An Approach for the Analysis of Security Standards for Authentication in Distributed Systems

H. A. Eneh and O.Gemikonakli

School of Computing Science, Middlesex University,
White Hart Lane, London N17 8HR, UK

Abstract. In this paper, we present our analysis of the leading standards for authentication in distributed systems and the inference rule used for our analysis. The inference rule here is similar to that used in the finite proof system of [3] and thus, is of the same family. However the rule of [3] can only reveal vulnerabilities of simpler protocols similar to Woo and Lam. Our inference rule proved that Kerberos version 5 remains vulnerable in scenarios of an attacker having reasonable communication and computational power especially in a single broadcast network. This vulnerability can aid a masquerade participating in the protocol. We also prove the possibility of a masquerade attack when an intruder participates in the SAML protocol. Though our inference rule, as part of our pre-emptive protocol tool is still in early stages of development, it has the potential to reveal subtle flaws that may not be detected by inference rules of the same family.

1 Introduction

Analysis of security products is becoming the most rapidly evolving discipline. This trend is necessitated by the continuous growth of attacks against computer network resources. Security mechanisms and their standards are required to ensure that both systems and network resources are not subject to abuse. The requirements of security mechanisms include those of authentication, integrity, confidentiality, etc. In any case, it is highly necessary to reaffirm that security mechanisms offer the services they are purported to as their failure can have devastating effects. Some authentication standards have gained high level of acceptance but nonetheless, it remains necessary to continue to review their operational procedure. In this paper, the procedure of interest is that of protocols for security standards of Kerberos and SAML. Protocols are the sequence of communication and computation steps followed by communicating principals to secure communication. Attackers, who also can participate in protocols, take advantage of the information deducted from the protocols steps. Inference rules are used to determine the information that principals, including attackers, can deduce from protocols' steps. In the perspective of protocols for distributed systems authentication services, analysis becomes more complicated due to functional requirements of such systems. A distributed system is characterised by a number of factors such as autonomy, size, heterogeneity, and fault tolerance [13],[24]. These characteristics of distributed systems present a requirement that a

A. Eneh H. and Gemikonakli O. (2005).
An Approach for the Analysis of Security Standards for Authentication in Distributed Systems.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 21-30
DOI: 10.5220/0002574600210030

user operating a client system proves his identity for each service desired and a superfeted requirement that servers prove their identity to clients. This scenario is the purpose for authentication mechanisms whose standards are discussed and analysed in this paper.

Our view of an attacker is that of having sufficient communication and computation power as well as enough time to exercise undue control of the distributed system or network. An environment with the possibility of such attackers is described as that of high mutual suspicion. The solution in [4] offered a framework for three-party authentication in environments of mutual suspicion. We present here an inference rule for analysis of authentication standards such as Kerberos and SAML. Our method stems from those of [3] but with differences such as consideration for sets of information that can be used for inductive reasoning besides deductions. Owing to limitations of space, this paper is succinct but we hope to render a greater detail in our technical report soon to be ready for publication. The balance of this paper is organised in sequence as follows; standards and standard making bodies, authentication in distributed information processing systems, Kerberos, basic operation of Kerberos, Kerberos security considerations, security assertions language (SAML), basic operation of SAML, SAML security considerations, related work, a method for analysis of protocols, discussions, further work, and conclusion.

1.1 Need for the Continuity of Standards

This section explains the merits of standards, reasons for not discarding standards even when they show some weaknesses. The proliferation of computers and distributed computing has made it impossible for a single vendor to monopolise customers. From operational point of view, standards seek to promote interoperability of components made by different vendors which abound in inter and intra organisational computer network domains. An age long belief in the telecommunication industry is that standards are required to govern the physical, electrical, and procedural characteristics of communications equipment [24]. In as much as standards show a number of pitfalls such as the tendency to freeze technologies and the possibility of having multiple standards for a single purpose, the merits of having standards are a lot more deserving. These merits include among others those of market expansion and extension made possible by broader acceptance, in addition to conformity and flexibility of equipment use as a result of components being interoperable irrespective of vendors. Standards can be conceptually categorised as voluntary, regulatory, or regulatory use of voluntary standards [24]. The distinctive features of these concepts are such that manufacturers and/or vendors, have the freedom to exercise their discretion to adopt voluntary standards, governments and their agencies setup regulatory standards such as health, safety, and economy, while regulatory use of voluntary standards is a synergy of government agencies and manufactures. Their goal is to minimise standardisation tasks on government agencies, reduction of standards numbers, and promotion of co-operation between government agencies and manufacturers.

