World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:6, 2007

Authenticast: a source authentication protocol for multicast flows and streams

Yacine Challal, Abdelmadjid Bouabdallah

***Abstract*—The lack of security obstructs a large scale deployment of the multicast communication model. Therefore, a host of research works have been achieved in order to deal with several issues relating to securing the multicast, such as confidentiality, authentication, non-repudiation, integrity and access control. Many applications require authenticating the source of the received traffic, such as broadcasting stock quotes and videoconferencing and hence source authentication is a required component in the whole multicast security architecture.**

In this paper, we propose a new and efficient source authentication protocol which guarantees non-repudiation for multicast flows, and tolerates packet loss. We have simulated our protocol using NS-2, and the simulation results show that the protocol allows to achieve improvements over protocols fitting into the same category.

***Keywords*—Source Authentication, Non-repudiation, Multicast Security.**

I. Introduction

THE straightforward solution to guarantee non-repudiation for multicast flows is to digitally sign each packet of the stream sent by a multicast source. This solution cannot be used in practice. Indeed, as the data flow of multicast applications is mainly of *streaming* nature it is practically impossible to sign each packet of the stream since existing digital signature mechanisms are very slow and computationally expensive, and thereby using them would not meet the real-time transmission requirement of such applications. In order to minimize delays induced by digital signatures on multicast messages (both in signing and verification processes), many works [3] [11] [10] [1] [7] [4] [8] propose the concept of amortizing a single digital signature on a part or the whole stream sent by a source. This concept consists in signing a first multicast message and chaining subsequent messages to this signed message (using hash mechanisms) in order to amortize characteristics of the signature (authentication and non-repudiation) on the overall messages in the chain. As most of multimedia applications use an unreliable transport layer to send multicast data, some of these solutions are not efficient in practice. In fact, if an intermediate message of the constructed chain is lost, the chain may be broken and the first signature can no longer be amortized on messages that follow the lost one. To overcome this weakness, other solutions propose to use redundant chaining of multicast messages in order to tolerate the loss of some packets.

In this paper, we propose a new and efficient protocol called AUTHENTICAST which authenticates the source of a multicast flow with non-repudiation and tolerates packet loss. As stated above, our protocol uses the concept of amortizing a single digital signature on many packets using hash chaining. Hash chaining is the underlying idea behind many proposed protocols [3] [11] [10] [1] [7] [4] [8]. These protocols use either static chaining [10] or random chaining [7]. In contrast, in our protocol, we use a hybrid chaining (static chaining combined with random chaining), and the carried out simulations using NS-2 show that our protocol achieves better tolerance to packet loss compared with the two previous approaches.

Yacine Challal and Abdelmadjid Bouabdallah are with Compiegne University of Technology, Heudiasyc lab., France (phone:33-3-44-23-44-23; e-mail: ychallal@hds.utc.fr, bouabdal@hds.utc.fr).

II. Related works

Recent source authentication schemes rely mainly on using MACs and hashes combined with digital signatures. MAC-based approaches [2] [1] [9] are generally used when only source authentication (without non-repudiation) is required. Whereas, hash / digital signature based approaches [3] [11] [3] [7] [5] are generally used when non-repudiation is required beyond source authentication. Since our protocol uses a hash-based technique to sign multicast streams, we will discuss particularly some protocols within this approach in the following paragraphs.

A. Terminology

We define some terminology to simplify the following discussion: if a packet P_j contains the hash of a packet P_i, we say that a **hash-link** connects P_i to P_j, and we call P_j a **target** packet of P_i. We define the **scope** of a hash-link between two packets P_i and P_j as the value $|i-j|$. A **signature packet** is a sequence of packet hashes which are signed using a conventional digital signature scheme. A hash-link points from a packet P_k to a signature packet S_l, if S_l contains the hash of P_k. We assume that some of the packets are dropped between the sender and the receiver. We designate by **redundancy degree** the number of times that a packet hash is embedded in subsequent packets to create redundancy in chaining the packet to a signature packet. A packet P_i is **verifiable**, if it remains a **path** (following the hash-links) from P_i to a signature packet S_j. We designate by **verification ratio**: the number of verifiable packets by the number of received packets. The verification ratio is a good indicator of the **verification probability** which means the probability for a packet to be verifiable given that it is received: P(packet is verifiable/packet is received).

