On Investigating the Security and Fairness of a Fair Exchange Protocol using logic-based verification

Tom Coffey, Marian Ventuneac, Thomas Newe, Ioan Salomie
Data Communications Security Laboratory
University of Limerick
Ireland
tom.coffey@ul.ie, marian.ventuneac@ul.ie, thomas.newe@ul.ie, ioan.salomie@ul.ie

Abstract – **Traditionally, informal and intuitive techniques have been used in the design and verification of cryptographic protocols. However, informal verification alone can lead to subtle protocol flaws and weaknesses remaining unidentified. Conversely, formal verification techniques provide a systematic approach to discovering protocol flaws and weaknesses. This paper presents an investigation into the security and fairness of a fair exchange protocol using logic-based verification. The paper introduces properties of non-repudiation protocols. A logic-based analysis technique, suitable for verifying these protocols is outlined. The verification process is demonstrated by way of a case study on a fair non-repudiation protocol to determine if the protocol is secure.**

I. INTRODUCTION

Cryptographic protocols are designed to provide security services, such as key distribution, authentication and non-repudiation, over insecure networks. These protocols are an indispensable component in providing services for applications on mobile and fixed networks such as: certified e-mail, secure e-business, and inter-bank transactions.

The design process of cryptographic protocols is particularly complex and error-prone. The surprisingly significant number of published protocols that have subsequently been found to contain various flaws [1],[2], [3], sometimes several years after the original publication, highlights the complexity of the design process. The absence of formal verification of these protocols can lead to flaws and security errors remaining undetected.

Formal methods provide means to verify such protocols thoroughly, adding confidence in the correctness of a protocol to a level unrivalled by informal methods. The use of logics has been shown to be effective in detecting flaws in the design of numerous protocols [4],[5],[6],[7], [8].

Non-repudiation services enable accountable and undeniable data exchange between two or more principals [9]. This involves the generation of non-repudiation information to prove that the originator sent the data and that the recipient received the data. In [10],[11],[12], security protocols were proposed to provide non-repudiation services for data exchange using computer networks as a communication medium.

This paper introduces properties of non-repudiation protocols. Logic-based analysis techniques, suitable for verifying these protocols are outlined. The verification process is demonstrated by way of a case study on a fair non-repudiation protocol [11].

A modal logic [7], which combines the logics of knowledge and belief, is used as part of the verification process. The formal verification of the protocol is presented and the results are discussed. The results indicate a weakness in the protocol, highlighting the importance of formal verification as part of the security protocol design process.

II. LOGIC-BASED ANALYSIS OF PROTOCOLS

Logic-based formal verification has been shown to be effective in detecting design flaws in security protocols that can lead to serious protocol failure [3]. The process of logic-based verification can be summarized in the following steps:

1. Formalisation of the protocol messages;
2. Specification of the initial assumptions;
3. Specification of the protocol goals;
4. Application of the logical postulates.

The first step involves specifying the protocol under investigation in the language of the logic by expressing each protocol message as a logical formula. This step is known as protocol formalisation or idealisation. A formal description of the protocol, obtained by formalisation, attempts to show the purpose components of each message so as to avoid any ambiguity.

In step two the initial protocol assumptions, which reflects the beliefs and possessions of the involved principals at the beginning of each protocol run, are specified.

The desired protocol goals are then expressed in the language of the logic. These goals are specified in terms of the beliefs and possessions of the protocol participants at the end of a successful protocol run.

Step four of the verification concerns the application of logical postulates to establish the beliefs and possessions of protocol principals. The objective of the logical analysis is to verify whether the desired goals of the protocol can be derived from the initial assumptions and protocol steps. If such a derivation exists, the protocol is successfully verified (within the scope of the logic); otherwise, the verification fails.

III. REASONING ON PROPERTIES OF NON-REPUDIATION PROTOCOLS

Non-repudiation protocols are generally analysed in terms of *non-repudiation*, *timeliness* and *fairness*.

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 [13].