1.2 Authentication in Distributed Information Processing Systems

The case for authentication in distributed computing systems is such that by convention, the information relating to organisational and technical standards are found in documents published by NIST, ANSI, and ISO/IEC, which are government agencies, while OASIS, Liberty, and WS-Secure are some of the voluntary consensus industry standards. It is also worthy to mention that IETF, an organisation under Internet Society that oversees protocol engineering and development on the Internet with voluntary membership, comprises working groups just like its siblings namely Internet Architecture Board (IAB) and Internet Engineering Steering Group (IESG). For authentication purposes and within the IETF's area of security, the following working groups exist; Kerberos, IPSec, X.509, S/MIME, and TLS.

1.2.1 Kerberos

Kerberos is a very popular distributed system and open network authentication service. Kerberos as an authentication system was based on Needham-Schroeder protocol [20] and became part of the MIT's project Athena of the mid 1980s. Kerberos is now in its fifth release, version 5, an improvement of version 4. Though, version 4 is still in commercial implementation, it exhibited vulnerabilities such as reliance on symmetric encryption, dependence on IP addresses, and others that were attributable to the Athena environment [12]. The success of password guessing and replay attacks against Kerberos and weaknesses as a result of Kerberos' requirement of a trusted path have been clearly identified as limitations of Kerberos [21][1]. A lot of the shortcomings of version 4 have been addressed in version 5 and currently, version 5 is treated as the standard Kerberos even in this paper.

Basic Operation of Kerberos
Authentication in Kerberos requires a client, say C, to send a request to the authentication server, AS, requesting credentials for a given server, application server V. The AS responds with the requested credentials consisting of a ticket and a session key encrypted with the client's key [12]. Kerberos exchanges may also be in the presence of a ticket granting server, TGS. We adopt the notation $X \rightarrow Y : Z$ to represent message transmission from a principal X to a principal Y with the content Z. A more detailed treatment of Kerberos operation is given in [24].

1. $C \rightarrow AS : Options \| ID_c \| Realm \| ID_{tgs} \| Times \| Nonce_1$
2. $AS \rightarrow C : Realm \| ID_c \| Ticket_{tgs} \| E_{kc}[K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}]$
3. $C \rightarrow TGS : Options \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c^b$
4. $TGS \rightarrow C : Realm_c \| ID_c \| Ticket_v \| E_{kc,tgs}[K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v]$
5. $C \rightarrow V : Options \| Ticket_v \| Authenticator_c^c$
6. $V \rightarrow C : E_{kc,v}[TS_2 \| Subkey \| Seq\#]$

Kerberos Security Considerations
There are some limitations of Kerberos and issues to consider when implementing an authentication service based on Kerberos. The following have been elicited from RFC 1510 and its improvements [12];

1. Denial of service attacks: as preventing this type of attack is beyond the capability of Kerberos, the detection and solution remain the responsibility of administrators and users or could be delegated to other applications.
2. Secrecy: participating principals are required to maintain the secrecy of secret keys. Failure in this regard can grant an intruder the impetus to masquerade as a targeted principal.
3. Dictionary attacks: Kerberos does not provide security for password guessing attacks. Users are advised to adopt relevant password management procedures.
4. Clock synchronisation: message times are used to ascertain freshness to avoid replay attacks. There is need for the principals' clocks within the network to be loosely synchronised to enable the servers, in particular, to reliably detect replays.
5. Identifiers recycling: the identifiers of the principals that exit the network should be removed to avoid the possibility of a new principal inheriting the privileges of another principal that had earlier exited. It is remarkable that principal identifiers are not recycled on a short-term basis in Kerberos.

1.2.2 Security Assertions Markup Language (SAML)

SAML is an OASIS standard that encodes security assertions and related protocol messages in XML format [22]. Fundamentally, the web platform includes some basic standards; XML, SOAP, WSDL, and UDDI. The industry significance of SAML is towards the reduction of user management costs. These costs can be reduced by using the instrument of federated identity management enabled by single sign-on systems such as SAML. The way SAML works is that a user signs on with a trusted asserting party, credible authority. The credible authority speaks to others, relying parties, about the authenticity of the user. The openness of the SAML standard made it possible for other protocols such as Liberty to be built upon it [10]. For efficiency of use, SAML is browser-based to eliminate the necessity of protocol-specific software packages. The SAML single sign-on protocol constrains the methods used to transfer messages with several security properties and builds on SOAP with its SOAP over HTTP binding [19].

