1-1-2014

Performance evaluation of a technology independent security gateway for Next Generation Networks

Fudong Li
Plymouth University

Nathan Clarke
Edith Cowan University

Steven Furnell
Edith Cowan University

Is-Mkwawa Mkwawa
Plymouth University

10.1109/WiMOB.2014.6962183

Performance Evaluation of A Technology Independent Security Gateway for Next Generation Networks

Fudong Li*, Nathan Clarke*†, Steven Furnell*† and Is-Haka Mkwawa*
*Centre for Security, Communication and Network Research, Plymouth University, United Kingdom
†Security Research Institute, Edith Cowan University, Western Australia
Email: info@cscan.org

***Abstract*—With the all IP based Next Generation Networks being deployed around the world, the use of real-time multimedia service applications is being extended from normal daily communications to emergency situations. However, currently different emergency providers utilise differing networks and different technologies. As such, conversations could be terminated at the setup phase or data could be transmitted in plaintext should incompatibility issues exit between terminals. To this end, a novel security gateway that can provide the necessary security support for incompatible terminals was proposed, developed and implemented to ensure the successful establishment of secure real-time multimedia conversations. A series of experiments were conducted to evaluate the security gateway through the use 40 Boghe softphone acting as the terminals. The experimental results demonstrate that the best performance of the prototype was achieved by utilising a multithreading and multi-buffering technique, with an average of 582 microseconds processing overhead. Based upon the ITU-Ts 150 milliseconds one way delay recommendation for voice communications, it is envisaged that such a marginal overhead will not be noticed by users in practice.**

I. INTRODUCTION

Real-time multimedia services, such as voice over IP and image transmitting, have become indispensable applications for the Next Generation Network (NGN), providing an economical and important all IP based communication channel for the modern society [1]. Real-time multimedia services can be utilised not only for daily communications (e.g. calling friends and family, video conferencing with business partners) but also in emergency situations (e.g. transmitting images of a car accident to first responders) [2-4]; therefore, private and sensitive information would be inevitably be transmitted in various conversations. Furthermore, real-time multimedia services can be established from terminals within the NGN and also across domains with other communication networks (e.g. Private Mobile Radio (PMR) and Public Switched Telephone Network (PSTN)) [5, 6]; consequently, certain incompatibility issues could be developed when users from different networks try to communicate with each other.

Within the NGN environment, the Session Initiation Protocol (SIP) is the predominant choice for setting real-time conversations in the signalling plane and Real-time Transfer Protocol (RTP) or Secure RTP (SRTP) are the de facto standards for transmitting the conversation in the form of IP packets in the media plane [7-9]. Nonetheless, various media codecs (e.g. GSM, PCM) and security controls (e.g. AES_CM_128_HMAC_SHA1_80, AES_CM_192_HMAC_SHA1_32) can be utilised by terminals to encode/decode and protect real-time media contents respectively. A conversation could be terminated prematurely should the calling parties use different codecs and/or security controls, even if they were in life threatening situations and intended to utilise real-time multimedia services to communicate with emergency responders; or the conversation could be established but without proper security protections, leaving the information transmitted in plaintext.

With the aim of overcoming the issue caused by mismatched codecs and/or security controls, gateways should be implemented to bridge the communication between incompatible terminals either within the NGN or from other networks. Since the late 1990s, a number of media gateways that provide transcoding support (i.e. convert one codec to another) have already been defined and developed for both research and commercial purposes, including [10-14]. In comparison, little work has been carried out on solving the problem posed by incompatible security controls (i.e. the confidentiality and integrity of the information will not be protected). Therefore, this paper will present a working prototype of the Technology Independent Security Gateway (TI-SGW) that is designed to fulfil the purpose of cross-ciphering (i.e. transform one security control to another) for incompatible terminals [15]. The paper will focus upon presenting an evaluation of the gateways performance characteristics.

The rest of the paper is structured as follows: section 2 presents related work in the domain of security controls for real-time multimedia services; section 3 illustrates the security gateway architecture and its function. The prototype of the gateway, its performance characteristics and the impact of these performance characteristics will be discussed in section 4, 5 and 6 respectively; and the conclusion and future work will be presented in section 7.

