ZIMMERMANN REAL-TIME TRANSPORT PROTOCOL FOR HIERARCHICAL KEYS GENERATION OF ENCRYPTED SCALABLE VIDEO STREAMING

M. N. Asghar, H. Majid, R. Kausar, I. S. Bajwa and M. Fleury*

Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, 63100, Punjab, Pakistan

* School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ - UK

Corresponding author Email: mamona.asghar@iub.edu.pk

ABSTRACT: Effective key management has been a fundamental prelude before encryption of single-layer video streams in the public Internet. In addition to single-layer video, multi-layer video can provide scalable on-demand video streaming, especially within a mobile network. The objective of the proposed mechanism was to reduce the number of keys exchanged for setting up an encrypted scalable video stream. The method proposed had access to the highest enhancement layer to which users were entitled, together with access to all layers below down to the base layer. A low-latency key exchange mechanism i.e. Zimmermann Real-time Transport Protocol (ZRTP) was used for the hierarchical keys' generation of scalable video in conjunction with an in-house form of selective encryption. The results confirmed that the keys' generation time was reduced by 7 µs than the previous techniques. It was also concluded from the results that the selective encryption method was decoder compatible and produced zero bit-rate overhead.

Keywords: Key management, Scalable Video, Selective Encryption.

(Received 10-12-2016 Accepted 09-03-2017)

INTRODUCTION

Video services can benefit from the ability to stream to multiple, heterogeneous devices according to the available bandwidth and constraints at the receiver. Different frame rates, display spatial resolutions, and fidelities can be selected from within a single scalable video stream if, when the video is first compressed, a scalable extension of a video codec is employed (Schwarz *et al*., 2007) . The most recent standardized codec, the High Efficiency Video Coding (HEVC) standard now has two scalable extensions (Helle *et al*., 2013a and Hong *et al.*, 2013b). However, video streaming of copyrighted digital content over networks is vulnerable to plagiarism attacks because of the ease of copying and modification of an unprotected video stream. In Digital Rights Management (DRM), cryptography remains the first line of defence (Rosenblatt *et al.*, 2003). As a result, there have been a number of initiatives to add cryptographic confidentiality to scalable video streams (Stütz and Uhl, 2012).

Selective encryption (SE) rather than full encryption brings benefits to commercial, scalable video streaming. When properly configured, SE can operate without any bit-rate overhead and can reduce encryption latency to a minimum (Asghar *et al.*, 2014a). As a result, interactive, two-way streaming applications, which require synchronization between the participants to avoid overlapped communication, benefit from the reduction in encryption delay (Blokowski and Steinmetz, 1996). For some purposes SE applied to complete video frames may be insufficient to prevent recognition of some participants. In which case, additional face recognition should be combined with region-of-interest (ROI) encryption (Unterwager and Uhl, 2014). Notice also that if SE provides video decoder compatibility (Asghar *et al.*, 2014b), a Media Aware Network Element (MANE) is able to extract a partial bit-stream from the scalable bit-stream without the need to decrypt the bit-stream (Deng *et al.*, 2014).

One important difference considered in this paper between single-layer video and the scalable video is the question of how to handle keying for multiple layers. There is a risk that scalable video key management can become too complicated. This paper presents a method in which a user has access to the highest enhancement layer, together with access to all layers below that down to the base layer. No access is granted to enhancement layers above the highest layer a user is granted access to. The layer entitlement could be linked to the capabilities of the target device that the user is employing but it could also be linked to the subscription level of the user. In doing so, more subscription levels are made available in comparison to the two levels made available in (Lian, 2008). That scheme, which used post-processing rather than codec-based layering, limited itself to basic and enhanced TV.

A number of researchers have previously investigated the key management of H.264/SVC.

However, they have not done so within the context of standardized protocols, as this paper achieves. The authors propose a key management scheme employing features embed in the video frames in order to generate keys (Park *et al.,* 2008a). The header of the first Group of Pictures (GoP) (Ghanbari, 2003) rather than the video content itself started the keying chain. Enhancement layer keys are formed (Park *et al.,* 2008b) by cryptographically hashing the base-layer key. That scheme allowed a different permanent key to be created for each user and considerably reduced the computation of keying material. Then the authors describe a method of restricting the storage of keys to a master key at the server and a layer key at a client (Fan *et al.,* 2010). The author in (Abu-Zahra *et al.,* 2010) proposes a method of creating concatenated keys of the same length whether containing the individual keys of: the base layer (BL) with enhancement layer (EL1); or the base layer with EL1 and EL2; and so on with the Advanced Encryption Standard (AES).

