

New Real Time Multicast Authentication Protocol

Riham Abdellatif¹, Heba K. Aslan², and Salwa H. Elramly³

(Corresponding author: R. Abdellatif)

Information and Systems Department, National Institute of Standards¹

Tersa St., El Haram, P.O. Box 136 Giza-Code 12211, Egypt

Informatics Department, Electronics Research Institute, Cairo, Egypt²

Electronics and Communications Engineering Department, Ain Shams University, Cairo, Egypt³

(Email: {rehlatif, hebaaslan}@yahoo.com), sramlye@netscape.net

(Received June 12, 2009; revised and accepted Nov. 15, 2009)

Abstract

Multicast gives professional large-scale content distribution by providing an efficient transport mechanism for one-to-many and many-to-many communications. There is a number of security issues in multicast communication directly related to the specific nature of multicast. In our paper, we concentrate on the multicast authentication problem. There are four important requirements of multicast communication protocols: to perform authentication in real-time, to resist packet loss and pollution attacks, to have low communication and computation overheads, and to have resistance to replay attacks. In this paper, a protocol for authenticating multicast data applications is proposed. In order to provide authentication, the proposed protocol uses both public key signature and symmetric key encryption. The proposed protocol resists packet loss by using erasure code functions over the signature. To resist pollution attacks, our protocol computes the symmetric encryption of the erasure code output. To resist replay attacks, a counter number is added to each packet. The proposed protocol is compared to other multicast authentication protocols. The comparison shows that the proposed protocol has low computation and communication overheads. The proposed protocol called Latif-Aslan-Ramly1 (LAR1) is analyzed using Burrows, Abadi and Needham (BAN) logic. The analysis shows that it achieves the authentication goals without bugs or redundancies.

Keywords: Authentication, multicast communication, real-time

1 Introduction

Multicast enables efficient large-scale content distribution by providing an efficient transport mechanism to communicate between one-to-many and many-to-many commu-

nications [4, 15, 20]. Over the years, multicast has been the topic of many researches.

Today, applications that are of multicast nature have increased (for example: video conference, distance learning, pay per view TV, financial stock quote distribution, etc). One of the important tasks for the use of multicast communication is the authenticity. Authenticity means that the receiver must be able to verify the identity of data's source. First level of functionality is for the receiver to be able to verify that the data is from a group member [7, 16, 18]. The next level of functionality is for the receiver to be able to verify that it is from an authorized sender. The most precise functionality is for the receiver to be able to determine the exact identity of the sender.

In case of multicast communication, authentication is a difficult problem, since it requires that a large number of recipients must verify the data sender. Assume a group containing n members. A simple solution is to use a shared symmetric key between the sender and each receiver to calculate different Message Authentication Codes (MACs). Then, the sender appends the calculated MACs to the group message. Upon receiving the message, each receiver ensures the authenticity of the message using the MAC calculated by the key shared between it and the sender. This solution has a high communication overhead since in order to ensure the authenticity of a message n MACs must be appended to it. Another solution is to use the private key of the sender to sign a hash of the entire message. This solution suffers from the high computation and communication overheads since the signature algorithms require large computation and produce large output signatures. The above mentioned solutions do not resist packet loss, since the loss of any packet of the message will cause the inability to authenticate the received packets. This is due to the fact that the MAC or the signature is calculated over the whole message. In order to resist packet loss, one solution is to calculate MAC