II. SECURITY CONTROLS FOR REAL-TIME MULTIMEDIA COMMUNICATIONS

It is well documented that SRTP is the fundamental protocol for securing the real-time multimedia communication on

TABLE I. A LIST OF CRYPTO SUITES OF THE SRTP

Crypto Suites
AES_CM_128_HMAC_SHA1_80
AES_CM_128_HMAC_SHA1_32
AES_F8_128_HMAC_SHA1_80
AES_192_CM_HMAC_SHA1_80
AES_192_CM_HMAC_SHA1_32
AES_256_CM_HMAC_SHA1_80
AES_256_CM_HMAC_SHA1_32

the NGN; and that SRTP relies upon the combination of crypto suites and key exchange methods to ensure the confidentiality and integrity of multimedia traffic. In this section, existing crypto suites and key exchange methods for securing the real-time multimedia communications will be examined.

A. *Crypto suites of the SRTP*

A crypto suite is a mixture of encryption and Message Authentication Code (MAC) algorithms that offer confidentiality, integrity and authentication for a piece of information. The default encryption algorithm of the SRTP is Advanced Encryption Standard (AES) that can operate in two modes (i.e. Segmented Integer Counter Mode AES (i.e. AES_CM) and AES in f8 mode) and three key sizes (i.e. 128, 192 and 256); and the standard message authentication and integrity algorithm of the SRTP is HMAC-SHA1 that can utilise two key sizes (i.e. 32 and 80) [7]. Based upon the combination of the encryption, message authentication and integrity algorithms and various key sizes, a set of crypto suites can be obtained for the SRTP (as presented in table 1) [16, 17].

B. *Key exchange protocols for the SRTP*

A key exchange protocol is the method by which cryptographic keys are exchanged between users, permitting cryptographic algorithms to be utilised. A number of key exchange protocols have been proposed to incorporate with the SRTP, constituting the security key exchange process between various terminals. In general, a key exchange protocol can utilise one of the three ways to manage the security key: through a Key Management Server (KMS), via the signalling plane and through the media plane.

Two key exchange protocols utilise the KMS approach to manage their key information: MIKEY pre-shared key and MIKEY-public key encryption [18, 19]. In both cases, a KMS is required for distributing security key material to terminals. Initially, individual terminals authenticate with the KMS to obtain their security key material (e.g. a pre-shared key or a private key) which is then utilised for securing the media transmission once the call set up phase is completed in the signalling plane; also the security key transmission process between individual terminals and the KMS should be protected by additional methods (e.g. a digital signature or a bootstrapping server function) [18, 20].

Three key exchange protocols rely upon the signalling plane to perform the key exchange process: Session Description Protocol (SDP) Security Descriptions for Media Streams (SDES), IMS Authentication and Key Agreement and Otway-Rees based key management [16, 20]. In this way, the key material is exchanged between terminals during the call setup SDP negotiation process in the signalling plane. Therefore, these protocols can only be utilised when the signalling plane is protected; otherwise, the confidentiality and integrity of the key material could be compromised.

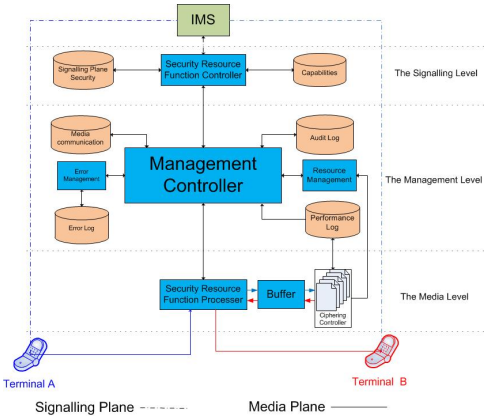

Fig. 1. The TI-SGW Architecture

ZRTP is a media plane key management protocol that utilises the Diffie-Hellman key exchange method to establish security key materials between terminals [21]. Terminals start to discover whether their peers also support the ZRTP once they obtain their IP addresses during the call set up phase; the media communication will be protected by ZRTP if all terminals in the conversation are compatible with the protocol.

