Unequal Authenticity Protection (UAP) for Rate-Distortion-Optimized Secure Streaming of Multimedia Over Wireless Networks

Zhi Li
Department of ECE
National University of Singapore
Email: lizhi@nus.edu.sg

Qibin Sun
Institute for Infocomm Research
A-STAR, Singapore
Email: qibin@i2r.a-star.edu.sg

Yong Lian
Department of ECE
National University of Singapore
Email: eleliany@nus.edu.sg

***Abstract*— This paper presents a new notion of authenticating degraded multimedia content streamed over wireless networks – Unequal Authenticity Protection (UAP). Multimedia content differs from other data in that the importance of different bits within a bitstream often varies. Therefore, given limited resources, a natural solution is to apply better authenticity protection to more important bits, and vice versa. In this paper, a quantitative relationship between the optimal authentication probability and the given resource budget is firstly derived, followed by a proposed authentication graph which realizes the idea of UAP. Simulation results further confirm the validity of the proposal.**

I. Introduction

With the increasing availability of bandwidth in the new-generation mobile networks, the rapid growing of wireless delivery of video, image or audio content has created an urgent need for authenticating degraded multimedia content caused by channel distortions.

We consider two services provided by an authentication mechanism – *data integrity* and *non-repudiation*. A naive solution to authenticating a potentially long stream is to sign each network packet using digital signature such as DSA. However, the problem is that signing algorithms nowadays are computationally expensive, and it is not worthy to compute and verify one signature for each packet. One solution undergoing intensive research is *signature amortization through hash-chaining* [1]–[5]. The rationale is: since it is too expensive to sign every packet of the stream, we can organize packets into groups and sign only one packet within each group. The authenticity of the rest packets is guaranteed in the following way: if we compute the hash of packet P_i and append it to P_{i+1} before signing P_{i+1}, then the authenticity of P_{i+1} also guarantees the authenticity of P_i. In this manner, each packet is hash-chained with the succeeding packets up to the signature packet. Then the authenticity of the signature packet will "propagate" through all the rest packets within the group. In addition, in order to ensure that the authentication chain is not broken due to packet loss, each packet may assign its hash to multiple other packets. In this way, designing the entire authentication scheme can be abstracted as constructing a directed acyclic *Authentication Graph* (AG) of parameter (V, G), where V is the set of nodes (or network packets) and G is the set of directed edges. (Throughout this paper, we use the terms *node* and *packet* interchangeably.)

Based on the hash-chaining algorithm, in this work, we are concerned with designing an authentication system that is best suited for multimedia stream. Multimedia content differs from other data in that the importance of different bits within a bitstream often varies. Common media compression algorithms produce bits that contribute quite differently to the reconstructed media quality. For example, in wavelet progressive coding, the LL subband coefficients contribute more to the reconstructed quality than the HL, LH and HH subband coefficients; in DCT-based video coding, the transmission errors in intra-coded macroblocks cause more severe quality damage than those in the inter-coded macroblocks. Therefore, *Unequal Error Protection* (UEP) is a natural solution of protecting multimedia stream against channel distortions. Similarly, this idea can also be extended to authenticating multimedia stream over an unreliable channel. At the receiver end, the media authenticity needs to be verified before decoding. However, for most of the signature amortization algorithms (except for [6], which provides guaranteed authentication for each packet, but also leads to huge communication overhead), there is always a probability that some bits are not verifiable due to packet loss, even if we can apply channel coding techniques to reduce the non-verifiability. Naturally, a good solution is to apply better protection (*i.e.,* more hash chains) to bits that is more important to the reconstruction quality (or information sensitivity, and *etc.*) while apply less protection to other bits. In this work, we present the idea of *Unequal Authenticity Protection* (UAP) – a better strategy for authenticating multimedia content streamed over wireless networks. Based on the idea of UAP, we are able to consider the authentication solution in a rate-distortion-optimized way in the sense that given limited resources, we want to optimally allocate resources to mitigate the impact of packet non-verifiablity on the end-to-end reconstructed multimedia quality.

The rest of this paper is organized as follows. Section II introduces the criteria that measure the efficiency of an AG construction. Section III derives a theoretical upper bound of the achievable authentication probability. Section IV presents a method of constructing the AG to approach the theoretical upper bound. In Section V, MUC – one AG construction that realizes the notion of UAP is presented, followed by the discussion of optimal hash bit allocation. Simulation results are given in Section VI. Conclusions are drawn in Section VII, followed by the discussion about the future work.

