HTSS: Hash Tree Signature Scheme for Multicast Authentication

Kannan Balasubramanian,
Professor,
Department of Computer Science and Engineering,
Mepco Schlenk Engineering College,
Sivakasi.

R. Roopa Anbu Malar,
PG Scholar,
Department of Computer Science and Engineering,
Mepco Schlenk Engineering College,
Sivakasi.

ABSTRACT

Many issues concern with secure multicasting are confidentiality, authentication, non-repudiation and data integrity including access control. Group-oriented applications such as video-conferencing, broadcasting stock quotes, software distribution or audio/video transmission are made possible through multicasting as an efficient communication mechanism. The deployment of these kinds of efficient communication mechanism is hindered because of lack of security. These limitations kindle the research minds to contribute towards secure multicasting. Hash Tree Signature Scheme (HTSS) is a newly proposed mechanism for multicast authentication that aims at providing packet authentication along with data integrity, non-repudiation and protection against key exposure. The scheme follows asymmetric cryptographic approach using tree-chaining technique that implements the tree construction for key generation and signature amortization for secure packet transmission. Performance evaluation is based on signing rate, providing non-repudiation and protection against key exposure. The HTSS is proposed in four different modes and its compatibility with the different issues in multicasting is discussed. The different modes discussed are sign-each, fixed delay, continuous and dynamic mode.

Keywords

Forward security, hash-tree, multicast authentication, network security, signature amortization.

1. INTRODUCTION

An efficient communication mechanism, for the group-oriented applications which includes video conferencing, broad-casting stock quotes, group games and video on demand, is through multicasting. The strength of this communication model is the group address by which the groups are identified and the nodes on the network can be dynamic i.e. May join or leave the network freely. Though multicasting promotes efficient communication it is also vulnerable to many attacks and remains a challenging task for researchers. A minimum security level [19] is always a basic consideration which should be the compulsory requirement for a multicast communication model.

Some of the basic issues that are to be considered during multicasting, providing with the security level that are minimum, are data integrity, data origin authentication and non-repudiation. Data integrity is concern with the correctness of data packets, whether they are modified or not. Data origin authentication is where the receiver assures that the data is from the claimed sender. Non-repudiation is a property which is able to prove itself which is the sender to the third party.

There are numerous kinds of symmetric and asymmetric cryptographic authentication mechanisms proposed to achieve secure multicasting. Though the previous works address most of the issues concern with secure multicasting, there are considerable pit falls which makes the multicasting channel unsecure and vulnerable.

The Hash Tree Signature Scheme (HTSS) is the proposed technique for secure multicasting. The scheme uses a tree constructed for keys and the approach using the asymmetric keys for the flow signing. It also exploits the Digital Signature Algorithm in order to provide secure transmission of data along the insecure network.

2. PREVIOUS WORKS

"Secure multicasting", one of the major issues in efficient multicasting, is achieved through many proposed schemes involving both symmetric and asymmetric cryptographic mechanism. It is also explained as MAC based protocols and digital signature based protocols which correspond to using one secret key and pair of keys for authentication mechanism. The MAC based protocol that uses a secret key that both the sender and the receiver share is the simplest method providing a better performance in authentication but does not provide with source authentication and non-repudiation. Alternatively the asymmetric approach that uses two keys namely the public key and private key aim at attaining security concerning data integrity, source authentication and non-repudiation at a high computational cost.

One of the schemes proposed for multicasting is TESLA [3], [4], which implements MAC based protocol [7], a symmetric cryptographic approach, to provide with the authentication. The mechanism is proved to be computationally efficient yet it does not provide with the non-repudiation which is one of the minimum security issue that should be provided with in a multicast communication.

There are many mechanisms which uses the asymmetric cryptographic mechanism for secure multicasting. The technique using a tree structure resulting in links like a chain [5] is an implementation of digital signature on many packets, having a combined signature rather than being signed individually. Though quite efficient it involves buffering of data at the sender as well as at the receiver.

The other asymmetric cryptographic scheme implementing similar technique as symmetric cryptography uses short keys for signing packets; the scheme is explained in DiffSig [6]. This technique provides authenticity for a short period of time. It also involves time synchronized flow which concludes that the packet becomes obsolete when it arrives after the stipulated time within which it should arrive though the packet is cryptographically correct.