It is not the authors intention to compare the advantages and disadvantage of the aforementioned crypto suites and key exchange protocols for the SRTP, but merely to present the variety of security controls which could be utilised for securing real-time multimedia services within the NGN environment. As mentioned in the introduction, a real-time multimedia session will not be established or protected if terminals utilise different security controls. Therefore, a security gateway that can ensure the confidentiality, integrity and authentication of a real-time multimedia session of security incompatible terminals is required. Such a security gateway will be described in the next section.

III. A TECHNOLOGY INDEPENDENT SECURITY GATEWAY

With the purpose of providing security support for any incompatible terminals to establish real-time multimedia communications, a novel technology independent Security Gateway was previously proposed by the authors [15].

As illustrated in Figure 1, the proposed security gateway architecture contains a number of internal modules, which enable the gateway to provide secure and timely security support. Based upon the nature of each internal module, they can be categorised into three levels: the signalling, media and management components. The signalling plane components are mainly responsible for negotiating with the signalling plane regarding

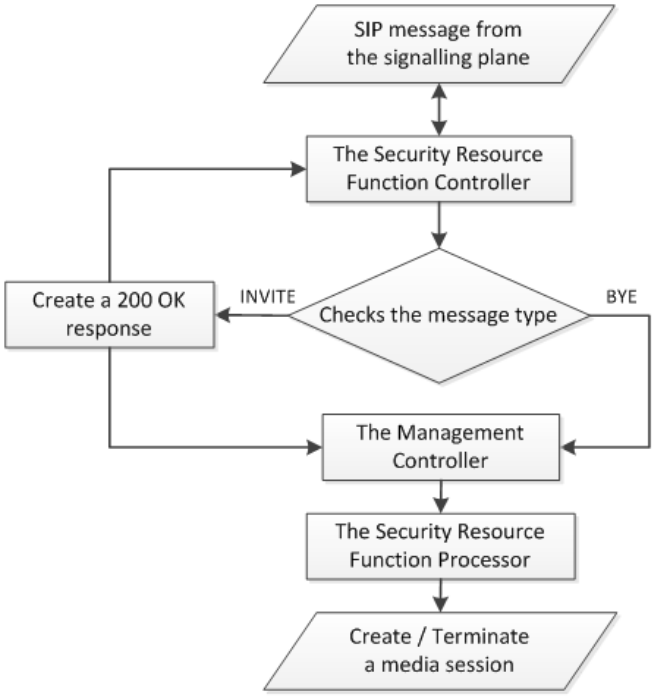

Fig. 2. A high-level information flow within the TI-SGW

various parameters for the establishment and management of a conversation. The media components are in charge of setting up appropriate media communication channels based upon the information negotiated in the signalling plane for incompatible terminals. The management controller controls the components from both the signalling and media planes; it also provides additional functionalities, such as resource management, error control and performance monitoring.

When a real-time multimedia session involving incompatible terminals occurs, the presence of the TI-SGW will be required. A high-level of information flow between the signalling plane and the TI-SGW and how the information is utilised by the gateway is illustrated in figure 2. When the signalling sends a SIP message to the TI-SGW, the Signalling Controller checks the type of the message and responds accordingly. If it is an invite message, the Signalling Controller replies with a 200OK message that contains the security capabilities (e.g. AES_CM_128_HMAC_SHA1_80 and security key information) and connection information (e.g. IP address and port number) of the gateway; in the meantime, the 200OK message will be forwarded to the Management Controller which utilises the information to create media thread for the Media Controller. Based upon the information provided by the Management Controller, the Media Controller can build up receivers and senders with appropriate security policies for decrypting incoming SRTP traffic from one terminal (e.g. caller) and encrypting outgoing traffic for the other terminal (e.g. callee). If it is a BYE message, the Management Controller utilises it to close related threads at the media level accordingly.

Based upon the TI-SGW architecture and the information flow logic, a prototype of the gateway is developed. Details of the prototype are fully described in section 4.

IV. TI-SGW PROTOTYPE

A working prototype of the proposed security gateway was developed based upon the architecture design presented in Section 3. The prototype, designed specially to deal with real-time multimedia traffic, is capable of providing security support for incompatible terminals in a timely fashion. According to the design of the TI-SGW architecture, the prototype was developed in Linux with three segments: the Security Resource Function Controller was developed in Java, while the Management Controller and the Security Resource Function Processor were developed in C. The connection between the Management Controller and the Security Resource Function Controller was created via a socket to minimise any potential communication delays.