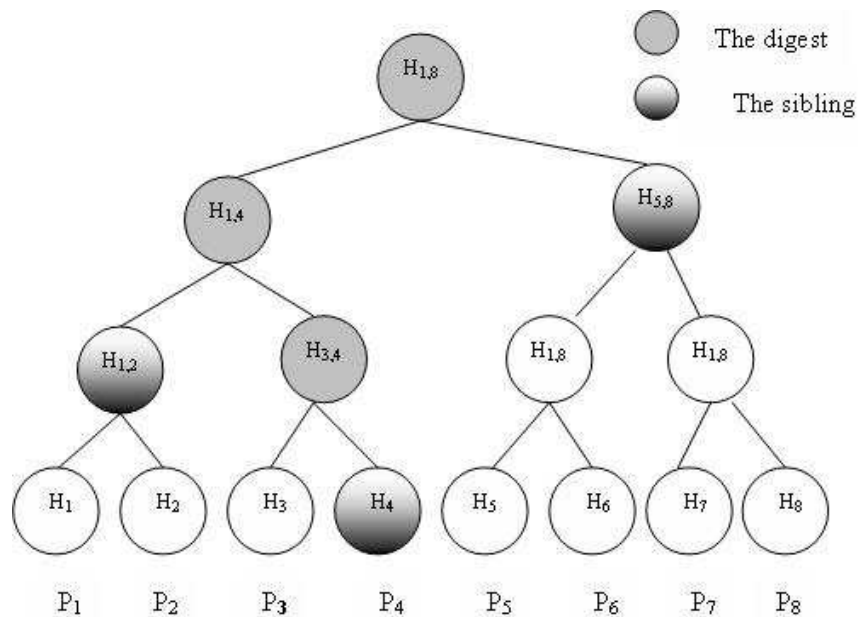


Figure 1: Tree chaining of Wong and Lam scheme [19]

or signature for every packet. This solution will suffer from a large amount of communication and computation overheads. Two solutions for providing multicast authentication are proposed. The first is to amortize signature over several packets [6, 8, 9, 12, 19]. The second is to design more efficient signature schemes [13, 14]. While designing more efficient signature schemes overcomes the computational overhead problem, it still suffers from the communication overhead problem. On the other hand, amortizing signature over several packets overcomes the communication overhead problem.

In this paper, we propose LAR1 which is a new protocol for multicast authentication. LAR1 uses both public key signature and UMAC function. The proposed protocol has low computation and communication overheads. Furthermore, it resists packet loss, pollution and replay attacks. The paper is organized as follows. In the next section, related works are reviewed. Then, a description of the proposed protocol is given in Section 3. Next, in Section 4, logical analysis of LAR1 is detailed. In Section 5, a comparison of the proposed protocol with other protocols is discussed. Finally, the paper concludes in the Section 6.

2 Related Work

The techniques proposed to solve the multicast authentication problem are divided into two main categories: design more efficient signature schemes and amortize the signature over several packets. For the first scheme, we have a technique called BIBA [13] proposed by Perrig. Perrig proposed a one-time signature and broadcast authentication protocol. It has a low verification overhead

and a relatively small signature size. BIBA enhances the computation overhead, but its communication overhead is still large. Reyzin and Reyzin [14] proposed a one-time signature scheme, which is faster than BIBA and has a slightly lower communication overhead. The two schemes are unsuitable for real-time applications for their large communication overhead.

For the second solution, that is to amortize the signature over several packets, we have an efficient scheme that has been proposed by Wong and Lam [19]. Although this scheme overcomes the computational problem, it suffers from the communication overhead problem. Wong and Lam proposed a tree chaining technique, this technique divides the stream of data packets into blocks of m packets ($P_1, P_2, P_3, \dots, P_{m-2}, P_{m-1}, P_m$) and forms a tree arrangement of degree 2 to perform authentication as shown in Figure 1. Each block of certain number of messages can be authenticated with one signature. Each leaf node is a message digest of a data packet, and the parent nodes are message digests of the two children nodes. The root node is the message digest for the block, which is signed once for the entire group. To verify the packets, the receiver recreates the path from the received packet up to the root, computes the digest of each node, and compares the computed root to the signed received root. Therefore, the sibling of each node along the packet's path must be sent to help the receiver to authenticate the packet. Although this work solves the problem of packet loss, it has a big problem that it needs high computation and communication overheads.

Perrig et al. proposed TESLA protocol [11]. TESLA protocol provides authentication without considering packet loss rate. It has many advantages like: Low computation overhead, Low per-packet communication over-

head, Strength to loss, every packet which is received in time can be authenticated and no buffering is needed at sender.