A.1 Simple Off-Line Chaining

The main idea of the solution proposed by Gennaro and Rohatgi in [3] is to divide the stream into blocks and embed in the current block a hash of the following block (which in turn includes the hash of the following one and

so on...)(see figure 1). This way the signer needs to sign only the first block and then the properties of this single signature will *propagate* to the rest of the stream through the hash-chaining. We note that in order to construct this chain, the sender needs to know the entire stream in advance (off-line). With this solution, the authentication information is reduced to one hash per block and the sender signs only the hash of the first block. However, this solution is not fault tolerant: if a block is lost, the authentication chain is broken and hence all subsequent blocks can no longer be authenticated.

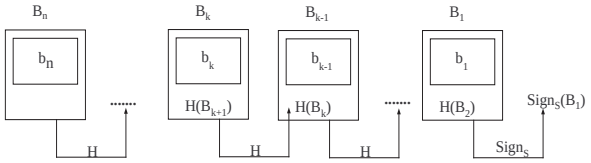

Fig. 1. Simple off-line hash-chaining (Example)

A.2 EMSS: Efficient Multi-Chained Stream Signature

Perrig et al. [7] introduced the notion of *redundant hash-chaining* which means that each packet of the stream is *hash-linked* to several *target* packets. Thus, even if some packets are lost, a received packet is verifiable if it remains a hash-link path that relates the packet to a signature packet. For a given packet, EMSS chooses target packets randomly. Hence, EMSS provides more or less probabilistic guarantees that it remains a hash-link path between the packet and a signature packet, given a certain rate of packet loss in the network. In order for the sender to continuously assure the authentication of the stream, the sender sends periodic signature packets. To verify authenticity of received packets, a receiver buffers received packets and waits for their corresponding signature packet. The signature packet carries the hashes that allow the verification of few packets. These latter packets carry, in turn, the hashes that allow to verify other packets, and so on until the authenticity of all received packets is verified.

A.3 Periodic Chaining Approach

Modadugu and Golle [10] have proposed to use a similar strategy to EMSS, but target packets of a given packet are chosen in a *deterministic* way rather than randomly. The proposed *deterministic topologies* of packet hash-links are designed to be optimized to resist a *burst loss*. The goal of the proposed schemes is to maximize the size of the longest single burst of loss that the authentication scheme can withstand (Once few packets have been received after a burst, the scheme recovers and is ready to maintain authentication even if further loss occurs). The authors have proposed a construction called C_a, a periodic authentication scheme of period 1 defined as follows: the hash of packet P_i is appended to two other packets: P_{i+1} and P_{i+a}. The last packet P_n is signed. C_a is called a chain of strength a (see figure 2 for an example of a C_3 authentication scheme).

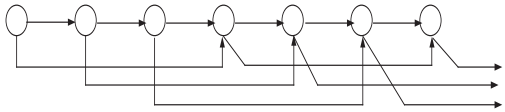

Fig. 2. A periodic chain of strength 3: C_3

III. AUTHENTICAST

A. Overview and Motivation

In this section, we present our protocol which uses the concept of amortizing a single digital signature over multiple packets using redundant hash-chaining in a way that enhances the verification ratio of received packets.