The Security Resource Function Controller was developed based upon an existing open source SIP stack implementations (i.e. IMS-communicator) [22]. The Security Resource Function Controller is able to register the gateway with the signalling domain of the NGN through mutual authentication. Once the registration process is completed, the Security Resource Function Controller can be utilised to establish various call setup processes with the signalling plane of the NGN via different types of SIP messages stated in RFC 3261 [8]. As mentioned in the previous section, the Security Resource Function Controller forwards the information of the call setup to the Management Controller which will deal it accordingly (e.g. create/terminate media connections).

The Management Controller and the Security Resource Function Processor were developed based upon a number of C open source libraries: Pthread, Socket, Semaphore and Libsrtp [23]. Pthread was utilised by the Management Controller to create thread for media receivers and senders; semaphore was used by the Security Resource Function Processor to ensure the access to the critical section is guaranteed, reducing potential packet lost issue; libsrtp was capitalised by the Management Controller to provide security capabilities to the security gateway with majority of crypto suites presented in Table 1; and socket was utilised by Security Resource Function Processor for the setup of media receivers and senders at the socket level. In addition, a code snippet of how the Security Resource Function Processor operates after the Management Controllers create threads for media connection is illustrated in Figure 3.

It is envisaged that the Security Resource Function Controller and the Management Controller will not take much of the processing powering as the SIP is a lightweight text-based protocol and management of the security gateway would be performed occasionally with light activities (e.g. viewing how many existing connections). In comparison, the Security Resource Function Processor could require more resource as it deals constantly with media connections in real-time. Furthermore, a higher amount of processing power would be demanded as the number of connections increases. In order to explore the association between the number of threads and buffers with the processing power and their impact upon the

Receiving thread:

```
While (connection is alive) {
    If (incoming traffic ==true) {
        Lock the buffer;
        Store information in the buffer;
        Unlock the buffer;
    }
}
```

Sending thread:

```
While (connection is alive) {
    If (Buffer is not empty) {
        Lock the buffer;
        Decrypt information with terminal sender's security policy;
        Encrypt the information with terminal receiver's security policy;
        Send the information to terminal receiver;
        Unlock the buffer;
    } else {
            Wait for a period of time;
            }
}
```

Fig. 3. Logics for receiving and sending threads

real-time multimedia conversations, two versions of the sending mechanism of the Security Resource Function Processor were developed: a) Multi-thread Multi-buffer (MM) and b) Single thread Single buffer (SS). MM provides a dedicated thread and buffer for each sending thread while SS indicates that only one thread and one buffer is created for the sending function regardless the number of terminals. The receiving function utilises the multi-thread approach (i.e. each receiving port is a unique thread).

V. PERFORMANCE EVALUATION

With the aim of evaluating the TI-SGW, a set of experiments was conducted within a local private network. The security gateway prototype was configured on an Intel Pentium 4 computer (specification: Duo core 2.80GHZ processor and 2.9GB memory) with Ubuntu 12.04. Boghe was chosen as the terminal as it supports two crypto suites (i.e. AES_CM_128_HMAC_SHA1_80 and AES_CM_128_HMAC_SHA1_32) [24]. The Open IMS core was utilised to simulate the signalling part of the NGN [25]; the Open IMS core handles user registration and call set up processes. With the aim of obtaining meaningful experiment outcomes, 40 Boghe terminals (i.e. 20 pairs) were utilised: callers and callees were configured to use different crypto suites but the same codec (i.e. Pulse Code Modulation (PCM)); the capability of the TI-SGW could not be thoroughly tested if a lower number of terminals was chosen, e.g. 20 terminals; nonetheless, experiment outputs could be inaccurate should a higher number of terminals was picked, e.g. 60 terminals. Lastly, SDES was chosen as the key exchange protocol between the Security gateway and terminals due to its simplicity.