There are many research works proposing schemes to deal with the packet loss and the security measures [8] in multicasting the data packets. Packet loss is one major issue in multicasting [11] and is also the minimum requirement in distribution of larger amount of data in smaller packets.

The forward security [13] is one property of the secure network. Work on protocols to provide with forward security with less computational overhead is a very challenging task. It mainly aims at protecting the keys. Forward security both during signing and verification [14] is an important issue to be taken care of during multicasting.

At present there are various challenging issues that are to be considered while multicasting which involves the scheme should be computationally efficient with low delay and also should address various complex attacks. It is not an easy task to make multicasting most secure without taking into account all worst cases of attacks and the efficiency in performances. The scheme should additionally involve protecting the keys from being exposed along with the data integrity, data origin authentication and non-repudiation.

3. NETWORK MODEL

Network for the proposed scheme is considered to be an insecure network involving one sender (S) and many receivers (R). There may be intermediate modes between the sender and the receiver that forwards the packets but do not assure any security. The packets may get compromised, dropped or get modified during the transmission. It is assumed that the packets which are generated at the sender are not known to the sender until the packet is ready for reception. It is also assumed that those packets generated are sent at slow rate that is supported by both sender and receiver.

The sender and receiver are time synchronized and buffering of data is found at sender and receiver end.

In this assumed network the multicasting is carried out following a multicast authentication based on asymmetric mechanism that aim at providing data integrity, data origin authentication, non-repudiation along with protection against key exposure.

The tree-chaining technique implemented [1] for key generation aim at protecting against the key exposure but the amortized signature of the packet [2] protects the data packets from being compromised.

4. PROPOSED WORK

The proposed work, the Hash Tree Signature Scheme (HTSS), aims at addressing all the minimum security issues concern with multicasting. It provides multicast authentication and makes it computationally efficient. The techniques also provides source authentication and integrity along with non-repudiation to allow a recipient prove the source to a third party.

The scheme implements asymmetric keys on tree-chaining technique. An amortized signing procedure on packets in order to make it more secure that protects the correctness of the message packet. The HTSS involve in providing non-repudiation for a longer period of time.

The HTSS is also explained under three different modes of operation. It includes (1) Sign-each mode, (2) Continuous mode, (3) Fixed delay mode and (4) Dynamic mode. The sign each mode involves a block containing only one packet and each packet is signed individually and thus verified individually at the receiver. The continuous mode is where a block may contain certain number of packets. The packets when generated are immediately signed and sent. The fixed delay mode is where the packets generated are signed and sent in a particular interval of time. Thus there is a fixed delay in sending and receiving the packets. The dynamic mode generalizes the continuous and fixed delay mode, includes constrains but nevertheless providing a better performance.

The scheme is evaluated based on the signing rate involving different key lengths used for various message sizes. The mechanism also involves different asymmetric cryptographic algorithms and hashing techniques based on which the signing rate is determined. These evaluations on the performance of the scheme are carried out for various proposed scheme and their efficiency is measured.

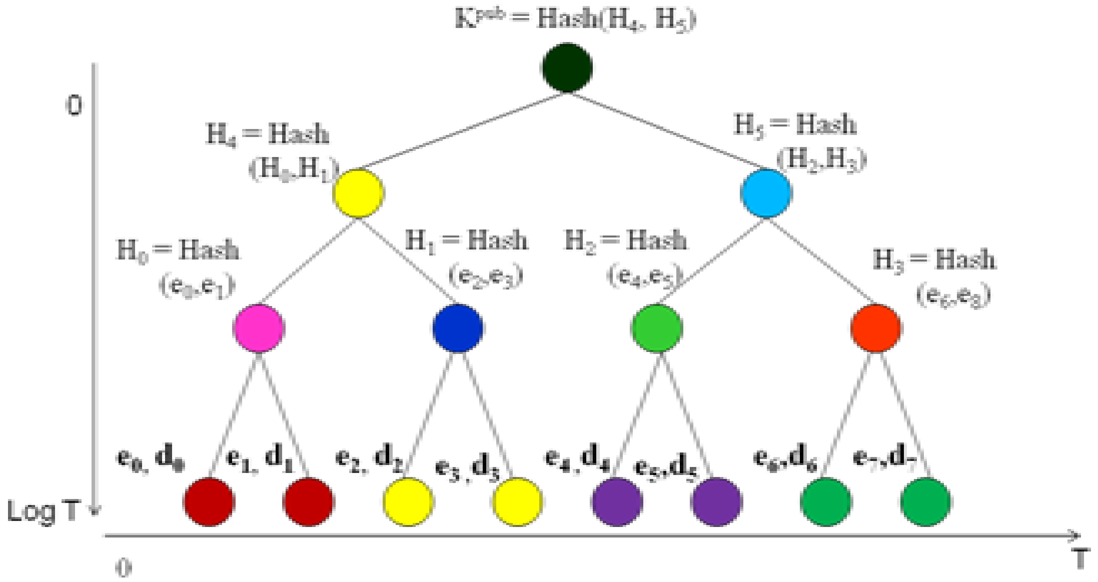
K^{pub} = Hash(H_4, H_5)
0
H_4 = Hash (H_0,H_1)
H_5 = Hash (H_2,H_3)
H_0 = Hash (e_0,e_1)
H_1 = Hash (e_2,e_3)
H_2 = Hash (e_4,e_5)
H_3 = Hash (e_6,e_8)
e_0, d_0
e_1, d_1
e_2, d_2
e_3 , d_3
e_4 , d_4
e_5,d_5
e_6,d_6
e_7,d_7
Log T
0
T