Timeliness states the protocol should terminate in a finite amount of time, thus avoiding any of the involved entities to wait an indefinite amount of time in order to receive the expected item or evidence.

Fairness of a non-repudiation protocol states that none of the parties involved in the exchange should gain an advantage over the other one during a run of the protocol [14],[15],[16].

In order to formally analyse a non-repudiation protocol, general goals are defined in the following format: entity A receives message *m* in a certain time frame. The assumptions and constraints necessary to achieve a successful exchange are stated. Following the definition of goals and assumptions, the protocol is analysed in respect to achieving the stated goals. The analysis of generic goals provides a straightforward method to verify protocol security. If the proposed goals are achieved, we state the protocol is secure.

By establishing the protocol security, the goals are further refined to reason about additional properties such as non-repudiation and fairness. By performing the security analysis, we determine implicitly the time range of each goal. This enables the reasoning on timeliness of the analysed protocol, as a condition that the protocol will terminate in a finite amount of time. Proving the time range of each step indicate that none of the actively involved entities will wait an indefinite amount of time in order to receive its expected item.

In order to analyse the non-repudiation property, two sub-goals are defined such as non-repudiation of origin (NRO) and non-repudiation of receipt (NRR) sub-goals. NRO sub-goal states that recipient B has to receive all its expected items (a message and its proof of origin signed by originator A), or nothing at all. NRR sub-goal states that A has to receive all its expected items (proof of receipt), or nothing at all. The protocol is deemed fair if A and B both receive their expected items at the end of the protocol run, or neither of them receive useful items. This is expressed as a logical conjunction of NRO and NRR sub-goals.

fairness: (NRO $\wedge$ NRR) $\vee$ ($\neg$NRO $\wedge$ $\neg$NRR)

If any other state is achieved (i.e. A has expected items, and B does not, expressed as $\neg$NRO $\wedge$ NRR), the protocol is said to be unfair.

IV. VERIFICATION LOGIC

The verification technique used in the analysis in this paper applies the logic of Coffey-Saidha [7] to analyse the correctness of the protocol. The logic combines the modal logics of knowledge and belief.

This logic provides a means of verifying public-key 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 logic provides a belief operator and two knowledge operators. One knowledge operator is propositional and deals with the knowledge of statements or facts. The other knowledge operator is a predicate and deals with the knowledge of objects (e.g. cryptographic keys, ciphertext data, etc.)

The inference rules provided are the standard inferences required for natural deduction. The axioms of the logic express the fundamental properties of public-key cryptographic protocols such as the ability of a principal to encrypt/decrypt based on knowledge of a cryptographic key. The axioms also reflect the underlying assumptions of the logic, which are as follows:

- The communication environment is hostile. That is, the data communication system itself is assumed to be reliable so that message loss and transmission errors cannot occur without some interference from a hostile party.
- The public-key cryptosystem is 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 so that the following equations hold true:

  $d(e,k_\Sigma), k_\Sigma^{-1}) = x$ and $e(d(x,k_\Sigma^{-1}),k_\Sigma) = x$

  where d and e are the decryption and encryption functions respectively and k_Σ and k_Σ^{-1} are the public and secret keys respectively. The cryptosystem is collision-free so that it is not possible to create the same ciphertext from two different pieces of plaintext.
- A public key used by the system is considered valid if it has not exceeded its validity period and the corresponding secret key is known only to its rightful owner.
- If a piece of data is encrypted/decrypted, then the entity which performed the encryption/decryption must know that data (the data can be plaintext or ciphertext). Since only one entity may know a decryption key, then if x is decrypted using the key k_Σ^{-1}, then Σ must know x.

As an example, the following axiom states that if a principal i knows a piece of data x and i knows the public key k_Σ, then i can encrypt x to produce the ciphertext data $e(x,k_\Sigma)$

$$L_{i,t}x \wedge L_{i,t}k_\Sigma \rightarrow L_{i,t}(e(x,k_\Sigma))$$

where *L* denotes knowledge of an object.

