Securing RSVP and RSVP-TE Signaling Protocols and Their Performance Study

Jing Zhi, Chung-Horng Lung, Xia Xu
Dept. of Systems and Computer Engineering
Carleton University, Ottawa, Canada
chlung@sce.carleton.ca

Anand Srinivasan, Yi Lei
EION Inc.
945 Wellington St., Ottawa, Canada
anand@eion.com

Abstract

RSVP and RSVP-TE are signaling protocols used to set up paths and/or support Quality of Service (QoS) requirements in IP and MPLS-based networks, respectively. This paper analyzes an authentication mechanism for securing the RSVP and RSVP-TE control messages, and studies their performance. This design and implementation of the authentication mechanism, which is based on RFC2747, using four commonly adopted hash algorithms - MD5, RIPEMD160, SHA-1, and SHA-256, not only improves security, but also provides useful information from the performance aspect. The time for authenticating the signaling messages depends on the algorithm used, and increases slightly in the order of MD5, SHA-1, RIPEMD160 and SHA-256. The performance of the RSVP-TE with multiple sessions was measured.

1. Introduction

With the emergence of the distributed multimedia applications such as audio/video conferencing, the traditional networks based on the best-effort service delivery model could not satisfy the requirements of the real-time performance needed by these applications. Resource ReSerVation Protocol (RSVP) [1,2] provides one of the solutions in IP networks by setting up network bandwidth reservation. RSVP-TE is an extension of the RSVP protocol, which is used in Multi-Protocol Label Switching (MPLS)-based networks to establish and maintain explicitly routed Label Switched Paths (LSPs). MPLS has received tremendous attention in both industry and academia for its ability to support traffic engineering, protection and restoration, and virtual private networks.

Along with the increased connectivity and new services, computer networks have also allowed technically advanced intruders opportunities to carry out a variety of attacks that threaten the integrity of its infrastructure and violate the privacy of its users [3,4]. Corrupted or spoofed reservation requests can lead to theft of services by unauthorized parties or denial of services. Lately, research groups have been looking for way of securing the RSVP messages. Different security protocols are proposed [5,6]. The RSVP protocol is embedded with the following security features [6]:

- Message Integrity and router (node) authentication;
- User Authentication;
- Non-repudiation;
- Confidentiality;
- Replay;
- Traffic Analysis; and
- Denial of Service.

Current efforts to make the RSVP protocol more secure is not only far from complete, but also not yet thorough. Since the specification of the RSVP [2] became an IETF standard in 1997 [1], less attention has been paid to this area.

The security services with RSVP assure the integrity and authenticity of services in networks. However, the overhead required by the RSVP message flow, the traffic scheduling procedure and its security mechanisms actually degrade the performance of the network [7,8]. It is expected that securing the RSVP and RSVP-TE messages may also bring the overhead and performance side effects to the network. However, to the authors' knowledge, presently there is no report on the influence of security protocols on the performance of RSVP and RSVP-TE.

This work will investigate ways to secure RSVP in IP networks and RSVP-TE in the MPLS framework in the EION Open IP Environment [9]. An integrity and authentication of the RSVP and RSVP-TE messages using a one-time-use sequence number, authentication key and keyed hash algorithms was implemented. This paper focuses on the influence of the security protocol on the performance of the two protocols in terms of time and space complexity.

This paper is organized as follows: section 2 discusses an approach to securing RSVP and RSVP-TE protocols based on authentication. Sections 3 and 4 present performance analysis of authenticating RSVP and RSVP-TE messages, respectively. Finally, section 5 summaries the paper.

2. Securing RSVP and RSVP-TE Protocols

2.1 RSVP and RSVP-TE protocols

RSVP is a signaling protocol used to request specific QoS from the network for data flow and by a router to deliver QoS requests to all nodes along the path(s) of the data flow. It is also used for routers to establish and maintain the requested service. The QoS parameters, such as bandwidth, are processed by a traffic control module that includes: (1) a packet classifier, (2) admission control, and (3) a packet scheduler. After being analyzed and processed by the traffic control module, the reserved information such as bandwidth and buffer space is sent to RSVP for processing reservation.