0-7803-9390-2/06/$20.00 ©2006 IEEE

II. Measurements

The properties of the constructed AG determine the efficiency of the hash-chaining scheme. Some parameters of the AG includes: *i)* number of edges (*i.e.*, number of succeeding hash-chained nodes) for each node, *ii)* how the nodes are chained, *iii)* the total number of nodes amortizing one signature packet, and *iv)* number of nodes directly chained to the signature packet. The efficiency of a specific hash-chaining scheme is measured by:

Authentication Probability (AP) for each particular node P_i, denoted by ξ_i. AP, the measure of robustness against packet loss, is defined as $\Pr(P_i \text{ is verifiable} | P_i \text{ is received})$. In this work, we consider an *i.i.d.* Binary Symmetric Channel (BSC) for wireless channel, and thus a resultant *i.i.d.* packet loss model (in many cases, bursty errors can be converted to random errors through interleaving), then the AP is equal to $\Pr(P_i \text{ is verifiable})$. AP of a node is determined by the status of its succeeding hash-chained nodes. More precisely, if we denote the event that P_i is *verifiable* by Λ_i, and the event that P_i is *received* by Π_i, then:

$$\xi_i = \Pr(\Lambda_j \Pi_j + \Lambda_k \Pi_k + ...) \tag{1}$$

where P_j, P_k,... are P_i's succeeding hash-chained packets. Generally the more hash-chained packets it has, the higher the AP. In general, within a AG, different nodes has different AP's; therefore, $\xi_{\min} = \min_i(\xi_i)$ is a proper measure of the whole scheme's AP.

Communication overhead, including the signature packet and the hash bits appended to each node. The signature packet size is determined by the number of nodes directly chained to the signature packet. In practice, we preset the number of directly chained nodes before constructing the AG, so that the signature packet size is fixed. For the hash bits, if we use a standard SHA-1 scheme, the size of each hash is 160 bits.

Verification delay, determined by the total number of packets amortizing one signature packet.

Transmitter and receiver buffer size, determined by how the nodes are chained with each other.

Among all these measures, we are particularly interested in the interplay between AP and the hash bits overhead. More precisely, given the hash bit budget, we want to find the theoretical upper-bound of $\xi_{\min}$.

III. Theoretical Upper Bound of AP

To examine the theoretical upper-bound of $\xi_{\min}$, we have the following proposition:

Proposition: Let P_j and P_k be any two nodes in the AG, then:

$$\Pr(\Lambda_j \Lambda_k) \geq \Pr(\Lambda_j)\Pr(\Lambda_k) \tag{2}$$

The equality holds when Λ_j and Λ_k are independent.

***A Sketchy Proof**: For any two nodes P_j and P_k in the AG, they may or may not have common hash-chained nodes. In case of the later, the events Λ_j and Λ_k are independent of each other, and therefore Eq. (2) holds with equality. The case that they have one common nodes are illustrated in Fig. 1. From Eq. (1) we can show $Pr(\Lambda_{c-1}|\Lambda_c) > Pr(\Lambda_{c-1})$ and $Pr(\Lambda_{c-2}|\Lambda_{c-1}) > Pr(\Lambda_{c-2})$ (where P_{c-1} is P_c's hash chained packet and so on). Hence, we can show $Pr(\Lambda_{c-2}|\Lambda_c) > Pr(\Lambda_{c-2})$. As such, we can prove $Pr(\Lambda_j|\Lambda_c) > Pr(\Lambda_j)$ and $Pr(\Lambda_k|\Lambda_c) > Pr(\Lambda_k)$. The last equation leads to $Pr(\Lambda_c|\Lambda_k) > Pr(\Lambda_c)$. Therefore, we have $Pr(\Lambda_j|\Lambda_k) > Pr(\Lambda_j)$, which is equivalent to $Pr(\Lambda_j\Lambda_k) > Pr(\Lambda_j)Pr(\Lambda_k)$.* □

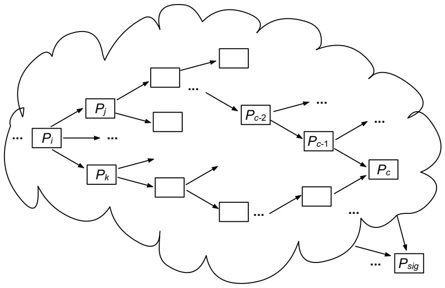

Fig. 1. Illustration of the AG with two nodes P_j and P_k having one common hash-chained node P_c.