However, it requires time synchronization between the sender and the receiver; and in order to satisfy the security condition, the sending rate must be slower than the network delay from the sender to the receiver. In TESLA, the stream is divided into blocks of m packets. Then, the sender selects a random number and uses it as a key; say km then by applying a pseudo random function (F) m times, the sender can calculate m keys; where $k_{m-1} = F(km)$ and $k_{m-2} = F(k_{m-1})$ and so on. When the receiver received these keys, it uses them to authenticate the received stream by calculating the MACs.

Finally, Ritesh Mukherjee proposed the symmetric message authentication scheme [10]. This scheme is based on symmetric message authentication codes (MACs) to add the required data source authentication and data integrity check to secure the group communication. It uses asymmetric encryption, and a symmetric MAC. To provide authentication, a symmetric key is shared among all group members. This symmetric key is a unique shared secret used for authentication. Every time a sender wants to send a message to the group, it adds an index to the packet, a counter “ c ” and a random number “ k ”. The index is a number assigned by the group manager to a particular sender for uniquely identifying it during a multicast session. It then encrypts the packet with the asymmetric encryption key, and then it calculates a MAC on the ciphertext using the symmetric key. After that it attaches “ k ” and the MAC to the packet. The packet structure is shown in Figure 2.

Encrypted data				MAC, k
<i>index</i>	k	c	Data packet	

Figure 2: Packet structure of Ritesh Mukherjee scheme [10]

The receiver computes the MAC of the encrypted sent data, check the pollution attack by ensuring that the packet was not modified during transit, then decrypts the data and gets the random number k , now the receiver compares this k with the sent random number k , the sent index identifies the particular sender, this achieves the data source authentication. The symmetric key used is generated by the group manager and distributed to the group members with the private decryption key when the members join the multicast group. This protocol consumes large computation overhead. The receiver needs to calculate the MAC of the cipher, make a comparison operation, make a decryption operation, and make another comparison, which may not be practical in case of real

time applications. In the next section, a description of LAR1 is given.

3 Latif-Aslan-Ramly Multicast Authentication Protocol (LAR1)

In this section, we present a description of LAR1 and its features.

3.1 Description of LAR1

The detailed procedure of LAR1 protocol is given below.

At the sender side:

The stream is divided into blocks of m packets ($P_1, P_2, P_3, \dots, P_{m-2}, P_{m-1}, P_m$). The sender applies the digital signature on the group key K_g . The digital signature is done by any public key system like RSA [17]. The sender applies the erasure code function on the signature. The output of the erasure code function is partitioned into m symbols: $S_1, S_2, S_3, \dots, S_m$ as shown in Figure 3. Assume the erasure code function can resist packet loss of rate R . Each time the sender wants to send a message to the group, he adds a counter “ c ” to the packet. The system uses this counter to rearrange the received packets, and to resist replay attacks. Then the sender encrypts the output of the erasure code function $S_1, S_2, S_3, \dots, S_m$ using a symmetric key system like AES [5] by the group key K_g . The output of encryption will be divided into m symbols which will be appended to the m data packets before sending. Then, the sender calculates the UMAC [1] value of P_i , concatenated with the corresponding S_i key, and the corresponding counter c_i as shown in Figure 3. This last operation will be divided also into m symbols that will be appended with the m data packets. The use of MAC algorithm has the same security strength as hash functions with lower output length [17]. Finally, the sender output will be: the data packet appended with its counter, the UMAC output and the output of the encryption algorithm.

At the receiver side:

Each received packet P^i contains the following data.

Packet P_i itself, the counter c_i , the corresponding output of the UMAC function and the corresponding output of the encryption operation. Upon receiving the stream, each receiver makes a decryption operation on the encrypted data. Now, each receiver has the ability to authenticate each packet. First, it computes the UMAC function of the packet, the counter and S . If it equals the same received UMAC symbol, this implies that the received packet has been sent by one of the group members. If the two values are different, this implies that the received packet is corrupted and it is discarded. The receiver can calculate the authentication information using