The language of this logic has the following syntax:

a, b:	general propositional variables
Φ:	an arbitrary statement
Σ, 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_1, t_n:	times
$e(x,k_\Sigma)$:	encryption function, encryption of x with key k_Σ
$d(x,k_\Sigma^{-1})$:	decryption function, decryption of x with key k_Σ^{-1}
$d(h(x),k_\Sigma^{-1})$:	digital signature of entity Σ for x.
K:	knowledge operator. $K_{\Sigma,t}\Phi$ 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: $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (complementation) and $\rightarrow$ (material implication). The symbols $\forall$ and $\exists$ denote universal and existential quantification respectively. $\in$ indicates membership of a set and / denotes set exclusion. Details of the logical postulates (axioms and inference rules) for the Logic are provided in [7].

V. A FAIR NON-REPUDIATION PROTOCOL

A non-repudiation protocol for achieving fair non-repudiation of receipt (NRR) and non-repudiation of origin (NRO) of an exchanged message is presented by Zhou and Gollmann in [11].

The protocol enables two principals A and B to exchange a message and to generate undeniable evidences for B to prove A originated *m*, and for A that B received *m* respectively. Fair non-repudiation is achieved by means of usage of an online *trusted third party* (TTP).

Informal notation of the protocol:

A,B, TTP: principals involved in the exchange.

m: message sent from A to B

c: commitment ciphertext for message *m*.

SK: message key defined by A.

h(m): a one-way hash function of message *m*.

l: a unique label chosen by A to link all the messages in a protocol run.

eK(m): encryption of message *m* with key *K*

$sS_A(m)$: digital signature of message X with the private key S_A.

$A \rightarrow B$: *m*: principal A sends message *m* to principal B.

$A \leftrightarrow TTP$: *m*: principal A fetches message *m* from principal B using *"ftp get"* operation.

f_{NRO}, f_{NRR}, f_{SUB}, f_{CON}: flags stating the purpose of a certain message.

$EOO = sS_A(f_{NRO}, B, l, c)$: evidence of origin of *c*

$EOR = sS_B(f_{NRR}, A, l, c)$: evidence of receipt of *c*

$sub\_K = sS_A(f_{SUB}, B, l, t, SK)$: evidence of submission of key *SK*

$con\_K = sS_{TTP}(f_{CON}, A, l, t, SK)$: evidence of confirmation of key *SK*.

Using the above notation, the steps of the protocol are shown in an informal way.

1. $A \rightarrow B$: f_{NRO}, B, l, c, EOO
   where ciphertext $c=SK(m)$
2. $B \rightarrow A$: f_{NRR}, A, l, c, EOR
3. $A \rightarrow TTP$: f_{SUB}, B, l, SK, sub_K
4. $B \leftrightarrow TTP$: $f_{CON}, A, B, l, SK, con\_K$
5. $A \leftrightarrow TTP$: $f_{CON}, A, B, l, SK, con\_K$

The protocol relies on splitting the exchange message *m* in two parts: a ciphered message $c=eSK(m)$ and the corresponding encryption key *SK*. A sends ciphertext *c* to B, waiting for B's acknowledge of receiving the ciphertext. Next, A signs and sends key *SK* to an online TTP. TTP confirms the key *SK* by signing it with its private key and makes it public to enable both A and B to retrieve it. B requires *SK* to decrypt ciphertext *c* and as part of NRO evidence, and A to complete its NRR evidences.

At the end of a protocol run, B should hold message *m* and NRR evidence, and A should hold NRO evidence. NRO evidence consists of two parts: A's signature on ciphertext *c* (evidence of origin of *c*) and TTP's signature on key *SK* (evidence of confirmation of *SK*). NRR evidence consists also of two parts: B's signature on ciphertext *c* (evidence of receipt of *c*), and TTP's signature on key *SK* (evidence of confirmation of *SK*).

In case of a dispute, an external arbiter uses NRO evidence to prove that A originated message *m*, ciphertext *c* as well as key *SK*. NRR evidence is used to prove that B received ciphertext *c* and received (or is able to receive) key *SK*. The protocol authors claim that the protocol provides non-repudiation of origin and receipt, as well as fair exchange.

