IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2017, accepted August 9, 2017, date of publication August 29, 2017, date of current version September 19, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2743019

Authentication of Scalable Video Coding Streams Based on Topological Sort on Decoding Dependency Graph

QIANG MA[1], LING XING[2], AND LONGSHUI ZHENG[1]
[1]School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China
[2]School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China

Corresponding author: Ling Xing (xingling_my@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61171109, in part by the Applied Basic Research Programs of Sichuan Science and Technology Department under Grant 2014JY0215, and in part by the Key Scientific Research Program of Henan Higher Education under Grant 18A510009.

ABSTRACT The security of streaming data should be ensured in current complex data era in order to provide a trusted and secure network environment. To authenticate the scalable video coding (SVC) streams by fully utilizing its decoding relationship without reducing its scalability, we establish an acyclic and directed decoding dependence graph (DDG) on the logical units of SVC steams. By applying the topological sort on DDG, we obtain the hash appendence mode for different layers of the streams (i.e., spatial and temporal layers). We propose a secure and efficient SVC authentication method based on the deduced hash appendence mode. With regard to the quality layer, we consider the corresponding quality data packets with unequal importance, and propose a grouping authentication strategy with constrained group lengths. We form the optimization problem of minimizing the authentication cost, and solve it by an iteration method. Simulation results show that our authentication approach can achieve much less computation cost and much lower overhead, while it can preserve higher verification rates and better recovered quality of video as compared with other state-of-the-art methods.

INDEX TERMS Multimedia security, streaming authentication, scalable video coding, decoding dependency graph, grouping authentication strategy.

I. INTRODUCTION

During the big data era, multimedia streaming has prevailed in current network-based services and applications (e.g., video conference, video broadcasting, Internet Protocol Television) [1]–[5]. Video streams are usually transferred over the Internet and are supported by network streaming protocols. However, the error-prone and unsafe network environments bring security issues to the streamed video content. The recipient needs to ensure the integrity of the received video packets before replaying [6]–[8]. Although many research have been conducted in video streaming authentication, an efficient and secure video authentication is a demanding task since it has to take into consideration many factors (e.g., computation cost, delay, verification rates and frame quality reduction) [9]–[11].

To date many architectures and methods have been proposed to authenticate video streaming [12]. As far as the features of video streams are concerned, authentication methods can be roughly categorized into two kinds, i.e., for non-scalable streams and for scalable streams.

Gennaro and Rohatgi [13] proposed the pioneer work of authenticating streams via hash chain. The hash chain is vulnerable to network packet loss since the loss can break down the chain, which makes the recipient fail to authenticate the incoming streams. Later, some improvements had been made to the hash chain to strengthen its robustness against packet loss, e.g., Wong and Lam [14] suggested the tree-based streaming authentication, Park *et al.* [15] bettered tree-based approach with sub-tree sets in order to increase verification rate, Zhang *et al.* [16] proposed the butterfly-based method which arranged a group of video packets into a butterfly graph. In order to decrease the authentication overhead, some methods adopted the amortization signature,

e.g., SAIDA (Signature Amortization using Information Dispersal Algorithm) [17].

Considering the existing unequal importance of packets for video recovery, some researches proposed video authentication methods with unequal protection. Wang *et al.* [18] used unequal error protection for streams authentication, in which quality-driven strategy was adopted to better the resource allocation. Zhu and Chen [19] addressed the H.264/AVC (Advanced Video Coding) authentication by means of source-channel adaptive scheme, which allocates bit rates to video packets according to their importance towards video quality degradation.

As for the authentication methods catered for scalable streams, various kinds of methods have been proposed. Mokhtarian and Hefeeda [20] used FEC (Forward Error Correction) channel coding to authenticate SVC (Scalable Video Coding) streams, where hashes of higher layer's packets are concatenated and are encoded into codeword being attached into lower layer's packets. Similarly, Zhao *et al.* [21] adopted ECC (Erasure Correction Code) for base layer's packets hashes and hash appendence was also processed in a linear chain mode. But Zhao's work did not provide scalability for the quality layer, while Mokhtarian's work endeavored to group quality packets and authenticate each group individually. The grouping authentication was meant to decrease the overhead and provide authentication scalability at quality layer. However, its main drawback is that it neglects the unequal importance of different layer quality packets. The reconstructed video quality and verification rate at the recipient could be improved.

Some SVC streams authentication methods adopt the hybrid methods. Wei *et al.* [22] used the content-based robust hash to extract enhancement layer features and cryptographic message authentication code (MAC) from base layer. The hybrid tag consisted of hash and MAC. Then tag was conveyed securely to the recipient. This method allows coarse authentication. But the tradeoff between hash's robustness and security is difficult to decide. Some SVC streams authentication adopts joint coding and authenticating. Yi *et al.* [23] proposed a joint coding and streaming authentication model. But the model does not explicitly employ the decoding relationship. Moreover, it is quite hard to be adapted to SVC streams authentication.

Other scalable authentication methods do not take into consideration the decoding relationship as methods mentioned above. On the contrary, the scalability of their methods means the multi-layer authentication. Atrey *et al.* [24] proposed a scalable signature for video authentication, where authentication could be performed at frame, shot or video level by means of interpolation. Tew *et al.* [25] developed tag-based authentication on three layers, i.e., coded units, quantization parameters and prediction mode. The tag is generated from video statistical features. However, this kind of authentication cannot be directly applied to SVC, since the streams' decoding structure is overlooked and extracted sub-streams may not be verified at the receiver side.

In this paper, we propose an efficient and secure authentication method for SVC streams. In summary, our contributions are threefold. First, we construct an acyclic and directed decoding dependency graph (DDG) for spatial and temporal layers. The graph provides the decoding order for the recipient. The authentication hash appendence mode is obtained by the topological sort. Second, we formulize the authentication cost by a minimal optimization for the quality layer of SVC streams. The optimization incorporates the philosophy of unequal protections across different quality packets. We obtain the optimal grouping strategy by solving the problem with an iteration method. Third, we propose the architecture and algorithms for the authentication. We implement the scheme by simulation and compare our method with state-of-the-art methods in terms of computation cost, overhead, verification rates and video quality.

The rest of this paper is organized as follows. We provide a brief preliminary on the SVC structure in Section II. In Section III we present the proposed architecture, where the DDG and topological sort on DDG are explained in detail. We describe the grouping method for quality layer of SVC in Section IV. Section V provides the security analysis of the authentication scheme. Section VI evaluates the performance, and section VII concludes this paper.

II. PRELIMINARY

The video coding standard of H.264/SVC was proposed by Joint Video Team (JVT), which was cofounded by ITU and ISO/IEC joint video team. H.264/SVC has been widely accepted in various applications, e.g., network-based video conference, video surveillance, video streaming service. It is the extension of previous video coding standard H.264/AVC. Sometimes it is called the G appendance towards H.264/AVC. H.264/SVC adds scalability to H.264/AVC and allows sub-streams extraction. Moreover, it achieves much higher compression rate than H.264/AVC, which means under the same bit rate condition the H.264/SVC has much better video quality [26].