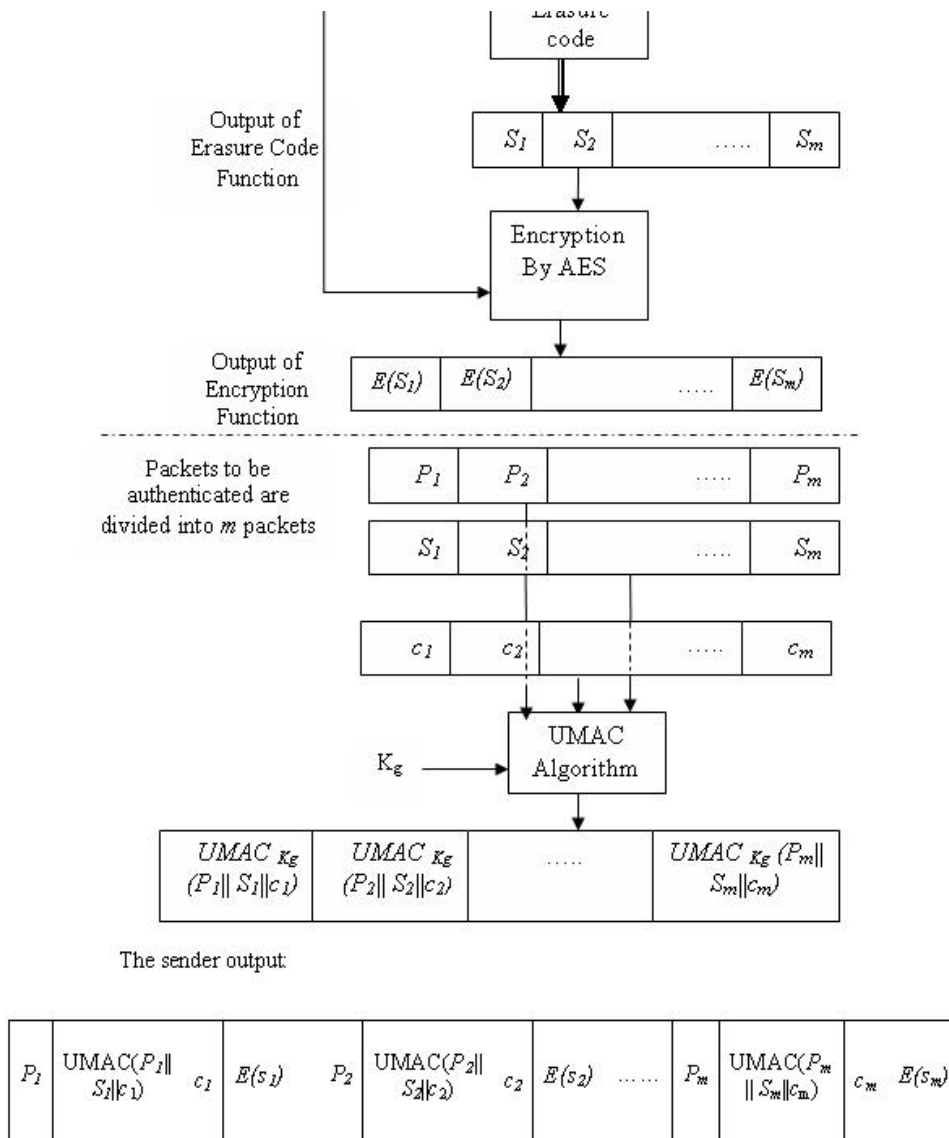


Figure 3: LAR1 protocol

the erasure decode function, after receiving $m(1 - R)$ correct packets. Then, it checks the digital signature and determines the exact message sender. The receiver stores the received packets in a buffer of length m for example. So the new arrived packet counter c_1 must fall within the range c_1 to c_{1+m} . If the received counter number already has an existing packet in the buffer then the received packet is considered a duplicate and is discarded. The counter c_i is sent free to give the ability to the receiver to compute the UMAC function, and is sent included in the UMAC function to preserve its security from intruders.

3.2 Features of LAR1 Protocol

- The digital signature is distributed along the data packets after applying the erasure code function on it so the system avoids the problem of signature loss

and sending the signature more than one time.