Basic Operation of SAML
As stated earlier, SAML works by a user signing on with a trusted asserting party that speaks to other relying parties about the credibility of the user. By convention, the asserting party is regarded as the SAML responder. User's target or application server is regarded as the SAML requester, while the user's browser host is regarded as the user agent that supposedly conducts the initial user authentication. Fig. 1 shows the message flows, as indicated in [22]. Fig. 1 is used to illustrate the basic operation of the SAML protocol. Applying the same symbolic representations that were used to describe the operation of Kerberos, the following statements explain the message exchanges among the entities of the SAML protocol shown in Fig. 1.

Message 1. User Agent → SAML Requester : Request to initiate protocol exchange
Message 2. SAML Requester → User Agent : artefact containing SAML request delivered by HTTP GET request to the SAML responder.

Message 3. SAML Requester → SAML Responder : HTTP GET (artifact)
Message 4. SAML Responder → SAML Requester : Samlp:ArtifactResolve>request containing (artifact) of message 3.
Finally, if all goes well when necessary conditions are met, then
Message 5. SAML Requester → SAML Responder : <Samlp:ArtifactResponse>

SAML Security Considerations
The following are the major highlights of SAML security considerations elicited from [11] and [22];

1. The existence of user agent intermediary implies that the requester and responder cannot rely on the transport layer for end-to-end authentication, integrity and confidentiality protection, and must authenticate the messages received instead.
2. SAML provides for a signature on protocol messages for authentication and integrity for such cases that they are required.
3. Whereas it is binding that the recipient verifies that the location at which message is received matches the destination XML attribute in the root SAML element (i.e. destination URL), for signed messages; this binding SHOULD NOT be used in the user agent intermediary for purposes of confidentiality. If confidentiality is necessary, SSL 3.0 or TLS 1.0 should be used to protect message in transit. In general, this binding relies on message-level authentication and integrity protection via signing and does not support confidentiality from the user agent intermediary.

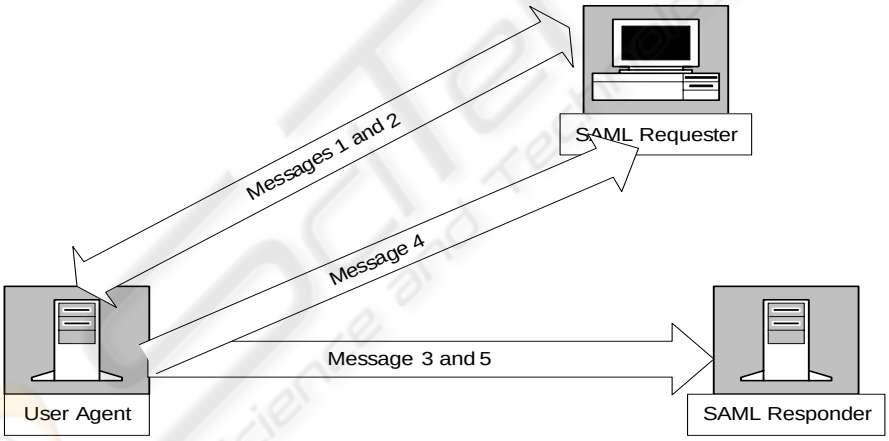

Fig. 1. Message exchanges of the entities of the SAML protocol.

2 Related Work

The work of [3] set the precedence for our current work. Their formal and automatic approach takes protocol specifications as input and generates possible flaws and their

attack scenarios. Three steps were used by [3] for analysis of cryptographic protocols; the generation of protocol roles from classical protocol description, the automatic deduction of intruder computational abilities as a finite proof system, and finally, the verification process using the roles and the proof system.

Reference [8] used closed networks to compare the performance of variants of Kerberos, using a public key infrastructure or adding a proxy server. [15] outlined the significance of adopting appropriate definition for authentication prior to use in order to avoid misplacement of trust in certain authentication protocols. [15] asserted that the strength of the authentication protocols range from fundamentally satisfying the requirement of aliveness to a more secured assurance of full agreement with recentness. Due to the feasibility of attacks such as connection hijacking/replay, man-in-the-middle, and HTTP referrer attack, [7] stated that further analysis of the SAML protocol is still very necessary despite the fact that SAML single sign-on protocol is well designed and carefully described.