Apart from organizing the efficient creation of per-layer keys, there needs to be a means of supplying keying material to a media-security protocol. Of these protocols, Zimmermann Real-time Transport Protocol (ZRTP)'s (Zimmermann *et al.,* 2011) is used in the proposed method. (Asghar et al., 2012) examined the possibility of employing the Multimedia Internet KEYing (MIKEY) (Arkko *et al.,* 2004) for scalable video encryption key management but for the reasons explained in the Results and Discussion Section, ZRTP is preferred.

The authors (Kolesnikov and Gurbani, 2015) discouraged the use of standard key management protocol by arguing that the ZRTP had a complex key distribution mechanism and provided medium level security. So, it is expensive to implement and having vulnerabilities when associating with real-time protocol sessions. However, (Siqi *et al.,* 2015) proved it wrong after presenting the formal analysis of ZRTP protocol and suggested it an appropriate choice for Real-time sessions. The authors (Asghar *et al.,* 2016) have recommended the ZRTP protocol even for the key management issues of pay-TV providers due to its ease of computation.

MATERIALS AND METHODS

Key Management Protocol: The Zimmermann Real-time Transport protocol was intended to achieve key agreement prior to using a multimedia security protocol. The ZRTP performed all key negotiations for Real-Time Protocol (RTP) media streams, so it was independent of any signaling protocol. This property implied that ZRTP was independent of attacks on the signaling path. It employed a Short Authentication String (SAS), which was a cryptographic hash of two DH values, for users to compare with each other's key and, hence, detect any Man-in-the-Middle (MITM) attack.

ZRTP worked in three major key agreement modes: 1) DH mode, 2) Pre-shared mode and 3) Multi-stream mode. The DH mode was crucial to the security of ZRTP, as the other two modes rely on the initial establishment of a shared secret in the DH mode. The message exchange sequence of the DH mode is summarized in Figure-1. The mechanism of the DH mode proceeds through a set of phases. It began with a discovery phase. A ZRTP endpoint EP1 initiateed the exchange by sending a Hello message to another endpoint, EP2, in order to confirm the existence of the other endpoint and discovered the common encryption algorithm and other capabilities that they both support.

Figure-2 shows the proposed key generation mechanism in a proper sequence. By means of a key derivation function, both EP1 and EP2 generate a ZRTP session key and Secure Real-Time Transfer protocol (SRTP) (Baugher *et al.,* 2004) keying material in the manner shown in Figure-2. Notice that different SRTP keying material was required for each of two RTP streams, one for each direction. The standard SRTP keying material is derived from the shared secret, s0, along with other hashed values. The salts were a remedy against a pre-computation attack when computing SRTP session keys. A message authentication code (mac) was a means of ensuring message integrity. Only the top-level keys were generated directly from s0, after which per-layer keys for encryption or decryption must be separately generated. A summary of the properties of the keys involved i.e the key length, number of keys generated per user has been given in Table-1.

Apart from ZRTP session key and SRTP keying material generation, an SAS was created by both parties. An SAS was a way of guarding against an MITM attack. An SAS was calculated as a hash of the ZRTP messages (responder‟s Hello, Commit, DHpart1, and DHpart2). The commit phase then took place. Confirm1, Confirm2 and Confirm2 ACK messages were exchanged between the end points. These messages were exchanged in response to the successful completion of the key negotiation process. For the termination phase of multimedia encryption, a GoClear message was used. The message did not terminate the session but changed the state of an RTP stream from being encrypted to unencrypted.

The generation of per-layer keys for SVC was performed in hierarchical fashion such that the encryption key of an upper layer could be employed in the generation of the keys of the lower layers.

For the highest enhancement layer EL_n to be made available to a receiver, the ZRTP shared secret s0 was employed to generate its session key, $ekey_n$. $ekey_n$ can subsequently generated the encryption key $ekey_{n-1}$ of lower enhancement layer EL_{n-1}. This process continued in recursive fashion, as in (1), (2) and (3):

$ekey_n$ was generated from *KDF(s0, KDF_Context, ZRTP_Sessionkey, length of key)* (1)
$ekey_{n-1}$ was generated from *KDF($ekey_n$, KDF_Context, ZRTP_Sessionkey, length of key)* (2)
$ekey_{n-2}$ was generated from *KDF($ekey_{n-1}$, KDF_Context, ZRTP_Sessionkey, length of key),* (3)

where KDF was the ZRTP Key Derivation Function (KDF). The ZRTP KDF was a cryptographic hash function that creates a key of length 'length of key'. In the proposed method, the per-layer keys were of the same length as s0, that is of length 128 bits. The KDF_Context was a concatenation of the ZIDs of the initiator and responder, along with *mh*. *mh* is a cryptographic hash of all the messages exchanged in the DH key establishment process.