VI. CASE SUDY: ANALYSIS OF FAIR NON-REPUDIATION PROTOCOL

The non-repudiation protocol described in section V is analysed in this section to determine if the protocol is secure.

A. Protocol formalisation

The protocol is formalised in the language of the logic as described in section III. The steps of the protocol are rewritten using the language of the logic.

Step1: $K_{B,t1}(R(B, t_1, m_1) \wedge C(m_1, \{f_{NRO}, B, l, c\} \wedge EOO))$

Step2: $K_{A,t2}(R(A, t_2, m_2) \wedge C(m_2 \{ f_{NRR}, A, l, c\} \wedge EOR))$

Step3: $K_{TTP,t3}(R(TTP, t_3, m_3) \wedge C(m_3 \{ f_{SUB}, B, l, SK\} \wedge sub\_K))$

Step4: $K_{B,t4}(R(B, t_4, m_4) \wedge C(m_4 \{ f_{CON}, A, B, l, SK\} \wedge con\_K))$

Step5: $K_{A,t5}(R(A, t_5, m_4) \wedge C(m_4 \{ f_{CON}, A, B, l, SK\} \wedge con\_K))$

Step 1 states, B knows at time t_1 that it will receive a message containing the recipients B identity, a unique label *l*, ciphertext *c* and *EOO* signed by A

Step2 states, A knows at time t_2 that it will receive a message containing the recipients A identity, an unique label *l*, the ciphertext *c* and *EOR* signed by B.

Step3 states, TTP knows that at time t_3 it will receive message m_3 containing B's identity, label *l*, key *SK* used to generate ciphertext *c* and *sub_K* signed by A.

Step4 states, B knows that at time t_4 it will receive message m_4 containing label *l*, key *SK* used by A to generate ciphertext *c* and *con_K* signed by TTP.

Step5 states, A knows that at time t_5 it will receive message m_4 containing label *l*, key *SK* used by A to

generate ciphertext c and con_K signed by TTP.

B. Protocol goals

The goals of the protocols are formalised, stating that each involved principal has to receive its expected message in a certain time frame.

G1: $K_{B,t1}(\exists t, t<t_1, S(A, t, m_1) \wedge C(m_1, \{f_{NRO}, B, l, c\} \wedge EOO))$

G2: $K_{A,t2}(\exists t, t_1 < t < t_2, S(B, t, m_2) \wedge C(m_2, \{f_{NRR}, A, l, c\} \wedge EOR))$

G3: $K_{TTP,t3}(\exists t, t_o < t < t_3, S(A, t, m_3) \wedge C(m_3, \{f_{SUB}, B, l, SK\} \wedge sub\_K))$

G4: $K_{B,t4}(\exists t, t_3 < t < t_4, S(TTP, t, m_4) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$

G5: $K_{A,t5}(\exists t, t_3 < t < t_5, S(TTP, t, m_4) \wedge C(m_4 \{f_{CON}, A, B, l, SK\} \wedge con\_K))$

Goal G1 states that B knows at time t_1 that entity A has sent message m_1 prior to time t_1 and that message m_1 contains label l, c and EOO.

Goal G2 states that A knows at time t_2 that entity B has sent message m_2 prior to time t_2 and that message m_2 contains label l, c and EOR.

Goal G3 states that TTP knows at time t_3 that entity A has sent message m_3 prior to time t_3 and that message m_3 contains label l, c, a key SK and sub_K.

Goal G4 states that B knows at time t_4 that entity TTP has sent message m_4 prior to time t_4 and that message m_4 contains label l, key SK and con_K.

Goal G5 states that A knows at time t_5 that entity TTP has sent message m_4 prior to time t_4 and that message m_4 contains label l, key SK and con_K.