In total, 6 sets of experiments were conducted by utilising the combination of several parameters within the sending function of the security gateway: the thread and buffer (i.e. MM or SS), the waiting period before checking the buffer (i.e. 1 millisecond, 10 millisecond), and a random time between 1 and 10 millisecond. Also, several operational characteristics were chosen to evaluate the TI-SGW: CPU usage, the overall processing (i.e. the duration between a SRTP packet enters and leaves the TI-SGW), and the decryption and encryption time of a SRTP packet.

Figure 4 illustrates the CPU usage of the TI-SGW under the 6 experiments. In general, a higher amount of processing power is required as the number of terminal pairs increases. The MM based approach requires more processing power than the SS based method does for each additional pair. The waiting period for checking the buffer has little effect on the SS based methods in terms of the Security gateway CPU usage.

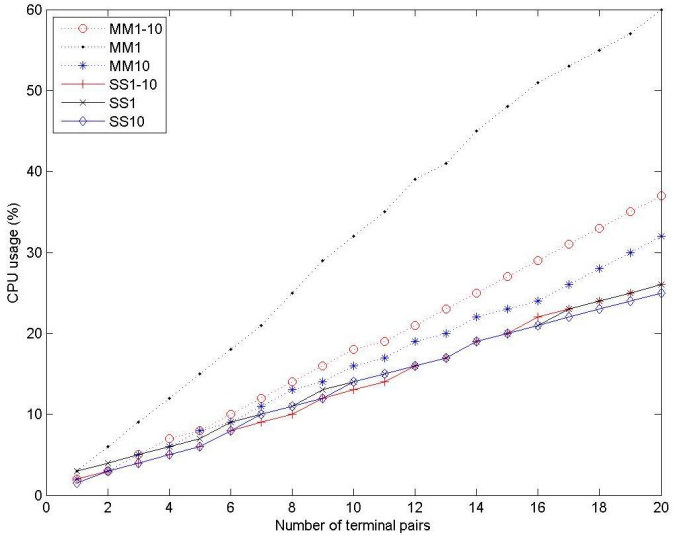

Fig. 4. TI-SGW CPU usage

TABLE II. THE AVERAGE PROCESSING TIME FOR OVERALL, DECRYPTION AND ENCRYPTION PER SRTP PACKET IN MICROSECOND

	MM1-10	MM1	M10	SS1-10	SS1	SS10
Overall	3,566	581.5	5,067	3,758	949	6,349
Decryption	20.4	22.2	19.9	15.2	16.3	15.1
Encryption	12.7	13.1	12	10.7	11	10.6

In comparison, the same parameter has a stronger influence on the Security gateway CPU usage when the MM based approach was utilised. For example, when 20 pairs of terminal were connected to the Security gateway, its CPU usage were 60% and 32% for the waiting period of 1 millisecond and 10 milliseconds respectively.

Figure 5 presents the outcome of the cumulative distribution function on the overall processing time of the Security gateway for SRTP packets of the first pair of terminals in various scenarios. In total, about 200,000 SRTP packets were processed for the first pair of terminals, representing a 25-minute conversation (i.e. from the starting to the ending time of each experiment). As shown in the figure, both MM and SS approaches have less influence on the overall SRTP processing time than the waiting periods do. When the waiting period was chosen for 1 millisecond, 80% of the SRTP packets were processed within 1 millisecond frame. In comparison, the Security gateway had to take 8-10 milliseconds to process 80% of the SRTP packets when the 10 milliseconds waiting period were chosen.

Figure 6 and 7 show the decryption and encryption processing time for the SRTP packets of the first pair of terminals from all 6 experiments. In general, the security gateway spends between 10-30 microseconds to decrypt 90% of the SRTP packets in all scenarios; while the encryption process only takes around 10-20 microseconds for the same amount of SRTP packets [26, 27].