Encryption took place based on a set of keys, as derived in the process defined in equations (1), (2), and (3). Only the additional frames added to an enhancement layer are encrypted. Thus frames were not re-encrypted. For a three-layer encoding, these keys would be applied as described in 'expressions' (4), (5) and (6).

$ekey_2$ was applied to the encryption of *(Frames of EL_2 less the Frames of EL_1)* (4)

$ekey_1$ was applied to the encryption of *(Frames of EL_1 less the Frames of BL)* (5)

$ekey_0$ was applied to the encryption of the *Frames of BL.* (6)

Thus, the general expression for the application of keys to encryption during temporal scalability was:

$ekey_n$ was applied to the encryption *of (Frames of EL_n less the Frames of EL_{n-1})* (7)

To decrypt, $ekey_n$ was also the decryption key in the symmetric encryption employed herein. Therefore, upon forming ekeyn from s0 according to (1), the decryption process could generate the per-layer decryption keys for the lower layers down to the base layer according to the process defined by (2) and (3).

Selective Encryption: The form of selective encryption (SE) presented by the authors (Asghar *et al.,* 2014) was confined to the entropy coding stage of H.264/AVC. In that way, the SE avoided any side-effects in the statistical properties of the compressed bit-stream because entropy coding was the final stage before output. The method of (Asghar *et al.,* 2014) selected either syntax elements of the Context Adaptive Variable Length Coding (CAVLC) or of the Context Adaptive Binary Arithmetic Coding (CABAC) methods of entropy coding supported by H.264. The proposed selective encryption scheme (Asghar *et al.,* 2014) for H.264 scalable layers employed the Cipher, Advanced Encryption Standard (AES) (Federal, 2001) in a Cipher Feedback Mode (AES-CFB). The AES Cipher was based on a modified substitution permutation network. The ZRTP key length for the layer wise encryption (i.e. encryption key (Ekey-n) and decryption key (Dkey-n) by referring Table-1) was the same as AES key length, which were 2^{128} possible keys with a key size of 128 bits. The syntax elements chosen for encryption were elements such as signs of transform coefficient levels that were assumed to be randomly distributed in value.

All the results were taken with H.264/SVC reference software JSVM 9.19.10 version encoder. The experiments were performed on an Intel Core i3 processor with 4 GB RAM. The SE method for H.264/SVC streams was tested at the onset of evaluation by using a set of reference video sequences. For implementation of scheme, the YUV video sequences were downloaded from the URL (ftp.tnt.uni-hannover.de/pub/svc/testsequences/). A total of 300 frames of the News sequence, 260 frames of the Football sequence and 300 frames of the Mobile sequence were chosen for SE. For ease of testing, the sequence was configured in Common Intermediate Format (CIF) (352 × 288 pixels/frame) @ 25 Hz, with standard 4:2:0 chroma sampling and a variable bit-rate. The frame format was IBBP…., that is a periodic intra-coded frame every 15 frames, with intermediate bi-predicted B-frames and one-way predicted P-frames. A BL (layer 0) and four ELs (layers 1–3) were employed.

RESULTS AND DISCUSSIONS

The results of applying SE to all the layers of the video sequences was shown in Figure 3. For effective evaluation the objective metrics Peak Signal Noise Ratio (PSNR) and Structural SIMilarity (SSIM) were calculated. PSNR is included for comparison with the work of others. SSIM index is a well-known method of measuring the perceived quality of an image. Original frames form Figures-3a and 3b. From the evidence of Figures-3c and 3d, these would be unwatchable in the distorted form resulting from SE and subsequent decoding, without decryption. SE was also applied separately to each layer of the Football sequence, configured as previously. Figure-4 illustrates the effect of applying SE on a selected frame, shown in Figure-4a. In this frame for speed of encoding CAVLC was used in the base layer and CABAC was used in the three enhancement layers. Figure-4b shows the impact of SE only on the base layer. The key for a base-layer encryption was provided by ZRTP. Further keys were applied with the increasing number of layers. The result videoing at the end of the multi-layered encryption was more resistant to estimation of the contents of the video by an attacker. Figure-4c represented the proposed scheme on a base layer and an enhancement layer. Further, Figure-4d showed the impact of SE when utilizing a base layer and two enhancement layers. Table-2 was a comparison of the mean PSNR of the Football sequence according to the number of layers employed. It is apparent from Table-2 that the PSNR changes as each

new layer was added, whether decoding takes place without SE or after application of SE upon the video sequence. (Notice that the dBs are rounded to one decimal place. As a result in the final column of Table-2, the dB value of the V component did not appear to change with additional ELs, even though there was indeed a small change as more ELs were added.)

