Symmetry in Security Protocol Cryptographic Messages – A Serious Weakness Exploitable by Parallel Session Attacks

Anca Jurcut*, Tom Coffey*, Reiner Dojen*

* Department of Electronic & Computer Engineering,
University of Limerick,
Ireland.
E-Mail: anca jurcut @ul.ie, tom.coffey@ul.ie, reiner.dojen@ul.ie

***Abstract** – This paper is concerned with detection and prevention of weaknesses in the design of security protocols. These weaknesses can be exploited by an attacker mounting attacks that compromise the security of the protocol.*

A novel theory defining weaknesses caused by the symmetry of cryptographic messages in protocols is introduced. This theory incorporates new rules describing the cases when the symmetry of messages has a structural weakness that is exploitable by parallel session attacks. Further, the rationale behind the Symmetry rules is presented and the structures of detected generic attacks for each rule are provided. Additionally, the Symmetry rules are applied to a protocol that is vulnerable to a parallel session attack. It is demonstrated that the proposed theory successfully detects the weaknesses caused by the symmetry of protocol messages, which lead to parallel session attacks.

Keywords: *Security protocols, weaknesses, parallel session attacks, attack detection, symmetry, cryptographic messages, Symmetry rules.*

I. INTRODUCTION

With the ever increasing use of distributed applications for meeting customer demands, there is a commensurate increase in the reliance of electronic communication over networks. Since many of these applications require the exchange of highly sensitive information over computer networks, the need for security protocols to ensure the protection and integrity of data is critical.

Many published security protocols have been subsequently been found to contain security weaknesses [1], [2], [3], [4], [5], [6], which can be exploited in an attack on the protocol.

A. Related Work

Formal verification of security protocols is concerned with proving that the goals of protocols are established and demonstrating the presence of any weaknesses that may be exploitable by mountable attacks.

Formal verification techniques can be categorized in two main classes: deductive reasoning [2], [7], [8], [9], [10], [11] and state exploration methods [12], [13], [14], [15]. Existing deductive reasoning methods include theorem proving and logic-based techniques. Deductive techniques are based on theories that represent the protocol faithfully comprising sets of axioms and deduction rules. Analysis of the protocol in that theory entails proving one or more theorems. State exploration methods take a quite different approach, which is more akin to simulation and testing. In state exploration, a protocol is characterized as the set of all its possible traces. Given the security protocol specification as input, the verification method explores as many execution paths of the protocol as possible, checking at each reachable state if some conditions hold.

In addition to deductive methods and state exploration, the possibility of using compiler-like static analysis methods such as type checking and data flow analysis for the purpose of formal verification of security properties has been investigated [16], [17]. However, these techniques are subject to several restrictions and are effective for verification of authenticity properties only. In addition to protocol verification, support in the development of security protocols such as: (i) protocol design guidelines and practices [18] (ii) protocol repair methods [19] for patching already disclosed faulty security protocols and (iii) protocol synthesis [20] have been proposed.

B. Contribution of this Work

In this paper we introduce a novel theory for the detection of weaknesses in the design of security protocols that may lead to parallel session attacks. The proposed theory is part of a new taxonomy of rules for detection of replay and parallel session attacks. This taxonomy includes new sets of rules addressing problems related to message symmetries, message freshness, handshake construction, signed statements and certificates. The weaknesses considered in this paper

are caused by the symmetry of cryptographic messages in the protocols.

The introduced theory incorporates four new rules describing the cases when the symmetry of cryptographic messages in a protocol is a serious weakness that is exploitable by parallel session attacks. The rationale behind these rules is detailed and the general structure of the detected attacks for each such rule is provided. Further, the applicability of the new rules for detection of the presence of weaknesses caused by symmetry of messages is demonstrated by application to a protocol vulnerable to a parallel session attack. Additionally, solutions to overcome these protocol weaknesses are proposed.