- The system has a resistance to packet loss as long as it is below a certain loss rate R .
- Each packet has a witness that helps us to decide to use this packet in the erasure decoding function or not. This witness is the result of applying the UMAC function on the result of the decryption operation concatenated with the sent packet. This means that LAR1 protocol overcomes the pollution attack problem.
- Each packet has a witness that helps us to decide to use this packet in the erasure decoding function or not. This witness is the result of applying the UMAC function on the result of the decryption operation concatenated with the sent packet. This means

that LAR1 protocol overcomes the pollution attack problem.

In the next section, logical analysis of LAR1 using BAN logic [3] is detailed.

4 Logical Analysis of LAR1

We use BAN logic to validate our model. This is done using logics developed specifically for the analysis of knowledge and belief. The analysis of our protocol is performed as follows:

- Assumptions about the initial state are given.
- Description of the protocol using Logical formulas is written.
- The logical formulas are applied to the assumptions and description formulas, in order to discover the beliefs held by the parties in the protocol.

We will assume that we have a group G that consists of the members: A, B, C, D, \dots . We will take A and B as an example to prove the authentication operation. We can represent the protocol in one step. The following formula for describing the operation of the protocol:

$$A \rightarrow B : \{P_i, c_i, H(P_i, C_i, \{K_g\}_{K_a1}), \{\{K_g\}_{K_a1}\}_{K_g}\}_{K_g}, \quad (1)$$

where

- A : User A
- B : User B
- P_i : A data packet number i .
- c_i : A counter number i .
- K_g : A group key.
- K_a^1 : Private key of A .
- $\{K_g\}_{K_a1}$: Digital signature.
- $H(P_i, c_i, \{K_g\}_{K_a1})$: The UMAC of the data packet concatenated with the counter concatenated with the digital signature.

We will consider the multicast authentication is completed between principals A and B , if there is a data packet “ X ” which the receiver B believes that it is sent by A . Thus authentication between A and B will be completed if: $B| \equiv A| \equiv X$, $B| \equiv X$, where the symbol $| \equiv$ means believes.

Now, we transform the protocol as follows:

$$A \longrightarrow B : \{P_i, \#c_i, H\{P_i, \#C_i, \{K_g\}_{K_a^{-1}}\}, \{\{K_g\}_{K_a^{-1}}\}_{K_g}\}_{K_g}, \quad (2)$$

where; $\#(X)$: X is fresh. It means X has never been said before the current run of the protocol. The initial assumptions are given by:

$$A| \equiv^{K_a} \rightarrow A. \quad (3)$$

$$B| \equiv^{K_a} \rightarrow A. \quad (4)$$

$$B| \equiv A \rightarrow^{K_g} \rightarrow B. \quad (5)$$

$$B| \equiv A| \Rightarrow P_i. \quad (6)$$

$$B| \equiv \#c_i. \quad (7)$$

$$B| \equiv A| \Rightarrow c_i. \quad (8)$$

Equation (3) indicates that A believes that K_a is the public key of A , Equation (4) indicates that B believes that K_a is the public key of A , Equation (5) indicates that A and B share the key K_g , Equation (6) indicates that B believes that A has jurisdiction over P_i , Equation (7) indicates that B believes that c_i is fresh, and Equation (8) indicates that B believes that A has jurisdiction over c_i . Using Equations (2) and (5), and applying the message-meaning rule [3], we obtain:

$$B|A, C, D, \dots| \sim (P_i, c_i, H(P_i, c_i, \{K_g\}_{K_a1}), \{\{K_g\}_{K_a1}\}_{K_g}), \quad (9)$$

where; $P| X$: means P at some time sent a message that contained X . we do not know exactly when the message was sent. Using Equation (1) and applying the interpretation rule (I2) in [2], where I2 formula is given below:

$$\frac{P| \equiv (Q| \sim (X, Y))}{P| \equiv (Q| \sim X), P| \equiv (Q| \sim Y)}$$

and

$$\begin{aligned} B| \models A, C, D, \dots| \sim P_i. \\ B| \models A, C, D, \dots| \sim \{\{K_g\}_{K_a^{-1}}\}_{K_g}. \\ B| \models A, C, D, \dots| \sim \{K_g\}_{K_a^{-1}}. \end{aligned} \quad (10)$$