Security Analysis: This section included comments on some security aspects of the scheme, though it is not intended to be comprehensive. There was a cryptanalysis of the SE scheme presented and employed with ZRTP key exchange. For example, in terms of resistance to a brute-force attack, the proposed scheme was secured enough to deal with the attack because of the following reasons:

- The encryption pattern will change if only one bit of the key is altered. The encryption key was XORed with the video data in every AES encryption round. The encryption key was provided by ZRTP from s0. After 12 hours the data will be encrypted with another key.
- In the proposed scheme, AES was used with a 128-bit key. The strength of this key was based upon the following parameters: the number of alternative keys was 2^{128} which was 3.4×10^{38}; the time required for 1 decryption/μs is 2^{127} μs which is 5.4×10^{24} years and the time required for 10^6 decryptions/μs is 5.4×10^{18} years. These time durations were so large that the video data were safe against brute force attacks.

The key-exchange method using ZRTP key agreement protocol provided security against Man in the middle (MITM) attack. As previously mentioned in Section 2, ZRTP employed an SAS instead of a PKI for authentication purpose. The confirmation of SAS was performed in one of two ways: 1) verbally agreeing upon SAS by means of a phone call or some other method of personal contact; or 2) automatic agreement upon SAS based on an optional digital signature contained in either the Confirm1 or Confirm2 messages. The SAS was16-bits in length, hence, an attacker had only one chance in 2^{16} = 65,536 of guessing the SAS, assuming that the SAS numbers were randomly distributed, which was a property of a cryptographic hash. If SAS matches between the two parties, it was safe to assume that no MITM had taken place.

Suppose also that an attacker has some knowledge of the encryption keys and tried to guess their exact values in order to use them for decryption. Figure-5 demonstrated the effect when SE was applied but slightly altered keys were used in decryption. Evidently the resulting decoded video would still be unwatchable.

Computational Overhead: The time consumed in generating keys was considered to be the computational overhead of ZRTP. The generation time of all the keys related to the user is dependent on the number of scalable layers used. For authentication purposes the shared secret s0 is newly generated for each session. After generation of s0, the srtpkey, srtpsalt, mac, zrtpkey and ekeys are generated (Figure-2).

Figure-6 showed times taken to generate the keys according to the number of layers employed. For example, to create keys for a BL and one EL takes 42 μs, while in (Asghar *et al.,* 2012) MiKEY scheme took 49 μs, so there was a saving of 7 μs in video comprises of two layers. Increasing the number of layers to ten results in a keying overhead of 102 μs with an average saving of 4 μs per layer. Table 3 showed the comparative time difference between proposed ZRTP and MiKEY protocol. Thus, the keying overhead had a negligible impact upon the overall encryption and decryption time as compared to previous techniques (Park *et al.,* 2008a and Fan *et al.,* 2010), which was shown on a per-layer basis in Figure-7.

Li *et al.,* 2009 and Park *et al.,* 2008a provided complete security management system for multiple layers data with complex key management schemes, without any reference to standard key management protocols. In all the compared studies there were more than one key generated on layer basis; hence the problem of overhead for managing multiple keys for each layer was not solved. Only the authors in study (Asghar *et al.,* 2012) proposed a scalable keys management scheme with MiKEY Protocol. But the results discussed above in Figure-6, 7 and Table 3 showed that the proposed security system in this paper had a remarkable achievement than the previous approaches. A MiKEY (Asghar et al., 2012) implementation was more complex than ZRTP because of MiKEY's need to employ another signalling protocol. Though the various key exchange modes allowed MiKEY to address a variety of scenarios, including group communication such as in multicast, they potentially added to its complexity, which itself could be a security weakness.

