An Efficient Signature Scheme for Securing Multicast Communication

R.Krishnaveni
National Engineering College
Kovilpatti

N.Shanthi
National Engineering College
Kovilpatti

ABSTRACT
Multicasting is an efficient communication mechanism for group oriented applications such as video conferencing, distance learning etc. Assuring certain level of security over multicast communication has become vital in Internet applications. Various cryptographic algorithms have been used to provide different security services. One among the cryptographic scheme is the digital signature which ensures authentication, message integrity and sender non repudiation. Digital signature algorithm such as RSA/DSA with SHA-1 can be used. In order to prevent passive attack, confidentiality has to be ensured. Confidentiality can be achieved by using the symmetric key encryption approach. The main objective of this paper is to provide all the above security services with low communication overhead. RSA with SHA-1 is one of the efficient public key cryptographic algorithms to provide security and therefore, its efficient hardware implementation is of great importance. This paper also proposes the design of SHA-1 implementation in FPGA.

Keywords
Authentication, client, confidentiality, encryption, server, verification

1. INTRODUCTION
Wireless networks have become increasingly popular in the communication industry. In emergency applications such as rescue and mission and military applications, there is a need to deploy a wireless network that can be formed instantly. Ad-hoc network is one such type of network. Ad-hoc networks are autonomous networks comprised of wireless mobile devices. Exchange of sensitive information over access paid services and unprotected wireless links demand the deployment of security in wireless networks. In recent years, secure communication has become an important subject of research. The main security service for wireless network is to provide confidentiality, authentication, authorization and data integrity. Most of the group oriented applications are based on multicast communication. Multicasting is an efficient method to deliver data from a sender to a group of receivers.

Secure data over multicast communication is a challenging task. In this paper, the problem of secure multicast of data streams over wireless ad hoc network is addressed.

This paper deals with the schemes for securing multicast communication and its implementation. In this paper, issues in multicast security and security services and the multicast scenario are briefly explained in section no.2. Algorithms used in the signature scheme are presented in section no.3. The basic digital signature scheme is discussed in section no.4. In section no.5, the proposed scheme providing confidentiality is explained. The implementation of the multicast server and client and their results are given in section no.6.1, 6.2 and 6.3. The hardware implementation of SHA-1 is shown under subsection no.6.4. The pros and cons of the schemes are discussed in section no.6.5. Finally the paper is concluded in section no.7.

2. MULTICAST SECURITY

In the multicast communication, groups are identified by a group address (multicast address), and any node of the network may join or leave the group freely. So, membership in a multicast group is dynamic, allowing any hosts to enter and leave the multicast session without the permission or knowledge of other hosts. This inherent benefit of multicast communication presents some vulnerability making it susceptible to attacks unless they are secured.

In the multicast model, senders may not be the members of the multicast group. This means that any host can send data to the multicast group [1]. Further, group members need to be able to verify that messages received are from the intended source. Multicast source authentication solutions are needed to provide this functionality. Another security issue in multicasting is that data sent to the group may transit via many insecure channels. Thus, eavesdropping opportunities are abundant [2].

Multicast communication need to be secured against threats through the application of several fundamental security services [24] which are discussed below:

- Data integrity: Each receiver should be able to assure that received packets have not been modified during transmissions.
- Data origin authentication: Each receiver should be able to assure that each received packet comes from the real sender as it claims.
- Non-repudiation: The sender of a packet should not be able to deny sending the packet to receivers in case there is a dispute between the sender and receivers.
- Confidentiality: Data should not be made available to the unauthorized users.

The first three security services can be provided by the asymmetric encryption techniques called digital signature as in [3]. The sender generates a signature for each data with its private key (PR), which is called signing process, and each receiver checks the validity of the signature with the sender's public key (PU), which is called verification process. If the verification succeeds, the receiver knows that the data is authentic. Confidentiality can be achieved by using symmetric encryption technique.

3. ALGORITHM

3.1 SHA-1

SHA-1 (Secure Hash Algorithm) [19] has been widely used to provide message integrity. SHA-1 gets an input message smaller than 2^{64} bit and performs message padding by dividing the message into 512 bit blocks. Five 32-bit buffers are used which are initialized with the predefined value. These five buffers (160-bit) are used to hold the intermediate and final results of hash function. Each message block (x_i) is processed in four stages and each stage consists of 20 steps of operation as shown in Figure 1 [20]. The algorithm computes a 32-bit word W_0, W_1... W_{79} for each of the 80 steps from the message block itself.

