $B_{B,t0}(\forall i, i\in\{TSA,NRS\},\forall t, t0<t<t9, S(i, t, m) \wedge C(m,dsgB) \rightarrow m=e(x, k_B) \wedge C(x,dsgB))$
(15) $K_{NRS,t0}(\forall i, i\in\{ENT\}, \neg L_{i,t0}n1\_NRS)$ (16) $K_{NRS,t0}(\forall i, i\in\{ENT\}, \neg L_{i,t0}n2\_NRS)$
Assumptions (1-10) reflect that the various entities know each other's public keys. (11) states that only A knows msg at t0, similarly by (12) only B knows dsgB at t0. (13) states A's belief that all messages containing msg and sent prior to t9 will be encrypted using A's public key. Similarly (14) states B's belief that all messages containing dsgB and sent prior to t9 will be encrypted using B's public key. Finally assumptions (15) and (16) state that the nonces n1_NRS and n2_NRS generated by the NRS are fresh and unkonwn to any entity at the onset of the protocol execution.

4.3 Theorems for Protocol Analysis

For the verification, two theorems are used, derived by using the axioms of the CS logic.

4.3.1 Theorem 1

$\exists t\exists x\exists t',t'<t\ K_{\Sigma,t}(\forall i, i\in\{ENT\}\neg L_{i,t'}x) \wedge K_{\Sigma,t}((\exists y(R(\Sigma,t,y) \wedge C(y,d(x, k_{\Psi^{-1}})))) \rightarrow K_{\Sigma,t}(\exists z(\exists t'',t'<t''<t$ $S(\Psi,t'',z)\wedge C(z,d(x,k_{\Psi^{-1}}))))$. This theorem states that if x is a piece of data unknown before a time t' and if an entity Σ receives a message containing the decryption of x under a key $k_{\Psi^{-1}}$, then Σ knows that entity Ψ sent a mesage containing the decryption after time t'.

4.3.2 Theorem 2

$\exists t\exists x\exists t',t'<t\ K_{\Sigma,t}(\neg L\Psi_{,t'}x)\wedge K_{\Sigma,t}(\forall y(\forall t'',t'<t''<t\ S(\Sigma,t'',y)\wedge C(y,x)\rightarrow\exists h,h\in\{ENT/\Psi\}y=e(x,k_h)))\wedge B_{\Sigma,t}(\forall z($ $\forall j,j\in\{ENT/(\Sigma,\Psi)\}\forall t''',t'<t'''<t\ S(j,t''',z)\wedge C(z,x)\rightarrow\exists l,l\in\{ENT/\Psi\}y=e(x,k_l)))\rightarrow B_{\Sigma,t}(\neg L_{\Psi,t}x))$.This theorem states that if x is a piece of data unknown to entity Ψ at time t' and if all subsequent messages sent prior to time t containing x are encrypted using keys not belonging to Ψ, the Ψ cannot know x at time t.

4.4 Application of the logical axioms and inference rules

The protocol messages are now examined with a view to verifying the four protocol goals. Each step is rewritten in the language of the logic. The digital signatures are in bold.

Step 1: A → TSA : e({type1, A, B, *msg*, d(*h*(type1, A, B, *msg*),k_A^{-1})}, k_{TSA}). After sending message m1, A knows that he has sent a message containing msg and that his message is encrypted under k_{TSA}.

Step 2 : TSA → A : e({p_poo, TSA, ts1, d(*h*(p_poo, TSA, ts1),k_{TSA}^{-1})}, k_A). Where p_poo = {type1, A, B, *msg*, d(*h*(type1, A, B, *msg*),k_A^{-1})}. If A receives m2, then since the message is encrypted under k_A, A's belief of assumption (13) is not refuted. By Axiom A7b, A is able to decrypt the message and recover it's contents.

Step 3 : A → NRS : { "N_R_req" }. This message does not advance the knowledge of any principle as it simply initiates communication with the NRS.

Step4 : NRS → A : e(n1_NRS, k_A). A is able to recover n1_NRS by axiom A7b using k_A^{-1}.

Step5: A→NRS : e({n1_NRS,POO,p_por, d(*h*(n1_NRS,POO,p_por),k_A^{-1})}, k_{NRS}). Where POO = {p_poo, TSA, ts1, d(*h*(p_poo, TSA, ts1),k_{TSA}^{-1})} and p_por = {type2, B, A, *h*(POO)}. Originator A knows that he has sent a message containing msg and that this message is encrypted under k_{TSA}. If the NRS receives the message it can conclude that it has received a message containing A's digital signature and original nonce n1_NRS decrypted with A's private key. It knows nonce n1_NRS to be fresh at time t0, so it can deduce that the message is fresh. Thus using Theorem 1 the following is deduced, satisfying **Goal 3**:$K_{NRS,t5}(\exists y,\exists t, t0< t < t5, S(A,t,y) \wedge C(y,x))$ where x={n1_NRS, POO, p_por, d(h(n1_NRS,POO,p_por), k_A^{-1})}.

Step 6 : NRS → B : e(n2_NRS, p_por, d(*h*(n2_NRS,p_por),k_{NRS}^{-1}), k_B). The NRS communicates the nonce n2_NRS and p_por to B. It can recover {n2_NRS, p_por} using axiom A7b and its private key k_B^{-1}.