We note the existence of two types of hash chaining: the static chaining and the random chaining. In the static chaining, target packets that will carry a hash of the current packet are defined a priori. An example of such chaining is the C_a construction proposed by Golle and Modadugu in [10]. Authors proved that with a C_a scheme, bursts of length up to $a-1$ do not disconnect any packet from the signature. However, in practice, we have only an idea about the average length of bursts. Thus, it is difficult to determine the best value of the parameter a of a C_a construction in an actual streaming session over an unreliable transport layer. In the random chaining, target packets that will carry a hash of the current packet are defined using a uniform random function. One solution that uses this approach is EMSS, proposed by Perrig et al. in [7]. This approach complies with the random bursty packet loss pattern and simulation results of the authors of EMSS [7] show that it is possible to verify up to 90% of received packets using 6 hashes per packet in an environment with 60% of packet loss and an average length of bursts equal to 10.

In order to take advantage of both approaches, we proposed the AUTHENTICAST protocol which combines the static with the random chaining schemes in a way that increases the probability that a received packet be verifiable.

B. AUTHENTICAST-*Hash-Chaining*

To explain how does AUTHENTICAST chain packets, consider a bursty packet loss model with an average burst length equal to b. Suppose that the outgoing degree of a node of the stream is k, and the maximum authorized scope is d. With AUTHENTICAST, we chain the nodes in two steps:

1. First step: we apply a C_{b+1} construction. It means that the target nodes of P_i are P_{i+1} and P_{i+b+1} (see figure 3-(a)).
2. Second step: we apply a random chaining using the remaining $k-2$ edges. It means that the $k-2$ remaining targets of P_i are P_{i+l} with l a random integer

comprised between 2 and d. In the example of figure 3-(b), we consider a AUTHENTICAST-hash-chaining with $k = 4$.

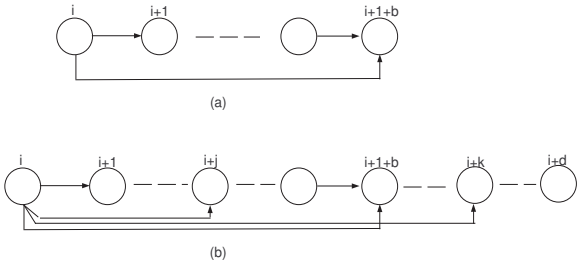

Fig. 3. AUTHENTICAST hash-chaining with $k = 4$.

This hybrid hash-chaining increases the verification probability. Indeed, it is easy to see that since packets are lost in a bursty way [6], the received packets are also received contiguously (see figure 4). And hence, if each packet is chained systematically to its subsequent packet (static chaining), then if only one packet is verifiable then all the packets that follow it (in the same contiguous received segment) are also verifiable: in figure 4, packets P_{f-1} to P_{f-n} are verifiable because P_f is verifiable (it holds a path to the signature packet). This is why our hybrid scheme increases the probability of verifiability of a received packet compared to the purely random or static hash-chaining techniques.

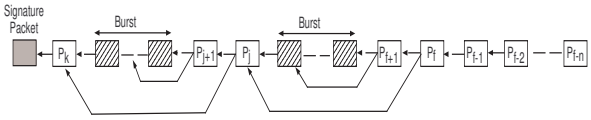

Fig. 4. Hybrid hash-chaining impact on verification probability

C. AUTHENTICAST *Protocol*

When a packet is presented to be sent, the source chain it to subsequent packets, using the AUTHENTICAST hybrid hash-chaining technique. To assure continuous non-repudiation, the source sends a signature packet periodically. When a receiver receives a packet, it buffers it waiting for a signature packet. Upon receiving the signature packet, the receiver starts a verification procedure which consists in comparing the hash of each received packet with its hash carried by one of the received packets, recursively. If the two quantities are equal, the packet is considered authentic. Otherwise, it is considered not authentic. If there is no hash carried by one of the received packets, corresponding to a received packet, this latter is considered *not verifiable*. This case happens when the hash-chain relating a packet to a signature packet is completely broken because of packet loss.

