Performance Analysis of Selective Authentication Schemes for IPTV Networks

Conference Paper · October 2014

CITATIONS
0

READS
120

3 authors, including:

Khaled Y. Youssef
Beni Suef University
29 PUBLICATIONS **30** CITATIONS

SEE PROFILE

Performance Analysis of Selective Authentication Schemes for IPTV Networks

Ahmad H. Al-Sharif
Electrical Engineering Dept.
Faculty of Engineering (Shoubra)
Benha University, Egypt
Email: ahmad.hammad@feng.bu.edu.eg

Khaled Y. Youssef
Head of Programs Management, Lecturer
Télécom Egypt Group, Thebes Academy
Cairo, Egypt
Email: khaled.yousef@tedata.net

Hala A. K. Mansour
Professor, Electrical Engineering Dept.
Faculty of Engineering (Shoubra)
Benha University, Egypt
Email: hala.mansour@gmail.com

***Abstract*—The growing demand on IPTV as well as the strong adoption of smart devices as tablets and smartphones in presence of broadband wireless technologies as LTE, leads to increasing challenges on the need to authenticate media content to be delivered specially over wireless networks. In this paper, we focus on security areas in IPTV networks specially authentication schemes. In general, a short survey is held for IPTV authentication schemes versus challenges of transmission media. In addition, mathematical modelling is held for seven authentication schemes and their respective performance impact against referential metrics including processing capabilities and existing algorithms limitations. Moreover, comparisons are done among the considered schemes in light of four performance metrics namely computational cost, communication overhead, receiver buffer size, and authentication delay.**

***Keywords*—IPTV, security, performance, authentication schemes, multimedia authentication , public key cryptography**

I. Introduction

IPTV is defined as the secure and reliable delivery of entertainment video and related services across an access agnostic, packet switched network that employs the IP protocol to transport the audio, video and control signals [1]. In contrast to video over the public internet, with IPTV deployments, network performance and security should be tightly managed to ensure a superior entertainment experience. IPTV is distinguished from internet television by its ongoing standardization process (e.g., European Telecommunications Standards Institute) and preferential deployment scenarios in subscriber-based telecommunications networks with high speed access channels into end-user premises via set-top boxes or other customer-premises equipment.

With broadband links and connected devices, consumers are increasingly watching videos on laptops, tablets, smartphones and other IPenabled devices both inside and outside home. Moreover, mobile viewing is clearly increasing as more consumers make the shift to watch VoD and shift from stationary to portable devices. The TV viewers are seeking more ways to watch what they want, when and where they want to; which create greater demand for choice and point to point-to-point transmission of content. At the same time, greater picture quality on all video screens are increasingly demanded versus current home TV screens quality.

Generally, a switched IP network allows for the delivery of significantly more content and functionality in addition to better bandwidth efficiency. In a typical TV or satellite network, using broadcast video technology, all the content constantly flows downstream to each customer, and the customer switches the content at the set-top box. The customer can select from as many choices as the Telecom, cable or satellite company can stuff into the ”pipe” flowing into the home. A switched IP network works differently. Content remains in the network, and only the content the customer selects is sent into the customers home. That frees up bandwidth, and the customers choice is less restricted by the size of the pipe into the home.

Accordingly, the performance and security of networks are becoming a tightly coupled requirement for growing IPTV services especially with the broadband multiple access capabilities over wireless networks (e.g. LTE). Detailed analysis of various authentication schemes is needed in order to discover the merits and shortcomings of each scheme. Moreover, side-by-side comparisons of authentication schemes along multiple performance metrics provide guidelines on choosing the most suitable scheme for a given multimedia streaming application, and offer insights for further research on the stream authentication problem.

In this paper, we focus on security areas in IPTV networks specially authentication schemes. In general, a short survey is held for IPTV authentication schemes versus challenges of transmission media. In addition, mathematical modelling is held for seven authentication schemes and their respective performance impact against referential metrics including processing capabilities and existing algorithms limitations. Moreover, comparisons are done among the considered schemes in light of four performance metrics namely computational cost, communication overhead, receiver buffer size, and authentication delay.