RSVP is receiver-oriented; the receiver of the data flow is responsible for the initiation of the resource reservation. Periodically, the RSVP process at a router scans the path

0-7803-8932-8/05/$20.00 © 2005 IEEE

state to create a new PATH message to be forwarded to the neighboring router along a path to the receiver. And then the receiver periodically sends a RESV message to establish and update the reservation state. The reservation state will be automatically invalidated after timeout unless the RESV signal is refreshed.

PATH and RESV are two of the RSVP message types, which are composed of sequences of objects. The PATH message, as an example, has the following format [1],

<PATH Message> :: = <Common Header> [<INTEGRITY>] <SESSION> <RSVP_HOP> <TIME_VALUE> [<POLICY_DATA> …] [<sender descriptor>]

where the Common Header, and the objects such as INTEGRITY, and SESSION, are specified in RFC 2205. The INTEGRITY object is used for an authentication to check the integrity and authenticity of the RSVP PATH messages. The RESV message also includes an INTEGRITY object that is used for authenticating the RSVP RESV message.

In MPLS networks, the RSVP-TE is a signaling protocol that adds a number of extensions to the original RSVP to set up LSP tunnels and support traffic engineering applications. RSVP-TE supports all RSVP functions, but adds new objects to PATH messages and RESV messages, such as Explicit Route, Label Request Objects, Record Route Object, Session Attribute Object, etc. The authentication of RSVP-TE messages is also implemented by the INTEGRITY object.

2.2 Authenticating RSVP and RSVP-TE messages

There are three main security concerns for RSVP: 1) Message integrity and node authentication; 2) User authentication; 3) Secure data stream [1]. In concern with the message integrity and node authentication, security protocols of SDS/CD [5], RSVP-SQoS [6] and hop-by-hop authentication were proposed. Wu et al. [5] primarily dealt with intruders that are malicious routers on the reservation path. Talwar, et al. [6] designed a secure RSVP protocol, RSVP-SQoS, to secure RSVP transmission of QoS parameters in three phases.

Authentication of the RSVP [1] uses the embedded INRTEGRITY object in the RSVP message in a hop-by-hop manner. The specific security goals are as follows:

- Prevention of forgery and modification of RSVP and RSVP-TE messages;
- Data origin authentication of sending routers; and
- Prevention of message replay

In this work, the authentication mechanism is implemented and conforms to RFC 2747. The INTEGRITY object carries an authenticating digest of the RSVP message, computed using a secret authentication key and a keyed-hash algorithm. It is also tagged with a one-time-use sequence number, which allows the message receiver to identify playbacks and hence to thwart replay attacks.

The security strength of the authentication mechanism is based on two aspects: the strength of the algorithm and the length of the key. In RSVP message authentication, the authentication algorithm was suggested to use hash algorithms and required to be altered in case of a security breach [10]. In MD4 family, due to the fact that MD4 and RIPEMD have lost their significance in application (refer discussion in Section 2.4.1), there is a need to select algorithms with different lengths of digest and different complexities. MD5 has 16-byte digest and is the most popular hash function used today although one of its properties, collision resistant, is breached [11]. Research showed that it did not pose a threat in actual applications [12,13]. SHA-1 conforms to the FIPS PUB 180-1 Secure Hash Standard (SHS) with 20-byte long digest, while SHA-256 conforms to the FIPS PUB 180-2 Secure Hash Standard (SHS) with 32-byte long digest [14]. RIPEMD160 is developed with 20-byte long digest. Among these algorithms, no breach of RIPEMD160, SHA-1 and SHA-256 has been found to date. In this work, the standard implementations were used for the experiments.

Secret keys were used for the authentication. They were generated by a random-pseudo number generator. Keys with lengths of 8, 16, 32 and 64 bytes were generated during experiments. 30 to 1100 keys (each has its own key id) were generated for changing keys when the current key exceeds its lifetime. The number of keys is also one of the security parameters. The key management, including key generation, storage, distribution/transfer, and deletion, is an important and challenging issue in cryptosystems. In this work, the authentication algorithms use the secret key mechanism, and the key is distributed in the network by the operator. Other factors of keys such as more secure distribution and storage are not within the scope of this study.

2.3 Performance measurements

Network topology. Figure 1 illustrates the experimental network for RSVP message flow during reservation setup.