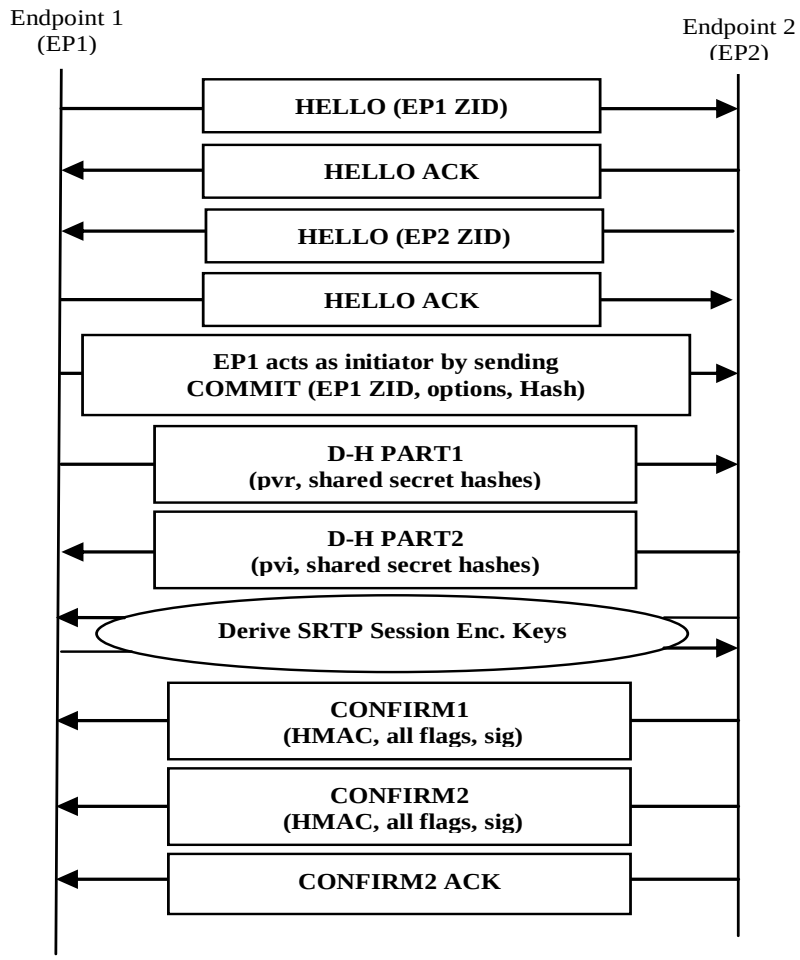

Figure 1. Message exchange sequence of ZRTP's DH mode

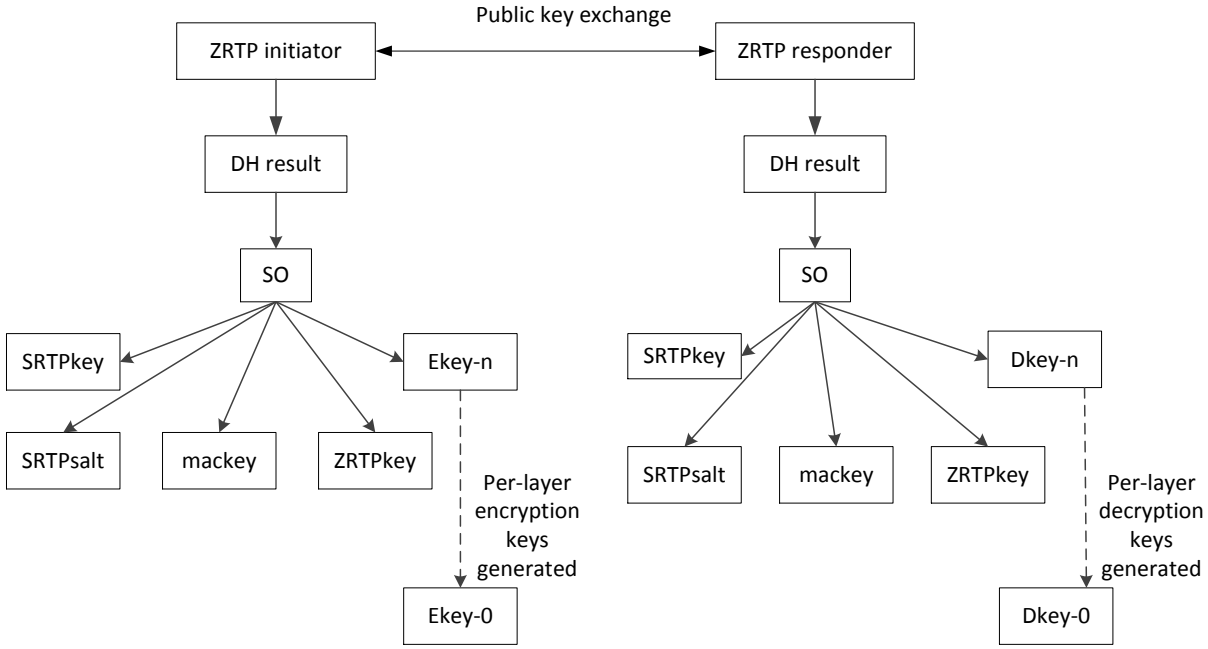

Figure 2. Proposed Key derivation Flow-Chart for hierarchical keys of scalable video