C. Initial assumptions

The initial assumptions are stated prior to the protocol verification, following the initial assumptions made by the protocol's authors.

i. $L_{B,to}\, k_A$ *ii.* $L_{A,to}\, k_B$
iii. $L_{TTP,to}\, k_A$ *iv.* $L_{B,to}\, k_{TTP}$ *v.* $L_{A,to}\, k_{TTP}$
vi. $K_{A,to}(\forall i, i \in \{ENT/A\}, \forall t, t<t_1, \neg L_{i,t}\, l)$
vii. $K_{A,to}(\forall i, i \in \{ENT/A\}, \forall t, t<t_3, \neg L_{i,t}\, SK)$
viii. $K_{TTP,to}(\forall i, i \in \{ENT\}, \forall t, t<t_o, \neg L_{i,t}\, l)$

Assumption *i-v* states that each of the involved parties A, B and TTP knows that the others public keys are valid at time t_o.

Assumption *vi* states that A generates a unique (fresh) label l and he knows that no other entity has knowledge of l before he reveals it at time t_1 (freshness of l).

Assumption *vii* states that A generates a session key SK and he knows that no other entity has knowledge of SK before he reveals it at time t_3 (freshness of SK).

Assumption *viii* states that TTP knows he will receive a unique label l in order to make public the associated key SK with label l.

Once the goals of the protocol and initial assumption have been stated, each of the formalised steps is analysed using the axioms and inference rules of the logic, as well as the initial assumption, in order to deduce the proposed goals.

D. Application of Logical Postulates

Step1: $K_{B,t1}(R(B, t_1, m_1) \wedge C(m_1, \{f_{NRO}, B, l, c\} \wedge EOO))$,
where $EOO = d(\{f_{NRO}, B, l, c\}, k_A^{-1})$

Applying Axiom A2:
$R(B, t_o<t_1, m_1) \wedge C(m_1, \{f_{NRO}, B, l, c\} \wedge EOO))$
By application of Axiom A6 and inference rule R2:
$L_{B,t1}\, m_1 \wedge$
$K_{B,t1}(\exists i, i \in \{ENT/B\}, \exists t, t<t_1, S(i, t, m_1) \wedge C(m_1, \{f_{NRO}, B, l, c\} \wedge EOO))$
Using inference rule R3 yields:
$K_{B,t1}(\exists i, i \in \{ENT/B\}, \exists t, t<t_1, S(i, t, m_1) \wedge C(m_1, \{f_{NRO}, B, l, c\} \wedge EOO)$
where $EOO = d(\{f_{NRO}, B, l, c\}, k_A^{-1})$ (1)

Assumption *i* states that B knows at time t_o A's public key k_A. By Axiom A9, only A has knowledge of his private key k_A^{-1}. Therefore, manipulation of Axiom A8(b) enables B to identify A as the sender of message m_1 since A signed EOO with its private key k_A^{-1}, as shown in (2)

$K_{B,t1}(\exists t, t<t_1, S(A, t, m_1) \wedge C(m_1, \{f_{NRO}, B, l, t, c\} \wedge EOO))$ (2)
where $EOO = d(\{f_{NRO}, B, l, c\}, k_A^{-1})$

Goal G1 is fulfilled.

Step2: $K_{A,t2}(R(A, t_2, m_2) \wedge C(m_2 \{ f_{NRR}, A, l, c\} \wedge EOR))$,
where $EOR = d(\{f_{NRR}, A, l, c\}, k_B^{-1})$

Applying Axioms A2, A6 and inference rules R2 and R3 yields:
$K_{A,t2}(\exists i, i \in \{ENT/A\}, \exists t, t<t_2, S(i, t, m_2) \wedge C(m_2, \{f_{NRR}, A, l, c\} \wedge EOR))$ (3)
By Assumption *vi*, A knows no one (except itself, since A generates label l) has knowledge of label l before t_1. By manipulating Axiom A4, it is deduced that no one has knowledge of message m_2 (which contains label l) before time t_1.
$K_{A,t2}(\forall i, i \in \{ENT/A\}, \forall t, t<t_1, \neg L_{i,t}\, m_2)$ (4)
Using expression (4) and Axiom A5:
$K_{A,t2}(\forall i, i \in \{ENT/A\}, \forall t, t<t_1, \neg S(i, t, m_2))$ (5)
Combining time dependencies from (3) and (4):
$K_{A,t2}(\forall i, i \in \{ENT/A\}\ \forall t, t_1<t<t_2, S(i, t, m_2) \wedge C(m_2, \{f_{NRR}, A, l, c\} \wedge EOR))$