IV. SIMULATIONS

We carried out simulations using NS-2 to evaluate the performance of AUTHENTICAST and compare it with EMSS [7] and the C_a approach [10]. First, we used the two state Markov chain model [12] to extend NS-2 with a new queuing behavior to simulate a bursty packet loss pattern. Indeed, many studies show that packet loss is correlated, which means that the probability of loss is much higher if the previous packet is lost. Yanik et al. show that a k-state Markov model can model Internet packet loss patterns [12]. For our simulation purposes, the two-state Markov chain model is sufficient, since it can model simple patterns of bursty loss well [12]. In what follows, we consider a bursty packet loss pattern with bursts having an average length equal to 6. Then, we considered a stream of 10.000 packets with a signature packet every 500 packets. Unless it is mentioned otherwise, we use a maximum edge scope equal to 250, an outgoing degree per node equal to 4 and a packet loss ratio equal to 40%. For C_a, we used $a = 2\times$ average burst length =12. We are interested in the probability of a received packet to be verifiable: P(packet is verifiable / packet is received). We evaluate and compare our protocol regarding the following criteria.

1. *The average verification probability:*
   In order to determine the average verification probability of AUTHENTICAST, EMSS and C_a, we carried out intensive simulations using the above settings. Figure 5 illustrates the results of the simulation.

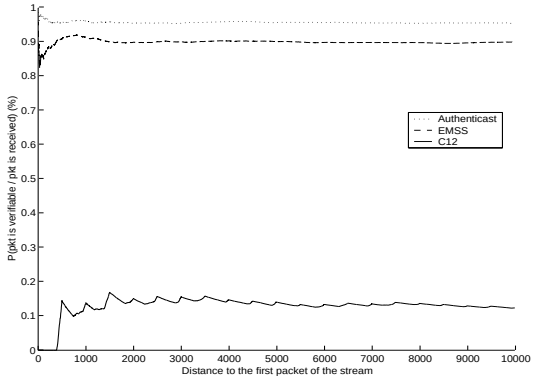

Fig. 5. Average verification probability

Note that EMSS reaches 90% of verification probability, in average, using only 3 hashes. With the same amount of authentication information (3 hashes), AUTHENTICAST reaches up to 95% of verification probability in average. This means that almost 500 packets among the 10.000 packets of the stream, that were unverifiable using EMSS, are verifiable using AUTHENTICAST, and this is without any increase in the authentication information size. In other words, if we use EMSS to reach 95% of average verification probability, we have to use at least 4 edges per node. Thus, AUTHENTICAST saves one hash per packet. If we consider a hash code of 20 bytes, it means that, AUTHENTICAST allows to save 200Kbytes of authentication data.

2. *Robustness against packet loss:*
To determine the influence of packet loss ratio on our protocol, we varied the packet loss ratio from 5% to 60% and we computed the average ratio of verifiable packets from received packets, for the three protocols: AUTHENTICAST, EMSS and C_{12}. Figure 6 shows the results of the simulation.

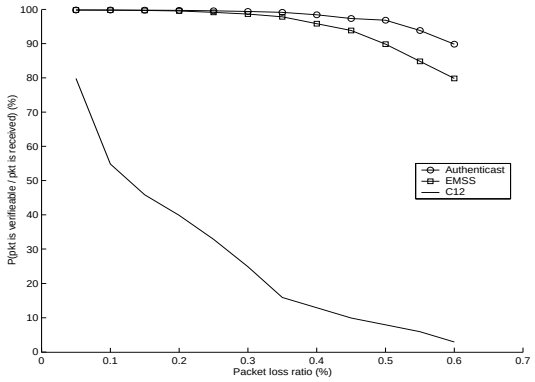

Fig. 6. Impact of packet loss ratio on the average verification probability

Note that AUTHENTICAST behaves better than the two other protocols since it allows 90% of verification probability in the average with 60% lost packets, compared with 80% for EMSS and 3% for C_{12}.

3. *The authentication information overhead:*
The amount of authentication information (number of hashes and signatures in our case), is an important criteria, since it determines the bandwidth overhead induced by using the protocol. Hence, less the authentication information size is, better is the protocol if it guarantees the same verification probability. We carried simulations using the default settings above and we varied the outgoing degree per node (number of hashes per packet). Then we calculated the average verification probability. Figure 7 illustrates the results.