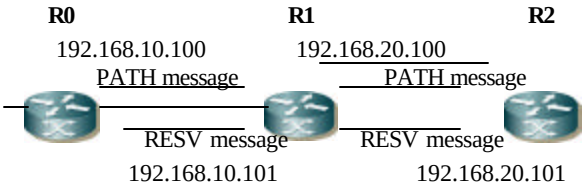

Figure 1 Network topology for the experiment

R0, R1, and R2 all represent either IP routers or label switched routers (LSRs). The incoming and outgoing IP-address interfaces of the routers are also shown in Figure 1. The parameters of the computers that run these router programs are shown in Table 1.

Table 1 System parameters of the computers that run the router program

Router	CPU	Speed	Memory	O.S.
R0	Pentium III	410.745 MHz	261MB	Redhat Linux 8.0
R1	Pentium III	799.807 MHz	261MB	Redhat Linux 8.0
R2	Intel Celeron	1933.586 MHz	260MB	Redhat Linux 8.0

Methodology. In this work, various processing times were measured and are described in Table 2.

Table2 Definitions of different time measurements

Time measured	Definition
Time for generating keys	Δ_K, Delay in time when the key generation starts and ends
Time for computing message digest	Δ_D, Delay in time when certain algorithms start and end
Time for Path message encoding	Δ_P, Delay in time when the authentication encoding function is called and returned; a period of encoding *PATH* message
Time for RESV message encoding	Δ_R, Delay in time when the authentication encoding function is called and returned; the period of encoding *RESV* message
Time for PATH message decoding	Δ_{Pd}, Delay in time when the authentication decoding function is called and returned; a period of decoding *PATH* message
Time for RESV message decoding	Δ_{Rd}, Delay in time when the authentication decoding function is called and returned; a period of decoding *RESV* message
Time RSVP-TE connection setup	Δ_T, Delay in time when the sender sends a *PATH* message and receives a *RESV* message

For all the measurements, the experiment was set up in a way that was similar to the realistic usage of the routing system in the field, except that there was no background data traffic. There were only control messages, including OSPF and RSVP or RSVP-TE messages. The process scheduling was based on the kernel scheduling. No process priority was changed during the experiment. The time values were obtained using the system function: *gettimeofday( )*. Each experiment was repeated about ten times. The average values were presented as the results in this paper.

3. Performance analysis of authenticating RSVP messages

3.1 Space Complexity of the Authentication Protocol

An RSVP INTEGRITY Object has a fixed size for fields of RSVP_OBJ, flag, a reserved bit, Key Identifier, Sequence Number. Their sizes are 4, 1, 1, 6, and 4 bytes, respectively. The message digest is algorithm-dependent. In this work, the authentication uses Keyed Hashing for Message Authentication (HMAC) [RFC 2104] with Message Digest algorithms: MD5, RIPEMD-160, SHA-1, and SHA-256. The generated digests have sizes of 16, 20, 20, and 32 bytes, respectively for HMAC-MD5, HMAC-RIPEMD-160, HMAC-SHA-1, HMAC-SHA-256. Thus, the total sizes of the INTEGRITY Objects are 32, 36, 36, and 48 bytes, respectively, by combining hash algorithms of MD5, RIPEMD-160, SHA-1, and SHA-256. So the overheads of 32, 36, 36, and 48 bytes, respectively, for MD5, RIPEMD160, SHA-1, and SHA-256, are needed for each authentication request message.

3.2 Time for Generating Message Digest

The time for authenticating the RSVP PATH message and generating its message digest was measured (for R_0 and thereafter, if not otherwise specified), as shown in Figure 2.

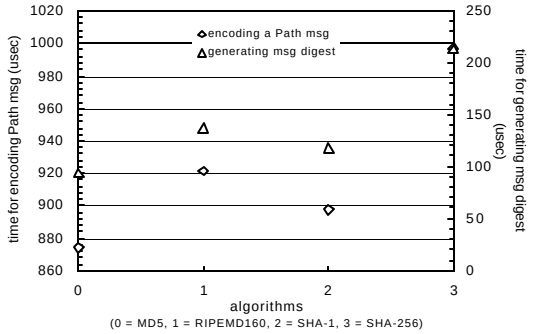

Figure 2 Time measured for encoding PATH message and time for generating message digest as a function of different hash algorithms (number of key – 30; Length of key – 8 bytes)

It can be seen that MD5 has the shortest time of generating the message digest and encoding the whole *PATH* messages, and SHA-256 has the longest. SHA-256 has the longest hash value, at 32 bytes, and the most complicated block processing procedures among these hash algorithms. Thus SHA-256 is the most secure, but brings more overhead. The time for generating the message digest is 94 µsec, 137 µsec, 118 µsec, and 213 µsec, respectively, for MD5, RIPEMD160, SHA-1 and SHA-256.