Now consider the case that P_j and P_k are the two succeeding nodes of P_i. From Eq. (1),

$$\begin{aligned} \xi_i &= \Pr(\Lambda_j \Pi_j + \Lambda_k \Pi_k) \\ &= \Pr(\Lambda_j \Pi_j) + \Pr(\Lambda_k \Pi_k) - \Pr(\Lambda_j \Pi_j \Lambda_k \Pi_k) \\ &= \Pr(\Lambda_j)\Pr(\Pi_j) + \Pr(\Lambda_k)\Pr(\Pi_k) \\ &\qquad\qquad - \Pr(\Lambda_j \Lambda_k)\Pr(\Pi_j)\Pr(\Pi_k) \end{aligned} \tag{3}$$

From Eq. (2),

$$\begin{aligned} \xi_i &\leq \xi_j \Pr(\Pi_j) + \xi_k \Pr(\Pi_k) - \xi_j \xi_k \Pr(\Pi_j)\Pr(\Pi_k) \\ &= 1 - \Big(1 - \xi_j \Pr(\Pi_j)\Big)\Big(1 - \xi_k \Pr(\Pi_k)\Big) \end{aligned} \tag{4}$$

That is, ξ_i is optimal when the dependency of Λ_j and Λ_k are fully de-correlated. We further assume the packet loss rate e is the same for every node, *i.e.*, $\Pr(\Pi_j) = \Pr(\Pi_k) = ... = 1 - e$. In addition, since we are interested in finding $\xi_{\min}$, the best case happens when $\xi_{\min} = \xi_i = \xi_j = \xi_k = ... = \xi_{\text{opt}}$. Then:

$$\xi_{\text{opt}} = 1 - \Big(1 - \xi_{\text{opt}}(1 - e)\Big)^2 \tag{5}$$

In general, when P_i have m succeeding nodes, the optimal AP can be found by solving:

$$\xi_{\text{opt}} = 1 - \Big(1 - \xi_{\text{opt}}(1 - e)\Big)^m \tag{6}$$

IV. Constructing the Optimal AG

Now that we have obtained ξ_{opt}, the theoretical upper bound of AP, the following work is to find a method of constructing the AG, such that the resulting AP can approach ξ_{opt}. Here we consider a group of packets that share one signature. Since the signature packet P_{sig} is of primary importance, we protect with strong FEC. In this work, for simplicity, we assume that P_{sig} is always received. Therefore, the packets directly chained to

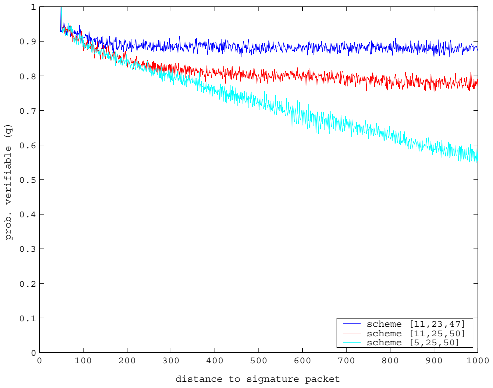

Fig. 2. Comparisons of AP's constructed by schemes $[11, 23, 47]$, $[11, 25, 50]$ and $[5, 25, 50]$.

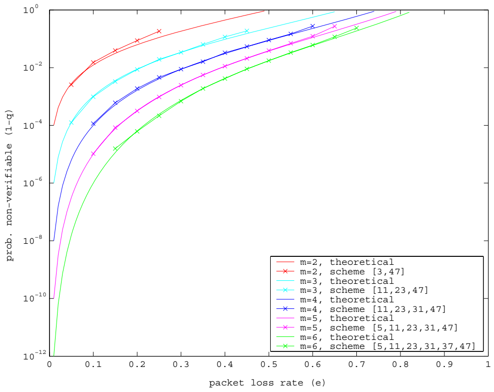

Fig. 3. Comparisons between the performance of some chosen schemes and the theoretical upper bound of AP.

P_{sig} have AP of 1. We call these packets *Pilot Packet*. Usually for each group the number of pilot packets M_{pp} are preset so that the size of P_{sig} is fixed.