Reference [23] categorised the approaches for the analysis of cryptographic protocols. They showed that majority of the efforts adapted to model the requirements of a protocol family using logics developed for the analysis of knowledge and belief. This category is classified as type III [18]. The approach presented in [23] stems from the use of BAN logic [2]. The object of the logics and their likes is to ascertain that authentication protocols assure mutual authentication between authenticating principals as well as qualitative key management. However, [6] warned that using logics should be approached with a lot of caution. Evidence exists for the improper use of the logics as cited in the discussion of BAN logic [23] and by the observations of [16]. The BAN logic, for instance, places authentication protocols in the perspective of the state of belief among principals rather than proving security. On the other hand are specialised tools that are characterised by state exploration. These tools appear to have attracted higher research attention somewhat as a result of Lowe's use of model checker, FDR, to find a man-in-the-middle attack against the Needham-Schroeder public key protocol [14]. Despite the developments in the analysis of security protocols; exploring knowledge and belief, and exploring finite state conditions, it has been frequently pointed out in literature that the strength of security protocols is yet to sublime and that the security protocols problem remains undecidable [17]. Thus, it is still highly necessary to increase vigour in the development of security protocols in general especially with the tremendous increase in number and complexity of attacks.

3 A Method for Analysis of Protocols

The entire phases of the proposed approach to the analysis and design of protocols have been described in [5]. The phases are organized in a cyclic framework as a way to depict automatic continuity. The phases include; specification, refinement, design, protocol specification analysis, implementation, and implementation verification. The phases are enshrined in a cyclic pattern to depict continuity and that the output of one phase is input to the subsequent phase. The phases are repeated whenever flaws are discovered by the automatic system itself or externally, changes in industry standard specifications or attack discovered by manual means. For brevity, *specification* is the

initial phase, *refinement* deals with the fine-tuning of the specification to remove ambiguities such as granting and denying a transaction in a complete run of a protocol, *design*, or otherwise the design with specification, deals with the actual design or adoption, as the case may be, of the protocol that matches the specification, *protocol specification analysis* deals with all forms of automatic or manual discovery of attacks against the designed protocol for the authentication mechanism to be deployed, and finally, the *implementation* and *implementation verification* phases deal with the automatic code generation and confirmation respectively. The protocol specification analysis is the core phase of the analytical method.

3.1 Inference rule for reasoning about protocols

This section describes the inference rule used within the pre-emptive protocol verifier, PPV, which is the system used for protocol specification analysis, the core of the analytical method discussed in [5]. PPV verifies protocols by modelling all instances of attacks that can be mitigated against a protocol by initially identifying all roles involved in a protocol, which primarily include transmitting, receiving, and encrypting/decrypting messages. It then builds logic to decide the extent of information that could be available to an attacker, and finally proceeds to verification by considering the possible effects, potentials, the roles an attacker or principal can have with available information. It uses an inference rule of the form $[X|Y]\pi \Rightarrow M$. This inference rule has the implication that whenever the condition π is true, any principal, including an intruder, who supplies the set of information $x \in X$ or $y \in Y$ to a protocol obtains the message $m \in M$; X and Y represent the set of legitimate and malicious messages respectively. By also considering the use of nonces, unique identifiers, as a parameter for condition π as well as freshness condition, this approach slightly differs from those of [3]. Besides, it extends coverage to more protocol types due to the nature of this inference rule.

3.2 Application of inference rule to protocols of authentication standards

The inference rule introduced above is applied to Kerberos and this is presented here. The description here is succinct due to space limitations. A technical report due to be out soon will contain a more detailed discussion of the application of inference rule to test the robustness of Kerberos protocol.

3.2.1 Kerberos

As shown earlier, in an instance of public key Kerberos operation, a protocol of six steps is followed to authenticate a client C to an application server/verifier V. Here the scenarios considering only the messages that serve authentication purposes, which are the identifiers for the principals ID_p and nonces are presented.

Step 1 is then seen as $C \rightarrow AS : ID_c \parallel ID_{tgs} \parallel Nonce_1$. The inference rule that applies to this step is of the form " " $\Rightarrow ID_c, ID_{tgs}, Nonce_1$, where " " denotes an empty set. This implies that an intruder who intercepts the communication between a

legitimate client who initiates the protocol and an authentication server is able to obtain the identities of the client and that of the server as well as the nonce generated by the client.