3.3 Time for Authenticating PATH Messages

Figure 3 shows the results for authenticating PATH messages.

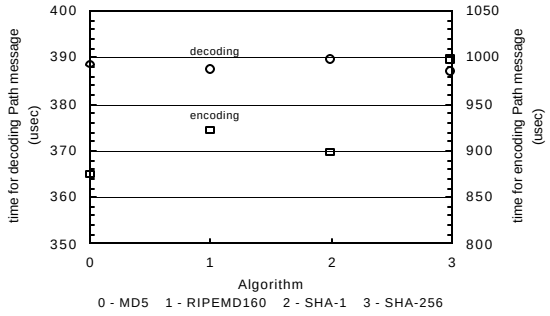

Figure 3 Time for encoding and decoding a RSVP PATH message (for R1 for comparison with the same CPU)

It can be seen that the time for encoding and decoding a PATH message varies for different algorithms; and the decoding time is shorter than the encoding time. This occurred when encoding a PATH message, because extra operations, such as encoding the RSVP_OBJ header and generating a sequence number, were undertaken.

Figure 3 reveals that the MD5 requires the shortest time for encoding, while the SHA-256 requires the longest. This may occur because MD5 leads to the shortest message digest value (16 bytes) and the SHA-256 creates the largest digest

value (32 bytes). Comparing RIPEMD160 with SHA-1, although both have the same length of message digest, SHA-1 shows a better performance than RIPEMD160.

3.4 Time for Authenticating RESV Messages

It can be seen that the times for encoding and decoding RESV messages are almost the same, about 387 μsec, and almost independent of the algorithms in this experiment environment. This indicates that the algorithms can be equally selected in terms of the time for encoding/decoding the RESV messages.

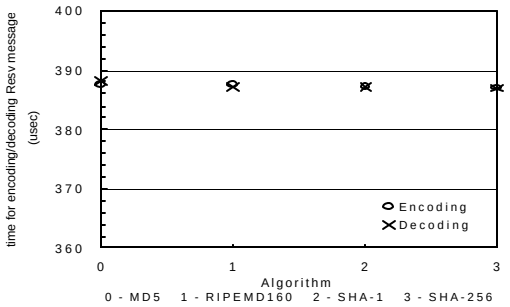

Figure 4 Time for encoding and decoding a RESV message (for R1 for comparison with the same CPU)

3.5 Discussion on Authenticating RSVP Messages

In IP networks, the reservation state is set up through timely refreshes between senders and receivers for the reservation connection. The RSVP message connection setup time can be defined as the delay between the time the sender first detects a PATH message and the time the sender receives a RESV message. The connection setup time for RSVP messages can be affected by the time for key generation, message digest computation, PATH message encoding and decoding, and RESV message encoding and decoding. From the measurement results, the time required for authenticating RSVP messages varies with different authentication key parameters and authentication algorithms. For a certain key used, the connection setup time could increase in an order of MD5, SHA-1, RIPEMD160, and SHA-256 if other factors in the network are the same.

4. Performance Analysis of Authenticating RSVP-TE Messages

The authentication mechanism is integrated into the RSVP-TE protocol to authenticate the signaling messages of RSVP_TE. Compared with RSVP messages, the additional objects of EXPLICIT_OBJ, LABEL_REQUEST _OBJ, and RECORD_ROUTE_OBJ are added to the RSVP-TE PATH and RESV messages. As is known, the time for generating authentication keys and message digest is only dependent on the authentication mechanism itself, and is independent of the protocols of RSVP or RSVP-TE. Thus, for RSVP-TE protocol, only the time for authenticating RSVP-TE PATH and RESV messages and the time for RSVP-TE connection setup were measured, which is shown below.

4.1 Time for Authenticating RSVP-TE PATH Message

Figure 5 shows the time required for encoding and decoding the PATH message with different algorithms.