The scalability of SVC is represented in three dimensions, i.e., spatial, temporal and quality. It can provide all three dimensional scalabilities at the same time. The scalability is realized by a base layer and several enhancement layers. Temporal scalability means that one GOP (Group of Pictures) consists of a base access unit (AU) and some enhancement AUs. Each AU is a decoded frame. An AU of higher layer is predicted by AU or AUs of lower layers. Fig. 1 illustrates a temporal prediction mode of GOP, which includes eight AUs with four layers. For instance, the AU 4 in temporal layer two T_2 is jointly predicted by the AU 2 in layer one T_1 and the AU 1 in layer zero T_0. The sub-stream is extracted by dropping some temporal AUs of higher layer. However, the base AU is a must for all streams. It provides video quality with the least tolerance.

The spatial scalability provides different frame resolutions for each AU. For each spatial layer of AU, scalability is implemented by inter-frames prediction. In order to

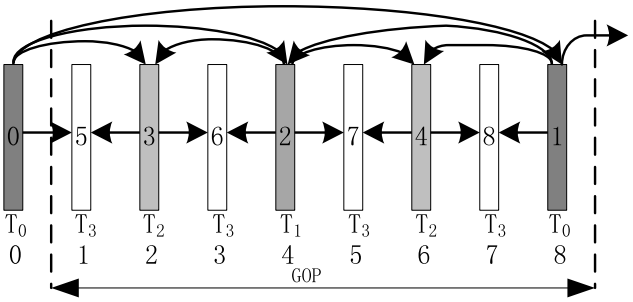

FIGURE 1. Prediction mode of temporal level within group of pictures.

decrease the memory consumption and the decoding complexity, SVC requires that all spatial layers across all AUs adopt the same decoding order [12]. The quality scalability means that different PSNR (Peak Signal to Noise Ratio) values can be provided for each temporal AU. Two approaches can be used for the quality scalability: one is CGS (Coarse Grain Scalability), which is akin to spatial scalability but has different quantization coefficients and resolutions; the other is MGS (Medium Grain Scalability), which separates DCT coefficients into different quality layers [19]. Compared to CGS, MGS is much more flexible. Because the coefficients of MGS are scanned in a Zigzag manner from the top-left corner, the sub-steams can drop arbitrary number of MGS packets. The remaining MGS packets are still useful for decoding. Note that the direct and lower frequencies of coefficients are kept in lower quality layers of MGS packets, which maintain the most important visual information for the reconstructed frame.

In order to transfer the video packets on the Internet, video streams are wrapped in form of NAL (Network Abstraction Layer) packets. NAL consists of two kinds, i.e., VCL (Video Coding Layer) NAL and non-VCL NAL. The former is used for video coded data, while the latter is used for video decoding parameters. The SEI (Supplemental Enhancement Information) non-VCL NAL packets can be adopted for extra information insertion. In our work we choose the SEI packets for authentication embedding, e.g., signature data transfer.

III. PROPOSED ARCHITECTURE

A. DECODING DEPENDENCE GRAPH CONSTRUCTION

Denote the graph DDG by $G(v, e)$, where v stands for the vertex set of logical units and edge set e stands for the decoding relationship between vertexes. The logical units in the DDG can be either AUs of temporal dimension or layers of spatial dimension. Suppose v is represented by $\{v_1, v_2, \ldots, v_n\}$, which means there are total n vertexes in the DDG. We map the decoding order of vertexes as the index order of v. In other words, the recipient decodes the vertexes $v_1, v_2, \ldots, v_n$, sequentially. Each edge of e is directed. The edge is denoted as $v_i \rightarrow v_j$, which states that vertex v_j is decoding dependent on v_i. Thus it means that vertex v_j is decodable only if vertex v_i is correctly received and verified.

We adopt the adjacent list to store DDG and denote it by L, whose structure is depicted in Fig. 2. The list head consists of two field, i.e., the out degree of current vertex and the list nodes' first edge point. The former reflects how many other vertexes are decoding dependent on this vertex, while the latter points to the first node of the linked list. The linked list is the one connecting the vertexes whose edges go to the current vertex. The element of the linked list, i.e., the list node, consists of vertex index (denoted by *VertexID*) and the point field (denoted by *Next*).

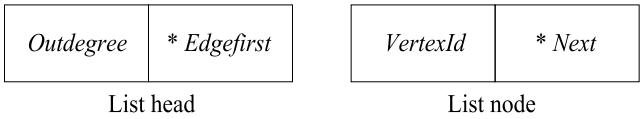

FIGURE 2. List head and List node composition of adjacent list.

The content of the adjacent list L is determined by the streams' coding structure. For instance, in SVC streams the temporal prediction mode of AUs can be either 'IBBP' or 'IPPP', where I stands for Intra-predicted frame, B stands for Bidirectional predicted frame and P stands for Predicted frame. As for the specific values of out degree in the list head, if the out degree of certain node is zero, it states that no other nodes are decoding dependent on this node; if the value is greater than one, it states that at least one other node is dependent on it; if certain node has the highest value, it means that this node is the most important one since it is highly decoding dependent by others. Usually this node is the base logical unit, i.e., the base layer of temporal or spatial layers.

B. TOPOLOGICAL SORT ON DDG

In order to obtain the authentication hash appendence mode, we apply the topological sort method on DDR according to the the adjacent list. If $L[i].Outdegree$ equals zero, we append the hash of vertex v_i (denoted by H_i) to the node with the highest value of *VertexId* (denoted by $max\{VertexId\}$) of the linked list pointed by $L[i].Edgefirst$. Meanwhile we decrease the value *Outdegree* of vertexes of the linked list pointed by $L[i]$. *Edgefirst* by one. Note that the value H_i is attached to the node with $max\{VertexId\}$, because the indexes of vertexes represent the decoding order as mentioned before. As for the recipient with low bandwidth, the SVC sub-stream can be of low bit rate. No higher layers' code data or authentication information are needed. Thus it decreases the communication overhead incurred by authentication embedding.

The topological sort algorithm on DDG (denoted by *TS*) is described in Fig. 3. The input to this algorithm is the adjacent list L and the output is the hash appendence mode (denoted by *Sort*). The algorithm iteratively deals with the node with zero*Outdegree* value and appends its hash onto the vertex with $max\{VertexId\}$. Compared with other ordinary topological sort methods, this sort is meant to obtain the decoding dependency order and its result is unique. While ordinary sort methods may have various outputs, they are mainly used

```
TS(L)
Input: L
1.   Initialize Stack S: For i=1 : n
2.     if (L[i].Outdegree = 0)
3.       Push(S, i);
4.   Authenticate: while ((i=Pop(S))!=NULL)
5.           P={VertexId};
6.           if (P!=NULL)
7.             j=max{P};
8.             Sort=Sort || {i ≺ j};
9.             For every p of P { L[i].Outdegree -=1;
10.                              if ( L[p].Outdegree =0 )
11.                              Push (S, p); }
12.          else Sort =Sort || {i};
Output: Sort
```

FIGURE 3. The topological sort algorithm on DDG.

for the linear list construction of the partial order within the vertexes.

In order to improve the algorithm's runtime efficiency, we choose the data structure stack to store the list of nodes with zero *Outdegree* value. These nodes are pending to be authenticated. Generally, the algorithm can be partitioned into two parts, i.e., the initialization and the authentication. During the initialization part, the adjacent list is first traversed. All nodes with zero *Outdegree* value are pushed into the stack. While in the authentication part, nodes popped from the stack are sequentially processed and assigned the hash appendance mode. In line 8 of Fig. 3, $i \prec j$ means the hash H_i is to be appended to the content of vertex v_j and symbol || means that the hash appendence mode is concatenated. Lines 9 to 11 are used to update the *Outdegree* value of all nodes after iterations. Line 12 is used to deal with the last node remaining in the adjacent list. The last node is hashed independently and it is inserted into stack S separately.