The second step, AS → C : ID_c, $Ticket_{tgs}$, $E_{kc}\{K_{c,tgs},Nonce_1,ID_{tgs}\}$ where $E_{kc}\{\}$ is an encryption operation using the public key of C. This implies that an intruder who listened to the first step and transmitted the message in an attempt to masquerade as a legitimate client C, according to the inference rule, will receive relevant messages in the form $[ID_c ,ID_{tgs}] \Rightarrow ID_c$, $Ticket_{tgs}$, $E_{kc}\{K_{c,tgs},Nonce_1,ID_{tgs}\}$ where the latter is the authenticator containing the id of the ticket granting server tgs and a session key shared between the tgs and client C and nonce all encrypted with the public key of the client. Supposedly, only an uncompromised legitimate client can decrypt the authenticator to obtain the session key to be shared with the tgs. This scenario poses some difficulty to an intruder in a different broadcast network from the tgs. However, if by chance the tgs is on the same broadcast network with the intruder, the intruder will only have to send a broadcast including the identity of a targeted principal perhaps learnt earlier by step three of the protocol that is C → TGS : Options$\|$ ID_v $\|$Times$\|$ $Nonce_2$ $\|$ $Ticket_{tgs}$ $\|$$Authenticator_c^b$, only a valid tgs will come responding with step four of the protocol that is

TGS→C :$Realm_c\|$ ID_c $\|Ticket_v$ $\|E_{kc,tgs}[K_{c,v}\|Times\|Nonce_2\|Realm_v\|ID_v]$

From the proposed inference rule, this implies that

$[ID_v$, $Nonce_2$, $Ticket_{tgs}$, $Authenticator_c^b]$ $Nonce_1\Rightarrow$ ID_c, $Ticket_v$, $E_{kc,tgs}\{K_{c,v},Nonce_2$, $ID_v\}$. The latter is a second authenticator. The intruder arriving at this stage now posses a valid ticket to be used with the application server V together with a session key to be shared with the verifier, which is obtained by decrypting the second authenticator with the session key shared with the tgs that the intruder obtained earlier on. The intruder from there can begin to impersonate the tgs as well as the client C. Steps five and six of the protocol are shown as follows;

C → V :Options$\|Ticket_v\|Authenticator_c^c$

V → C : $E_{kc,v}[TS_2\|Subkey\|Seq\#]$

Going by the inference rule, these are read as

$[ID_c$, $Ticket_v$, $E_{kc,tgs}\{K_{c,v},ID_v\}]Nonce_2$, $\Rightarrow$ $E_{kc,v}[TS_2\|Subkey\|Seq\#]$. Thus, an application server responds with parameters for secured communication to an intruder having been deceived to believe that it is communicating with a legitimate client.

Discussion

Kerberos version 5 poses a greater difficulty to masquerade attack than version 4 hence the development. However, in distributed system scenarios where the intruder possesses reasonable communication and computational power within a broadcast network belonging to the same administrative domain, the version five of the protocol stands compromisable. Again, depending on the nature of implementation of the AS and the tgs, the chances of the intruder impersonating a principal is higher where AS and tgs are on the same broadcast network or even implemented on the same host.

3.2.2 SAML

The SAML protocol discussed in section 1.2.2 is a sensational “speak-for” protocol; the asserting party speaks to a requester about the authenticity of a principal acting

through an agent. Using our inference rule, we are able to deduce that an intruder who intercepts the first message of the protocol is able to masquerade as a legitimate SAML requester to obtain the authenticating credentials of a principal acting through the user agent. This scenario presents the intruder with the opportunity to act as a responder to subsequent SAML requests. The intruder can therefore, readily produce other instances of the principal being spoken for. Due to the space limitations, it is not possible to present the detailed analysis regarding this.

4 Further work

Here, the theoretical grounds of an inference rule and its implications are presented. There still remains the need to fine tune the logic in such a manner to ease program coding so that the inference rule can be integrated into a finite proof system. The finite proof system is yet to be employed in the form of an automatic protocol verifier as part of the proposed cyclic framework [5], a major step in the pre-emptive protocol design tool.

There is need to empirically compare existing verification systems and even against the background of the Athena verifier to determine the best fit for the nature of proof system to be employed in the pre-emptive protocol design proposed.