II. Related work

A survey on authentication schemes for multicasting multimedia streams was discussed in [2]. The authors classified authentication schemes according to the core techniques underlying them. In [3], the author compared his proposed scheme with four other schemes focusing on verification probability of some specific cases. In [4], the author provided a quantitative analysis of eight authentication schemes for video streaming generally. However, he did not cover several dimensions of performance as processing platforms, video encoding types, and hashing algorithm which are handled in our research with focus on IPTV architecture in specific. The contribution of

978-1-4799-7476-4/14/$31.00 ©2014 IEEE

our research could be considered a corner stone for a dynamic authentication management layer enabling a next generation of IPTV services (balance the trade-off between security strength and performance degradation)

III. OVERVIEW ON AUTHENTICATION SCHEMES

The term "Stream authentication" refers to a process of verifying a sequence of stream packets transmitted over a public and lossy network has not been altered by an unauthorized third party, while tolerating packet loss which may occur in transit. To ensure that, a number of cryptographic functions and techniques are usually employed in authentication schemes such as one-way hash functions and digital signatures. A one-way hash function generates a fixed-length bit-string (hash value) for any given data with arbitrary size. These hash functions guarantee that even a single-bit change in the data will result in a totally different hash value. Therefore, hash function provides data integrity i.e. the recipient can be confident that the message has not been accidentally modified. Commonly used hash functions include MD5, SHA-1, SHA-256, and SHA-512 and they are all considered in our analysis. On the other hand, digital signatures provide data integrity, authentication and non-repudiation. A digital signature (e.g. RSA) is created with a sender private key, and verified with a corresponding public key of an asymmetric key-pair. Only the holder of the private key can create this signature, and normally anyone knowing the public key can verify it. Thus, the recipient can be confident that the message is generated by the desired sender and is not modified by anybody; moreover, the sender cannot repudiate the validity of his signature because nobody else knows the private key. Digital signatures are computationally expensive compared to hashing, so the basic idea of most authentication schemes is to amortize a digital signature among a group of packets to reduce computations and overhead, and at the same time remain robust against packet loss.

Media stream authentication approaches can be classified into two major groups, graph-based authentication [5]–[9] and error-correction coding based (ECC-based) authentication [3], [11], [12]. In graph-based authentication, only one packet in the stream (usually the first or last) is signed and the other packets are connected with the signature by a chain of hashes. They are referred to as block signatures schemes as there is one digital signature assigned to the whole block. On the other hand, ECC-based authentication performs error-correction coding on the authentication information and split them over all the packets of a block to resist against certain packet loss. Authentication information can be recovered as long as the packet-loss percentage does not exceed a threshold determined by the coding parameters.

In all consider authentication schemes in this paper, multimedia stream is divided into fixed-size packets P_1, P_2, ... Each n packets are called block and the sender generates and sends n packets including their authentication information. After sending all these packets, it repeats this procedure for the next block. When the receiver receives the packets for a block, it attempts to verify them. Because packets are processed in the unit of a block, we will briefly explain the main idea of how the packets are generated/verified for only a single block by each scheme.

A. Hash Chaining

Hash chaining, proposed by Gennaro and Rohatgi in [5], is the simplest authentication scheme. In hash chaining, the hash of each packet is appended to its previous packet, then the first packet of the block is signed. Because the first packet carries the signature, packets can be verified once they arrive after receiving the first packet and hence no receiver buffer is required for this scheme.

B. Augmented hash chaining

Golle and Nagendra [6] proposed the augmented hash chaining in order to tolerate packet loss. They proposed a family of schemes parameterized by two integers, a and p that affect resistance against bursty losses, receiver delay, receiver buffer size, sender delay and sender buffer size. In the augmented hash chaining, the hash of packet P_i is appended to two other packets, P_{i+1} and P_{i+a}, and the last node P_n is signed. Then $p-1$ additional packets are inserted between those of the original chain to create augmented chain. Since verification process depends on the delivery of the signature packet, augmented chaining sends n_{sig} copies of the signature packet.

C. Butterfly Hash Chaining

Zhang et al. proposed a butterfly-graph based stream authentication in [7]. The butterfly authentication divides n packets into n_c stages "columns", each stage contains n_r packets. The hash of each packet in each stage except stage 0 is appended to two other packets of the previous stage according to the stage it belongs to. In addition, all the hashes of packets in stage 0 are appended to a signature packet. Butterfly graph has its own limitations. Firstly, the total number of packets n depends on the number of rows n_r which means that butterfly graphs do not work for arbitrary number of packets. Secondly, the signature packet size grows with n_r. It may become larger than the maximum transfer unit (MTU) of the network and therefore it would be fragmented for transmission, increasing its loss probability. To overcome these limitations, the authors in [8] proposed a generalized butterfly-graph authentication scheme which is considered in our analysis. As in augmented chaining, the signature packet is sent n_{sig} times to increase its probability of being delivered.