Assumption *ii* states that A knows B's public key k_B at time t_o. Using Axiom A9 and manipulating Axiom A8(b) enables A to identify B as the sender of message m_2 since B signed EOR with its private key k_B^{-1}, as shown in (6)
$K_{A,t1}(\exists t, t_1<t<t_2, S(B, t, m_2) \wedge C(m_2, \{f_{NRR}, A, l, t, c\} \wedge EOR))$ (6)

where EOR = $d(\{f_{NRR}, A, l, c\}, k_B^{-1})$

Goal G2 is fulfilled.

Step3: $K_{TTP,t3}(R(TTP, t_3, m_3) \wedge C(m_3, \{f_{SUB}, B, l, SK\} \wedge sub\_K))$,
where $sub\_K = d(\{f_{SUB}, B, l, SK\}, k_A^{-1})$

Using Axioms A2, A6 and inference rules R2 and R3, expression (7) is deduced:

$K_{TTP,t3}(\exists i, i \in \{ENT/TTP\}, \exists t, t<t_3, S(i, t, m_3) \wedge C(m_3, \{f_{SUB}, B, l, t, SK\} \wedge sub\_K))$ (7)

By Assumption *vii* and manipulating Axioms A4 and A5:

$K_{TTP,t3}(\exists i, i \in \{ENT/TTP\}, \exists t, t_o<t<t_3, S(i, t, m_3) \wedge C(m_3, \{f_{SUB}, B, l, t, SK\} \wedge sub\_K))$ (8)

Assumption *iii* states that TTP knows A's public key k_A at time t_o. Applying Axiom A9 and manipulating Axiom A8(b) enables TTP to identify A as the sender of message m_3 since A signed *sub_K* with its private key k_A^{-1}, as shown in (9)

$K_{TTP,t3}(\exists t, t_o<t<t_3, S(A, t, , m_3) \wedge C(m_3, \{f_{SUB}, B, l, t, SK\} \wedge sub\_K))$ (9)
where $sub\_K = d(\{f_{SUB}, B, l, SK\}, k_A^{-1})$

Goal G3 is fulfilled.

Step4: $K_{B,t4}(R(B, t_4, m_4) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$
where $con\_K = d(\{f_{CON}, A, B, l, SK\}, k_{TTP}^{-1})$
Expression (10) is deduced by applying Axioms A2, A6 and inference rules R2 and R3 to step 4:

$K_{B,t4}(\exists i, i \in \{ENT/B\}, \exists t, t<t_4, S(i, t, m_4) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$ (10)

Assumption *iv* states that B knows TTP's public key k_{TTP} at time t_o. Using Axiom A9 and manipulating Axiom A8(b), B is able to determine the identity of message m_4 as being TTP, as shown in (11).

$K_{B,t4}(\exists t, t<t_4, S(TTP, t, m_4) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$ (11)

However, the expected time frame $t_3<t<t_4$ is not achieved. Since B has no knowledge about freshness of label l and key *SK*, A new assumption can now be made, Assumption *ix*.

ix. $B_{B,t0}(\forall i, i \in \{ENT/A\}, \forall t, t<t_3, \neg L_{i,t}(l, SK))$,
which states that at time t_o B believes that before t_3 only A had knowledge of both label *l*and key *SK*. Therefore, (11) is rewritten as a belief expression in (12).

$B_{B,t4}(\exists t, t_3<t<t_4, S(TTP, t, m_4) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$ (12)
where $con\_K = d(\{f_{CON}, A, B, l, SK\}, k_{TTP}^{-1})$

Goal G4 is not fulfilled.