The space and time complexity for *TS* are $O(2N + E)$ and $O(N + E)$, where N and E are the number of nodes and edges, respectively. Note that the coding style of streams remains the same across different GOPs and different spatial layers within one GOP. Thus it needs to run the *TS* on the GOP only once. Hash appendance modes for following GOPs and spatial layers can be referred to the first one and no more TS algorithm run is required. This is quite helpful to authenticate efficiently and decrease the delay needed at the recipient side.

C. AUTHENTICATION SCHEME

When authenticating the SVC streams, we first analyze from the streams' coding style the logical units graphs, i.e., the GOP, temporal layers, spatial layers and quality layers. Then we run the TS to obtain the hash appendance mode. We adopt the 'bottom-up' way to authenticate the corresponding streams packets. Fig. 4 depicts the authentication process for different temporal layers within one AU. Note that for quality layers of each spatial dimension we choose the group authentication method, which will be introduced

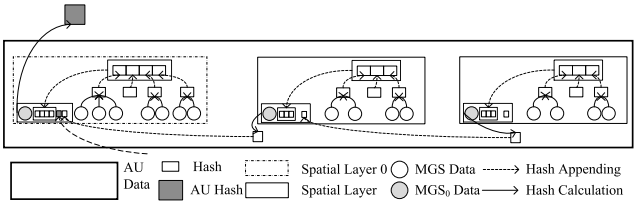

FIGURE 4. Authentication for one access unit.

in Section IV. The concatenated hashes of groups are appended in the base layer of quality dimension, which are mandatory for all sub-streams for authentication.

As for the temporal AUs within one GOP, the authentication architecture is shown in Fig. 5, where AU Data0 is the temporal base layer. Note that the hash of an AU is appended into the MGS base layer of the base spatial layer of its coding dependent AU. The dashed line means hash appending, while the real line means hash calculation. We obtain the final GOP hash from AU Data0 and sign this hash value in order to protect it from malicious manipulations. The signature serves as the starting point for the recipient to verify the incoming packets. Also in our authentication scheme, we choose to sign several GOPs instead of one GOP in order to decrease the computation cost and overhead. The NAL packet for signature is transferred separately from video code data. We repeat the transfer of the signature K times in order to avoid its transfer failure due to network packet loss. The value K is decided empirically.

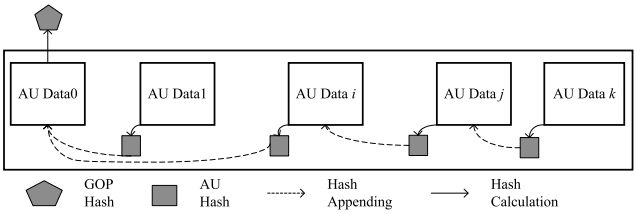

FIGURE 5. Authentication for one group of pictures.

The authentication scheme based on TS is shown in Fig. 6, which consists of four functions, i.e., AuthenticateStreams, AuthenticateGOP, AuthenticateAU and AuthenticateSL. The scheme is performed by the SVC streams server to embed authentication information. In the function of AuthenticateStreams, the input parameters n, k, *Key* and K stand for the number of GOPs to be grouped, the number of AUs within one GOP, the key to sign and the number of times to repeat signature transfer, respectively. In function of AuthenticateGOP, lines 2 to 4 are for TS execution if no decoding sort is available. Lines 6 and 9 are to authenticate the temporal AUs according to the forms of sort. Similarly, lines 2 to 4 of the function AuthenticateAU are used for TS execution to obtain the decoding sort of spatial layers. Note that for a specific SVC streams, the DDGs remain the same for all AUs layers across GOPs and for the spatial layers across AUs, thus these codes need run only once. Line 5 is for optimization

```
AuthenticateStreams(SVCStreams, n, k, Key, K)
1.  token[]=ParseSVCStreams(SVCStreams);  // Parse to obtain the boundary of units
2.  GOPSections[]=SeparateToSections(SVCStreams, token[], k); // Separate into GOP units
3.  GroupsofGOP[]=SeparateToGroups(GOPSections[], n); // Separate into Groups of GOP
4.  T=NULL;
5.  for group of GroupsofGOP[]
6.      for gop of group
7.          H= AuthenticateGOP(group);
8.          T=T || H;
9.      S=SignWithKey(T , Key);
10.     Embed T || S into stream before group in K copies;
11.     T=NULL;
AuthenticateGOP(group)
1.  AUs[]=ParseGroup(group);
2.  If (DecodingSort==NULL)
3.      VertexAUGraph= ParsetoGOPGraph(group);
4.      DecodingSort[]=TS(VertexAUGraph);
5.  for order of DecodingSort
6.      if ( order is form of {i≺ j})
7.          H_i= AuthenticateAU(AUs[i]);
8.          AUs[j] = AUs[j] || H_i;
9.      elseif (order is form of { i })
10.         H=AuthenticateAU(AUs[i]);
11. Return H;
AuthenticateAU(AU)
1.  SLs[]=ParseAU(AU);
2.  If (DecodingSort == NULL)
3.      VertexSLGraph=ParsetoAUGraph(AU);
4.      DecodingSort[]=TS(VertexSLGraph);
5.  GroupsInfo[][]=COptimization(AU)
6.  for order of DecodingSort
7.      if( order is form of {i ≺ j})
8.          H_i=AuthenticateSL(SLs[i], GroupsInfo[i]);
9.          SLs[j]= SLs[j] || H_i ;
10.     elseif(order is form of {i})
11.         H=AuthenticateSL(SLs[i] , GroupsInfo[i]);
12. Return H;
AuthenticateSL(SL, GroupsInfo[])
1.  H=NULL;
2.  for each ∈groupsInfo[]
3.      H=H || Hash(MGS(each));
4.  return H;
```

FIGURE 6. The proposed scheme for authenticate SVC streams.

of grouping quality layers. The function AuthenticateSL is mainly used for hashing grouped quality packets.

IV. GROUPING AUTHENTICATION FOR QUALITY LAYERS

In order to preserve the scalability of quality layers and to decrease the overhead brought by the authentication hashes, we adopt the grouping strategy to authenticate the quality layers packets. Mokhtarian and Hefeeda [20] addressed the grouping authentication in their proposed reducing overhead problem for SVC streams. Their work provided a theoretical analysis for the tradeoff between the groups' size and the scalability reduction. But they treated all quality layers as of equal importance. The performance of received and verified video quality could be improved.

While different quality layers possess unequal importance to the video recovery as mentioned in the preliminary, the group lengths for different layers should be different. An intuitional explanation is that data of the lower quality layer should be grouped with shorter lengths, and data of higher layer should be with longer lengths. Because data of lower layers store the transformation coefficients of direct or lower frequencies, which are more important to the frame recovery since they carry more important perception information. The shorter lengths for lower quality data allow more bit rates truncation points for lower quality of MGS video data. We improve the grouping authentication for quality layers with unequal importance attached to different layers. Our work differs from Mokhtarian's work [20] in a twofold aspect. First, we adopt a linear constraint on the quality layers to implement the unequal protection and second, we set lengths boundaries for all groups in order to speed up the grouping process.

