Verifying a Minimum-Knowledge Authentication Protocol for use with Power-Line Networks

T. Newe and T.Coffey
Department of Electronic & Computer Engineering,
University of Limerick,
Ireland.
E-Mail: (Thomas.Newe@UL.ie) (Tom.Coffey@UL.ie)
Fax: 353-61-338176

ABSTRACT

Transferring data over power-line networks presents a number of security challenges due to their open, insecure nature. Providing security services such as identity authentication and message integrity verification are examples of two services that are necessary for these networks.

Minimum-knowledge authentication schemes enable principals prove their identities, or the authenticity of their messages, using an interactive challenge-response process. Due to their strong security, relatively low data transfer requirements and computational complexity, minimum-knowledge schemes are ideally suited for power -line networks, where data transfer rates are limited. This paper discusses the implementation of a minimum-knowledge identification protocol and also investigates the formal verification of the identification protocol using a logic theory.

Keywords: Power-line, minimum-knowledge, authentication, security, formal methods, verification, logic, belief, knowledge.

1. INTRODUCTION

The power-line is an inhospitable and insecure environment for data transfer. As power-line communications becomes more popular it will become necessary to provide security services such as identity authentication, message integrity and data protection. The focus of this paper is on identity authentication for power line systems using a minimum-knowledge identification scheme.

Minimum-knowledge authentication schemes can provide both identity authentication and message integrity verification. Identity authentication comprises a series of random challenges made by a verifier to a claimant. The claimant computes a response to each challenge using its private key. By examining the claimant's response, the verifier is able to corroborate, with a high level of confidence the identity of the claimant. A claimant may also digitally sign a message using a minimum-knowledge scheme without significant dialogue with the verifier.

A number of minimum-knowledge schemes currently exist, such as: *Fiat-Shamir*[1]*, Ohta-Okamoto*[2] *and Guillou-Quisquater*[3]. These schemes vary in terms of the number of challenge-response cycles, the complexity of the modular calculations, the amount of secret information that must be stored, and data transfer requirements. The Ohta-Okamoto scheme is particularly suitable for use with power-line networks, since it minimises the amount of data exchanged during a verification cycle.

This paper outlines a realisation of an Ohta-Okamoto based identification scheme [4] and presents a formal evaluation of the security and trust of the scheme using the Coffey-Saidha logic of knowledge and belief [5].

2. FORMAL VERIFICATION USING A LOGIC THEORY

Traditionally, cryptographic protocols have been designed and verified using informal and intuitive techniques. However, an absence of formal verification can lead to flaws and security errors remaining undetected in a protocol. Formal verification aims at providing a rigid and thorough means of testing the correctness of a cryptographic protocol so that even subtle defects can be uncovered. A number of formal techniques have recently been developed for this purpose and can be broadly divided into two categories: (i) testing techniques and (ii) logical techniques. This paper employs a logical technique in the verification of the realised identification scheme.

2.1 *Structure of a Logic Theory*

A logic theory consists of the following [6]:

- A formal language, which allows protocol facts to be expressed in a form which can be manipulated by the logic.
- A set of inference rules, which allow new facts to be deduced from already known facts.
- A set of axioms, which are formal expressions of facts which apply to the logic. There are two types: logical and non-logical. Logical axioms are part of the logic theory itself and are not dependent on the system being analysed. Non-logical axioms on the other hand include statements expressing the fundamental properties of public/conventional key cryptosystems.
- Semantics, which specify whether logical statements will be true or false in practice.

2.2 *Verification using a Logic Theory*

Most analysis using formal logic theories follow the Hoare [7] structure for logical analysis. This involves a deductive reasoning approach based on the "application of valid rules of inference to sets of valid axioms". This proceeds as follows:

- Formally express protocol steps in the language of the logic.
- Formally express protocol assumptions.
- Formally express desired protocol goals.
- Starting with the assumptions, build up logical statements after each protocol step using the logical axioms and inference rules. Compare these logical statements with the desired protocol goals to see if the goals are achieved.

If the desired goals have not been achieved, then this points to a missing hypothesis in the protocol assumptions or the presence of some protocol flaw.

Section 3 outlines a realisation of the Ohta-Okamoto based identification scheme [4], section 4 presents a summary of the Coffey-Saidha logic [5], and section 5 applies this logic to the realised identification scheme.