The proposed theory, expanded to incorporate new rules for message freshness, handshake construction, signed statements and certificates, is meant to be formalized and incorporated into a new formal attack detection logic. This logic will characterize the general circumstances under which a potential attack may exist (e.g. the crucial role played by messages without certain secrecy attributes at specific stages of a protocol) and will define a logical formula that describes such a possibility. In addition, the proposed logic will provide the structures of the exploitable attacks that can be mounted on a protocol.

II. SECURITY PROTOCOLS AND ATTACKS

A security protocol is a communication protocol that is based on a cryptographic system. A protocol is a prescribed sequence of interactions between communicating parties designed to achieve certain goals. These parties may be users, hosts, mobile devices or processes and they are referred to as principals. An honest principal follows a particular communication protocol, while a dishonest principal (or attacker, intruder, spy, enemy, adversary, etc.) tries to manipulate the protocol to achieve an unfair advantage.

The attacker can be either passive (just listening to communication) or active (modifying message content, message order, dropping messages, etc). An active adversary is more dangerous than a passive eavesdropper. Further, an adversary might use simultaneous protocol runs to gain information. One has to keep in mind that an adversary will not play fair by any rules. Hence, a security protocol should attempt to achieve its goals under all possible circumstances.

An attack on a protocol is a sequence of actions performed by an attacker, by means of any hardware or software tool, in order to subvert protocol goals. The attacker is a dishonest participant (also called 'adversary' or 'intruder') that may be a member of the system, i.e., a legitimate user, or external to the system.

A security protocol should enforce the data exchange between honest participants, while the dishonest ones should be denied any benefit of it. However, security protocols can contain weaknesses which exploit the conceptual structure of the protocols and are independent on the cryptographic systems. These weaknesses make them vulnerable to a range of attacks such as freshness, parallel session and type-flaw attacks.

A *freshness attack* is one of the most common attacks on authentication and key-establishment protocols. If the messages exchanged in an authentication protocol do not carry appropriate freshness identifiers, then an intruder can get itself authenticated by replaying messages copied from a legitimate authentication session. The intruder, according to the Dolev Yao model [7], has complete control over the communication channels. The intruder knows the session keys and short-time secrets generated in previous runs of the protocol. In a system in which sequential runs of the protocols are performed, it is assumed that the intruder has the capacity to replay messages from an old session run of the protocol that might contain a possibly compromised session key. Examples of freshness attacks on security protocols include [2], [4]. Freshness attacks can be foiled by the use of nonces, run identifiers and timestamps.

A *parallel session attack* utilizes the parallel execution of multiple protocol runs, where the intruder uses messages from one session (*the reference session*) to synthesize messages in the other session (*the attack session*). Examples of parallel session attacks on security protocols include [3], [4], [6]. There are several forms of parallel session attacks such as *oracle attacks*, *man-in-the-middle attacks or multiplicity attacks.* In an *oracle attack* the intruder starts a new run of the protocol and uses one of the principals as an oracle. Thus, the intruder has the appropriate answers for the challenges in the main run of the protocol. *A man-in-the-middle attack* occurs when two principals believe they are mutually authenticated, when in fact the intruder masquerades as one principal in one session and as the other principal in another. There are two runs of the protocol, but the honest principals think that there is only one. *A multiplicity attack* is a parallel session attack that can occur when the principals disagree on the number of runs they think they have successfully completed with each other.

A type flaw attack involves the replacement of a message component with another message of a different type by the intruder (e.g. accepts a nonce as a key). Examples of type flaw attack on security protocols include [3], [5].

III. CAUSES AND PREVENTION OF EXPLOITABLE WEAKNESSES IN SECURITY PROTOCOLS

In this section we outline the BAN simplified version of Yahalom protocol [2] and the parallel session attack published on this protocol by Syverson in [8]. Further, the causes of the exploitable weaknesses leading to this attack and the solutions to prevent these weaknesses are discussed. Additionally, similar weaknesses which are the cause of numerous such parallel session attacks [2], [4], [6], [10] on other protocols are briefly introduced.