It has to be noted that the encryption of the digital signature is not a redundancy; it is added to avoid pollution attacks. Also, from Equation (9) and using interpretation rule, we can obtain:

$$B| \models A, C, D, \dots| \sim H(P_i, c_i, \{\{K_g\}_{K_a^{-1}}\}_{K_g}). \quad (11)$$

From Equations (4) and (10), and applying the message meaning rule we can say:

$$B| \equiv A| \sim K_g. \quad (12)$$

We will assume that Equation (12) is used as an index for the rest of the group to know the identity of the sender, and by assuming that all the group is honest we can put the following formula assuming the group members are honest:

$$\frac{B| \equiv G| \sim H(X, Y), B| \equiv A| \sim X}{B| \equiv A| \sim Y}, \quad (13)$$

where

- G is a certain group in a multicast communication session.
- A is the intended sender.
- B is a certain receiver from the group.
- X, Y are the sent data.

From Equations (10), (11), and (13) we can say:

$$B| \equiv A| \sim (P_i, c_i). \quad (14)$$

From Equation (7) and using the freshness rule [2] ($\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$), therefore from Equation (14), we get:

$$B| \equiv \#(P_i, c_i). \quad (15)$$

From Equations (14) and (15), and by applying the nonce-verification rule we obtain:

$$B| \equiv A| \equiv (P_i, c_i). \quad (16)$$

Using Equation (16), and from [2] ($\frac{P| \equiv Q| \equiv (X, Y)}{P| \equiv Q| \equiv X}$):

$$B| \equiv A| \equiv P_i. \quad (17)$$

Using Equations (6) and (17), and by applying the jurisdiction rule we get:

$$B| \equiv P_i. \quad (18)$$

From Equations (17) and (18), we deduce the following: LAR1 achieves its goals without bugs or redundancies, and it is free from any type of known attacks like: replay attacks, message modification, insertion, or deletion. In the next section, comparison of LAR1 with other multicast authentication protocols is discussed.

5 Comparison With Other Protocols

We select some protocols that also support the real time communication; which are suitable in the case of most internet applications, they are: Wong-Lam [19], TESLA [11] and Ritesh Mukherjee [10] protocols. Table 1 shows the comparison of LAR1 with other multicast authentication protocols. In the table, we consider the following general assumptions.

- The stream of data is divided into blocks. Each block consists of m packets.
- Notations:

$Hout$: the length of the hash function;

Sig : the signature length;

E : the erasure code function;

$SymE$: one symmetric encryption operation;

$KeyLen$: a key length;

$SymD$: one symmetric decryption operation;

$PubD$: one public decryption operation;

$PrivE$: one private encryption operation;

$PubE$: one public encryption operation;

$PrivD$: one private decryption operation;

$EncLen$: the length of the symmetric encryption;

$count$: the counter length;

MAC : the length of the MAC;

$UMAC$: the length of the $UMAC$;

R : the rate loss;

$EncLen$: encryption length which is equal to the packet length;

RN : a random number.

- Both Wong-Lam and Mukherjee protocols can resist any packet loss. Concerning Wong-Lam protocol, it has the largest communication overhead.
- TESLA protocol has the lowest communication overhead. However, it can not resist any packet loss. Also, it can not resist both pollution and replay attacks.
- Comparing LAR1 protocol to TESLA, the proposed protocol has a comparable communication and computation overheads. Also, it can resist packet loss rate less than R . Further, it can resist both pollution and replay attacks. Finally, it doesn't need any synchronization.
- Comparing LAR1 protocol to Ritesh Mukherjee protocol, they both resist pollution and replay attacks, and don't need synchronization. Ritesh Mukherjee protocol can resist any packet loss. On the other hand, since Ritesh Mukherjee is based on public key encryption, our protocol has a lower communication and computation overheads. This is an important feature for real time communication.