From the analysis of Section III, we have seen that in order to achieve the optimal AP, we must de-correlate the dependency between packets. This can be achieved in either a deterministic or a statistical manner. In [5], Zhang *et al.* have proposed a deterministic method to construct the butterfly-graph-based AG. In their method, each packet has two succeeding hash-chained packets, which are organized in a way that it is guaranteed they have no common succeeding packet. Therefore, the two packets are fully de-correlated, resulting near-optimum AP's. However, in this scheme, the total number of packets has to be $2^L(L+1)$, where L is the number of layers. This constraint greatly circumscribes the choice of the group size. In [2], instead of analytically computing the AP of each packet, Perrig *et al.* have adopted an experimental approach to examine the dominant factors influencing AP. One of their main findings is that it is highly probable to construct a good AG by randomly choosing the chaining scheme. In this work, we extend their analytical approach. We follow their notations to use [a, b, c] to denote the scheme in which packet P_i is hash-chained to packet P_{i+a}, P_{i+b} and P_{i+c}, where a, b and c are called *chaining distance*. We find that it is easy to construct a good AG by making the chaining distances *relatively prime* with each other. For example, Fig. 2 illustrates the performance of chaining schemes $[11, 23, 47]$, $[11, 25, 50]$ and $[5, 25, 50]$ ($e = 0.4$, number of simulations $= 1000$).

We can see that for a good scheme, the AP's can be maintained at a constant level no matter how far away the packets are from the signature packet (*e.g.*, scheme $[11, 23, 47]$ of Fig. 2). This fact supports our assumption that $\xi_{\min} = \xi_i = \xi_j = \xi_k = ... = \xi_{\text{opt}}$. We call the scheme is *stable* if it has this property. In general, a scheme's stability varies with the packet loss rate e. If a scheme is stable for $e \leq 0.5$, we say the scheme's stable region is $[0, 0.5]$. Intuitively, a good scheme has the ability of statistically de-correlating the dependence between packets. However, since the correlation cannot be fully reduced to 0, the effect of dependence prevails when the packet loss rate is high.

In Fig. 3, we compare the performance of some chosen schemes for each m with the theoretical upper-bound of $\xi_{\min}$ (within the stable region only). We plot the probability that a packet is not verifiable, *i.e.*, $(1 - \xi_{\text{opt}})$ in the log scale for better illustration. The results show that under this statistical approach, the chosen schemes are able to achieve the optimal AP in most of the cases.

It is worth noting that in [2], Perrig *et al.* have proposed the idea of using *Information Dispersal Algorithm* (IDA) to further improve the AP. However, this improvement is at the expense of increasing the number of pilot packets, thus the size of the signature packet. In addition, our analysis in the previous section can be easily extended to IDA as well. For our work here, we will stick to the basic scheme for simplicity.

V. MUC - A Realization of UAP

In the previous sections, we have derived the quantitative relationship between the optimal AP and the hash-bit overhead (Eq. (6)). This expression is important, since given the channel condition (*i.e.*, packet loss rate e) and the required AP, we can quantitatively compute the hash overhead needed to achieve this AP. We have also identified some schemes of AG construction to achieve this optimal AP. However, we see that these schemes produce equal AP's for all packets. In order to produce packets of unequal AP's, one solution is to group packets and use different m's for different groups. In this section, we propose a construction of AG with controllable unequal AP's – *Multi-layer Unequal Chaining* (MUC).

Fig. 4 illustrates the structure of MUC. In MUC, the packets are organized in multiple layers. In layer L_i, each packet is hash-chained to i other succeeding packets based on the good chaining scheme described in the previous section. For each layer, there are some pilot packets which are directly chained to the signature packet P_{sig}. In this manner, each layer is similar to the construction of equal AP described in the previous section. Therefore, the AP's of different layers can be computed by Eq. (6). We let fixed fraction of packets to be the pilot packets (*e.g.*, 5%) so that the signature packet size

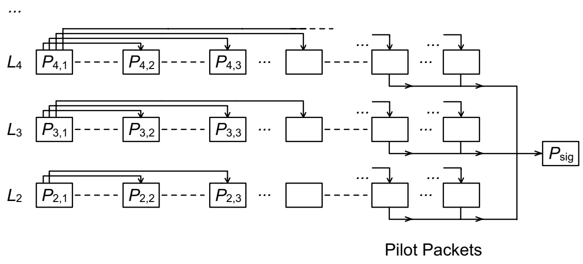

Fig. 4. Structure of the MUC AG.

is also fixed. Note that increasing the number of layers will lead to decreasing the number of pilot packets for each layer. As a result, the de-correlating effect will reduce. However, correspondingly the chain length will also reduce. Our experiment results show that this can properly balance the reduction of the de-correlating effect. As a result, for each layer the chaining stability can still be maintained. Another point to note is that it is undesirable to chain packets across different layers. For example, it appears that we could chain lower-layer (LL) packets to higher-layer (HL) packets to further improve the LL packets' AP. However, this creates the LL packets' dependence to HL packets. As a result, the loss of a HL packet becomes more expensive since it now also influences the LL packets' AP. Therefore, it is better to leave each layer unchained with one another. Based on this AG construction, the rest problem is how to allocate the packets to each layer most effectively. We seek to maximize the quantity $\overline{\xi}_{\text{opt}} = \max_{\{\xi_i\}} \left(\frac{\sum_{i=1}^{M} W_i \xi_i}{\sum_{i=1}^{M} W_i} \right)$ which is a representation of the scheme's robustness against packet loss, given an overall hash bit budget.