3. REALISING THE OHTA-OKAMOTO IDENTIFICATION SCHEME

3.1 *Ohta-Okamoto Identification Scheme*

The Ohta-Okamoto identification scheme is based on the difficulty of extracting the L^{th} roots *mod n,* when the factors of n are unknown. If L is large (≥ 30 bits) and $n = p * q$: where p and q are private large prime numbers (≥ 256 bits), then extracting modular roots is as difficult as factoring the public modulus n, to obtain p and q.

The Ohta-Okamoto identification scheme requires the existence of a trusted centre, such as a government, university, bank, etc., to generate and publish a public modulus n and a public integer L. Parameters n and L are common to all principles in the domain of the trusted centre. Each claimant generates and publishes its public key I, where $I^{-1} = S^L \ (mod\ n)$, and S is the claimant's secret random integer, $S \in Z_n$ where Z_n denotes $\{0...n\text{-}1\}$.

Verification of a claimant's identity by a verifier in this scheme is achieved by executing the following steps t times. If the value of L is sufficiently large ($\geq 2^{30}$, as is normally the case) then $t = k = 1$.

1. The claimant generates a random number $R \in Z_n$, and sends $X = R^L \pmod n$ to the verifier.
2. The verifier challenges the claimant with a random number $E \in Z_L$.
3. The claimant responds with Y, where $Y = R * S^E \pmod n$.
4. The verifier checks $Y^L * I^E \equiv X \pmod n$.

The verifier only accepts the claimant's proof of identity if step 4 returns true.

3.2 *Realised Ohta-Okamoto Based Identification Scheme*

In realising the identification scheme, for use with multiple claimants and verifiers, a number of additional requirements need to be considered:

1. The trusted central authority is responsible for generating the following information: (i) n - the public system-wide modulus ($n \geq 512$ bits), (ii) L - the public system-wide integer ($L \geq 2^{30}$), (iii) S – each claimant's secret key ($S \in Zn$), (iv) I_c - each claimant's public key, where $I_c^{-1} = S^L \pmod n$ and (v) a new parameter c ($\geq 2^{16}$) which acts as a unique claimant's identification number. The verifier(s) will use c to retrieve each claimant's public key I_c.
2. Each claimant (/client) is personalised using parameters n, L, S. Each verifier has a copy of parameters n and L as well as access to each claimant's public key I_c, either from a key database located in the verifier (for off-line operation), or from a trusted on-line server (see fig. 1).

The identification procedure of a claimant by a verifier is outlined in fig. 1. In step 1 of the procedure, the

claimant transmits its identification number c to the verifier. The claimant's public key I_c is retrieved from a database using parameter c. The public key may be retrieved off-line using a local database or on-line using a trusted server. Steps 2-5 of the identification procedure are the same as steps 1-4 described in section 3.1 above.

On verification of $Y^L * I_C^E \equiv X \pmod{n}$ in step 5 (a & b), the verifier establishes that the claimant possesses secret S (thus establishing the claimants identity), since:

$$
\begin{aligned}
Y &= R * S^E \pmod{n} \\
Y^L &= R^L * S^{LE} \pmod{n} \\
&= X * S^{LE} \pmod{n} \qquad \{\text{since } X = R^L \pmod{n} \} \\
&= X * I_c^{-E} \pmod{n} \qquad \{\text{since } I_c^{-1} = S^L \pmod{n} \} \\
Y^L * I_c^E &= X \pmod{n}
\end{aligned}
$$