Table 2 shows the communication overhead per packet for the multicast authentication protocols. In Table 2, the following parameters are assumed: $m = 128$ packets, $Hout = 16$ bytes (assuming MD5 algorithm), $Sig = 128$ bytes (assuming RSA algorithm), $UMAC = 4$ bytes (assuming UMAC algorithm), $count$: variable we will select it 1 byte, $R = 0.4$, $EncLen = 16$ bytes (assuming Advanced Encryption Standard algorithm (AES)), $MAC = 8$ bytes, $KeyLen = 16$ bytes, and $RN = 16$ bytes.

From Table 2, our protocol has the lowest communication overhead compared to other protocols, except in case of TESLA protocol; our protocol has a communication overhead comparable to it. On the other hand, TESLA protocol needs time synchronization between the sender and the receiver; which is a very big problem. Furthermore, it has no resistance against packet loss, pollution and replay attacks.

Table 1: To-be tested audio files

	Wong-Lam	bfTESLA	Ritesh Mukherjee	LAR1
Computation overhead at sender	$PrivE + (2m - 1) * H$	$mMAC$	$mPubE + mMAC$	$mUMAC + E + SymE + PrivE$
Computation overhead at receiver per packet	$PubD + (log_2 m - 1) * H$	MAC	$PrivD + MAC$	$SymD + UMAC + PubD/m$
Communication overhead/ packet	$(log_2 m + 1) * H_{out} + Sig$	$MAC + KeyLen$	$MAC + count + 2 * RN$	$UMAC + EncLen * (1 + R) + count$
Resistance to packet loss	Any	No	Any	Loss rate $< R$
Resistance to pollution attacks	Yes	No	Yes	Yes
Resistance to replay attacks	No	No	Yes	Yes
Need synch.	No	Yes	No	No

Table 2: Communication overhead per packet in bytes

	Wong-Lam	TESLA	Ritesh Mukherjee	LAR1
Communication Overhead	256	24	41	27.4

Our protocol has a low computation and communication overheads compared to the presented multicast authentication protocols. Furthermore, it resists packet loss, pollution and replay attacks. Finally, it doesn't need synchronization as in TESLA.

6 Conclusions

Nowadays, Multicast communication becomes an important field. There are research topics in this field, which have many alternatives and challenges. Its importance is a logic result after its wide applications, but because of its properties, its security becomes somehow hard. In the present paper, a new protocol (LAR1) for multicast group authentication is presented. In order to provide authentication, LAR1 uses both public key signature and symmetric key encryption. It resists packet loss by using erasure code functions over the signature. To resist pollution attacks, our protocol computes the symmetric encryption of the erasure code output. To resist replay attacks, a counter number is added to each packet. To evaluate the performance of our protocol, we compare it with some of other previously proposed protocols; they are Wong-Lam, TESLA and Ritesh Mukherjee protocols. LAR1 can resist pollution attacks, packet loss and replay attack. It achieves these goals with low computation and communication overheads compared to known multicast authentication protocols. We make verification to LAR1 protocol using BAN logic. The verification results show that it achieves its goals, free from redundancy and bugs.

References

- [1] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: fast and secure message authentication," *Proceedings of Crypto'99*, LNCS 1666, pp. 216-233, Springer-Verlag, Berlin, 1999.
- [2] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 18, no. 1, pp. 18-36, Feb. 1990.
- [3] L. Buttyan, S. Staamann, and U. Wilhelm, "A simple logic for authentication protocol design," *11th IEEE Computer Security Foundations Workshop*, pp. 9-11, Rockport, MA, June 1998.
- [4] T. K. Chua and D. C. Pheanis, "Bandwidth-conserving multicast VoIP teleconference system," *International Journal of Network Security*, vol. 7, no. 1, pp. 42-48, 2008.
- [5] J. Daemen and V. Rijmen, "The Rijndael block cipher: AES proposal," *First Candidate Conference (AES1)*, pp. 343-348, Ventura, California, Aug. 1998.
- [6] R. Gennar and P. Rohatgi, "How to sign digital streams," *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, LNCS 1294, pp. 180-197, Springer-Verlag, Berlin, 1997.
- [7] S. Gharout, Y. Challal, and A. Bouabdallah, "Scalable delay-constrained multicast group key management," *International Journal of Network Security*, vol. 7, no. 2, pp. 160-174, 2008.
- [8] P. Golle and N. Modadugu, "Authenticating streamed data in the presence of random packet loss," *Proceedings of the ISOC Network and Distributed System Security Symposium*, pp. 13-22, IEEE Computer Society Press, USA, 2001.