VI. Simulation Results

To examine the efficiency of the MUC construction, we have selected 16 test images of size 512×512 as the input source. We perform block-based DCT on each image; each 8×8 block is packetized as one network packet. The variance of DCT-coefficients is taken as the weight of each packet. In addition, the central $1/9$ area is defined as ROI and the packet weights are doubled. In all, there are 4096 packets amortizing one signature (in practice the group size need not be so large; here we want to examine as how large the group size can be). The number of pilot packets is chosen to be 5% of the overall packets.

In Fig. 5 and Fig. 6, we present the simulation results for *lena* image (all other sources have similar results). Fig. 5 illustrates $\overline{\xi}_{\text{opt}}$ under various given average hash chains per packet $\overline{m}$ and some packet loss rate e for *i)* UAP, and *ii)* Equal Authenticity Protection (EAP). It is clearly shown that UAP has better performance than EAP. In Fig. 6, we compare the expected result based on the optimal bit allocation algorithm in Section V, and the observed result through simulation. We can see that the observed result is very close to the expection, except for some cases when given low $\overline{m}$. This is because these conditions are outside the stability region of the chosen scheme. In practice, we can always employ further constraint to avoid these cases.

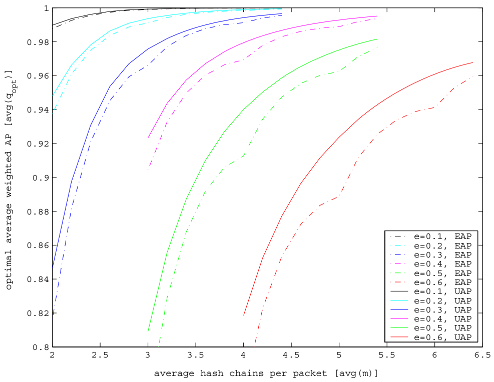

Fig. 5. Simulation results: Comparison between UAP and EAP.

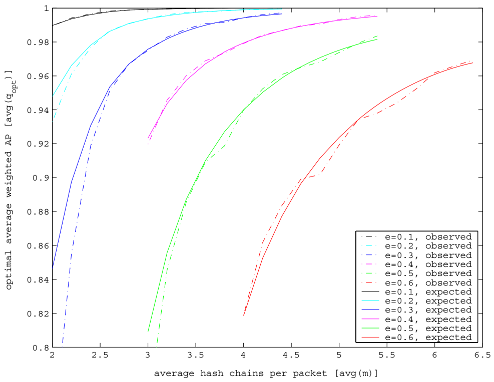

Fig. 6. Simulation results: Comparsion between the expected result through the optimal bit allocation algorithm and the observed result through simulation.

VII. Conclusions and Future Work

This paper presented *unequal authenticity protection* – a new notion of multimedia stream authentication. We have given theoretical analysis as well as a practical AG construction to realize this idea. In our future work, we shall integrate this scheme into a complete source/channel coding system, to optimally allocate resources and achieve a joint optimization of end-to-end multimedia quality.

References

[1] R. Gennaro and P. Rohatgi. How to sign digital streams. In *Advances in Cryptology*, pages 180–197, Aug 1997.

[2] A. Perrig, R. Canetti, J. D. Tygar, and D. Song. Efficient authentication and signing of multicast streams over lossy channels. In *IEEE Symposium on Security and Privacy*, pages 56–73, May 2000.

[3] P. Golle and N. Modadugu. Authenticating streamed data in the presence of random packet loss. In *Network and Distributed System Security Symposium*, pages 13–22, Feb 2001.

[4] S. Miner and J. Staddon. Graph-based authentication of digital streams. In *IEEE Symposium on Security and Privacy*, pages 232–246, May 2001.

[5] Z.S. Zhang, Q.B. Sun, and W-C. Wong. A proposal of bufferfly-graph based stream authentication over lossy networks. In *IEEE International Conference on Multimedia and EXPO*, Jul 2005.

[6] C.K. Wong and S. Lam. Digital signatures for flows and multicasts. *IEEE/ACM Transactions on Networking*, 7(4), August 1999.