A. MINIMAL COST PROBLEM FORMULIZATION

The quality layers grouping authentication is applied to each AU individually. Suppose the MSG data for an AU has total N bytes to transfer, i.e., $\{d_1, \ldots, d_i, \ldots, d_N\}$. The purpose of the grouping authentication is to divide the N bytes into I groups, i.e., $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_I\}$, where each group $\boldsymbol{x}_i$ is formed by the consecutive bytes. Hash functions are applied to groups of set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots, \boldsymbol{x}_I\}$ and each group can be seen as a basic authentication unit. We denote the authentication communication overhead for each group by S bits. Thus an AU needs total $I \times S$ bits for hashes overhead.

Let $d(f)$ represent the time duration required for an AU transfer. The value $d(f)$ is related to the frame rate of SVC streams. Denote the bandwidth cost for a single byte by b, and then the following relation holds:

$$b = size(t)/d(f) \tag{1}$$

where $size(t)$ represents the size of a byte. Suppose total i MGS bytes need to be conveyed and bandwidth consumption for the quality layers data is denoted by b_i, then we have

$$b_i = i \times b \tag{2}$$

For each MGS group, denote the bandwidth allocated for overhead is s, then

$$s = size(S)/d(f) \tag{3}$$

It can be seen that the cost brought by the grouping authentication consists of two parts, i.e., the bandwidth waste used for authentication overhead and the scalability reduction by the grouping authentication. The latter occurs because under the channel bandwidth constraint, the discarding of certain MGS packets would result in the authentication failure for other received MGS packets, if the discarded MGS packets and the received packets are within the same group. For example, if the recipient maintains the channel bandwidth range of $[b_i, b_j]$ and the needed bandwidth for an authenticated AU is b_{i-1}, then bandwidth cost for this recipient would be $b_i, b_{i+1}, \ldots, b_j$. Thus we consider the channel bandwidth distribution of all recipients and try to find the optimal overhead method. The cost function for the grouping authentication is defined as follows:

$$cost = s \times I + \sum_{i=1}^{I} \sum_{l=1}^{|\boldsymbol{x}_i|} l \times b \times \int_{B_{i-1}+bl}^{B_{i-1}+b(l+1)} p(x)dx + \sum_{i=1}^{I} (1 - B_{i-1}/N) \times b \times |\boldsymbol{x}_i| \tag{4}$$

where $p(x)$ is the *p.d.f* of the channel bandwidth, $|\boldsymbol{x}_i|$ is the number of bytes of ith group, B_{i-1} is the required accumulated bandwidth for the front $(i-1)$ groups, i.e.,

$$B_{i-1} = b \times \sum_{t=1}^{i-1} |\boldsymbol{x}_t| \tag{5}$$

The third part in the cost function represents the group length constraints for different quality layers. If the group is made up of lower MGS data, the constraint would be larger and thus the group length is meant to be shorter. On the contrary, if higher MGS data forms the group, the constraint is much looser, which makes the group length larger. Note that considering the ease of computation complexity and the performance of evaluation results, we choose the linear decay function to impose the unequal protection to MGS packets data.

The question then is to find an optimal group size I and an optimal group set $\{|\boldsymbol{x}_1|, \ldots, |\boldsymbol{x}_i|, \ldots |\boldsymbol{x}_I|\}$ to minimize the overhead cost. The formulization of optimal problem is as follows:

$$\begin{aligned} &min\ cost \\ &s.t. \begin{cases} \sum_{i=1}^{I} |\boldsymbol{x}_i| = N \\ 1 \le |\boldsymbol{x}_i| \le N \end{cases} \end{aligned} \tag{6}$$

B. OPTIMIZATION PROBLEM SOLUTION

Intuitively, the aim of optimization of (6) is to ensure that after the grouping authentication, the allowable sub-streams should have the bit rates which fall into the recipients' channel bandwidth ranges with a probability as high as possible. This optimization reduces the channel bandwidth waste and improves the scalability of SVC streams. Note that the optimal parameters needed to be found are the group size and the group set. The group size is related to the channel remaining bandwidth, which is the result of the maximum channel bandwidth (B) minus the bandwidth needed for all layers of MGS packets (b_{AU}). For instance, if value B is less than value b_{AU}, some MGS packets of higher layers should be dropped. Let the discarded MGS packets size be denoted by D and the dropping of these data brings the cost c', which is defined as follows:

$$c' = \sum_{l=1}^{D} l \times b \times \int_{B_I+bl}^{B_I+b(l+1)} p(x)dx \tag{7}$$

Note that the discarded MGS packets need no authentication and therefore the overall cost is

$$cost' = cost + \sum_{l=1}^{D} l \times b \times \int_{B_I+bl}^{B_I+b(l+1)} p(x)dx \tag{8}$$

We take into consideration the ease of RTP encapsulation for streams' NAL packets and intend to avoid fragmentizing the NAL packets. Thus we only allow the size of an authentication group to be integer times of the size of NAL size (denoted by U). Let the ratio of group size to the size of NAL be denoted by r, which is defined as

$$r = |\boldsymbol{x}_i|/U \tag{9}$$

and r is chosen from an integer set $\{1, 2, \ldots, p\}$. The value p should not be too large because the loss of certain MGS packet of the group, whose size is p times of U, causes much reduction of verification rate. Since the number of this group's remaining MGS packets, which cannot be verified for the recipient, is relatively large. Nor value p should be too small, because too small p would result in too much authentication overhead. Since MSG packets are only allowed to be divided into small groups. The value p is decided empirically in our work to best the verification rate.

Given the parameter p, we obtain the group size's lower bound and upper bound, which are denoted by I_τ and I_υ, respectively. Thus the following relations hold:

$$I_\tau = \lfloor (N-D)/(Up) \rfloor \tag{10}$$
$$I_\upsilon = \lfloor (N-D)/U \rfloor \tag{11}$$

where operator $\lfloor * \rfloor$ means round down function. Then (6) can be rewritten as follows:

$$\begin{aligned} &min\ cost' \\ &s.t. \begin{cases} \sum_{i=1}^{I} |\boldsymbol{x}_i| + D = N \\ |\boldsymbol{x}_i|/U = r, \quad r \in \{1, 2, \ldots, p\}\ p \in Z^+ \\ I_\tau \le I \le I_\upsilon \\ 0 \le D \le N \end{cases} \end{aligned} \tag{12}$$

where the value D represents the number of dropped MGS data, which is calculated as follows:

$$D = \begin{cases} (b_{AU} - B - s \times I)/b & if\ b_{AU} > B + s \times I \\ 0 & else \end{cases} \tag{13}$$

An effective solution to (12) is that we first decide the group size and then we try to find the optimal group set. If the group size is fixed, how to choose the optimal group set turns to be a combination problem. An intuitive method to the combination problem is by means of enumeration. However, this kind method would consume too much computation and cause unbearable delay for the recipient. The problem of grouping N MGS data into I groups is akin to the problem of dividing an integer number n into k parts, where the sum of k parts should equal n. Moreover, the cost functions of groups are independent of each other. Therefore the enumeration method to solve groups division is unfeasible.