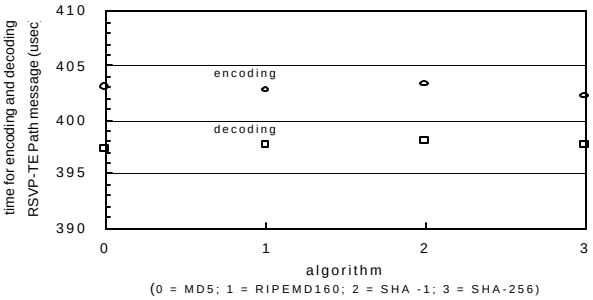

Figure 5 Time required for encoding and decoding RSVP-TE PATH message with different algorithms

Figure 5 shows that the time for encoding RSVP-TE PATH message is larger than that of decoding. And the times required for encoding and decoding RSVP-TE PATH messages are almost independent of the algorithms within the experimental error, which is similar to that of decoding RSVP message, shown in Figure 3.

4.2 Connection Setup Time for RSVP-TE Messages

The connection setup time (Δ_T) for RSVP-TE messages in setting up an LSP in MPLS-based network is defined as the delay between the time the sender first detects a PATH message and the time the sender first receives a RESV message. The first PATH and RESV messages of a session were measured. In this experiment, the connection setup time was measured both with and without the authentication mechanism in RSVP-TE messages, designated Δ_A and Δ_{NA}, respectively. When there is only one session and the authentication mechanism is disabled in the network, the connection setup time was Δ_{NA} = 57.3 msec for RSVP-TE. The connection setup time was Δ_A = 61.7 msec when the authentication mechanism was enabled with MD5. The time ratio (ϕ) of the authentication mechanism versus the mechanism without authentication was calculated as:

$$\phi = \Delta_A / \Delta_{NA} \qquad \text{(Eq. 1)}$$

The time ratio for setting up an LSP by RSVP-TE messages is 1.08 when there is only one session in the network. Multiple sessions were then set up, and only one LSP was established in each session. The connection setup time of an RSVP-TE message was measured for different number of sessions. Figure 6 shows the average connection setup time of an RSVP-TE message as a function of the number of sessions with and without authentication mechanism. It can be seen that the connection setup time increases as the number of sessions increases. The time ratio, ϕ, was calculated and shown in Table 3 for MD5 at some of session numbers. In considering the experimental error, the least square fits of the experimental values with polynomials are also shown as the solid lines in this figure. In case of the authentication, the fitting equation obtained from Excel is:

$$y = -0.0837x2 + 6.6139x + 68.739 \qquad \text{(Eq. 2)}$$

with the R-squared value of 0.99. In case of the non-authentication, the fitting equation gotten from Excel is:

$$y = -0.0612x2 + 5.4318x + 33.753 \quad \text{(Eq. 3)}$$

with R-squared value of 0.98. The estimated time ratio, ϕ_{cal}, of the connection setup time was calculated from the fitting curves using Eq. 2 and Eq. 3, respectively, and was also shown in Table 3 for MD5. The estimated time ratio qualitatively decreases as the number of session increases.

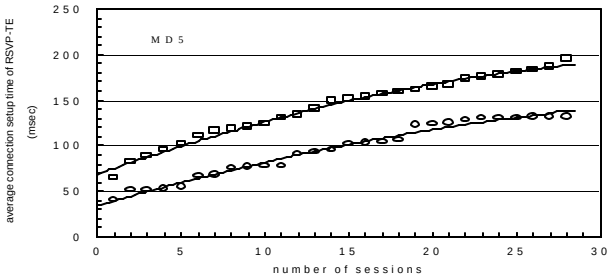

Figure 6 Average connection setup time of RSVP-TE as a function of the number of sessions and the fitting curves

The connection setup time was also measured for other authentication algorithms of RIPEMD160, SHA-1, and SHA-256. The average connection setup times using different authentication algorithms show similar behaviors with the increase in the number of sessions. The least square fits were also applied to these algorithms. Results of the estimate time ratios, ϕ_{cal}, together with the measured time ratios, ϕ, are shown in Table 3 for different authentication algorithms at a selective number of sessions.