- [9] C. Karlof, N. Sastry, Y. Li, A. Perrig, and J. Tygar, "Distillation codes and applications to Dos resistant multicast authentication," *Proceedings of the ISOC Network and Distributed System Security Symposium*, pp. 37-56, IEEE Computer Society Press, USA, 2004.
- [10] R. Mukherjee and J. Atwood, "Scalable solutions for secure group communications," *Computers and Security*, pp. 3525-3548, Science Direct, Nov. 2007.
- [11] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 56-73, IEEE Computer Society Press, USA, May 2000.
- [12] A. Perrig, D. Song, and J. Tygar, "ELK, a new protocol for efficient large group key distribution," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 47-62, Oakland, California, USA, May 2001.
- [13] A. Perrig, "The BiBA one-time signature and broadcast authentication protocol," *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 28-37, Philadelphia Pennsylvania, USA, Nov. 2001.
- [14] L. Reyzin and R. Reyzin, "Better than BIBA: Short one-time signatures with fast signing and verifying," *Proceedings of the 7th Australian Conference on Information Security and Privacy*, pp. 144-53, Melbourne, Australia, July 2002.
- [15] R. Srinivasan, V. Vaidehi, K. N. Srivathsan, L. Ramesh Babu, and C. Karunagaran, "SeReRoM: Secured reliable routing scheme for multicasting," *International Journal of Network Security*, vol. 5, no. 1, pp. 82-88, 2007.
- [16] R. Srinivasan, V. Vaidehi, R. Rajaraman, S. Kanagaraj, R. Chidambaram Kalimuthu, and R. Dharmaraj, "Secure group key management scheme for multicast networks," *International Journal of Network Security*, vol. 11, no. 1, pp. 30-34, 2010.
- [17] W. Stallings, *Cryptography and Network Security: Principals and Practices*, Third Edition, Prentice-Hall, 2003.
- [18] L. Wang and C. K. Wu, "Efficient key agreement for large and dynamic multicast groups," *International Journal of Network Security*, vol. 3, no. 1, pp. 8-17, 2006.
- [19] C. Wong and S. Lam, "Digital signatures for flows and multicasts," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 502-513, 1999.
- [20] Q. Zhang and K. L. Calvert, "A peer-based recovery scheme for group rekeying in secure multicast," *International Journal of Network Security*, vol. 6, no. 1, pp. 15-25, 2008.
- Riham Abdellatif** graduated from Faculty of engineering Ain Shams University in 2000, obtained MSc in 2004 with a Master of Electronics and Communications Cairo University. She is now an assistant research in the Department of Information and Systems, National Institute of standard and technology, Giza, Egypt. Her research interests are information technology, and design of security standards, design of protocols and analysis and validation of security protocols.
- Heba K. Aslan** is an Associate Professor at the Electronics Research Institute, Cairo- Egypt. She received her B. Sc., M. Sc. And Ph. D. from Faculty of Engineering, Cairo University in 1990, 1994 and 1998 respectively. Her research interests include: Key Distribution Protocols, Authentication Protocols, Logical Analysis of Protocols and Intrusion Detection Protocols.
- Salwa Elramly** graduated 1967, obtained MSc. Degree 1972 from the faculty of Engineering, Ain Shams University, then her PhD degree 1976 from Nancy University, She is now professor Emeritus with the electronics Department, Faculty of Engineering, Ain Shams University; where she was the Head of the Department (2004-2006). Her research field of interest is communication systems and signal processing specially speech signal processing, digital signal Processing, wireless communications, Coding, Encryption, and Radars. She is a Senior Member of IEEE and Signal Processing chapter chain in Egypt.