Table 1. Summary of ZRTP keys for SRTP and scalable video streams

ZRTP keys	Key length (bits)	Method of generation	Key life
s0 (shared secret)	128	Diffie- Hellman	For session
SRTPkey	128	KDF(s0)	12 hours
SRTPsalt	112	KDF(s0)	12 hours
mackey	128	KDF(s0)	Unique for every user
ZRTPkey	128	KDF(s0)	Unique for every user
Ekey-n	128	Custom KDF	For session
Dkey-n	128	Custom KDF	For session

(a) (b) (c) (d)

Figure 3. Effect of SE on frames within reference video sequences: (a) Mobile frame no. 149 (b) Football frame no. 138 (c) Effect of SE: PSNR (Y=11.1 U=13.7 V=13.8) dB SSIM= 0.0872 (d) Effect of SE PSNR (Y= 11.1 U=15.8 V=21.2) dB SSIM = 0.2394

(a) (b) (c) (d)

Figure 4. Effect of SE on a layer by layer basis: (a) Football frame no. 56 (b) SE of BL (c) SE of BL, EL1 (d) SE of BL, EL 1-3

Table 2. Average video distortion (PSNR) (dB) of the Football sequence with and without SE, for luma (Y) and chrominance (U and V) components.

	PSNR Y (dB)	**SE PSNR Y (dB)**	**PSNR U (dB)**	**SE PSNR U (dB)**	**PSNR V (dB)**	**SE PSNR V (dB)**
Base layer 0	32.5	9.8	38.4	13.9	40.4	21.2
Layer 1	33.5	10.2	40.7	14.4	42.2	21.6
Layer 2	34.6	10.3	40.9	14.7	42.8	21.6
Layer 3	35.1	10.4	41.1	14.8	42.1	21.6

(a) (b) (c)

Figure 5. Effect of altering the encryption key (ekey) upon decryption and decoding and the resulting video distortion (PSNR (dB)) (a) Foreman frame no. 56: PSNR (Y= 36.1, U= 41.9, V= 43.1) dB (b) ekey changed by 1 nibble: PSNR (Y= 10.3, U= 23.1, V= 20.6) dB (c) ekey changed by 1 byte: PSNR (Y=13.5, U= 20.8, V=17.6) dB

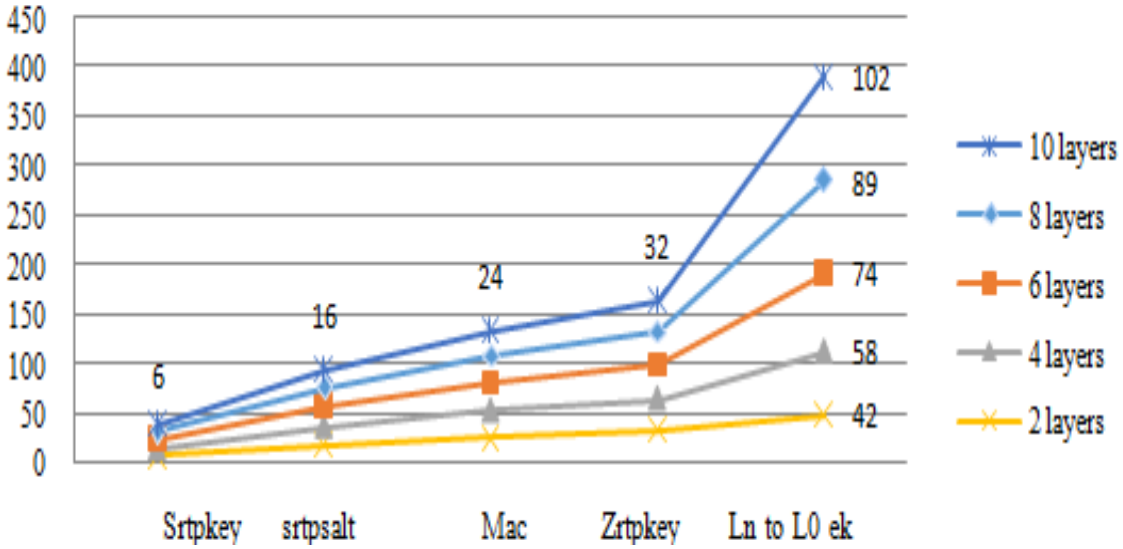

Figure 6. Key generation computation times (X-axis shown Keys and Y-axis shows time in µs)