The average processing time on a SRTP packet during the overall, decryption and encryption processes in all experiments

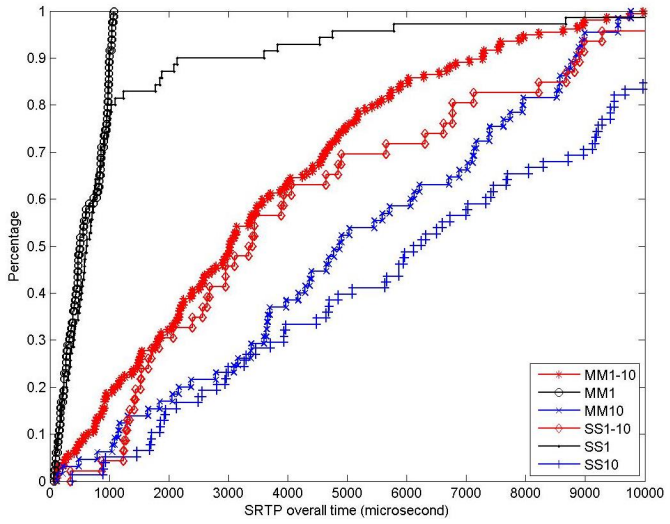

Fig. 5. The overall processing time for SRTP packets of the first pair of terminals

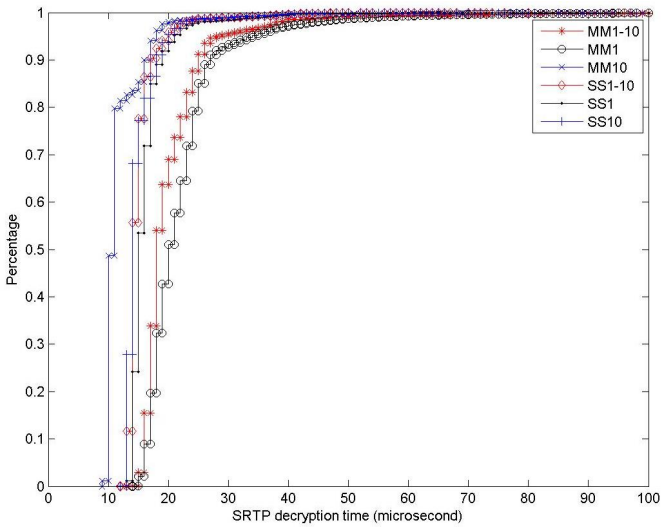

Fig. 6. Decryption processing time for SRTP packets of the first pair of terminals

is summarised in Table 2. In general, the MM approach with 1 millisecond waiting period achieved the best overall processing time of 581.5 microseconds. In comparison, it takes the SS approach with a 10 milliseconds waiting period around 6 milliseconds; interestingly the same approach achieved the best performance on average decryption and encryption time with 15.1 and 10.6 microseconds respectively.

VI. DISCUSSION

Based upon the presented experiment outcomes, it demonstrates that the security gateway prototype is capable of providing support for incompatible terminals for the NGN platform. Generally, the multi-threading multi buffer based approach provides better performance than the single threading buffer based technique does in terms of overall delay introduced by the security gateway (as demonstrated in Table 2). This is

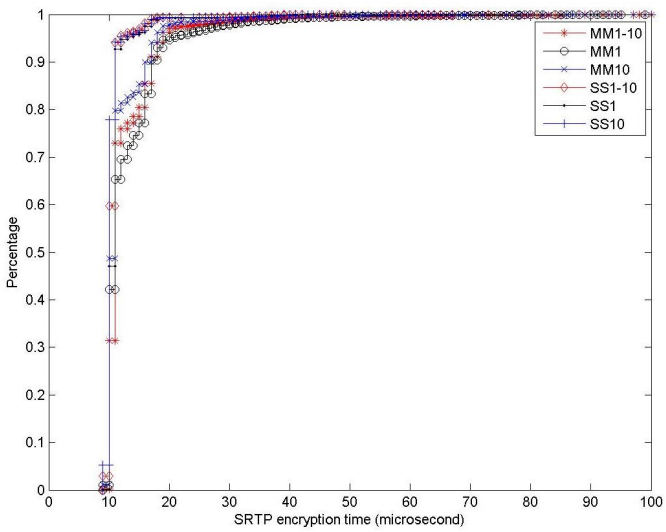

Fig. 7. Encryption processing time for SRTP packets of the first pair of terminal

expected as the MM approach provides dedicated threads for each pair of terminals while all the terminals have to share the single processing resource in the SS technique. Nonetheless, the shortcoming of the former approach is that it requires more CPU resources for supporting the same number of terminal pairs than its counterpart does as the additional threads were created (as demonstrated in Figure 4).