A. BAN simplified version of Yahalom Protocol

The BAN simplified version of Yahalom protocol (Figure 1) was proposed in [2].

1. **A → B : A, Na**
2. **B → TTP : B, Nb, {A,Na}Kbttp**
3. **TTP → A : Nb, {B,Kab, Na}Kattp, {A,Kab,Nb}Kbttp**
4. **A → B : {A,Kab,Nb}Kbttp, {Nb}Kab**

Figure 1. The BAN simplified version of Yahalom Protocol

The target of this protocol is to enable two principals A and B in the protocol to share one common key Kab distributed by the authentication server TTP, which is trusted by both principals. Each principal is assumed to share a symmetric key with the server TTP (Kattp - the symmetric key shared between A and TTP and Kbttp - the symmetric key shared between B and TTP, respectively). Principal A starts the protocol by communicating to B its intention to share a new session key with it (step 1). Principal B generates a fresh nonce, Nb and creates a new message containing the identity of A and its nonce Na encrypted with the key Kbttp, shared with the trusted server. B sends its identity, the fresh nonce Nb and the encrypted message to the server (step 2). Server TTP deciphers the encrypted message, obtains the identity of A and generates a new fresh key Kab. It also builds two messages encrypted with Kattp and Kbttp, and sends the whole message to principal A (step 3). Finally, A deciphers the first encryption and checks whether it contains nonce Na. If this is the case, A sends to B the second message encrypted with Kbttp and mutually authenticates to B by sending nonce Nb encrypted with the new fresh session key (step 4).

B. The Parallel Session Attack on BAN simplified version of Yahalom Protocol

The BAN simplified version of Yahalom protocol is subject to a parallel session attack [8] as outlined in Figure 2. The intruder I intercepts an initial message from A to B. On receiving this message, I initiates a new protocol run with principal A (denoted by ii) using nonce Na as a challenge, while masquerading as B. The second protocol run proceeds as follows: intruder I modifies the message from A to TTP, replacing the new nonce N'a by the old nonce Na. I also intercepts the message from TTP to B and simply replays the encrypted components ({A, Kab, Na}Kbs and {B, Kab, Na}Ka) of this message back, as encrypted components of message 3 in the first run (denoted by i), but with the order of components switched.

At the end of the attack A is convinced it has engaged in two protocol runs with B. However, B is absent in both protocol runs and the intruder I is able to spoof principal A into thinking that it has exchanged a key with principal B, when in fact B is absent.

i.1. A -> I(B) : A, Na
ii.1. I(B) -> A : B, Na
ii.2. A -> I(S) : A, N'a, {B, Na}Kas
iii.1. Omitted
iii.2. I(A) -> S : A, Na, {B, Na}Kas
iii.3. S -> I(B) : Na, {A, Kab, Na}Kbs, {B, Kab, Na}Kas
i.2. Omitted
i.3. I(S) -> A : Ni, {B, Kab, Na}Kas, {A, Kab, Na}Kbs
i.4. A -> I(B) : {A, Kab, Na}Kbs, {Ni}Kab

Figure 2. The attack on BAN simplified version of Yahalom Protocol

C. Discussion on the Causes and on the Prevention of the Exploitable Weaknesses Leading to This Attack

The above attack can be mounted due to the symmetry of the cryptographic messages exchanged in steps 3 and 4 of the protocol between TTP and principals A and B. The weakness exploited by this attack is that the two cryptographic expressions: {B,Kab,Na}Kattp (step 3) and {A,Kab,Nb}Kbttp (step4) have the same structure. Hence, an intruder I can impersonate B in a parallel run, where the cryptographic expression {B,Kab,Na}Kattp obtained in step 3 of a parallel run is used in step 3 of the first protocol run. As the cryptographic expression required in step 3 of the first run has the same structure (is symmetric) with {B,Kab,Na}Kattp of step 3 of the parallel run, the principal A cannot distinguish the replayed {B,Kab, Na}Kattp in step 3 of the first run, send by the intruder, from a legitimate cryptographic expression sent by TTP.