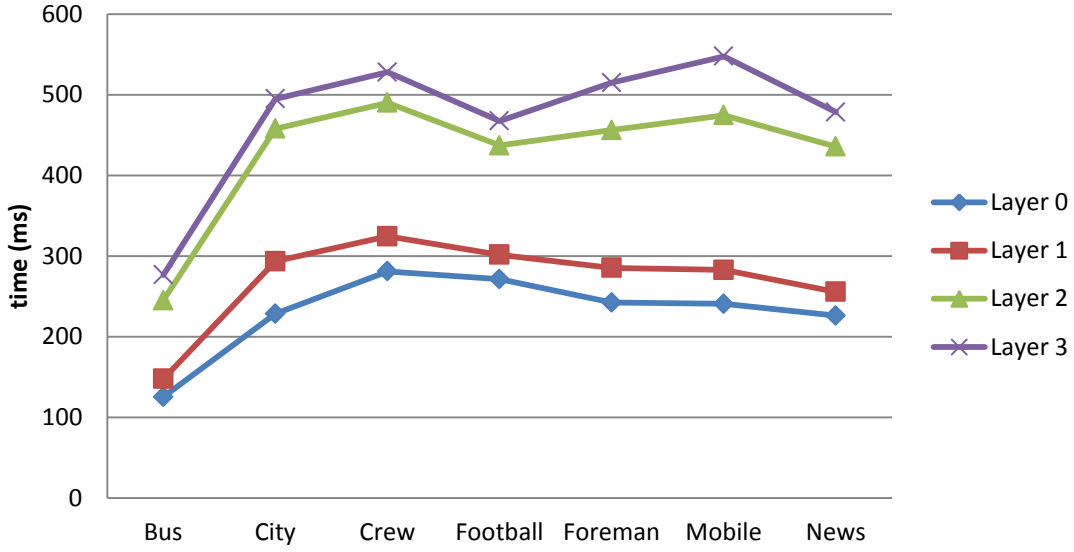

Figure 7. Layer encryption and encoding times for various reference video sequences (times in ms)

Table 3. Layer wise ZRTP vs. MIKEY Keys generation overhead (in µs)

No. of Layers	MiKEY	ZRTP
2 layers	49	42
4 layers	64	58
6 layers	79	74
8 layers	93	89
10 layers	Not given	102

Conclusion: This paper pointed out the issues of keys management for scalable video data which has been divided into multiple layers. A unique method for deriving the single encryption key for each layer was presented in the methodology section with standard key management protocol. A principal contribution of this paper has been to show how ZRTP could be extended to scalable video, specifically to H.264/SVC, though there was no basic obstacle to its use for scalable H.265/HEVC. If selectively-encrypted video streams were employed, as herein, then further factors need to be considered. It was not as easy to arrive at a method that was decoder compatible and also results in no bit-rate overhead. To make further progress based around ZRTP, the research challenge was to augment ZRTP with a key-exchange mechanism suitable for group video distribution, as currently ZRTP could be employed to set up point-to-multipoint sessions but was not convenient if multiple session leavers and joiners occur.

REFERENCES

Abu-Zahra, A., Z. Shahid, A. Rattrout, W. Puech (2010). Independent protection of different layers in spatially scalable video coding, Procedia Computer Sci., 10, 240-246.

Adams, C., and S. Lloyd (2002). Understanding PKI: Concepts, standards and deployment considerations, Boston, MA: Addison-Wesley. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©2002, ISBN:0672323915.

Arkko, J., E. Carrara, L. Lindholm, M. Naslund, and K. Norrman (2004). MIKEY: Multimedia Internet KEYing, IETF, RFC 3830.

Asghar M. N., M. Fleury, S. Makki, (2016). Interoperable conditional access with video selective encryption for portable devices, Multimedia Tools and Applications. DOI 10.1007/s11042-016-3725-3, 1-14.

Asghar, M. N., and M. Ghanbari (2012). MIKEY for keys management of H.264 scalable video coded layers, J. of King Saud Univ. - Computer Info. Sci., 24(2), 107-116.