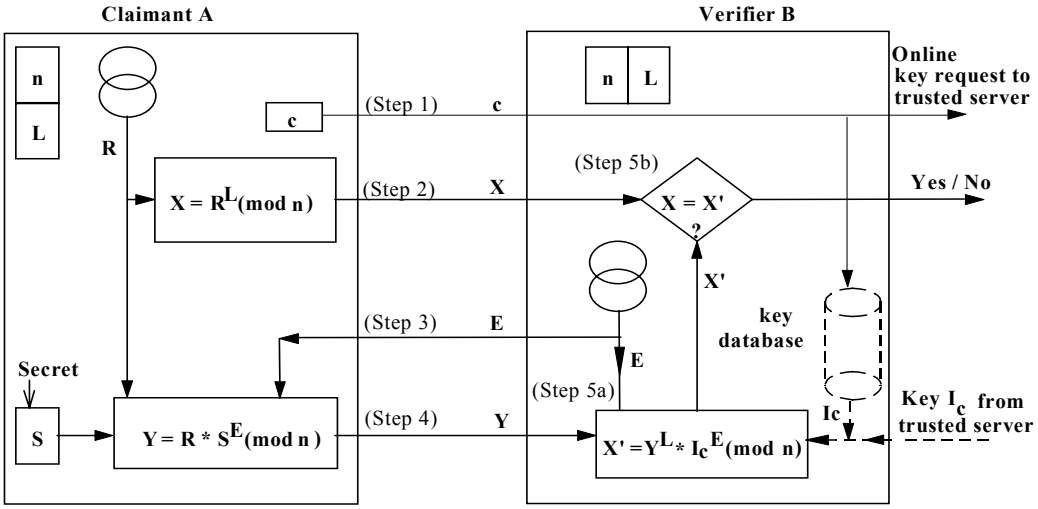

Fig.1: Realised Claimant Identification Procedure

3.3 *Security*

The realised scheme outlined above is well suited for use with power-line networks, given that:

- Only one cycle of the scheme is required if L is chosen to be $\geq 2^{30}$ since the security of the scheme is determined by $L^{-(t*k)}$ [4].
- Using a security level of 2^{-30} or 2^{-72} requires 134 or 139 bytes of information to be transferred between claimant and verifier [4]. Therefore as the security level of the scheme is increased there is relatively little increase in its data transfer requirements.

4. THE LOGIC OF COFFEY-SAIDHA

This logic [5] 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. A belief operator and two knowledge operators is provided by the logic. 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 are sufficiently low-level to 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.

These axioms 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.

- A public key used by the system is considered valid if it has not exceeded its validity period and only its rightful owner knows the corresponding secret key.
- If a piece of data is encrypted/decrypted, then the entity which performed the encryption/decryption must know that data.

This logic is capable of analysing a wide variety of public-key cryptographic protocols such as the Ohta-Okamoto minimum-knowledge scheme presented in section 3. This is because the constructs of the logic are general purpose as seen below, and therefore provide the user with increased flexibility allowing him to develop his own theorems.

4.1 *The Logic Language*

The language of a logic is used to formally express the logical postulates and protocol facts. The language for the Coffey-Saidha logic is as follows:

- **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 private key of entity Σ
- **t, t', t''...** time
- **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}
- ***K***, propositional knowledge operator of Hintikka. $K_{\Sigma,t}\Phi$ 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 that the object *x* contains the object *y*. The object *y* may be cleartext or ciphertext 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*.

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.

4.2 *Inference Rules*

The logic incorporates the following rules of inference [5,8]:

R1:		From $\vdash p$ and $\vdash (p \rightarrow q)$ infer $\vdash q$
R2:	**(a)**	From $\vdash p$ infer $\vdash K_{\Sigma,t}p$
	(b)	From $\vdash p$ infer $\vdash B_{\Sigma,t}p$

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

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

R3:	From $(p \wedge q)$ infer p
R4:	From p and q infer $(p \wedge q)$
R5:	From p infer $(p \vee q)$
R6:	From $\neg(\neg p)$ infer p
R7:	From (from p infer q) infer $(p \rightarrow q)$

4.3 *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, in this case they apply to public-key systems.

4.3.1 Logical Axioms

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

A1: (a) $\exists t \exists p \exists q (K_{\Sigma,t} p \wedge K_{\Sigma,t}(p \rightarrow q) \rightarrow K_{\Sigma,t} q)$

(b) $\exists t \exists p \exists q (B_{\Sigma,t} p \wedge B_{\Sigma,t}(p \rightarrow q) \rightarrow B_{\Sigma,t} q)$

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

The axioms A1(a) and A1(b) are applications of the Modus Ponens to the knowledge and belief operators. 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) $\exists t \exists x \exists i, i \in \{ENT\} (K_{i,t} x \rightarrow \forall t', t' \geq t \; K_{i,t'} x)$

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

Axioms A3(a), A3(b) and A3(c) relate to the monotonicity of knowledge and belief, which asserts that knowledge and belief, once gained, cannot be lost.