We notice that the cost function of (12) has the characteristic of iteration within, i.e., the computation of cost for the front I groups can be broken into two sections: the computation of cost for front $(I-1)$ groups and the cost brought by the Ith group. The following relation holds.

$$\begin{aligned} cost(I, N') = {} & cost(I-1, N' - |\boldsymbol{x}_i|) \\ & + min\{s + \sum_{l=1}^{|\boldsymbol{x}_i|} l \times b \times \int_{B_{I-1}+bl}^{B_{I-1}+b(l+1)} p(x)dx \\ & + (1 - B_{I-1}/N') \times b \times |\boldsymbol{x}_i|\} \end{aligned} \tag{14}$$

where the $cost(I, N')$ is the cost for the N' MGS data grouped into I groups and $N' = N - D$.

Thus we adopt the iteration method to solve (12). Denote the cost matrix by C, in which the element $C(I, X)$ represents the cost for MGS data with X size grouped into I groups. In order to simplify the calculation of size that each group should have, we also define group cost matrix F, whose element $F(I, P \times U)$ represents the cost for the Ith group which has size of $P \times U$ MGS data. Note that element $F(I, P \times U)$ corresponds with the minimal part of (14), where $|\boldsymbol{x}_i|$ equals $P \times U$.

The pseudo-code for function cost(i, x) is shown in Fig. 7, where $X(i)$ is the MGS data size of the ith group. The input x to this function is N'/U. When i or x is less than zero, or the group number is larger than data size, the current iteration is void. We return a *MaxValue* to make this run null. Note that the elements of the lower triangle of matrix C are all zero.

```
1. cost (i, x)
2. if (i<0 || x<0 || i>x)  i=1, x=1;     return MaxValue;
3. if (i==1 && x>0)    return 0;
4. if(x>1 && x>0) { for r=1,...,p
5.                      S(r)=cost(i-1, x-r)+ f(i, rU);
6.                  X(i) = U × arg min(S(1),...,S(r)); }
7. X(1) = N' − Σ_{t=2}^{i} X(t)
```

FIGURE 7. The pseudo-code for cost function.

We claim that the iteration method can find an optimal solution to (12). Since when we limit the group size into the range $[I_\tau\ I_\upsilon]$, there must exist at least one legal group set. The code in Fig. 7 sequentially obtains the group set as $\{X(i), \ldots, X(1)\}$ for each possible grouping policy. Then it compares the costs of all candidate grouping policies and chooses the one with the least cost. If the number of the grouping candidates is equal or more than two, we choose the one with the largest group size.

The memory usage of the minimal optimization mainly includes two matrixes, i.e., the cost matrix C and the group cost matrix F. The former matrix needs storage of $O(I_\upsilon \times \lfloor N'/U \rfloor)$ and the latter needs storage of $O(I_\upsilon \times p)$. Thus the spatial complexity of the algorithm is $O(I_\upsilon \times \lfloor N'/U \rfloor) + I_\upsilon \times p)$. The algorithm needs to loop $(I_\upsilon - I_\tau + 1)$ times to calculate those two matrixes. Denote the number of comparisons for each AU during the optimization by T, where the comparisons are used to calculate the lower triangle of cost matrix. T is analyzed as follows:

$$
\begin{aligned}
T &= \sum_{I=I_\tau}^{I_\upsilon} (\lfloor N'/U \rfloor \times I - \frac{(I+1)(I-2)}{2} + I \times p) \\
&= \sum_{I=I_\tau}^{I_\upsilon} ((\lfloor N'/U \rfloor + p + \frac{1}{2}) \times I - \frac{1}{2} I^2 + 1) \\
&= (\lfloor N'/U \rfloor + p + \frac{1}{2}) \times \frac{(I_\upsilon + I_\tau)(I_\upsilon - I_\tau + 1)}{2} \\
&\quad - \frac{I_\upsilon(I_\upsilon + 1)(2I_\upsilon + 1) - (I_\tau - 1)I_\tau(2I_\tau - 1)}{12} \\
&\quad + (I_\upsilon - I_\tau + 1) \qquad (15)
\end{aligned}
$$

Note that the maximum number of groups $\lfloor N'/U \rfloor$ equals the upper bound I_υ. Thus the temporal complexity of the algorithm is $O(I_\upsilon^3)$, with its constant factor being $(1/3 - 1/(2p^2) + 1/(6p^3))$.

V. SECURITY ANALYSIS OF THE AUTHENTICATION SCHEME

When video streams are transferred over the Internet, they are prone to various attacks, e.g., frame alteration, frame deletion, frame removal and frame insertion by malicious attackers. The attackers deliberately destroy the integrity of the conveyed video, which results in recipients' distorted perceptual understanding of the video contents. Although there exist network packet dropping for normal circumstances, the authentication information should still be usable for the recipient verification of the sub-streams. The security of the authentication scheme should ensure that the all sub-streams are able to be verified for all recipients.

We state that the proposed SVC authentication streams are safe. Recall that we adopt the 'bottom-up' way to keep the authentication information of higher layers into the packets of lower layers, and the authentication of lower layers are kept into the base layer packet. The base layers, either the spatial or the temporal layer, are the necessary parts for any sub-streams, which provide the verification basis for higher layers. The hashes of base layers are signed by the sender, which are assumed to be successfully transferred to the recipients. Thus the recipients can form a verification process from the signature packets up to the quality layers.

From the perspective of the bundled GOPs, the successful verification of the signature is the necessary and sufficient condition for the successful verification of any sub-streams. As for the sufficient condition, if the recipient obtains the right signature, he/she gets the right hashes of the bundled GOPs and can use those hashes to verify the upcoming GOPs. On the contrary, if hash of certain GOP within the bundle is destroyed by attackers, the overall hashes of the bundle would not remain the same as the one in the signature, provided that the hash function and signature algorithm are safe. Thus the necessary condition holds.

From the perspective of the GOP, the successful verification of a GOP is the necessary and sufficient condition for the successful verification of frames within the GOP. As is seen from Fig. 5, the hash of a GOP is the hash value of the base AU data of that GOP. If the GOP is successful verified, then it says that that AU Data0 is safe. The base AU is mandatory for any sub-streams, which means any sub-streams have such verification starting point. With the relations of hash appendence, AUs of higher layers can be verified from lower layers. Thus the sufficient condition holds. As for the necessary condition, if certain AU of any sub-streams is not of integrity, then

its hash is no longer the same as the one stored within AU of lower layer. The recipient claims an error occurs if such mismatch is detected.

From the perspective of spatial or quality layers, the analysis of its security is similar to that of GOP. For instance, we observe from the Fig. 4 that the hashes of higher spatial layers are stored in the lower layers, which provide the verification basis for higher layers of any sub-streams. Note that as for the quality layers, the hashes of groups are concatenated and stored into the spatial base layer. Thus those hashes of groups within one AU are actually transferred to all recipients regardless of the sub-streams bit rates. In other words, even if certain sub-stream drops higher quality layer packets, the hashes of those layers are still available to the recipient of that sub-stream. This makes the base spatial layer become the verification basis of quality layers, which ensure that any sub-streams can verify the integrity of high quality layers.

From the above analysis we conclude that the authentication schemes are secure. The sub-streams are verifiable for the recipients in terms of bundle of GOPs, GOP, AUs, spatial and quality layers. The signature of the sender serves as the verification beginning for any sub-streams. Any attacks on the different logical units of sub-streams can be detected if and only if the signature is secure.