Asghar, M. N., M. Ghanbari, M. Fleury, and M. Reed (2014a). Sufficient encryption based on entropy coding syntax elements of H.264/SVC, Multimedia Tools and Applications, 73(23), 10215–10241.

Asghar, M. N., M. Ghanbari, M. Fleury, and M. Reed (2014b). Confidentiality of a selectively encrypted H.264 coded video bit-stream, J. of Visual Commun. and Image Representation, 25(2), 487-498.

Baugher, M., D. McGrew, M. Naslund, E. Carrera, and K. Norrman (2004). The Secure Real-time Transport Protocol (SRTP), IETF, RFC 3711.

Blokowski, G., and R. Steinmetz (1996). A media synchronization survey: Reference model, specification, and case studies, IEEE J. Sel. Areas in Commun., 14(1), 3-35.

Blom, R., E. Carrara, F. Lindholm, K. Norrman, and M. Näslund (2005). Key management and protection for IP multimedia, in Multimedia Security Handbook, eds. by B. Furht, and D. Kirovski, CRC Press, 169-196.

Deng, R. H., X. Ding, Y. Wu, and Z. Wei (2014). Efficient block-based transparent encryption of H.264/SVC bitstreams, Multimedia Systems, 20(2), 165-178.

Fan, M., C. Yuan, and Y. Zhang (2010). An efficient hierarchical key management of H.264/Scalable Video Coding, in Proc. of IEEE Int. Conf. on Intelligent Computing and Intelligent Syst., 757-760.

Federal (2001) Information processing standards publication 197, Advanced Encryption Standard (AES), available at http://csrc.nist.gov/ publicitions/fips/fips197/fips-197.pdf

Ghanbari, M., (2003). Standard codecs: Image compression to advanced video coding, Stevenage, UK: IET Press.

Helle, P., H. Lakshman, M. Siekmann, J. Stegemann (2013). A scalable video coding extension of HEVC, in Proc of Data Compression Conf., 201-210.

Hlavacs, H., W. Gansterer, H. Schabauer, J. Zottl, M. Petrashek, T. Hoeher, and O. Jung (2009). Enhancing ZRTP by using computational puzzles, Int. J. of Universal Comput. Sci., 14(55), 693-716.

Hong, D., J. Wonkap, J. Boyce (2012). Scalability support in HEVC, in Proc. of IEEE Int. Symp. on Circuits Syst., 890-893.

Kolesnikov, V. and V. Gurbani (2015). Efficient key management system and method, Patent no. US9106628 B2.

Lian, S. (2008). Digital rights management for the home TV based on scalable video coding," IEEE Trans. Consumer Electron., 54(3), 1287-1293.

Li, C., X. Zhou, and Y. Zong, (Oct. 2009) "Layered encryption for scalable video coding," in Proc.

IEEE Conf. Image Signal Process., 1–4.

Park, S. W., D. Yi, and S. U. Shin (2008a). A practical key management scheme using the frame based features for layered access control of H.264/SVC, in Proc. of Int. Symp. Info. Theory and its Applications. DOI: 10.1109/ISITA. 2008. 4895406.

Park, S. W., and S.-U. Shin (2008b). Efficient selective encryption scheme for the H.264/Scalable Video Coding (SVC), in Proc. of IEEE Int. Conf. on Networked Computing and Advanced Info. Management, 371-376.

Rosenblatt, W., W. Trippe, and S. Mooney (2003). Digital rights management: Business and technology, Foster City, CA: M and T Books.

Schwarz, H., D. Marpe, and T. Wiegand (2007). Overview of the scalable video coding extension of the H. 264/AVC standard, IEEE Trans. Circuits Syst. Video Technol., 17(9), 1103-1120.

Siqi. L., W. Wenbo, C. Qingfeng (2015). Formal Analysis of the Real-time Multimedia Network Protocol ZRTP, Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering), 8(1), 12-17.

Stütz, T., and A. Uhl. (2012). A survey of H.264 AVC/SVC encryption, IEEE Trans. Circuits Syst. Video Technol., 22(3), 325-339.

Unterwager, A., and A. Uhl (2014). Slice groups for post-compression region of interest encryption in H.264/AVC and its scalable extension, Signal Processing: Image Commun., 29(10), 1158-1170.

Zimmermann, P., A. Johnston, E. Avaya, and J. Callas (2011). ZRTP: Media path key agreement for unicast secure RTP, IETF, RFC 6189.