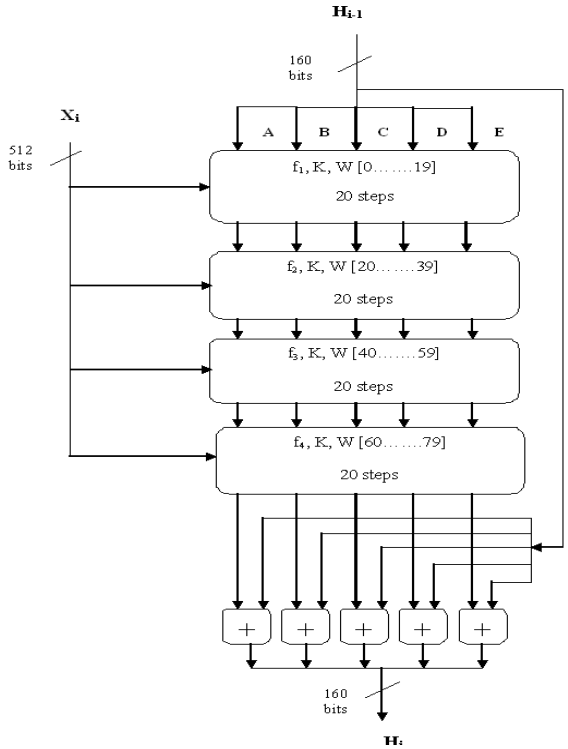

Fig 1: SHA-1 Compression function for a single 512-bit block

The words W_j are derived from the 512-bit message block as given in Equation (1),

$$W_j = \begin{cases} x_i^{(j)}, 0 \le j \le 15 \\ \left(W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}\right) <<< 1, 16 \le j \le 79 \end{cases} \quad (1)$$

where x<<<n indicates a circular left shift of the word x by n bit positions and j denotes the current step of operation.

The four SHA-1 stages have a similar structure but use different internal functions f_t and constants k_t, where $1 \le t \le 4$. Message blocks are processed by the function together with some stage dependent k_t. The output after 80 stages is obtained by adding to the input value H_{i-1} modulo 2^{32} in word-wise fashion. The operation within step j in stage t is given by Equation (2),

$$\begin{aligned} A &\leftarrow E + f_t(B,C,D) + (A <<< 5) + W_j + K_t \\ B &\leftarrow A, \\ C &\leftarrow B <<< 30 \\ D &\leftarrow C \\ E &\leftarrow D \end{aligned} \quad (2)$$

Table 1 shows the definition of f_t in each step. The logical operators (AND, OR, NOT, XOR) are represented by the symbols (^, v, ¯, $\oplus$) respectively.

Table 1: Function value for f_t

Step	Function Name	Function value
0≤t≤19	f_1(B,C,D)	$(B \wedge C) \vee (\bar{B} \wedge D)$
20≤t≤39	f_2(B,C,D)	$B \oplus C \oplus D$
40≤t≤59	f_3(B,C,D)	$(B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
60≤t≤79	f_4(B,C,D)	$B \oplus C \oplus D$

3.2 RSA

RSA is a very popular asymmetric key cryptographic algorithm used in many security protocols [14] [20]. In order to use RSA, a sender chooses two large random primes p and q to get $N = p \times q$, and then calculates two exponents e and d such that $e \times d = 1 \bmod \Phi(N)$, where Φ (N) = (p-1) (q-1). The sender publishes (e, N) as its public key and keeps d in secret as its private key. A signature of a message M can be generated as $\sigma = (h(m))^d \bmod N$, where h(.) is a collision-resistant hash function. The sender sends {M, σ} to a receiver that can verify the authenticity of the message by checking $\sigma^e = h(m) \bmod N$ .The strength of the algorithm is based on the difficulty of factoring numbers into prime factors.

4. BASIC SCHEME

The target is to authenticate multicast streams from a sender to multiple receivers. The sender signs each packet with a signature and transmits it to multiple receivers. Each receiver needs to assure that the received packets are really from the sender (authenticity) and the sender cannot deny the signing operation (non-repudiation) by verifying the corresponding signatures.