VI. PERFORMANCE EVALUATION

We implement the SVC streams authentication based on the proposed TS on DDG and apply the minimal overhead optimization on the quality layers. We compare our authentication method with other methods in terms of computation cost, delay, overhead and reconstructed video quality. The methods we adopt for comparison are FEC method [20] and ECC method [21] for short. They are chosen because these two authentication methods are quite related to our work and are also suitable for SVC streams. Note that the FEC method also applies the group authentication for quality layers but without unequal protections, while the ECC method does not provide the authentication scalability for the quality layer.

A. EXPERIMENTS SETUP

We build our simulation on an open source code JSVM (Joint Scalable Video Model) with version 9.19, which was developed by the H.264/SVC project of Joint Video Team (JVT). We select three raw videos from the JVT test video set, i.e., 'bus', 'city', and 'mobile'. The three videos are quite different in terms of contents changing degree and bit rates. Then we encode the raw videos by JSVM into SVC streams, where the frame rate is 15, the number of frames in a GOP is 8. The number of spatial layers is 2, which are of resolution of 352x288 and 176x144. The bit rates of 352x288 layer for the three video 'bus', 'city', and 'mobile', are 1.12Mbps, 2.03Mbps and 3.04Mbps, respectively, while the bit rates of 176x144 layer are 0.27Mbps, 0.61Mbps and 0.80Mbps. The number of quality layers is configured to be 4.

When authenticating the logical units of the SVC streams, we adopt the SHA-1 algorithm to obtain their digests, which yields a 20 bytes length hash value. We choose the RSA algorithm to sign hash of the bundled GOPs, which results in a 128 bytes length value. We use the Java programming language to implement the proposed authentication scheme. The hashing and signing algorithms are realized by functions from Java library. As for parsing SVC streams into different logical units, we adopt an open source Java NAL Parser and the svcAuth library [20]. The parsing process finds the boundaries between NAL units, GOP, AU, spatial and quality layers, which eases our authentication code programming. The signature of GOPs is encapsulated in NAL packets and it is embedded in the SVC streams. The type of the signature NAL packets is 6. Considering that the maximum of MTU is 1600 bytes and the RTP head needs 40 bytes, we configure the size of NAL date unit to be 1200 bytes.

We simulate the network packet loss during the SVC video streaming. Like the hypothesis in FEC method, we assume that the NAL packets of the sub-streams follow the loss pattern of independent and identical distribution, which means that the network channels are of random errors characteristics. As for the minimal overhead optimization of quality layers, we assume that the recipients' channel bandwidth follow a multimodal Gaussian distribution, as is shown in Fig. 8. The main concentrations of the bandwidth distribution are supposed to be about 0.8Mbps, 2.5Mbps and 4Mbps, which corresponds with the three recipients' abilities for obtaining the different bit rates sub-streams.

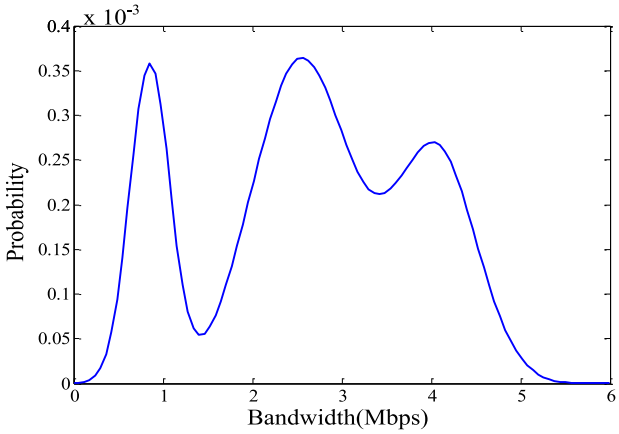

FIGURE 8. Bandwidth distribution for recipients' channel.

The overall SVC streams authentication simulation process is described in Fig. 9. Note that the authenticated SVC streams are used to generate three sub-streams by the proxy, which acts according to the recipients channel bandwidth. The recipients first verify the received packets to check whether they are authenticated.

If the hashes match, the packets are further used to reconstruct the frames. If not, they are discarded by the recipients. Besides the failed verification, another case for the discarding of packets is that the decoding dependent packets are not correctly received. We define the verification rate to quantitatively compare the effectiveness of different methods,

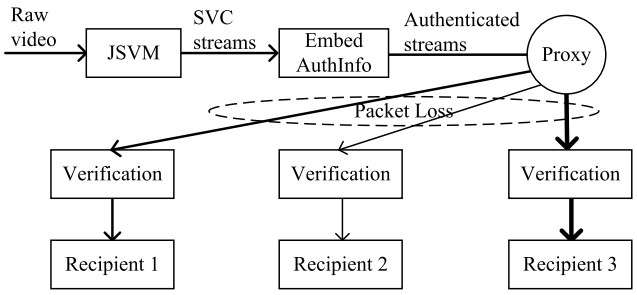

FIGURE 9. The overall simulation framework.

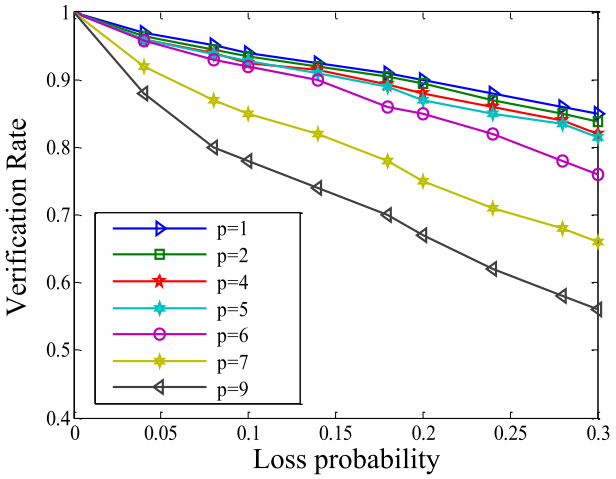

FIGURE 10. Verification rate for different loss probability under different *p*.

which is calculated as the ratio of correctly verified NAL data packets to the total received NAL data packets. The higher the verification rate is, the better the authentication method is.

As for the overhead optimization for quality layers, the parameter needing to decide is the maximum of group size p, which is used for value r's range in (9). We empirically choose this parameter by comparing the verification rates of the video 'city' under different values of p. The result is shown in Fig. 10, where value p goes from 1 to 9. Note that when p equals 1, there actually exists no grouping authentication for quality layers, whose result is seen as a benchmark for other cases when p is greater than 1. We observed that the verification rates for all values of p decrease when the probability increases. When p is greater than 7, the degradation of verification rate is too fierce. Therefore the group size maximum should not surpass 7. If the loss probability is less than 0.15, value p should better be set at 6; and if the probability is greater than or equals 0.15, it is more appropriate to set p at 5. These settings would make the loss of verification rate more bearable for the recipients. We also see these trends on the other two videos and thus we adopt these settings.

B. EXPERIMENTAL RESULTS

1) COMPUTATION COST

The computation cost includes two parts, the sender part and the recipient part. For the sender part, the computation cost is used to embed the authentication information into the streams, which consists of running the TS algorithm, grouping authentication for quality layers, calculating hashes and signature. As for the recipient part, he/she needs the computation to verify the signature and calculate the hashes of logical units of received streams. In order to ease the signature burden on the computation, we choose to sign a group of GOPs, which is the same as the work in FEC method. ECC method signs one GOP each time, where it is for decreasing the delay for the recipient. Also ECC method does not need grouping at the sender part for it provides no quality authentication scalability.