Step5: $K_{A,t5}(R(A, t_5, m_4) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$
where $con\_K = d(\{f_{CON}, A, B, l, SK\}, k_{TTP}^{-1})$

Following a similar reasoning as that for goal G4, (13) is deduced:

$K_{A,t5}(\exists i, i \in \{ENT/A\}, \exists t, t<t_5, S(i, t, m_4))$ (13)

By using Assumption *viii* and manipulating Axiom A4:

$K_{A,t5}(\forall i, i \in \{ENT/A\}, \forall t, t<t_3, \neg L_{i,t}\ m_4)$, which yields expression (14) by using Axiom A5:

$K_{A,t5}(\forall i, i \in \{ENT/A\}, \forall t, t<t_3, \neg S(i, t, m4))$ (14)

Expression (15) is deduced by combining time dependencies of (13) and (14), as following:

$K_{A,t5}(\forall i, i \in \{ENT/A\}, \forall t, t_3<t<t_5, S(i, t, m4)) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$ (15)

Assumption *v* states that A knows TTP's public key k_{TTP} at time t_o. Using Axiom A9 and manipulating Axiom A8(b), A is able to determine the identity of message m_4 as being TTP, as shown in (16).

$K_{A,t5}(\forall t, t_3<t<t_5, S(i, t, m4)) \wedge C(m_4, \{f_{CON}, A, B, l, SK\} \wedge con\_K))$ (16)
where $con\_K = d(\{f_{CON}, A, B, l, SK\}, k_{TTP}^{-1})$

Goal G5 is fulfilled.

E. Discussion

The security of the protocol is verified by reasoning first about fulfilment of the goals G1-G5. Goal G4 of the protocol fails because B doesn't know if either label *l* or key *SK* is fresh. Therefore, B can only achieve belief in their freshness. This highlights a vulnerability of the protocol making it susceptible to a replay attack [16].

As result of deducing goals G1-G5, the following sub-goals are inferred:

1. The fulfilment of each goal in its expected time frame enables reasoning about the timeliness of the protocol.
2. NRO and NRR sub-goals are defined by using subsets of the initial goals. Since the evidences of NRO consist of *EOO* and *con_K*, NRR sub-goal is fulfilled if B received *EOO* from A and *con_K* from TTP in the expected timeframes. Thus, using goals G1 and G4, NRO sub-goal can be expressed as:

   NRO sub-goal = G1 $\wedge$ G4 (17)

   Similarly, NRR is expressed as

   NRR sub-goal = G2 $\wedge$ G5 (18)

3. Fairness sub-goal is realised by reasoning on NRO and NRR sub-goals. If either NRO and NRR sub-goals or none of them are fulfilled during a run of the protocol, the protocol is fair. Expression (19) models the definition of fairness sub-goal:

   fairness = (NRO $\wedge$ NRR) $\vee$ ($\neg$NRO $\wedge$ $\neg$NRR) (19)

Therefore due to the failure of goal G4 the following are concluded:

1. NRO sub-goal is not fulfilled, since goal G4 is not fulfilled and G1 is achieved. The protocol does not provide non-repudiation of origin for principal B.
   $\neg$NRO = G1 $\wedge$ $\neg$G4

2. NRR sub-goal is fulfilled, by means of achieving

goals G2 and G5. The protocol provides non-repudiation of receipt for principal A.
NRR = G2 ∧ G5

3. Fairness sub-goal is not fulfilled, since NRO sub-goal failed. The protocol does not provide fair exchange.

¬fairness = (¬NRO ∧ NRR) ∨ (NRO ∧ ¬NRR)

VI. CONCLUSIONS

This paper discussed non-repudiation protocols and their verification. The verification process was demonstrated by way of a case study. This process shows how to reason about the security, timeliness and fairness of non-repudiation protocols.

The analyses demonstrated that the non-repudiation of origin sub-goal (G4) is not achieved. This also causes a failure in the fairness sub-goal. G4 can only be represented as a belief (trust) statement and not as a statement of knowledge (security). This means that key agreement is not properly implemented as part of the protocol in the generation of the label *l* or the key *SK*. This non-agreement allows a replay attack scenario to occur as shown in [16]. It can be concluded that the protocol under analysis is not fair and that non-repudiation of origin is not guaranteed.