This weakness can be overcome by breaking the symmetry of the two cryptographic expressions {B,Kab, Na}Kattp and {A,Kab,Nb}Kbttp.

The symmetry of the two messages can be broken either:

- By the addition of another component in the content of one of the cryptographic expression: for example, the identity of principal B in the content of {A,Kab,Nb}Kbttp resulting in {A,B,Kab,Nb}Kbttp.
- By changing order of the components in one of the cryptographic expression: for example {A,Kab,Nb}Kbttp can be modified as {Kab,A, Nb}Kbttp.

D. Similar Weaknesses Caused by Messages Symmetry

Similar weaknesses have been the cause of numerous parallel session attacks on other protocols, for example: the attack on Wide mounted frog [2]; the attack on the Lowe's modified version of the Wide-Mouthed Frog protocol discovered by Dojen, Jurcut, Coffey and Gyorodi [6]; the attack on KSL protocol [10]; the attack on Lee et al authentication: Nonce-based user authentication scheme using smart cards [4].

IV. SYMMETRY OF PROTOCOL MESSAGES – A SERIOUS WEKENESS EXPLOITABLE BY PARALLEL SESSION ATTACKS

In this section we introduce a novel theory regarding symmetry of protocol messages. This theory incorporates four rules (hereafter referred to as the Symmetry rules) describing the cases where the symmetry of messages in a protocol is a serious weakness exploitable by parallel session attacks. These Symmetry rules use four novel properties regarding cryptographic transformations: *parent cryptographic transformation*, *symmetry of a pair of cryptographic transformations*, *principal value type equivalent* and *travelling in opposite directions*.

These rules detect the presence of weaknesses that are exploitable by parallel session attacks if parent cryptographic transformations are symmetric and principal value type equivalent when travelling in opposite directions. The four rules in this category deal with different cases of travelling in opposite directions.

A. Definitions

Definition 1: A *cryptographic transformation* c is either a cryptographic expression ({x}k) or a hashed expression (H(x)).

Definition 2: A *parent cryptographic transformation* is a cryptographic transformation which is not contained within another cryptographic transformation.

Definition 3: Two messages x and y are *symmetric* if both contain the same components in the same order and of the same type.

Definition 4: Two keys k1 and k2 are *matching keys* if either:

- keys k1, k2 are both symmetric keys and have the same value
- keys k1, k2 are both symmetric keys and are shared with a TTP
- keys k1,k2 are both public keys
- keys k1,k2 are both private keys.

Definition 5: Two cryptographic expressions {x}k1, {y}k2 are *symmetric* if x, y are symmetric and keys k1,k2 are matching keys.

Definition 6: A principal G is the *generator* of a message x in a step Sr of a protocol P, if x cannot appear prior to step Sr. By convention, the generator is the sender of the step Sr, in which x appears for the first time.

Definition 7: Two messages, x and y are *principal value type equivalent* (*Pvte(x,y)*) if for each subcomponent x_i at position i of x that is of type principal there is a corresponding subcomponent y_i at the same position i of y that is also of type principal and at least one of the following also holds:

- if x_i is a trusted third party (TTP) then y_i is also a trusted third party (TTP)
- if x_i is the generator of x then y_i is the generator of y
- if x_i is not the generator of component x then y_i is not the generator of y
- if x_i is the intended recipient of x then y_i is the intended recipient of y
- if x_i is not the intended recipient of x then y_i is not the intended recipient of y.