Fig 1: Structure of the tree-construction called hash tree, for key generation.

5. HASH TREE SIGNATURE SCHEME (HTSS)

The Hash Tree Signature Scheme implements tree-chaining technique along with signature amortization. It involves the following steps:

(1) Tree construction for key generation.
(2) Packet encryption and signing with active keys.
(3) Signature amortization.
(4) Verification at the signer.

5.1 Tree construction for key generation

The scheme considers a time period t € {0, 1 … T}. Each and every time period is associated with a public key (e_i) and a private key (d_i) where i = t € {0, 1… T}. The keys are active for the particular time period and the packet generated at a particular time period is encrypted with the active private key and signed with the hash of active public keys. The scheme follows tree construction mechanism [15] for key generation.

5.2 Packet encryption and signing with the active keys

Signing is a simple procedure providing with forward security [13] and protecting the message packet [18]. During the packet signing the packet is first encrypted with the active key and the signing is done with the hash of the public key [10]. The keys and the hash value for signing the packets vary with time. The packet is signed and sent to the receiver for verification.

Considering the second time period, T = 2, the signature of the message M_2 in the corresponding time period is given as:

Sig (M_2) = (E (d_2, M_2), 2, e_2, e_1, e_0, H_1, H_2, K^{pub})

Parameter description:

E (d_2, M_2) – Encrypting the message with the active private key.

2 – The second time period.

e_2 – public for time period t_2.

e_1 – public for time period t_1.

e_0 – public for time period t_0.

H_1 – Hash of the public keys in time period t_0 and $t_{1.}$

H_2 – Hash of the public key in time period $t_{2.}$

$K^{pub\,-}$ Hash of H_1 and $H_{2.}$

These are sent to the receiver where the message is decrypted and verification is made for public key exposure.

5.3 Signature amortization

Signature amortization can provide resilience to packet loss [12], [17]. It is a technique where each packet is signed individually and a hash of a message, which is combined form of all message packets, is used to sign all the combined signed packets.

5.4 Verification at the receiver

Verification at the receiver is very simple. The data received are checked if they arrive in time, then the correctness of the message and the public key exposure is checked. They are done by determining the hashes separately as in the sender and comparing the result with the received hash. If the comparison remains true i.e. if hash values are found to be same with no difference then the message is proved correct with zero percent public key exposure. If the hash value generated at the receiver is different from the received value then the packet is concluded to be vulnerable and is completely ignored.

6. MODES: HASH TREE SIGNATURE SCHEME (HTSS)

The Hash Tree Signature Scheme (HTSS) is explained in four different modes:

(1) The Sign-each mode
(2) The continuous mode
(3) The fixed delay mode
(4) The dynamic mode

6.1 The Sign-Each mode

In sign-each mode there is only one packet in the block. Each packet is signed separately with the active public key. There is no amortization of signature in sign-each mode. The packets are sent to the receiver one by one and the verifications are done accordingly.