The Sender (A) generates the message (M) which is given as input to the hash algorithm (H) such as SHA-1. The output message digest value is encrypted using the private key of the sender and now the encrypted output serves as a digital signature. The signature is concatenated with the original message and it is sent to the receiver. The receiver (B), on obtaining the signature and message, perform the verification process. Receiver calculates the message digest value for the received message and decrypts the signature using the public key of the sender. Then, receiver compares the hash value and

decrypted output as shown in Figure 2. If both match, the receiver knows that the message has not been modified during transmission (data integrity) and message is authentic.

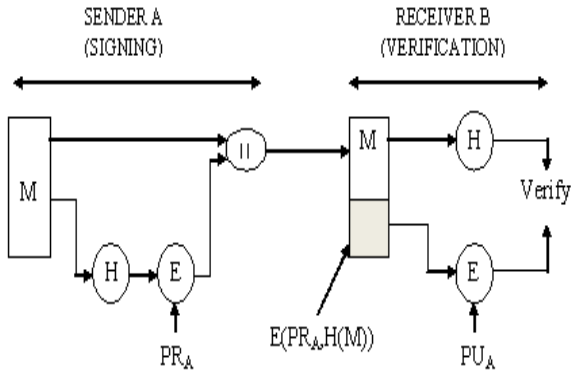

Fig 2: Digital signature signing and verification process

This scheme [14] provides digital signature because only the sender could have produced the encrypted hash code. Public key encryption algorithm such as RSA, DSA can be used for signing and verification process.

5. PROPOSED SCHEME

The above digital signature scheme does not provide confidentiality since the message (M) is transmitted in clear. If confidentiality as well as digital signature is desired, then the message plus the private-key-encrypted hash code can again be encrypted using a symmetric secret key [20]. Then, encrypted message is transmitted to the receiver. The receiver decrypts the received message using the shared secret key (K). The decrypted message is then verified using the public key (PU) of the sender as shown in Figure 3.

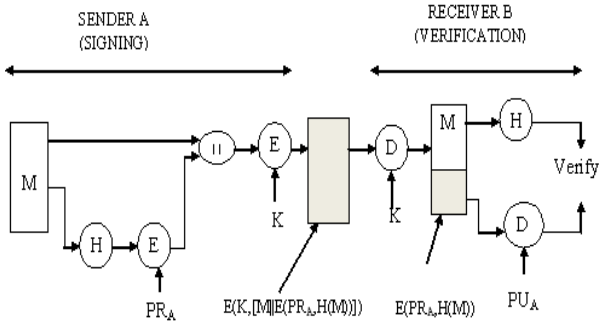

Fig 3: Scheme providing authentication and confidentiality

The secret key can be produced using DES algorithm. For DES, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output.

6. IMPLEMENTATION

In order to prevent unauthorized users from accessing the data, the server signs the data using digital signature scheme such as RSA with SHA-1 before transmission. The client can verify the authenticity of the message by using the verification process. Key size of 1024 bits is used. The scheme is implemented in Java [18], [21]. One server and four clients are used for implementation. Implementation of basic scheme and proposed scheme are given in results-I and II subsection. The simulation of SHA-1 [15] algorithm is shown in result – III subsection.

6.1 Multicast Server and Client

When a host wants to send data to a multicast group, it puts that data in multicast datagram [23], which is a UDP datagram addressed to a multicast group. Multicast data is sent via UDP, which, though unreliable, can be as much as three times faster than data sent via connection-oriented TCP. Once the data has been buffered out and packetized, the sending host launches the datagram onto the Internet. If any client wants to receive data from a particular multicast group, it should join that multicast group. The receiving host must also be listening on the proper port and be ready to process the datagram when it arrives.

6.2 Results-I

The server generates the RSA key pair and calculates the digital signature and then packetizes the digital signature and sent to the client as shown in Figure 4. The reception of datagram packet by the multicast client is explained in Fig.5. The client whoever joins that particular multicast group can be able to receive the datagram packet. The client retrieves the public key of the server from the publicly available file and then verifies the signature.

Fig 4: Multicast server

Fig 5: Multicast client

6.3 Results-II

To provide secure transmission in multicast communication, a new scheme is proposed which encrypts the signature before transmission and decrypts the encrypted signature after reception. The multicast server generates the RSA key pair and DES secret key and encrypts the digital signature using the secret key which is observed in Figure 6. Then the encrypted data is packetized and transmitted to the requested client. On receiving the datagram packet, multicast clients decrypts the received data and then verifies the decrypted data using the public key of the sender as inferred from Figure 7.