D. Tree chaining

Wong and Lam [9] proposed tree chaining authentication scheme where every packet carries the full information necessary to verify itself, which in some sense resembles signing of every packet individually. At the sender side, a balanced binary Merkle hash tree [10] is built over packets of each block. The packets hashes are the leaf nodes of the tree. Each interior node (Parent) is the hash of the concatenation of its children. The block signature is calculated on the root hash of the tree.

To verify a packet individually, the receiver needs to verify its path to the root. Therefore, each packet needs to carry its own authentication information (packet signature). In tree chaining, a packet signature consists of the (1) block signature, (2) the packet position in the block and (3) the siblings of each node in the packet's path to the root.

E. SAIDA

Park et al. [11] proposed SAIDA (Signature Amortization using Information Dispersal Algorithm) for stream authentication. Instead of generating one signature for the whole block and sending one signature packet, SAIDA uses the authentication information (the packets hashes and the signature) to create authentication tags and splits them over the packets of a block. In SAIDA, authentication tags are created as follows, the sender computes the hash of each packet $H_i = h(P_i)$ where $(1 \leq i \leq n)$. Then, it concatenates the calculated hashes $H_{1\sim n} = (H_1 \parallel \cdots \parallel H_n)$ and computes a signature on the hash of the hashes concatenation. Let m denote the minimum number of the received packets for successful verification, i.e., $m = \lfloor (1-r)n \rfloor$. The final step in SAIDA algorithm is that, the hashes concatenations $H_{1\sim n}$ and the signature on $h(H_{1\sim n})$ are divided into m pieces, FEC-coded into n pieces and split over all the n packets of the block. Any m pieces suffice to reconstruct the hashes and the signature to verify authenticity of the entire block.

F. eSAIDA

Park and Cho [12] proposed enhanced-SAIDA (eSAIDA) that is an improvement of SAIDA. In eSAIDA, beside the authentication tags, some of the packets contain a hash value and the average number of such packets is parameterized by an integer s where $(s \leq n)$. In eSAIDA, authentication tags are created as follows, the sender computes the hash of each packet $H_i = h(P_i)$ where $(1 \leq i \leq n)$ and then computes the hash of concatenation of each two successive hashes $H_{2j-1\sim 2j} = h(H_{2j-1} \parallel H_{2j})$ where $(1 \leq j \leq n/2)$ and n is even (when n is odd, a pre-defined dummy value, P_{n+1}, can be used). After that, it computes $H_{1\sim n} = (H_{1\sim 2} \parallel \cdots \parallel H_{n-1\sim n})$ and calculates a signature on $h(H_{1\sim n})$. $H_{1\sim n}$ and the signature on $h(H_{1\sim n})$ are divided into m pieces, FEC-coded into n pieces and split over all the n packets of the block. The fraction of packets containing their couple's hash is parameterized by x $(0 < x < 1)$ as an input, which governs a tradeoff between successful verification rate and communication overhead.

G. cSAIDA

The cSAIDA scheme [3], proposed by Pannetrat and Molva, is an improvement of SAIDA because it reduces the overhead by using the erasure codes twice times. In cSAIDA, the sender computes the hash of each packet $H_i = h(P_i)$ where $(1 \leq i \leq n)$. Then, the n hashes are FEC-coded into b $(b \geq n)$ pieces $(H_1, \ldots, H_n, H_{n+1}, \ldots, H_b)$ such that the first n pieces of the encoded b pieces are exactly the same as the original n hashes and any n subset of the b pieces are sufficient to reconstruct the original n hashes. The extra generated pieces $(H_{n+1}, \ldots H_b)$ are called parity pieces. Then, only parity pieces and a signature on the original hashes are concatenated, divided into m pieces and FEC-coded again into n pieces to be attached to all the n packets of a block. At the receiver, if m packets are successfully received, the signature on the original hashes and the parity pieces can all be successfully retrieved. So, the original hashes can be reconstructed so as to verify the whole block using the signature.