6.2 The Continuous mode

The continuous mode is where a block is assumed contain differing packets for differed period of time. The packets as and when they get generated it is grouped, encrypted, signed and sent. The amortized signature is also sent for checking the correctness of the message. The receiver on obtaining the message checks for packet loss, key exposure and correctness in the message.

6.3 The Fixed-Delay mode

The fixed delay mode is where the packets are buffered at the sender and are sent in a fixed delay fashion. The sender and receiver being time synchronized waits for the fixed delay to send and receive packets. On reception of the packets they are concluded to be secured if they reach in time else they are found malicious and are ignored.

6.4 The Dynamic mode

The dynamic mode generalizes both the continuous and the fixed delay mode. The packet generation is constrained with both the time and number of packets being generated. Thus here a ratio is maintained between the time taken in seconds and the number of packets generated. The conditions are checked for the threshold values. When either of the values is satisfied the packets are combined, signed and sent. The ratio of time taken in seconds to packets generated is considered as α , where $10 \le \alpha \le 100$ is the threshold range. In the implementation the results are obtained using 10 as the threshold value.

7. RESULTS AND DISCUSSION

The scheme implements RSA digital signature which is an asymmetric cryptographic approach for encrypting and decrypting the message packet while the SHA1 algorithm for the determination of hash [16], [20].

The evaluation is based on the signing rate calculated for different message size and key lengths. These are the evaluation graph which represents the performances of different schemes. The study is thus made using the signing rate.

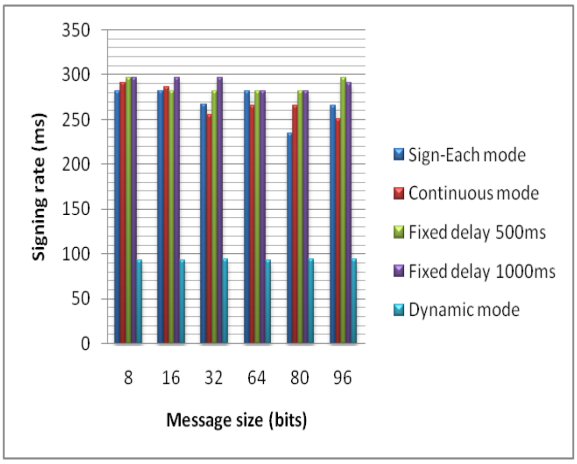

Fig 2: Signing rate: Using 16 bit key, message size in x-axis and time for signing in milliseconds in y-axis.

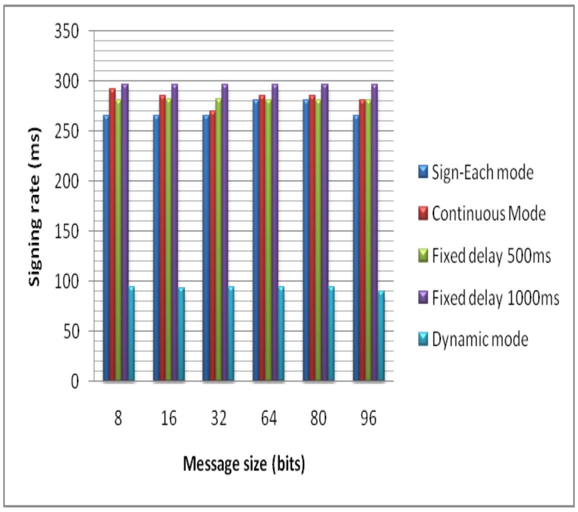

Fig 3: Signing rate: Using 32 bit key, message size in x-axis and time for signing in milliseconds in y-axis.

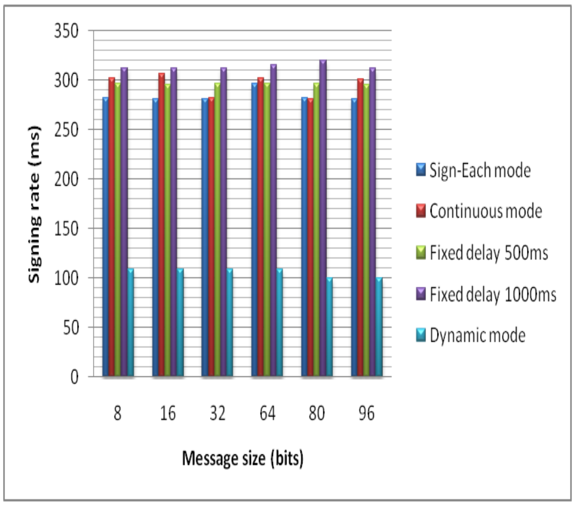

Fig 4: Signing rate: Using 64 bit key, message size in x-axis and time for signing in milliseconds in y-axis.