Table 3. The time ratio of authentication mechanism in RSVP-TE message for different sessions

No. sessions	Processing overhead (ϕ) **(%)**							
	MD5		**RIPEMD160**		**SHA-1**		**SHA-256**	
	ϕ	ϕ_{cal}	ϕ	ϕ_{cal}	ϕ	ϕ_{cal}	ϕ	ϕ_{cal}
2	1.63	1.84	1.95	2.23	1.80	2.05	1.89	2.03
5	1.89	1.68	2.09	1.84	1.87	1.62	1.89	1.77
10	1.60	1.54	1.67	1.55	1.46	1.32	1.70	1.57
15	1.47	1.47	1.42	1.42	1.23	1.23	1.44	1.49
20	1.32	1.42	1.26	1.36	1.08	1.23	1.44	1.47
25	1.38	1.38	1.41	1.35	1.29	1.28	1.48	1.47
28	1.48	1.36	1.45	1.36	1.48	1.34	1.54	1.49

The measured data from Table 3 were obtained under the same experimental conditions and reflects quantitatively the change of the time ratio with the number sessions using different authentication algorithms.

4.3. Discussion on Authenticating RSVP-TE Protocols

Different from RSVP, where the sender sends the PATH messages and the receiver sends the RESV messages along a hop-by-hop path, the RSVP-TE is used to set up an LSP on an explicit route. The ingress sends out the route information in a PATH message to request label binding from the egress. Upon receiving the PATH message, the RSVP-TE RESV message distributes labels upstream along the explicit route. By authenticating the PATH and RESV messages, the explicit route information (included in EXPLICIT_ROUTE object), the labels of the LSP, and resource reservation parameters such as bandwidth are authenticated. This contributes to the security assurance of the MPLS networks.

5. Summary

Security is crucial in communications networks. This paper investigated security aspect of the RSVP and RSVP-TE protocols used in IP and MPLS networks, respectively; because both protocols are used for network control purpose. The paper discussed how to incorporate the authentication mechanism into the protocols. Four different commonly used hash algorithms - MD5, RIPEMD160, SHA-1, and SHA-256 - were implemented for performance evaluation. The results can be used for reference if authentication is considered for these two protocols.

Acknowledgements

We would like to thank Dr. M. Zaid and Dr. R. Crawhall of NCIT, Ottawa and Dr. R. Munikoti and Dr. K. Kalaichelvan of EION Inc., for supporting this research. We also wish to thank Dr. P. Dhakal of EION Inc for RSVP-TE implementation help.

References

[1] Braden, R., Zhang, L., Berson, S., Herzog, S., "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", *RFC 2205*, 1997.

[2] Zhang, L., et. al, "Rsvp: A new resource reservation protocol", *IEEE Network*, 7 (5), pp. 8-18, 1993.

[3] Wang, G., "Inter-network Security", Proc. of ICCS'94 Conf., Vol. 3, pp. 14-18, 1994.

[4] Molva R., "Internet security architecture", *Computer Networks*, Vol. 31, 787-804, 1999.

[5] Wu, T.-L., Wu, S.F, Gong F.M., "Securing QoS: Threats to RSVP messages and their countermeasures", *Proc. of IWQoS*, pp.62-64, 1999.

[6] Talwar, V., Nahrstedt, K., and Nath, S. K., "RSVP-SQOS: A secure RSVP protocol", *Proc. of International Conference on Multimedia and Expo*, pp.579-582, 2001.

[7] Barzilai, T.P. & Kandlur, D.D., "Design and implementation of an RSVP-based quality of service architecture for an integrity services Internet", *J. on Selected Area in Commu.*, 16(3), 98.

[8] Anindya and Chiueh Tzi-cker, "Performance analysis of an RSVP-capable router", *IEEE Network*, 13 (5), pp.56-63, 1999.

[9] *Common System Plane, Developer's Guide, Open IP Environment 2.0.2*, EION Inc, 2000.

[10] Baker, F., Lindell, B., Talwar, M., "RSVP Cryptographic Authentication", RFC 2747, 2000.

[11] den Boer, B., "An attack on the last two rounds of MD4," *Advances in Cryptology, proc. Crypto'91*, pp. 194-203, 1992.

[12] Dobbertin , H., "The Status of MD5 after a Recent Attach", *CryptoBytes* Vol.2, No.2, pp. 1-6, 1996.

[13] Ragab, A. H. M., and Ismail, N. A., "An efficient message digest algorithm (MD) for data security", *Proc. of 10th Int'l Conf on Electrical and Electronic Tech.*, pp. 191-197, 2001.

[14] NIST, "FIPS 180-2: Secure Hash Standard (SHS)", available at http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf, 01.