IV. PERFORMANCE ASSESSMENT OF IPTV STREAM AUTHENTICATION SCHEMES

A. IPTV Service Performance Key Indicators

To analyze and evaluate the above schemes, four key performance metrics are defined for assessment and comparison between authentication schemes

(i) Computational Cost (C). It is defined as the summation of the CPU time required by the sender to generate authentication information for stream packets and the CPU time required by the receiver to verify that information. We focused on evaluating the computational cost at the receiver only as it may be a device with limited computational capabilities compared to powerful sender.

(ii) Communication Overhead (Ov). It is the additional number of bytes transmitted along with the media packets to enable the receiver to verify the received packets. The overhead may include digital signatures, or hashes. It is important to minimize this overhead, especially if the available network bandwidth is limited.

(iii) Receiver Buffer Size (Bfr). Some authentication schemes require the receiver to buffer a certain number of packets before the verification process starts. Quantifying the required buffer size specifies the minimum memory requirements, which is of great importance if the receiver device has limited capabilities

(iv) Authentication delay (D_{auth}). It is defined as the summation of sender and receiver delays, and they can be explained as follows:

(a) Sender delay: It is the delay placed on a packet before it can be transmitted due to authentication processing (e.g., processing a block of packets).
(b) Receiver delay: It is the delay from the time when a packet is received until it can be verified by the receiver.

B. User Quality of Experience (QoE) standard parameters

Quality of video content can be specified by three main parameters

(a) Resolution, which is the number of distinct pixels that can be displayed in one frame. Commonly used resolutions are 640×480 for Standard-Definition television (SDTV) and 1280×720 and 1920×1080 for High-Definition television (HDTV)
(b) Frame rate, which describes how many unique consecutive images are displayed per second in a video stream to give the illusion of movement. Common frame rates are 24 fps, 25 fps, and 30 fps.
(c) Color model. The purpose of a color model is to facilitate the specification of colors in some standard generally accepted way. A color model encodes a pixel color into value components (e.g RGB and YUV).

All these parameters affect the video bit rate which is the amount of data dedicated to a second of video. Video bitrate is bounded by the channel bandwidth constraints which may require lower values of video bit rate. One possible solution is to introduce convenient codec to play this role and to accommodate with channel limitations. Most commonly used

TABLE I: Parameters used in paper with notations and values

Parameter	*Value*	*Description*
t_S	variable	Time to verify a signature (Processor dependant)
t_H	variable	Time to compute a hash (Processor dependant)
n	variable	Block size
L	1400 byte	Packet size
k	64 byte 128 byte	Input block size for MD5, SHA-1 and SHA-256 Input block size for SHA-512
S	128 byte	Signature size (1024bit RSA)
H	16 byte 20 byte 32 byte 64 byte	Hash size (MD5) Hash size (SHA-1) Hash size (SHA-256) Hash size (SHA-512)
β	variable	Video encoding rate
α	variable	Packet rate (depends on β)
a, p	searched	Inputs to augmented chain
n_{sig}	$1/16n$ searched	Signature replications for augmented Signature replications for Butterfly
n_r	variable	Number of rows in butterfly
r	0.2	Packet loss ratio
m	$0.8n$	Minimum number of received packets for verification
x	$0 < x < 1$	Input to eSAIDA

video codecs include MPEG-2 , MPEG-4 and H.264. Encoding techniques are highly dependent on access technologies due to channels' bandwidth constraints for each access technology. Video encoding rates have a great impact on end-to-end packet transfer delay which is illustrated later in next section.

C. Notations and equations

Several parameters are used in our analysis, such as video encoding rate (β) and block size (n). For quick reference, all parameters used in this paper with their notations and values are listed in Table I. The packet size (L) should fit the size of a packet in the underlying packet network. In the case of Internet, it is usually not more than $1000 - 1500$ bytes [13], 1400 byte is chosen as the packet size. The packet generation rate (α) is calculated from the video encoding rate (β) by

$$\alpha = \beta / L \tag{1}$$

The computation costs of the digital signature and hashing operations depend on the receiver device's processor and are estimated using the European Network of Excellence for Cryptology II "ECRYPT II" benchmarks [14].

All analysis equations for all considered authentication schemes are listed in Table II.