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

The final logical axiom is concerned with the *C* predicate. 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:

4.3.2 Non-Logical Axioms

The non-logical axioms (A5-A10) 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 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*. 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 refers to the fact that a private key owner must know any data which has been decrypted using that private key.

5. VERIFICATION OF THE REALISED IDENTIFICATION SCHEME

In the analysis of the realised Ohta-Okamoto based scheme which follows, *tn* refers to the time at the end of protocol step *n*. For example, *t0* is the time at the start of the protocol execution and *t4* is the time at the end of the protocol execution.

5.1 *Goals of the Scheme*

Goal 1:	$K_{B,t1}(\exists t,\ t < t1,\ S(A,\ t,\ c))$
Goal 2:	$K_{B,t2}(\exists t,\ t0 < t < t2,\ S(A,\ t,\ X))$
Goal 3:	$K_{A,t3}(\exists t,\ t0 < t < t3,\ S(B,\ t,\ E))$
Goal 4:	$K_{B,t4}(\exists t,\ t3 < t < t4,\ S(A,\ t,\ Y))$

Fig 2: Goals of the Realised identification Scheme

Goal 1 states that *B* will obtain (have knowledge of) the claimants *A* identification number *c* before the end of step 1 of the protocol.

Goal 2 states that *B* will obtain (have knowledge of) the claimants *A* random calculation *X* before the end of step 2 of the protocol. This is a timely transmission indicating that *X* must not have been sent in any previous cycle of the protocol.

Goal 3 states that claimant *A* will receive (have knowledge of) the random challenge *E* from verifier *B* before the end of step 3 of the protocol. . This is a timely transmission indicating that *E* must not have been sent in any previous cycle of the protocol.

Goal 4 states that *B* will obtain (have knowledge of) the response *Y* from claimant *A* to the challenge *E* during step 4 of the protocol.

5.2 *Initial Assumptions*

1:	$\forall i,\ \forall t,\ i\in\{ENT\}K_{i,t}L$
2:	$\forall i,\ \forall t,\ i\in\{ENT\}K_{i,t}n$
3:	$\forall i,\ \forall t,\ i\in\{ENT\}K_{i,t}c$
4:	$\forall i,\ \forall t,\ i\in\{ENT\}K_{i,t}I_c$
5:	$L_{A,t0}S$
6:	$K_{A,t0}(\forall i,\ i\in\{ENT/A\},\forall t,\ \neg L_{i,t}S)$
7:	$K_{A,t0}(\forall i,\ i\in\{ENT/A\},\forall t,\ t < t2,\ \neg L_{i,t}R)$
8:	$K_{B,t0}(\forall i,\ i\in\{ENT/B\},\forall t,\ t < t3,\ \neg L_{i,t}E)$

Fig 3: Initial Assumptions of the Realised Identification Scheme

Initial assumptions 1 to 4 state that the public information of the claimant, such as *L*, *n*, *c* and I_c, are known to all entities while assumptions 5 and 6 state that its secret information, namely *S*, is known only to itself.

Assumptions 7 and 8 refer to the timely revelation of *E* and possibly *R*. These state that *R* and *E* are not known before the end of steps 2 and 3 respectively, thereby indicating that *R* and *E* are fresh random numbers.

5.3 *Message Exchanges*

In order to verify the goals of the protocol it is necessary to analyse its message exchanges. The message exchanges for the protocol are summarised as follows:

Step 1:	A $\rightarrow$ B:	c
Step 2:	A $\rightarrow$ B:	X
Step 3:	A $\leftarrow$ B:	E
Step 4:	A $\rightarrow$ B:	Y

Fig 4: Message Exchanges of the Realised Identification Scheme

Step1:

Looking more closely at the exchange of step 1: $K_{B,t1}(R(B, t1, c))$

By application of Axiom A6: $L_{B,t1}c \wedge K_{B,t1}(\exists i, i \in \{ENT/B\}, \exists t, t < t1, S(i, t, c))$

Now applying Inference rule R3: $K_{B,t1}(\exists i, i \in \{ENT/B\}, \exists t, t < t1, S(i, t, c))$

Using assumption 3 and the fact that *c* is the ID number of *A* for which no gain is made by falsifying it:

$\boldsymbol{K_{B,t1}(\exists t, t < t1, S(A, t, c))}$ **:Satisfies Goal 1**

Step 2:

If verifier B receives the message of step 2: $K_{B,t2}(R(B, t2, X))$

Applying Axiom A6 and inference rule R3: $K_{B,t2}(\exists i, i \in \{ENT/B\}, \exists t, t < t2, S(i, t, X))$

Using assumption 7 which reflects the timeliness of *X* (and *R*) and indicates the sender of *X*:

$\boldsymbol{K_{B,t2}(\exists t, t0 < t < t2, S(A, t, X))}$ **:Satisfies Goal 2**

Step 3:

If *A* receives the message of step 3: $K_{A,t3}(R(A, t3, E))$

Applying axiom A6 and inference rule R3: $K_{A,t3}(\exists i, i \in \{ENT/A\}, \exists t, t < t3, S(i, t, E))$

Using assumption 8, which reflects the timeliness and sender of *E*:

$\boldsymbol{K_{A,t3}(\exists t, t0 < t < t3, S(B, t, E))}$ **:Satisfies Goal 3**

Step 4:

If B receives the message of step 4: $K_{B,t4}(R(B, t4, Y))$

Applying axiom A6 and inference rule R3: $K_{B,t4}(\exists i, i \in \{ENT/B\}, \exists t, t < t4, S(i, t, Y))$

Given that *Y* can only be calculated if the other entity knows the secret *S* and random number *R* it can be stated using assumptions 5, 6 and 7 that: $K_{B,t4}(\exists t, t < t4, S(A, t, Y))$

Assumption 8 states that *E* is not known before step 3 therefore *Y* cannot be calculated until after step 3. This gives us the timely arrival of *Y* as follows:

$\boldsymbol{K_{B,t4}(\exists t, t3 < t < t4, S(A, t, Y))}$ **:Satisfies Goal 4**

5.4 *Summary*

From the above analysis it can be seen that the four desired goals of the protocol are fulfilled. If the protocol completes step 4 and the resulting calculation in step 5 (internal to the verifier) results in X = X' then a successful claimant identification has been made. If the protocol does not complete a full cycle, then neither principal has gained enough information to enable a false identity to be constituted.

CONCLUSIONS

In this paper we have presented a realised Ohta-Okamoto based identification scheme, which, because of its low data transfer requirements is ideally suited for use with power-line networks. We have also outlined the verification logic of Coffey and Saidha. This logic was used to represent the goals, initial assumptions and message exchanges of the realised identification scheme. Upon analysis of the message exchanges it was shown

that the four desired goals of the realised scheme are met, thereby proving the security properties of the realised scheme.

REFERENCES

[1] Fiat, A. and Shamir, A.: 'How to prove yourself : Practical Solutions to Identification and Signature Problems'. Proceedings Crypto `86, Lecture notes in Computer Science, Vol. 263, Springer Verlag, 1987

[2] Ohta, K. and Okamoto, T.: 'A modification of the Fiat-Shamir scheme'. Proceedings Crypto `88, Lecture notes in Computer Science, Vol. 403, Springer Verlag, 1987

[3] Guillou, L.C. and Quisquater, J.J.: 'A practical Zero-Knowledge protocol fitted to security microprocessors minimising both transmission and memory'. Proceedings Eurocrypt `88, Lecture notes in Computer Science, Vol. 330, Springer Verlag, 1988

[4] Coffey, T. and Newe, T.: 'Realisation of a Minimum-Knowledge identification and signature scheme'. Computers & Security, Elsevier, Vol. 17, No 3, 1998.

[5] Coffey T. and Saidha P: 'Logic for verifying public-key cryptographic protocols'. IEE Proceedings - Comput. Digit. Tech., Vol. 144, No. 1, January 1997. Pages 28-32.

[6] Burrows M., Abadi M., and Needham R.M: 'A Logic of Authentication'. DEC Systems Research Report No. 39. February 1990.

[7] Hoare C.A.R: 'An axiometric basis for computer programming'. Communications of the ACM. Vol.12, No.10, (October 1969), Pages 576-580, 583.

[8] Coffey T, and Saidha P.: 'Non-Repudation with mandatory proof of receipt.' ACM Computer Communication Review. Vol. 26, No. 1, January 1996. ISSN # 0146-4833.