Fig 6: Multicast server

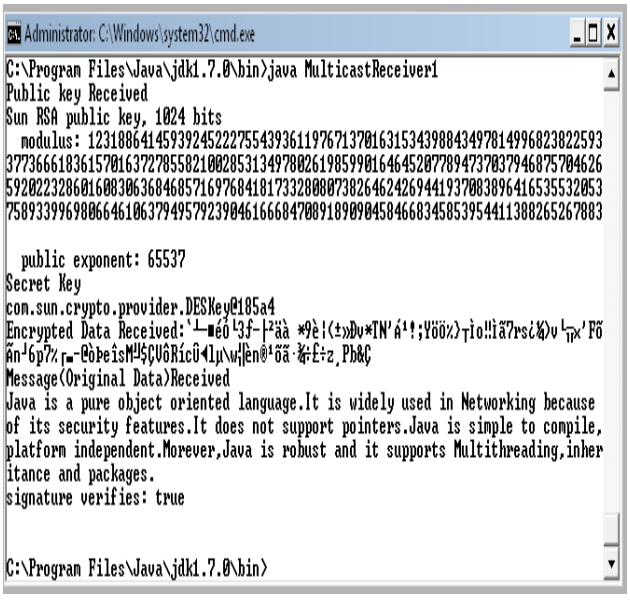

Fig 7: Multicast Client

6.4 Results-III

The module for SHA-1 is written in verilog and simulated using Active HDL. The timing waveform obtained during simulation is shown in Figure 8.The module is synthesized using Quartus-II software and the compilation report is given in Figure 9.

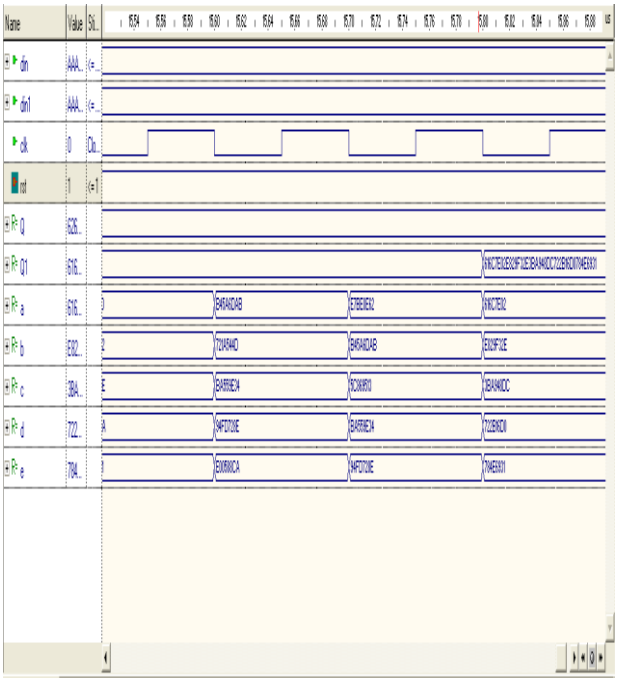

Fig 8: HDL simulation result for SHA-1

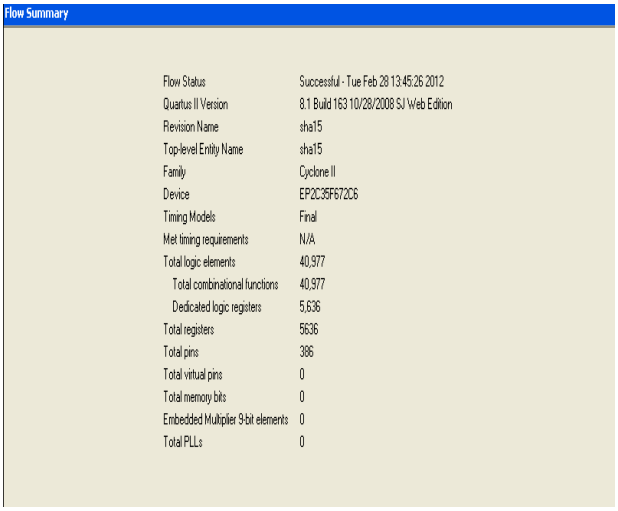

Fig 9: Compilation report for SHA-1 module

6.5 Discussion

Here server is one of the members of a particular multicast group and uses RSA with SHA-1 algorithm to provide the desired security services. Since SHA-1 produces a fixed output of 160 bits, communication overhead is reduced. Since there is no relationship among packets and each packet can be independently verifiable, this scheme reduces authentication latency. But the sender needs to sign each packet which costs more computation overhead. However the sender is usually a

powerful server and so per packet signature generation can be affordable.