The comparisons of the computation cost are shown in Fig. 11, where (a) is the result at the sender part; (b) is the averaged result of three recipients. We separately draw the result of each video. The time shown in Fig. 11 is the average time for a GOP. The symbol TS is short for our authentication method. We observed that the time of TS required for both the sender and recipients is much less than that of the other two methods. FEC method consumes the most computation cost since it needs to obtain the channel codes of all spatial layers. We also observe that along with the number of GOPs growing, the average time for each GOP of TS and FEC is decreasing slowing. This is due to that signing the bundle of GOPs amortizes the signature computation cost across several GOPs. However, we note that this does not mean that we can choose arbitrary large number of GOPs. Because the larger the number is, the more time the sender needs to cache previous GOPs in order to retain their hashes and send the signed NAL packets before the bundled GOPs.

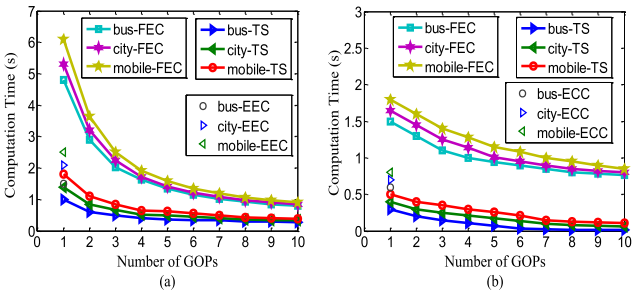

FIGURE 11. Computation cost for sender and recipient, (a) for sender and (b) for recipient.

2) DELAY

The delay caused by authentication can be seen both in the sender and the recipients. The sender delay is due to the GOPs caching, which is used to conduct the signing operations. The recipients need the delay to cache the received GOPs and to implement the verification. As for the sender, our method and FEC method should cache n GOPs before sending them, while ECC only needs cache one GOP. Although the delay results can be directly seen from Fig. 11, we here separately present the sender delay comparison in order to provide an intuitive explanation. The results of sender delay are shown

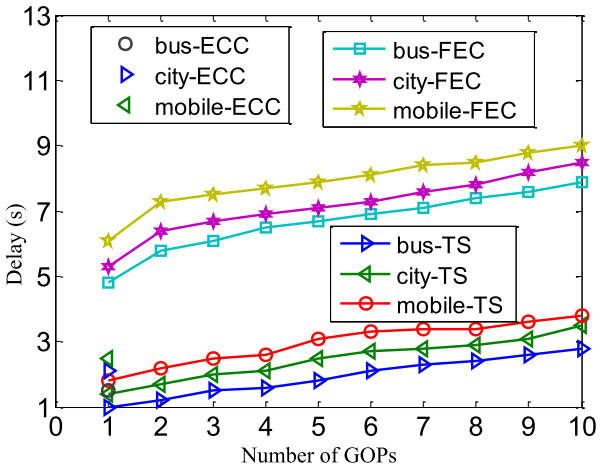

FIGURE 12. Delay for video sender.

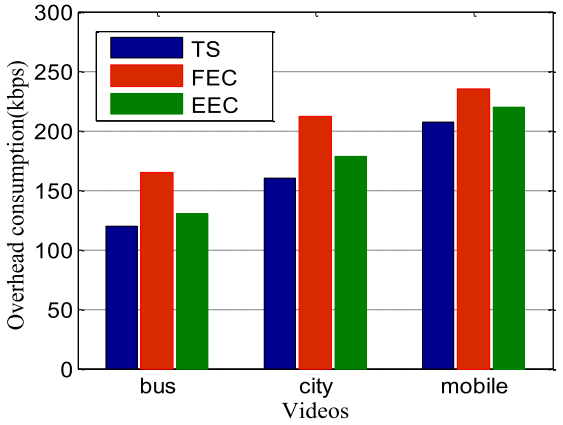

FIGURE 13. Overhead consumption comparison for three videos.

in Fig. 12, where the delay is averaged for all GOPs. For instance, when n equals 5, the delay of our method for the three videos ('bus', 'city', 'mobile') are 1.8s, 2.5s and 3.1s (s short for second), respectively. Meanwhile, delays of FEC method are 6.7s, 7.1s and 7.7s. The latter needs more time to calculate the channel coding and quality optimization. The ECC method needs delay of 1.5s, 2.1s and 2.5s for only one GOP. We conclude that at the sender side our method brings less delay than the other two methods.

As for an individual GOP, the FEC method needs to cache the whole temporal AUs of lower layers before any higher layers data can be used. This requirement is also mandatory for the ECC method. However, since our method takes advantage of decoding dependency between the logical units, we can use higher layers data instantly if and only if the authentication data of lower layers are correctly arrived. This saves us a lot of time when decoding the frames. We found that when n equals 5, for the video of 'mobile', the FEC method needs 1.15s to verify each GOP in average, which is larger than the time duration of each GOP replay (0.53s). Thus it needs about 6.45 Mega bytes for cache. Note that the ECC method requires an average recipient delay of 0.81s, which also results in the cache memory need. Our method requires no cache storage since the average delay for a GOP is only 0.41s, which is less than average GOP replay duration. We conclude that our method brings much less recipient delay time than others. Moreover it requires no additional cache memory for the recipients.

3) OVERHEAD

The overhead is caused by the hashes of logical units and the signature NAL packets, which consumes the bandwidth for all recipients. We compare the averaged overhead of the three videos in terms of bandwidth consumption. The result is depicted in Fig. 13. We observed that our method takes the least overhead consumption, while the FEC method requires the most. This is contributed to the FEC channel coding of all quality or temporal layers. The ECC method needs only one hash value for the quality layers of one AU, but its channel coding and signature on one GOP account for more overhead than ours.

4) PERFORMANCE AGAINST PACKET LOSS

We also compare the averaged verification rate and PSNR value for all videos under different packet loss probabilities. The results are drawn in Fig. 14, where (a) is for the verification rate and (b) is for the PSNR of averaged luminance values. The curves with 'NOAUTH' are the results when no authentication is applied onto the streams. It serves as the benchmark for the three methods. We see that all three methods achieve the same performances for the 'NOAUTH' one when no packet loss is introduced. As the packet loss probability slightly grows, the verification rates of all three methods decrease. The performances gap between our method and the other two enlarges when packet loss reaches about 0.15. The ECC performs the worst when packet loss gets to about 0.3. We conclude that our method incurs the least side effects to the verification rate than the other two.

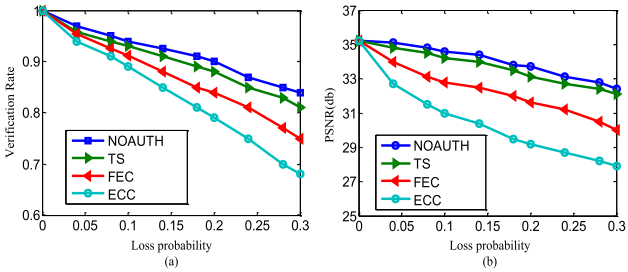

FIGURE 14. Overhead consumption comparison for three videos, (a) for verification rate and (b) for PSNR comparison.