5 Conclusion

This paper presents an inference rule used in the analysis of authentication standards. The inference rule provides for both inductive and deductive reasoning about protocols. This is an improvement over other inference rules of the same family that provide for only possible deductions from protocols. This is illustrated using the additional field in the message set [X|Y], X for deductions (legitimate messages) and Y for inductions (malicious messages). Considerations for Y, reveal the relative ease of inducing session keys and therefore, possible masquerade as trusted servers such as the tgs when applied to Kerberos authentication. The fact that even the highly secured networks and computing resources remain vulnerable due to evolution of attacks, makes it highly necessary to adopt methods that can render the positions of networks and computing resources unassailable to attackers.

References

1. Bellovin, M. and Merritt, M.: Limitations of the kerberos authentication mechanism, Winter In Proceedings USENIX Conference, Dallas, Texas, USA. (1991)
2. Burrows, M., M. Abadi, and Needhan, R.: A Logic of Authentication, ACM Transactions on Computer Systems, 1990. 8(1), (1990), pp. 18-36.
3. Debbabi, M. et al: From protocol specifications to flaws and attack scenarios: An automatic and formal algorithm, Proceedings of the 6th IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprise, 0-8186-7967-0/97, (1997)

4. Eneh, H. A., Singh, H., and Gemikonakli, O.: A three-way authentication framework for IEEE 802.11b networks, Proceedings of the 4th International Network Conference INC'04, Plymouth, UK, ISBN 1-84102-125-3, (2004), pp 345-352
5. Eneh, H. A. and Gemikonakli, O., Analysis of security protocols for authentication in distributed systems, Proceedings of IADIS International Conference on Applied Computing, IADIS'05, Volume 2, Algarve, Portugal, ISBN 972-99353-6-X, (2005), pp 301-305
6. Gligor, V. D. et al.: Logics for cryptographic protocols – Virtues and limitations, Proceedings of the 4th IEEE CSFW, (1991), pp 219-226
7. GroB, T.: Security analysis of the SAML Single Sign-on browser/artifact profile, http//www.acsac.org/, (2003)
8. Harbitter, A. and Menascé, D. A. A methodology for analysing the performance of authentication protocols, ACM Transaction on Information and System Security, Vol 5, No. 4, (2002), pp 458-491.
9. Heintze, N. and Tygar, J. D.: A model for secure protocols and their composition, IEEE Transactions on Software Engineering, 22(1), (1996), pp 16-30.
10. Hodges, J. and Wason, T.: Liberty architecture overview, http://www.projectliberty.org/specs/
11. Hughes, J and Maler, E.: Technical overview of the OASIS Security Assertion Markup Language (SAML) V1.1, Draft 03, http://www.oasis-open.org/committees, (2004)
12. Kohl, J. and Neuman, B. C.: The Kerberos network authentication service (Version 5), Internet Request for Comments RFC 1510., (1993)
13. Lampson, B. et al.: Authentication in distributed systems: Theory and Practice, ACM Transactions on Computer Systems, 10(4), (1992), 265-310.
14. Lowe, G.: An attack on the Needham-Schroeder public-key authentication protocol, Information Processing Letters, 56, (1995), 131-133.
15. Lowe, G.: A hierarchy of authentication specifications, Proceedings of the 10th Computer Security Foundations Workshop, (CSFW '97), 1063-6900/97, (1997)
16. Mao, W. and Boyd, C.: Towards formal analysis of security protocols, Proceedings of Computer Security Foundation Workshop VI, (1993), pp 147-158.
17. Meadows, C.: Extending formal cryptographic protocol analysis techniques for group protocols and low level cryptographic primitives, Proceedings of the 1st Workshop on Issues in the Theory of Security, Geneva, (2000), pp 87-92.
18. Meadows, C.: Applying formal methods to the analysis of a key management protocol, Journal of Computer Security, 1(1), (1992), pp 5-35.
19. Mishra, P.: Bindings and protocols for the OASIS security assertions markup language (SAML), http://www.oasis-open.org/committees/security/, (2002)
20. Needham, R. and Schroeder, M.: Using encryption for authentication in large network of computers. Communications of the ACM, 21(12), (1978), pp 993-999.
21. Neuman, B. and Ts'o, T.: Kerberos: An authentication service for computer networks, 32(9), (1994) pp 33-38.
22. OASIS: Authentication Context for the OASIS Security Assertion Markup Language, (2004)
23. Rubin, A. D. and Honeyman, P.: Formal methods for the analysis of authentication protocols, Technical Report, CITI TR 93 – 7, (1993)
24. Stallings, W.: Cryptography and network security: Principles and practices, Third Edition, Prentice Hall, New Jersey, (2002)