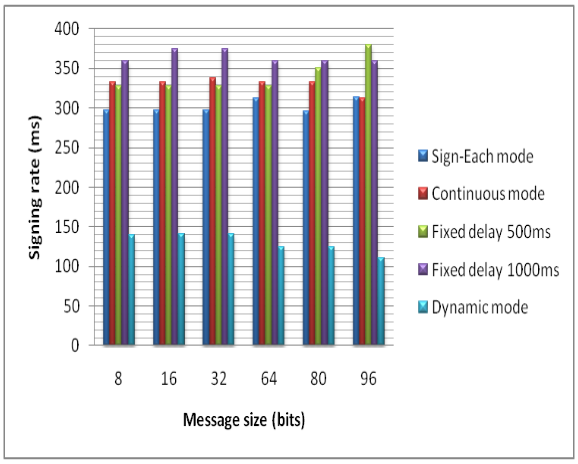

Fig 5: Signing rate: Using 128 bit key, message size in x-axis and time for signing in milliseconds in y-axis.

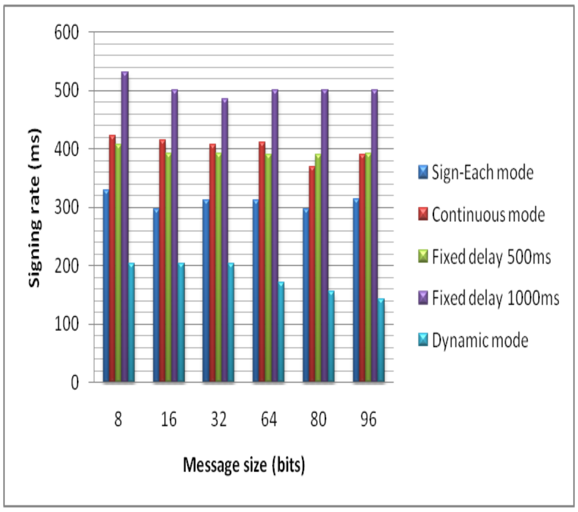

Fig 6: Signing rate: Using 256 bit key, message size in x-axis and time for signing in milliseconds in y-axis.

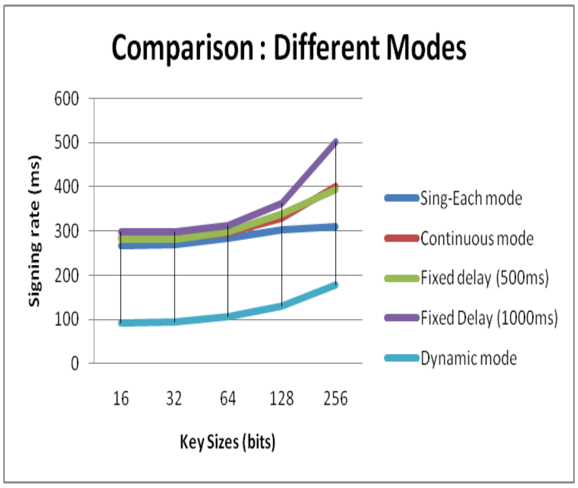

Fig 7: The four different modes are compared based on the key sizes.

The evaluation is bounded to the four different modes of operations. The HTSS is formulated for four different modes and the performance is evaluated based on the signing rate.

The performance graph is plotted from which the solution is derived for these modes.

The sign-each mode suites best while the signatory uses the larger bit sized key and when the messages are longer. Generally these messages increase the computational overhead when they are multicoated following any other modes of HTSS.

The continuous mode works with smaller messages signed using smaller size keys. This is where the short messages can be sent continuously without any time delay and are found more secure satisfying all the minimum requirements of secure multicasting.

The fixed delay mode suites the burst traffic of packets. There is a fixed delay in multicasting the burst data that manages the packet loss problem. The computational overhead is found higher as the message size increases and it becomes still higher when the key size is increased for the security purpose.