We observed that the PSNR decreases for all methods when the loss probability increases. But our authentication method is at most 0.4 db less than the 'NOAUTH' one, which is quite negligible in terms of the reconstructed quality of video. However, the FEC and ECC methods bring much higher losses to the PSNR, with ranges from 0.7 db to 4.5 db. This is due to their lower verification rate. Because much larger of received packets are dumped as useless since they cannot be correctly verified.

VII. CONCLUSION

In this paper, we have proposed an authentication scheme for H.264/SVC streams based on the decoding dependency graph, which is inferred from the streams' decoding relationship. We obtained the hash appendance mode by the topological sort on the graph and applied the hash appendance mode to the spatial and temporal layers. As for the quality layers, we developed a grouping authentication strategy with unequal protections. We formulized the optimal minimizing cost problem and solved it by an iteration algorithm. We showed by simulation results that our authentication scheme incurs less computation cost and lower overhead as compared with other methods. Meanwhile, it requires negligible delay for the receivers and causes negligible side-effects for the real-time streaming. Its verification rates are much higher and thus it maintains much better video qualities.

REFERENCES

[1] A. A. Yavuz, "An efficient real-time broadcast authentication scheme for command and control messages," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1733–1742, Oct. 2014.

[2] J. L. T. Woo and M. V. Tripunitara, "Composing kerberos and multimedia Internet KEYing (MIKEY) for authenticated transport of group keys," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 898–907, Apr. 2014.

[3] L. Xing, Q. Ma, and L. Jiang, "Microblog user recommendation based on particle swarm optimization," *China Commun.*, vol. 14, no. 5, pp. 134–144, May 2017.

[4] X. F. Gong, X. L. Wang, and Q. H. Lin, "Generalized non-orthogonal joint diagonalization with LU decomposition and successive rotations," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1322–1334, Mar. 2015.

[5] B. Du and L. Zhang, "A discriminative metric learning based anomaly detection method," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6844–6857, Nov. 2014.

[6] N. Nasiraee, J. B. Mohasefi, and M. Nasiraee, "DSBS: A novel dependable secure broadcast stream over lossy channels," *Wireless Pers. Commun.*, vol. 78, pp. 599–613, Sep. 2014.

[7] M. Franklin, R. Gelles, R. Ostrovsky, and L. J. Schulman, "Optimal coding for streaming authentication and interactive communication," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 133–145, Jan. 2015.

[8] B. Du and L. Zhang, "Target detection based on a dynamic subspace," *Pattern Recognit.*, vol. 47, no. 1, pp. 344–358, Jan. 2014.

[9] M. Fischlin, F. Günther, and G. A. Marson, *Data Is a Stream: Security of Stream-Based Channels* (Lecture Notes in Computer Science), vol. 9216. Berlin, Germany: Springer-Verlag, 2015, pp. 545–564.

[10] B. Du, Y. Zhang, L. Zhang, and D. Tao, "Beyond the sparsity-based target detector: A hybrid sparsity and statistics-based detector for hyperspectral images," *IEEE Trans. Image Process.*, vol. 25 no. 11, pp. 5345–5357, Nov. 2016.

[11] M. Usman, M. A. Jan, and X. He, "Data sharing in secure multimedia wireless sensor networks," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Tianjin, China, Aug. 2016, pp. 590–597.

[12] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[13] R. Gennaro and P. Rohatgi, "How to sign digital streams," *Inf. Comput.*, vol. 165, no. 1, pp. 100–116, Feb. 2001.

[14] C. K. Wong and S. S. Lam, "Digital signatures for flows and multicasts," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 502–513, Aug. 1999.

[15] Y. S. Park, T.-S. Chung, and Y. Cho, "An efficient stream authentication scheme using tree chaining," *Inf. Process. Lett.*, vol. 86, pp. 1–8, Apr. 2003.

[16] Z. Zhang, Q. Sun, and W.-C. Wong, "A proposal of butterfly-graph based stream authentication over lossy networks," in *Proc. IEEE Conf. Multimedia Expo*, Amsterdam, The Netherlands, Jul. 2005, pp. 784–787.

[17] J. M. Park, E. K. P. Chong, and H. J. Siegel, "Efficient multicast stream authentication using erasure codes," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 2, pp. 258–285, May 2003.

[18] W. Wang, D. Peng, H. Wang, H. Sharif, and H.-H. Chen, "A multimedia quality-driven network resource management architecture for wireless sensor networks with stream authentication," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 439–447, Aug. 2010.

[19] X. Zhu and C. W. Chen, "A joint source-channel adaptive scheme for wireless H.264/AVC video authentication," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 141–153, Jan. 2016.

[20] K. Mokhtarian and M. Hefeeda, "Authentication of scalable video streams with low communication overhead," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 730–742, Nov. 2010.

[21] Y. Zhao, S.-W. Lo, R. H. Deng, and X. Ding, "Technique for authenticating H.264/SVC and its performance evaluation over wireless mobile networks," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 520–532, 2014.

[22] Z. Wei, Y. Wu, R. H. Deng, and X. Ding, "A hybrid scheme for authenticating scalable video codestreams," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 543–553, Apr. 2014.

[23] X. W. Yi, G. Zheng, and M. Li, "Efficient authentication of scalable media streams over wireless networks," *Multimedia Tools Appl.*, vol. 71, no. 3, pp. 1913–1935, Aug. 2014.

[24] P. K. Atrey, W.-Q. Yan, and M. S. Kankanhalli, "A scalable signature scheme for video authentication," *Multimedia Tools Appl.*, vol. 34, no. 1, pp. 107–135, Jul. 2007.

[25] Y. Tew, K. Wong, and R. C.-W. Phan, "Multi-layer authentication scheme for HEVC video based on embedded statistics," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 502–515, Oct. 2016.

[26] I. Amonou, N. Cammas, and S. Kervadec, "Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1186–1193, Sep. 2007.

QIANG MA was born in Mianyang, China, in 1982. He received the B.S. degree in electronic engineering from the Southwest University of Science and Technology, China, in 2005, and the M.S. degree in automation from the University of Science and Technology of China, China, in 2008. He is currently pursuing the Ph.D. degree in wireless physics with the Institute of Electronic Engineering, China Academy of Engineering Physics, China.

Since 2008, he has been a Lecturer with the School of Information Engineering, Southwest University of Science and Technology, China. His current research interests include image hashing, multimedia security, and multimedia streaming process.

LING XING received the B.S. degree in electronic engineering from the Southwest University of Science and Technology, China, in 2002, the M.S. degree in electronic engineering from the University of Science and Technology of China, in 2005, and the Ph.D. degree in communication and information system from the Beijing Institute of Technology in 2008. In 2007, she was a Visiting Scholar with the Illinois Institute of Technology, Chicago, IL, USA.

She is currently a Professor with the School of Information Engineering, Henan University of Science and Technology, China. Her research interests include multimedia semantic mining, multimedia security, and information intelligent management.

LONGSHUI ZHENG received the B.S. degree in electronic science and technology from Shangqiu University, China, in 2014. He is currently pursuing the M.S. degree in communication and information system with the Southwest University of Science and Technology, China.

His interests include multimedia trust analysis, social media dynamics, and multimedia streaming process.