Definition 8: Two cryptographic transformation c1, c2 are *travelling in opposite direction* if they are exchanged directly or indirectly between two principals. The following four cases arise:

1. Two principals G and R exchange cryptographic transformations c1 and c2 directly.
2. A trusted third party TTP is involved, but does not create the cryptographic transformations c1, c2: The principals generating c1, c2, respectively are sending it to the TTP.
3. A trusted third party TTP is involved, but only creates one cryptographic transformation c2. The intended recipient of c2 is different from the generator of c1.
4. A trusted third party TTP is involved and creates both cryptographic transformations c1, c2. The intended recipient of c1 must be different to the intended recipient of c2.

B. Symmetry Rules

Consider two symmetric parent cryptographic transformation c1 and c2 belonging to a message exchange (denoted En(P)) of a protocol P. If c1 and c2 are exchanged directly or indirectly between two principals, the following rules can be defined:

Rule 1: *Two principals G and R exchange the symmetric parent cryptographic transformations c1, c2 directly. If c1 and c2 are principal value type equivalent pairs and are travelling in opposite directions, then a parallel session attack can be can be mounted on En(P).*

Rule 2: *Two principals G and R exchange the symmetric parent cryptographic transformations c1, c2 indirectly; a TTP is involved, but does not create the cryptographic transformations c1, c2*: *The principals G and R generating c1, c2, respectively are sending it to the TTP. If c1 and c2 are principal value type equivalent pairs and are travelling in opposite directions, then a parallel session attack can be can be mounted on the message exchange En(P).*

Rule 3: *Two principals G and R exchange the symmetric parent cryptographic transformations c1, c2 indirectly; a TTP is involved, but only creates one cryptographic transformation c2. The intended recipient of c2 is different from the generator of c1. If c1 and c2 are principal value type equivalent pairs and are travelling in opposite directions, then a parallel session attack can be can be mounted on the message exchange En(P).*

Rule 4: *Two principals G and R exchange the symmetric parent cryptographic transformations c1, c2 indirectly; a TTP is involved and creates both cryptographic transformations c1, c2. The intended recipient of c1 must be different to the intended recipient of c2. If c1*

and c2 are principal value type equivalent pairs and are travelling in opposite directions, then a parallel session attack can be can be mounted on the message exchange En(P).

V. RATIONALE BEHIND SYMMETRY RULES

This section details the rationale behind the four Symmetry rules, presenting a reasoning for each rule. In addition, a general structure of the detected attacks for each of these rules is provided.

A. Rule 1

Assume cryptographic transformations c1 and c2 are sent in arbitrary steps Sq and Sr, respectively of a message exchange of a protocol P, denoted by En(P). The principals G and R exchange the two cryptographic transformations directly and c1, c2 are principal value type equivalent pairs and are travelling in opposite directions.

An attacker can impersonate R in a parallel run, denoted by $E^2n(P)$, where the cryptographic transformation c2 obtained in step S^1r (denotes step Sr belonging to first run) is used in S^2q (denotes step Sq belonging to second run). As the parent cryptographic transformation required in S^2q is symmetric and principal type value equivalent with c2 of step S^1r, the principal G cannot distinguish the replayed c2 in S^2q send by the intruder from a parent cryptographic transformation c2 send by R in a genuine message exchange.

The general structure of this attack is shown in Figure 3.

$$E^1n(P)\begin{cases} S^1q: G \to R: ...,c1,... \\ \quad ... \\ S^1r: R \to I(G): ...,c2,... \end{cases} \begin{cases} S^2q: I(R) \to G: ...c2,... \\ \\ S^2r: G \to I(R): ...,c3,... \end{cases} E^2n(P)$$

Figure 3. The structure of generic attack for Symmetry Rule 1

B. Rule 2

Assume cryptographic transformations c1 and c2 are sent in arbitrary steps Sq and Sr, respectively of a message exchange of a protocol P, denoted by En(P). The principals G and R exchange the two cryptographic transformations indirectly: a TTP is involved, but does not create the cryptographic transformations c1, c2. The principal G generates c1 and principal R generates c2. The cryptographic transformations c1 and c2 are principal value type equivalent and are travelling in opposite directions.