The dynamic mode generalizes both the continuous and fixed delay mode. The continuous mode does not support longer messages and longer key sizes while the fixed delay mode is where, there is always a time delay in multicasting the packets. The dynamic mode deals with these two issues and contributes best results supporting any key size and message size yet the signing rate remains low with very less time delay.

8. CONCLUSION

The proposed scheme “Hash Tree Signature Scheme (HTSS)” aims at providing the all basic security requirements for multicasting along with the method to protect the secret key. The hash of the public key considered as the signature for packets make it more secure and is difficult to attack while the amortized signature for the encrypted packet is from the binary tree constructed for the public keys of the particular time period.

From the evaluation concludes that in continuous mode as the packets has to be grouped as and when generated it suffers extra process overhead, similarly in fixed delay mode the system incurs extra processing in cases of very less and very high number of messages generated. But the dynamic mode combines the advantages of both fixed delay and continuous mode. Thus the performance also is verified to be higher compared to all other mode.

9. ACKNOWLEDGMENTS

The authors are grateful to the Management and the Principal of Mepco Schlenk Engineering College, Sivakasi, for their constant support and encouragement.

10. REFERENCES

[1] Diana Berbecaru, Luca Albertalli, and Antonio Lioy, “The Forward DiffSig Scheme for Multicast Authentication,” in IEEE/ACM transaction on networking, Vol 18, No.6, December 2010.

[2] Yun Zhou, Xiaoyan Zhu, and Yuguang Fang, “MABS : Multicast Authentication Based on Batch Signature,” in IEEE transactions on Mobile Computing, Vol. 9, No. 7, July 2010.

[3] A. Perrig, R. Canetti, D. Song, and J. Tygar, “Efficient and secure source authentication for multicast,” in Proc. NDSS, 2001, pp. 35–46.

[4] A. Perrig, D. Song, R. Canetti, J. D. Tygar, and B. Briscoe, “Timed efficient stream loss-tolerant authentication (TESLA): Multicast source authentication transform introduction,” RFC 4082, June 2005.

[5] 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.

[6] N. Kang and C. Ruland, “DiffSig: Differentiated digital signature for real-time multicast packet flows,” in Proc. Trust Privacy Digital Business, 2004, LNCS 3184, pp. 251–260.

[7] A. Perrig, J. Tygar, D. Song, and R. Canetti, “Efficient authentication and signing of multicast streams over lossy channels,” in Proc. IEEE Security Privacy, 2000, pp. 56–63.

[8] P. Golle and N. Modadugu, “Authenticating streamed data in the presence of random packet loss,” in Proc. NDSS, 2001, pp. 13–22.

[9] S. Miner and J. Staddon, “Graph-based authentication of digital streams,” in Proc. IEEE Security Privacy, 2001, pp. 232–246.

[10] R. Gennaro and P. Rohatgi, “How to sign digital streams,” in Proc. Crypto, 1997, LNCS 1294, pp. 180–197.

[11] A. Pannetrat and R. Molva, “Efficient multicast packet authentication,” in Proc. NDSS, 2003.

[12] J. M. Park, E. Chong, and H. Siegel, “Efficient multicast packet authentication using signature amortization,” in Proc. IEEE Security Privacy, 2002, pp. 227–240.

[13] M. Bellare and S. Miner, “A forward-secure digital signature scheme,” in Proc. Crypto, 1999, LNCS 1666, pp. 431–448.

[14] G. Itkis and L. Reyzin, “Forward-secure signatures with optimal signing and verifying,” in Proc. Crypto, 2001, LNCS 2139, pp. 332–354.

[15] M. Szydlo, “Merkle tree traversal in log space and time,” 2003 [Online].Available: http : / / www.szydlo.com / logspacetime03.pdf, preprint version.

[16] B. Weis, “The use of RSA/SHA-1 signatures within encapsulating security payload (ESP) and authentication header (AH),” RFC 4359, Jan.2006.

[17] M. Yajnik, J. Kurose, and D. Towsley, “Packet loss correlation in the Mbone multicast network,” IEEE Global Internet Conference, Nov. 1996.

[18] A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.[19] R. Merkle, Secrecy, Authentication and Public Key Systems. Ann Arbor, MI: UMI Research Press, 1982, also appears as a Stratford University Ph.D. dissertation in 1979.

[19] Secure Hash Standard (SHS), NIST FIPS 180-2, 2004.