In hardware implementation of SHA-1, one clock cycle is spent for each step of operation. For each block of data, hash value is obtained at the end of every 80^{th} clock cycle.

7. CONCLUSION

Multicast enables efficient large-scale content distribution by providing an efficient transport mechanism for one-to-many and many-to-many communication. The properties that make multicast attractive, however, also make it a challenging environment to support Internet-based applications. To solve the security issue in multicast environment, various encryption techniques has been employed. In this paper, we have presented some security schemes for multicast communication. Secret key management issue has yet to be considered. Hardware implementation of RSA is yet to be implemented in future.

8. REFERENCES

[1] Judge, P., and Ammar, M. 2003. Security Issues and Solutions in Mulicast Content Distribution: A Survey, *IEEE Network Magazine*: 30-36.

[2] Challal, Y., Bettahar, H., and Bouabdallah, A. 2004. A Taxonomy of Multicast Data Origin Authentication: Issues and Solutions, *IEEE Comm. Surveys & Tutorials*: 34-57.

[3] Zhou, Y., and Fang, Y. 2007. Multimedia Broadcast Authentication Based on Batch Signature, *IEEE Comm. Magazine*: 72-77.

[4] Even, S., Goldreich, O., and Micali, S. On-Line/Offline Digital Signatures, *J. Cryptology*: 35-67, 1996.

[5] Perrig, A., Canetti, R., Tygar, J.D., and Song, D. 2000. Efficient Authentication and Signing of Multicast Streams over Lossy Channels, *Proc. IEEE Symp. Security and Privacy (SP '00)*: 56-75.

[6] Pannetrat, A., and Molva, R. 2002 . Authenticating Real Time Packet Streams and Multicasts, *Proc. Seventh IEEE Int'l Symp. Computers and Comm. (ISCC '02)*: 490-495.

[7] Pannetrat, A., and Molva, R. 2003. Efficient Multicast Packet Authentication, *Proc. 10th Ann. Network and Distributed System Security Symp. (NDSS '03)*.

[8] Rivest, R.L., Shamir, A., and Adleman, L. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*: 120-126.

[9] Harn, L. 1998. Batch Verifying Multiple RSA Digital Signatures, *IEE Electronic Letters*: 1219-1220.

[10] FIPS PUB 186. 1994. Digital Signature Standard (DSS).

[11] Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., and Pinkas,B. 1999. Multicast Security: A Taxonomy and Some Efficient Constructions, *Proc. IEEE INFOCOM*: 708-716.

[12] Perrig, A., Canetti, R., Song, D., and Tygar, J. 2001. Efficient and Secure Source Authentication for Multicast, *Proc. Network and Distributed System Security Symp. (NDSS '01)*.

[13] Merkle, R. 1980. Protocols for Public Key Cryptosystems, *Proc. IEEE Symp. Security and Privacy*.

[14] Zhou, Y., Zhu., X and Fang, Y. 2010. MABS: Multicast authentication based on batch Signature, *IEEE Trans. Mobile Computing*: 982-993.

[15] Hui, C.X., Zhi, D.J. 2010. Design of SHA-1 algorithm based om FPGA, *Proc. International Conference on Networks Security, Wireless Communications and Trusted Computing.*

[16] Harold, E.R . 2004. Java Network Programming, Third edition, O'Reilly publications.

[17] Harte, L. 2008. Introduction to Data Multicasting, Althos publications.

[18] Jaworski, J., and Perrone, P. 2000. Java Security Handbook, Sams Publications.

[19] Paar, C., Pelzl, J. 2009. Understanding Cryptography, Springer publications.

[20] Stallings, W. 2005. Cryptography and Network Security Principles and Practices, Fourth edition, Prentice hall publications.

[21] Pitt, E. 2006. Fundamental Networking in Java , Springer publications.