Essentially, the same rationale as in the case of *Rule1* can be considered: An attacker can impersonate R or/and G in a parallel run, $E^2n(P)$, where the cryptographic transformation c2 obtained in step S^1r is used in S^2q, or/and c1 of S^1q is used in S^2r, respectively.

The general structure of the attack is shown in Figure 4.

$$E^1n(P)\begin{cases} S^1q: G \to I(TTP): ...,c1 \\ \quad ... \\ S^1r: R \to I(TTP): ...,c2 \end{cases} \begin{cases} S^2q: I(R) \to TTP: ...c2,... \\ \\ S^2r: I(G) \to TTP: ...,c3,... \end{cases} E^2n(P)$$

Figure 4. The structure of generic attack for Symmetry Rule 2

C. Rule 3

Assume cryptographic transformations c1 and c2 are sent in arbitrary steps Sq and Sr, respectively of a message exchange of a protocol P, denoted by En(P). The principals G and R exchange the two cryptographic transformations indirectly: a TTP is involved, but only creates one cryptographic transformation, c2. The intended recipient of c2 is principal R and the generator of c1 is principal G. The cryptographic transformations c1 and c2 are principal value type equivalent and are travelling in opposite directions.

As in the case of previous two rules, an attacker can impersonate R in a parallel run, $E^2n(P)$, where the cryptographic transformation c2 obtained in step S^1r is used in S^2q. As the parent cryptographic transformation required in S^2q is symmetric and principal type value equivalent with c2 of step S^1r, the trusted third party cannot distinguish the replayed c2 in S^2q send by the intruder from a parent cryptographic transformation c2 send by R in a genuine message exchange.

The general structure of the attack is shown in Figure 5.

$$E^1n(P)\begin{cases} S^1q: G \to TTP: ...,c1,.. \\ \quad ... \\ S^1r: TTP \to I(R): ...,c2 \end{cases} \begin{cases} S^2q: I(R) \to TTP: ...c2,... \\ \\ S^2r: TTP \to G: ...,c3,... \end{cases} E^2n(P)$$

Figure 5. The structure of generic attack for Symmetry Rule 3

D. Rule 4

Assume cryptographic transformations c1 and c2 are sent in arbitrary steps Sq and Sr, respectively of a message exchange of a protocol P, denoted by En(P). The principals G and R exchange the two cryptographic transformations indirectly: a TTP is involved and creates both cryptographic transformations c1, c2. The intended recipient of c1 is principal G and the intended recipient of c2 is principal R. The cryptographic transformations c1 and c2 are principal value type equivalent and are travelling in opposite directions.

As in the case of the previous three rules an attacker can impersonate TTP in a parallel run, $E^2n(P)$, where the cryptographic transformation c2 obtained in step S^1r is used in S^2q. As the parent cryptographic transformation required in S^2q is symmetric and principal type value equivalent with c2 of step S^1r, principal R cannot distinguish the replayed c2 in S^2q send by the intruder from a parent cryptographic transformation c2 send by TTP in a genuine message exchange.

The general structure of the attack is shown in Figure 6.

$$E^1n(P)\begin{cases} S^1q : TTP \rightarrow G : ...,c1,.. \\ \quad ... \\ S^1r : TTP \rightarrow I(R) : ...,c2, \end{cases} \quad \begin{matrix} S^2q : I(TTP) \rightarrow R : ...c2,... \\ \\ S^2r : omitted \end{matrix} \Big\} E^2n(P)$$

Figure 6. The structure of generic attack for Symmetry Rule 4

VI. APPLYING THE PROPOSED SYMMETRY RULES FOR DETECTION OF WEAKNESSES EXPLOITABLE BY PARALEL SESSION ATTACKS

This section demonstrates the effectiveness of the proposed Symmetry rules for detection of the presence of weaknesses exploitable by parallel session attacks. The Symmetry rules are applied to BAN simplified version of Yahalom protocol (Figure 1).