V. Results

A. Impact of processor speed on computational cost

To analyze the impact of the receiver device's processor on computational cost, three processors are considered in our mathematical analysis which are

(1) armeabi (v7-A, Cortex A8) 2012 TI Sitara XAM3359AZCZ100 1×1000MHz

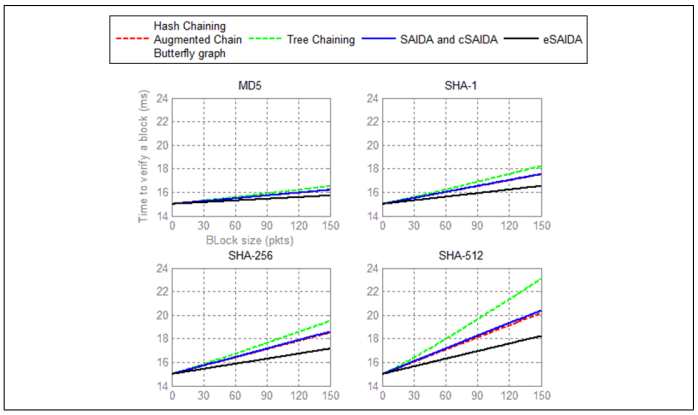

Fig. 1: Computational cost versus block size for A8 processor

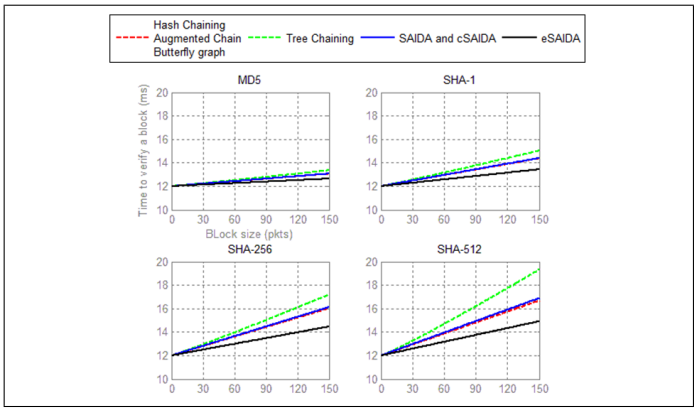

Fig. 2: Computational cost versus block size for A9 processor

(2) armeabi (v7-A, Cortex A9) 2012 VIA WonderMedia 8850 1×1200MHz
(3) amd64; IB+AES (306a9) 2012 Intel Core i5-3210M 2×2500MHz

They will be denoted in paper by A8, A9 and i5 respectively.

The computational costs for all considered authentication schemes are plotted in Fig. 1, Fig. 2 and Fig. 3 for the three processors as n varies from 0 to 150 packets using four hashing algorithms and they show that