The waiting periods were chosen based upon previous empirical studies. The CPU of the security gateway would be taken over by the checking process should a shorter period of time was chosen (e.g. 20 microseconds); while a degraded quality would be experienced by users if a longer period of time was allocated (e.g. 100 milliseconds). This phenomenon is also reflected by results presented in Figure 4. Nonetheless, based upon the ITU-Ts one way delay less than 150 millisecond recommendation on voice communications [28], it demonstrates that the overall delay that is introduced by the TI-SGW could be ignored as the overall processing time is less than 10 milliseconds for most of the scenarios (apart the result from SS1-10).

From the results presented by Figure 6, 7 and Table 2, they demonstrate that the decryption and encryption time on SRTP packets are significantly smaller than the overall processing time. Hence, more investigation should be carried on the topic of reducing the overall delay time but without compromising the processing powers. Furthermore, the decryption and encryption time on the SRTP packet should be the same in theory as the symmetric encryption method was utilised. However, the results show that the security gateway took more time on the decryption process than the encryption process. This could be caused by the implementation of the libsrtp library (e.g. the number of machine cycles for both processes could be different).

VII. CONCLUSION

This paper has identified the need for a security gateway that can provide ciphering support for incompatible real-time multimedia terminals for the NGN. Also based upon the security gateway architecture, a working prototype that is fit the purpose has been developed.

Based upon the experiment outputs, the results demonstrate that a gateway can provide security support for incompatible terminals. It is envisaged that the overhead introduced by the TI-SGW (less than 10 milliseconds) would have little impact on the real-time multi-media conversations based upon the ITU-T's 150 milliseconds one way delay recommendation on one way voice communication. To this end, future research will be focused upon two directions: reducing the overhead caused by the TI-SGW and obtaining real users opinion upon the TI-SGW and getting a measure for the Quality of Experience.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 284863 (FP7 SEC GERYON).

REFERENCES

[1] ETSI (2014) "Next Generation Networks", available: http://www.etsi.org/technologies-clusters/technologies/next-generation-networks, accessed: 27 May 2014

[2] 3rd Generation Partnership Project (3GPP), IP Multimedia Subsystem (IMS) Stage 2, Technical Specification 23.228, available: http://www.3gpp.org/ftp/Specs/html-info/23228.htm, accessed: 5 March 2014

[3] Carlberg, K. and Atkinson, R. (2004) IP Telephony Requirements for Emergency Telecommunication Service (ETS), RFC 3690, available from: http://tools.ietf.org/html/rfc3690, accessed: 1 April 2013

[4] HEERO (2014) Harmonised eCall European Pilot, available: http://www.heero-pilot.eu/view/en/home.html, accessed: 10 May 2014

[5] GERYON (2014) Next Generation Technology Independent Interoperability of Emergency Services, available: http://www.sec-geryon.eu, accessed: 10 April 2014

[6] SECRICOM (2014) The SECRICOM project, available: http://www.secricom.eu/funding-scheme, accessed: 15 April 2014

[7] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman (2004) The Secure Real-time Transport Protocol (SRTP), RFC 3711, IETF, available at: http://www.ietf.org/rfc/rfc3711.txt, accessed 20 November 2012.

[8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler (2002) SIP: Session Initiation Protocol, RFC3261, IETF, available at: http://www.ietf.org/rfc/rfc3261.txt, accessed: 06 January 2014

[9] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson (2003) RTP: A Transport Protocol for Real-Time Applications, RFC3550, IETF, available at: http://www.ietf.org/rfc/rfc3550.txt, accessed: 01 January 2014

[10] A.M. Grilo, P.M. Carvalho, L.M. Medeiros, and M.S. Nunes (1999) "VTOA/VoIP/ISDN telephony gateway," ATM, 1999. ICATM '99. 1999 2nd International Conference on, vol., no., pp.230-235, 1999 doi: 10.1109/ICATM.1999.786807

[11] A. Conte, L.P. Anquetil, and T. Levy (2000) "Experiencing Megaco protocol for controlling non-decomposable VoIP gateways," Networks, 2000. (ICON 2000).Proceedings. IEEE International Conference on, vol., no., pp.105-111, 2000 doi: 10.1109/ICON.2000.875776