A. Applying Symmetry Rules

According with *Definition 5* the parent cryptographic transformations {B,Kab,Na}Kattp, {A,Kab,Nb}Kbttp belonging to step S_3 and S_4, respectively are *symmetric* (1). Further, by *Definition 7* it follows that the previous pair of cryptographic transformations is *principal value type equivalent* (2). Additionally, by *Definition 8* the cryptographic transformations {B,Kab,Na}Kattp, {A,Kab,Nb}Kbttp are *travelling in opposite direction* (case 4), since TTP creates both cryptographic transformations and the intended recipient of {B,Kab,Na}Kattp is principal A, who is different to B, who the intended recipient of {A,Kab,Nb}Kbttp (3).

Combining (1), (2) and, (3) it follows by Symmetry Rule4 that a parallel session attack can be mounted on the message exchange $E_4(P)$, where $E_4(P)$ is considered to include all (four) steps of BAN simplified version of Yahalom protocol. Further, since $E_4(P)$ includes all (four) steps of the protocol it is implied that the parallel session attack can be mounted on the whole protocol.

B. Results

The application of our proposed Symmetry rules establishes that the BAN simplified version of Yahalom protocol is vulnerable to a parallel session attack. The parallel session attack allows an intruder I, to spoof principal A into thinking that it has exchanged a key with principal B, when B is in fact absent.

This parallel session attack can be prevented by breaking the symmetry of the two cryptographic messages. Application of the Symmetry rules to the corrected version of the protocol will show that no parallel session attack caused by symmetry can then be mounted.

VII. CONCLUSION

This paper introduced security protocols and common attacks that exploit weaknesses in the design of security protocols. The causes of exploitable weaknesses caused by symmetry of cryptographic messages in security protocols that lead to parallel session attacks were discussed. As a case study, BAN simplified version of Yahalom protocol was analyzed. Other references to similar weaknesses, causing such parallel session attacks on protocols were provided. A novel theory describing weaknesses caused by the symmetry of protocol cryptographic messages was proposed. This theory incorporates four new rules describing the cases when the symmetry of cryptographic messages in a protocol becomes a serious weakness exploitable by parallel session attacks. Further, the rationale behind these rules was detailed and the general structure of the detected attacks for each of these rules was provided. Furthermore, the applicability of the proposed rules for detection of the presence of weaknesses exploitable by parallel session attacks was demonstrated by application to the BAN simplified version of Yahalom protocol. It is shown that compliance with these Symmetry rules, successfully detects the weaknesses caused by the symmetry of protocol messages and leading to parallel session attacks. Additionally, solutions to overcome these weaknesses were proposed**.**

Acknowledgements

This work was funded by the Science Foundation Ireland - Research Frontiers Programme (11/RFP.1/CMS 3340).

REFERENCES

[1] Dazhi Sun and Zhenfu Cao. New cryptanalysis paradigm on a nonce-based mutual authentication scheme. International Journal of Network Security, Vol. 6, No. 1, pp. 116-120, 2008.

[2] M. Burrows, M. Abadi, R. Needham. A logic of authentication. ACM Transactions on Computer Systems TOCS, Vol. 8, No. 1, pp.18-36, 1990.

[3] Gavin Lowe, "Some new attacks upon security protocols", In IEEE Computer Society Press, editor, In Proceedings of the Computer Security Foundations Workshop VIII, pp. 162-169, 1996.

[4] Junghyun Nam, Seungjoo Kim, Sangjoon Park, Dongho Won, "Security Analysis of a Nonce-Based User Authentication Scheme Using Smart Cards", IEICE

Transactions Fundamentals, vol. E90-A, no. 1, pp.299-302, January 2007.