(1) The eSAIDA scheme is the most efficient scheme while

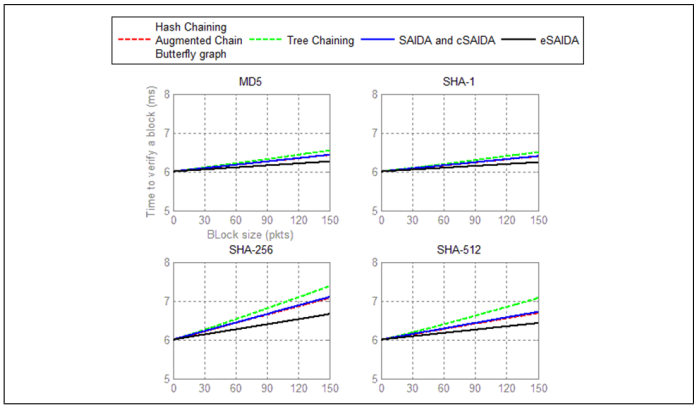

Fig. 3: Computational cost versus block size for i5 processor

TABLE II: Summary of authentication schemes performance metrics equations

	Computational cost (second)	Communication overhead (bytes per packet)	Delay (second)	Rx. buffer size (packets)
Hash Chaining	$C = t_S + t_H n \left\lceil \frac{L}{k} \right\rceil$ (2)	$Ov = \frac{S}{n} + H$ (3)	$D = \frac{n}{\alpha}$ (4)	$Bfr = 1$ (5)
Augmented	$C = t_S + t_H n \left\lceil \frac{L}{k} \right\rceil$ (6)	$Ov = \frac{n_{sig} S}{n} + 2H$ (7)	$D = \frac{n}{\alpha}$ (8)	$Bfr = n$ (9)
Butterfly	$C = t_S + t_H n \left\lceil \frac{L}{k} \right\rceil$ (10)	$Ov = \frac{n_{sig}(S + n_r H)}{n} + \frac{H(2n - n_r)}{n}$ (11)	$D = \frac{n}{\alpha}$ (12)	$Bfr = n$ (13)
Tree	$C = t_S + t_H \left(n \left\lceil \frac{L}{k} \right\rceil + \lceil n \log_2 n - n \rceil \left\lceil \frac{2H}{k} \right\rceil \right)$ (14)	$Ov = S + \lceil \log_2 n \rceil H$ (15)	$D = \frac{n}{\alpha}$ (16)	$Bfr = 1$ (17)
SAIDA	$C = t_S + t_H \left(n \left\lceil \frac{L}{k} \right\rceil + \left\lceil \frac{nH}{k} \right\rceil \right)$ (18)	$Ov = \frac{S + nH}{m}$ (19)	$D = \frac{2n}{\alpha}$ (20)	$Bfr = n$ (21)
eSAIDA	$C = t_S + t_H \left(\left\lceil \frac{L}{k} \right\rceil (1 + x)n/2 + \left\lceil \frac{nH}{2k} \right\rceil \right)$ (22)	$Ov = \frac{S + H\frac{n}{2}}{m} + xH$ (23)	$D = \frac{2n}{\alpha}$ (24)	$Bfr = n$ (25)
cSAIDA	$C = t_S + t_H \left(n \left\lceil \frac{L}{k} \right\rceil + \left\lceil \frac{nH}{k} \right\rceil \right)$ (26)	$Ov = \frac{S + \lceil rn \rceil H}{m}$ (27)	$D = \frac{2n}{\alpha}$ (28)	$Bfr = n$ (29)

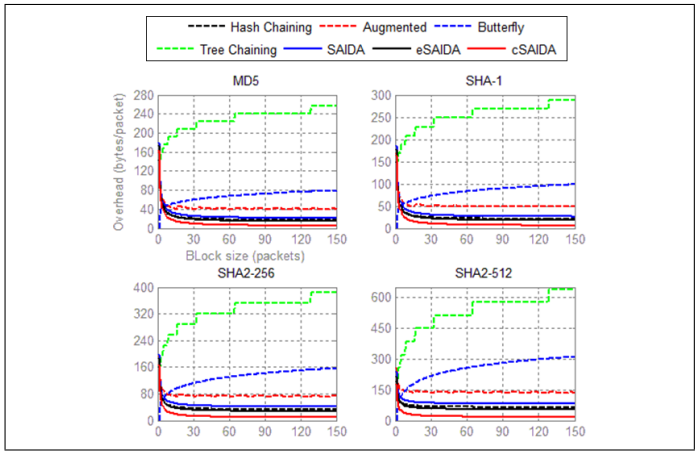

Fig. 4: Communication overheads versus block size

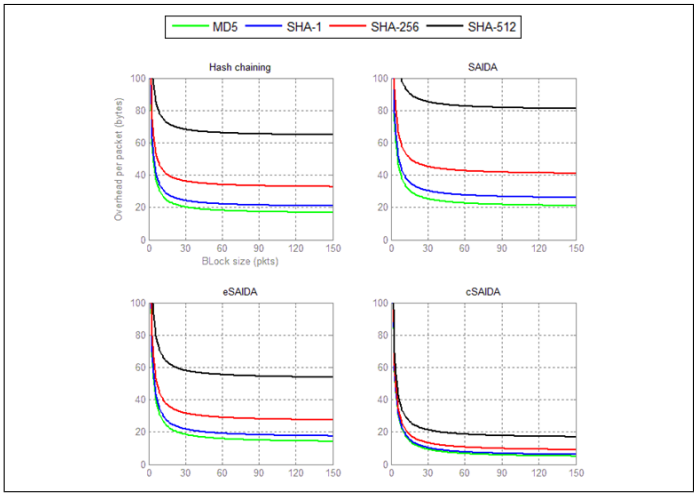

Fig. 5: Impact of hashing algorithms on overhead (1)

tree chaining is the worst.