[12] F.C. Castello, R. Balbinot, J.G. Silveira and P.M. Santos (2003) "A robust architecture for IP telephony systems interconnection," Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on , vol.2, no., pp. 593- 596 vol.2, 28-30 Aug. 2003, doi: 10.1109/PACRIM.2003.1235851

[13] Y. Guo, M. Liang, Y. Guo, and L. Zhang (2004) "A design scheme of PSTN media gateway," Signal Processing, 2004. Proceedings.ICSP '04. 2004 7th International Conference on , vol.3, no., pp. 2651- 2654 vol.3, 31 Aug.-4 Sept. 2004, doi: 10.1109/ICOSP.2004.1442327

[14] Asterisk (2014) Asterisk, available: www.asterisk.org, accessed: 08 May 2014

[15] F. Li, N.L. Clarke and S.M. Furnell (2013) A Technology Independent Security Gateway for Real-Time Multimedia Communication, Proceedings of the 7th International Conference on Network and System Security (NSS2013), 3-4 June 2013, Madrid, Spain, pp14-25, 2013

[16] F. Andreasen, M. Baugher, and D. Wing (2006) Session Description Protocol (SDP) Security Descriptions for Media Streams, RFC 4568, IETF, available at: http://www.ietf.org/rfc/rfc4568.txt, accessed 05 November 2012.

[17] D. McGrew (2011). The Use of AES-192 and AES-256 in Secure RTP, RFC 6188, IETF, available at: http://www.ietf.org/rfc/rfc6188.txt, date accessed: 27 November 2012.

[18] J. Mattsson and T. Tian (2011) MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY), RFC6043, IETF, available at: http://www.ietf.org/rfc/rfc6043.txt, date accessed 27 December 2012

[19] V. Cakulev and G. Sundaram (2011) MIKEY-IBAKE: Identity-Based Authenticated Key Exchange (IBAKE) Mode of Key Distribution in Multimedia Internet KEYing (MIKEY), RFC 6267, IETF, available at: http://www.ietf.org/rfc/rfc6267.txt, date accessed 27 November 2012.

[20] 3GPP TR 33.828 (2012) IP Multimedia Subsystem (IMS) media plane security, available from: http://www.3gpp.org/ftp/Specs/html-info/33828.htm, date accessed 27 November 2012.

[21] P. Zimmermann, A. Johnston (Ed.) and J. Callas (2011) ZRTP: Media Path Key Agreement for Unicast Secure RTP, RFC 6189, IETF, available at: http://www.ietf.org/rfc/rfc6189.txt, date accessed: 27 November 2012

[22] PT Inovacao (2014) IMS communicator, http://imscommunicator.berlios.de/, accessed:10 May 2014

[23] D. McGrew (2001) libSRTP 1.4 Overview and Reference Manual, available: http://srtp.sourceforge.net/libsrtp.pdf, accessed: 11 May 2014

[24] Boghe (2014) Boghe, available: http://code.google.com/p/boghe/, accessed 30 March 2014

[25] Fraunhofer FOKUS NGNI (2014) Open Source IMS, available: http://www.openimscore.org/, accessed 27 May 2014

[26] P.K. Nakarmi, J. Mattsson and G.Q. Maguire (2011) "Evaluation of VoIP media security for smartphones in the context of IMS," Communication Technologies Workshop (Swe-CTW), 2011 IEEE Swedish , vol., no., pp.123,128, 19-21 Oct. 2011, doi: 10.1109/Swe-CTW.2011.6082479

[27] A.L. Alexander, A.L. Wijesinha and R. Karne (2009) "An Evaluation of Secure Real-Time Transport Protocol (SRTP) Performance for VoIP," Network and System Security, 2009. NSS '09. Third International Conference on , vol., no., pp.95,101, 19-21 Oct. 2009, doi: 10.1109/NSS.2009.90

[28] ITU-T (2003) G.114: One-way transmission time, article number: E24508, available: https://www.itu.int/rec/T-REC-G.114-200305-I/en, accessed: 02 June 2014