[5] J. Heather, G. Lowe, and S. Schneider, “How to prevent type flaw attacks on security protocols”, IEEE Computer Society, pp. 255–268, 2000.

[6] Reiner Dojen, Anca Jurcut, Tom Coffey, Cornelia Györödi: On Establishing and Fixing a Parallel Session Attack in a Security Protocol. Intelligent Distributed Computing, Systems and Applications. Springer Berlin / Heidelberg, Vol. 162, pp. 239-244, September 2008.

[7] L. C. Paulson. The inductive approach to verifying cryptographic protocols. Journal of Computer Security, 6(1), 1998.

[8] E. Cohen. First-order verification of cryptographic protocols. Journal of Computer Security, 11(2), 2003.

[9] Coffey, T. and Saidha, P., “A Logic for Verifying Public Key Cryptographic Protocols”, IEE Journal Proceedings-Computers and Digital Techniques, Vol. 144, No. 1, pp.28-32, 1997.

[10] Dojen, R. and Coffey, T., "Layered Proving Trees: A Novel Approach to the Automation of Logic-Based Security Protocol Verification", ACM Transactions on Information and System Security (TISSEC), Volume 8, Issue 3, pp. 287-311, August 2005.

[11] A. Datta, A. Derek, J. Mitchell, and A. Roy. Protocol composition Logic (PCL). Electron. Notes Theor. Comput. Sci, 172: 311-358, 2007.

[12] G. Lowe, “Casper: A compiler for the analysis of security protocols,” *Journal of Computer Security*, vol. 6, pp. 53–84, 1998.

[13] D. Basin, S. *Mödersheim*, and *L.* Vigano. *OFMC: A symbolic model checker for* security protocols, *International Journal of* Information Security, 4(3):181–208, 2005.

[14] C. Meadows, P. Syverson, Formalizing GDOI group key management requirements in NPATRL. In Proceedings of the 8th ACM conference on Computer and Communications Security, 235-244, Nov. 2001.

[15] Dong Song, Sergey Berezin, and Adrian Perrig. Athena: a novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, 9:47–74, 2001.

[16] C. Bodei, M. Buchholtz, P. Degano, F. Nielson, H. Riis Nielson, Automatic validation of protocol narration, Proceedings of the 16th IEEE Computer Security Foundations Workshop (CSFW 2003), IEEE Computer Society Press, Washington, pp. 126–140, 2003.

[17] Kikuchi, D., Kobayashi, N.: Type-based verification of correspondence assertions for communication protocols. In: Proceedings of APLAS 2007. Volume 4807 of LNCS., Springer-Verlag ,191–205, 2007.

[18] M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols," *IEEE Trans. Software Eng.,* vol. 22, no. 1, pp. 6-15, Jan. 1996.

[19] Lopez Pimentel, J.C., Monroy, R and Hutter, D.: A Method for Patching Interleaving-Replay Attacks in Faulty Security Protocols. *Electronic Notes in Theoretical Computer Science,* 174:117-130, 2007.

[20] A. Perrig and D. Song, “Looking for Diamonds in the Desert - Extending Automatic Protocol Generation to Three-Party Authentication and Key Agreement Protocols,” in *Proc.13th IEEE Computer Security Foundations Workshop*, Cambridge, UK, 2000.

[21] D. Dolev and A. C. Yao. On the Security of Public Key Protocols. IEEE. T. on Information Theory, Vol. 29, No. 2, pp.198-208, 1983.

[22] Paul Syverson. A taxonomy of replay attacks. In *Proceedings of the 7th IEEE Computer Security Foundations Workshop*, pages 131-136. IEEE Computer Society Press, 1994.

[23] Axel Kehne, Jürgen Schönwälder, and Horst Langendörfer. Multiple authentications with a nonce-based protocol using generalized timestamps. In *Proc. ICCC '92*, Genua, 1992.