In conclusion, the investigation presented in this paper highlighted the importance of formal verification as part of the design process for security protocols

VII. ACKNOWLEDGEMENTS

This work was partly funded by the Irish Research Council for Science, Engineering and Technology (IRCSET) - Basic Research Award SC02/237.

VIII. REFERENCES

[1] D.E. Denning and G.M. Sacco, "Timestamps in key distribution protocols", *Communications of the ACM*, vol. 24, no. 81981, pp. 533-536.

[2] R.M. Needham and M.D. Schroeder, "Authentication revisited", *ACM Operating Systems Review*, vol. 21, no. 1, Jan. 1987, pp.7-7.

[3] T. Coffey, R. Dojen and T. Flanagan, "Formal verification: an imperative step in the design of security protocols", *Computer Networks (The International Journal of Computer and Telecommunications Networking)*, vol. 43, no. 5, Dec. 2003, pp 601-618.

[4] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," in *ACM SIGOPS Operating Systems Review*, vol. 23, no. 5, Dec. 3–6, 1989.

[5] L Gong, R. Needham and R. Yahalom, "Reasoning about belief in cryptographic protocols," *Proceedings of the IEEE Computer Security Symposium on Security and Privacy*, May 1990, pp. 234-248.

[6] K. Gaarder and E. Snekkenes, "Applying a formal analysis technique to the CCITT X.509 strong two-way authentication protocol," *Journal of Cryptology*, vol. 3, 1991, pp. 81-98.

[7] T. Coffey and P. Saidha, "A Logic for Verifying Public-Key Cryptographic Protocols", *IEE Proceedings of Computers and Digital Techniques*, vol. 144, no. 1, Jan. 1997, pp. 28-32.

[8] T. Newe and T. Coffey, "Formal Verification logic for hybrid security protocols", *International Journal of Computer Systems Science & Engineering*, vol. 18 no. 1, Jan 2003, pp 17-25.

[9] ISO/IEC 13888-1, "Information Technology-Security Techniques-Non-repudiation-Part 1: General", ISO/ IEC, 1997.

[10] T. Coffey and P. Saidha, "Non-repudiation with Mandatory Proof of Receipt", *Computer Communication Review*, vol. 26, no. 1, Jan. 1996, pp. 6-17.

[11] J. Zhou and D. Gollmann, "A fair non-repudiation protocol", *Proceedings of 1996 IEEE Symposium on Security and Privacy*, IEEE Computer Security Press, Oakland, California, May 1996, pages 55-61.

[12] O. Markowitch and S. Kremer, "An Optimistic Non-repudiation Protocol with Transparent Trusted Third Party", *Proceedings of ICIS 2001*, Lecture Notes in Computer Science, vol. 2200, Springer-Verlag, 2001, pp. 363-378.

[13] T. Coffey, P. Saidha and P. Burrows, "Analysing the Security of a Non-repudiation Communication Protocol with Mandatory Proof of Receipt", *Proceedings of International Symposium on Information and Communication Technologies*, Trinity College, Dublin, Sept. 2003, pp. 370-376.

[14] N. Asokan, V. Shoup and M. Waidner, "Asynchronous protocols for optimistic fair Exchange", *Proceedings of 1998 IEEE Symposium on Security and Privacy*, Oakland, California, May 1998, pp. 86-99

[15] O. Markowitch, D. Gollmann, and S. Kremer, "On Fairness in Exchange Protocols", *Proceedings of 5th ICISC 2000*, Lecture Notes in Computer Science, vol. 2587, Springer-Verlag, 2000, pp. 451-464.

[16] S. Gürgens, and C. Rudolph, "Security Analysis of (Un-) Fair Non-repudiation Protocols", *Formal Aspects of Security*, Lecture Notes in Computer Science, vol. 2629, Springer-Verlag, London, 2002, pp. 97-114.