(2) The worst computational cost variation is about 25% for A8 and A9 while it is about 13% for i5

B. Communication Overheads comparison

The communication overhead per-packet is plotted against the block size n using four hashing algorithms for all authentication schemes in Fig. 4 and it shows that

1) The cSAIDA authentication scheme has the least value of communication overhead
2) Overhead becomes almost stable for block sizes greater than 60 packets for all schemes, except for the tree chaining scheme where the overhead increases largely as the block size increases and butterfly which has a less increasing rate than tree chaining.

C. Impact of hashing algorithm on communication overhead

The communication overhead for each authentication scheme is plotted in Fig. 5 and Fig. 6 using the different hashing algorithms MD5, SHA-1, SHA-256 and SHA-512 and they show that, the cSAIDA overhead is the least affected overhead when changing the hashing algorithm; its overhead is increased by about 10 bytes when changing hashing algorithm from MD5 to SAH-512 whereas butterfly overhead changes by about 200 bytes, tree overhead changes by about 350 bytes and the other schemes overheads change by about 50 bytes.

D. End-to-End delay calculations

Our analytical results for end-to-end ($E2E$) delay are built based on the ITU-T's G.114 recommendation that suggests that the one-way E2E should be less than 150ms and has an upper limit of 400ms. These values are most important for interactive media (e.g. video conferencing) and can be increased to an upper limit of 2s. This will not affect the stream as it is one direction and not interactive. The $E2E$ delay is the summation of the following delays (digitization, packetization, transmission, propagation, jittering, and authentication verification). We as-

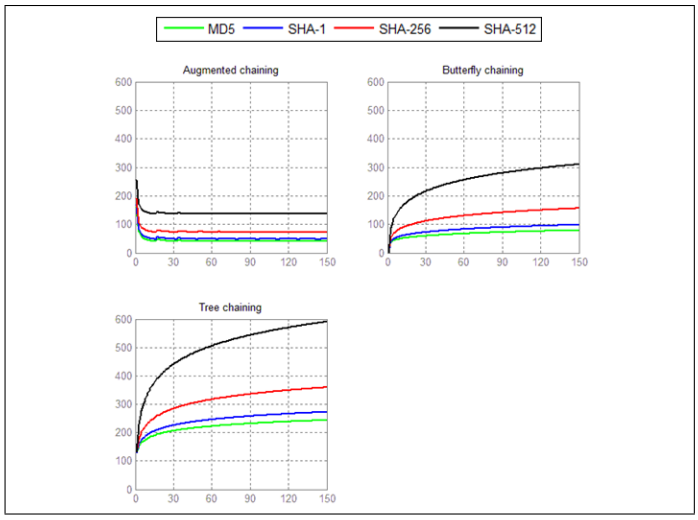

Fig. 6: Impact of hashing algorithms on overhead (2)

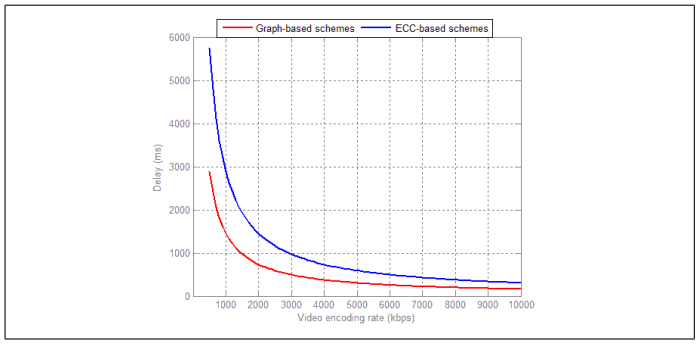

Fig. 7: E2E delay vs. encoding rate

sume that; the delays of digitization, transmission, propagation and jittering are negligible. So, $E2E$ delay can be calculated as:

$$E2E = D_{auth} \tag{30}$$

Equations (4), (8), (12), (16), (20), (24), (28) show that, ECC-based authentication schemes have a delay values that are double of that of graph-based authentication schemes. Fig. 7 illustrates $E2E$ delay vs. different values for video rates starting from 500 kbps till 10 Mbps for a block size of 128 packets and the hashing algorithm is SHA-1 and it shows that, graph-based schemes delay values becomes less than 2s when the bitrate is approximately 700 kbps or higher compared to 1500 kbps for ECC-based schemes delay values.