Step 7 : B → TSA : e(p_por, d(*h*(p_por),k_B^{-1}), k_{TSA}). Where d(h(p_por), k_B^{-1}) is the digital signature of B for the *proof of receipt.* After step 7, B concludes that when m7 was sent, which contains dsgB, it was encrypted by k_{TSA}. That is: $K_{B,t7}(\exists t, t0<t<t7, S(B,t,m7) \wedge \exists x, m7=e(x,k_{TSA}) \wedge C(x,dsgB))$.

Step 8 : TSA → B : e({s_p_por, TSA, ts2, d(h(s_p_por, TSA, ts2),k_{TSA}^{-1})}, k_B). Where s_p_por = {p_por, d(h(p_por),k_B^{-1})}. If B receives message m8, then since the message is encrypted under k_B, B can continue to believe that dsgB has not been sent in the clear.

Step 9 : B → NRS : e({ n2_NRS, POR, d(h(n2_NRS, POR),k_B^{-1}) }, k_{NRS}). Where POR = {s_p_por, TSA, ts2, d(h(s_p_por, TSA, ts2),k_{TSA}^{-1})}. The POR contains dsgB, which must remain protected until this step has completed. B can conclude that when it sent m9, which contains dsgB it was encrypted by k_{NRS}. That is:$K_{B,t9}$(∃t, t0<t<t9, S(B,t,m9) ∧ ∃x, m9=e(x,k_{NRS}) ∧ C(x,dsgB)). Thus combining the knowledge from step 7 and step 9 and the fact initial assumptions (12) and (14) have not been refuted, B concludes that A cannot have gained knowledge of dsgB if the TSA and the NRS remained trustworthy. That is: $K_{B,t9}$(∀t,t0<t<t9, S(B,t,m) ∧ C(m,dsgB) → ∃j,j∈{TSA,NRS}, m=e(x,k_j) ∧ C(m,dsgB)). Combining this with Theorem 2 allows **Goal 2** to be deduced **:** $B_{B,t9}$(¬$L_{A,t9}$dsgB). A similar examination of the states of knowledge and belief of A up to this point reveals that assumptions (11) and (13) have not been refuted and A can conclude that B cannot have gained knowledge of msg if the TSA and the NRS remained trustworthy. Again using Theorem 2, **Goal 1** can be deduced: $B_{A,t9}$(¬$L_{B,t9}$msg). If the knowledge of the NRS is examined at this stage it can be shown that : $K_{NRS,t9}$(R(NRS,t9,m9) ∧ C(m9,d(h(n2_NRS, POR), k_B^{-1}))). Also, based on initial assumption (14), the following statement can be deduced: $K_{NRS,t9}$(∀i, i∈{ENT}, ¬$L_{i,t0}$(h(n2_NRS, POR))).

Theorem 1 can be applied to these results to yield: $K_{NRS,t9}$(∃y, ∃t, t0<t<t9 S(B,t,y) ∧ C(y,d(h(n2_NRS, POR),k_B^{-1}))). Hence, the NRS knows that B sent the digital signature in m9 and can therefore be satisfied with the data signed. Thus **Goal 4** can be deduced. That is: $K_{NRS,t9}$(∃y, ∃t, t0<t<t9 S(B,t,y) ∧ C(y,x)). Where x = { n2_NRS, POR, d(h(n2_NRS,POR),k_B^{-1}) }.

Step 10 : NRS → B : e({ POO }, k_B).

Step 11 : NRS → A : e({ POR }, k_A). Messages m10 and m11 distribute the proof information to recipient and originator. It can be concluded that the four desired protocol goals are fulfilled. If the protocol completes step 9 a successful non-repudiable exchange is deemed to have occurred and the evidence distributed to both principles. No useful information is gained by either principle if the protocol does not complete.

5. Conclusions

In this paper, the correctness and security of the non-repudiation protocol outlined in [1] has been formally analysed. The verification of the protocol was performed with respect to the following goals:

i. The non-repudiation server is assured of the authenticity and timeliness of the evidence submitted to it by both communicating principals
ii. The originator may rightly believe that the recipient cannot discover the proof of origin until the non-repudiation server is in possession of the proof of receipt.
iii. The recipient may likewise believe that the originator cannot discover the proof of receipt until the non-repudiation server is in possession of the proof or origin.

The logical analysis presented in this paper shows that the NR protocol with mandatory proof of receipt fulfils its goals and therefore can be deemed to be correct and secure.

References

[1] T. Coffey, and P. Saidha, "Non-repudiation with Mandatory Proof of Receipt", Computer Communication Review, ACM-Sigcomm, Volume 26, No. 1, pp 6-17, 1996.

[2] N. Zhang and Q. Shi, "Achieving non-repudiation of receipt, The Computer Journal 39 (10) (1996) 844-853

[3] S. Kremer and O. Markowitch, "Optimistic non-repudiable information exchange", 21st Symp. on Information Theory in the Benelux, Werkgemeenschap Informaie-en Communicatietheorie, Enschede (NL), Wassenaar (NL) 2000, pp. 139-146

[4] T. Coffey, and P. Saidha, "A Logic for Verifying Public-Key Cryptographic Protocols", IEE Proceedings - Computers and Digital Techniques, September 1996.

[5] J. Hintikka, "Knowledge and belief: an introduction to the logic of two notions", Cornell University Press. Ithaca, 1962.

[6] S. Kremer, O. Markowitch and J Zhou, "An Intensive Study of Non-Repudiation Protocols" Computer Communications, 25(17):1606--1621, November 2002.