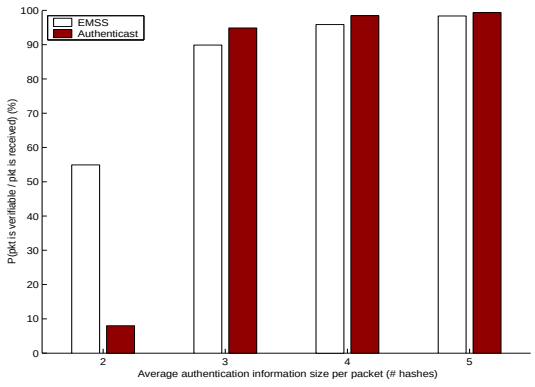

Fig. 7. Impact of the authentication information size on the average verification probability

With a degree higher or equal to three hashes per packet, we remark that the average verification probability of AUTHENTICAST is higher than the one of EMSS. Moreover, AUTHENTICAST allows to have the same average verification probability as EMSS with one edge less, which allows to save up to 200Kbytes of data (if the size of a single hash is 20 bytes): remark in figure 7 that with 4 edges only AUTHENTICAST reaches 98.60% of verification probability in average, whereas EMSS requires 5 edges to reach the same average verification probability. The case of two hashes per packet is a special case where chaining in AUTHENTICAST is restricted to the static step which does not comply with the random nature of bursty packet loss pattern.

V. Conclusion

To achieve non-repudiation for a multicast stream, we propose a new efficient protocol called AUTHENTICAST. Our protocol uses a hash-chaining technique to amortize a single digital signature over many packets. AUTHENTICAST takes advantages of both static and random hash-chaining schemes, and hence improves the probability that a packet be verifiable even if some packets are lost.

Simulation results using NS-2 show that our protocol resists to bursty packet loss pattern and assures with a high probability that a received packet be verifiable. Besides, the simulations and comparisons with other protocols show that our hybrid chaining technique is more efficient than using separately either static or random hash-chaining.

References

[1] F. Bergadano, D. Cavagnino, and B. Crispo. Individual Single Source Authentication on the MBone. *IEEE International Conference on Multimedia and Expo*, 2000.
[2] Ran Canetti, Juan Garay, Gene Itkis, Daniele Micciancio, and Moni Naor. Multicast Security: A taxonomy and Efficient Constructions. *INFOCOM*, 1999.
[3] Rosario Gennaro and Pankaj Rohatgi. How to Sign Digital Streams. *Information and Computation*, 165(1):100–116, February 2001.
[4] Sara Miner and Jessica Staddon. Graph-Based Authentication of Digital Streams. *IEEE Symposium on Security and Privacy*, 2001.
[5] J. M. Park, E. K. P. Chong, and H. J Siegel. Efficient Multicast Packet Authentication Using Signature Amortization. *IEEE Symposium on Security and Privacy*, 2002.
[6] Vern Paxson. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking*, 7(3):277–292, June 1999.
[7] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. Efficient Authentication and Signing of Multicast Streams over Lossy Channels. *IEEE Symposium on Security and Privacy*, 2000.
[8] Adrian Perrig. The BiBa One-Time Signature and Broadcast Authentication Protocol. *The 8th ACM Conference on Computer and Communications Security*, November 2001.
[9] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5, Summer 2002.
[10] Nagendra Modadugu Philippe Golle. Authenticating Streamed Data in the Presence of Random Packet Loss. *NDSS'01: The Network and Distributed System Security Symposium*, 2001.
[11] Chung Kei Wong and Simon S. Lam. Digital Signatures for Flows and Multicasts. *IEEE/ACM Transactions on Networking*, 7(4), August 1999.
[12] Maya Yajnik, Sue Moon, Jim Kurose, and Don Towsley. Measurement and Modeling of the Temporal Dependence in Packet Loss. *INFOCOM'99*, pages 345–352, March 1999.