VI. Conclusion

As a result of research conducted, it is proven mathematically that the performance and security are tightly coupled in IPTV services. Accordingly, the security of IPTV authentication scheme could be tuned according to the performance metrics goals. For example, the tree authentication scheme could be used with minimal impact on computational cost in light-weight hash algorithms (e.g. SHA-1) while in relatively higher security hash algorithms (e.g. SHA-512); it makes a significant change in computational performance. It is proved that it exceeds 25% in the time needed to verify the received block. As a result, performance impacts the selection of optimum security algorithm and authentication scheme has a significant impact on IPTV system performance depending on the type of hashing algorithm and security level used. In addition, the security strength impact on computational performance is studied versus different processing platforms and it's concluded that the computational performance is affected dramatically by processing platform used (e.g. computational performance is decreased by more than 60% if processor changes from A8 to i5 for instance). The above indicates the fact that, NO-One-Fits-For-All and the parameters of systems as processing power could be selection criteria for optimum algorithm to run on each platform and consequently corresponding security level.

The research results point out that eSAIDA has the lowest computational cost on i5 platform while tree chaining has the highest computational cost on A8 platform. From the other prospective, communication overhead is strongly affected by the used hashing algorithm. For instance, the cSAIDA overhead has the least value and it has also the least affected overhead when changing the hashing algorithm.

References

[1] ATIS IPTV Exploratory Group Report and Recommendation to the TOPS Council.Alliance for Telecommunications Industry Solutions. July 2006. http://www.atis.org/tops/IEG/ATIS_IPTV_EG_RPT_final.pdf

[2] Y. Challal, H. Bettahar, and A. Bouabdallah. A taxonomy of multicast data origin authentication: Issues and solutions. IEEE Communications Surveys and Tutorials, 6(3):34-57, July 2004.

[3] A. Pannetrat and R. Molva. Efficient multicast packet authentication. In Proc. of Network and Distributed System Security Symposium (NDSS'03), San Diego, CA, February 2003.

[4] K. Mokhtarian. Efficient and secure delivery of scalable video streams. Diss. School of Computing Science-Simon Fraser University, 2009.

[5] R. Gennaro and P. Rohatgi. How to sign digital streams. In Proc. of Advances in Cryptology (CRYPTO'97), vol. 1294 of LNCS, Santa Barbara, CA, August 1997. Springer-Verlag, pp 180-197

[6] P. Golle and N. Modadugu. Authenticating streamed data in the presence of random packet loss. In Proc. of Network and Distributed System Security Symposium (NDSS'01), San Diego, CA, February 2001, pp 13-22

[7] Z. Zhang, Q. Sun, and W. Wong. A proposal of butterfly-graph based stream authentication over lossy networks. In Proc. of IEEE International Conference on Multimedia and Expo (ICME'05), Amsterdam, The Netherlands, July 2005, pp 784-787

[8] Z. Zhishou, Q. Apostolopoulos, J.and Sun, S. Wee, and W. Wong. Stream authentication based on generalized butterfly graph. In Proc. of IEEE International Conference on Image Processing (ICIP'07), vol. 6, San Antonio, TX, September 2007, pp 121-124

[9] C. Wong and S. Lam. Digital signatures for flows and multicasts. IEEE/ACM Transactions on Networking, Vol. 7, No. 4, August 1999, pp 502-513.

[10] R. Merkle. A certified digital signature. In Proc. of Advances in Cryptology (CRYPTO'89), vol. 435 of LNCS, Santa Barbara, CA, August 1989. Springer-Verlag, pp 218-238

[11] J. Park, E. Chong, and H. Siegel. Efficient multicast stream authentication using erasure codes. ACM Transactions on Information and System Security, vol. 6 no. 2, May 2003, pp 258-285

[12] Y. Park and Yookun Cho. The eSAIDA stream authentication scheme. In Proc. of International Conference on Computational Science and Its Applications (ICCSA'04), vol. 3046 of LNCS, Assisi, Italy, May 2004, pp 799-807

[13] J. Lu. Signal Processing for Internet Video Streaming. Proc. of SPIE, Image and Video Communications and Processing, San Jose, CA, USA, January 2000, pp 246-259

[14] ECRYPT Benchmarking of Cryptographic Systems. "Measurements of hash functions, indexed by machine". June 2014. http://bench.cr.yp.to/results-hash.html