Message Integrity in the World Wide Web: Use of Nested Hash Function and a Fast Stream Cipher

Sheena Mathewa, K. Paulose Jacoba

aDepartment of Computer Science, Cochin University of Science and Technology, Kochi, Kerala, India

The focus of this work is to provide authentication and confidentiality of messages in a swift and cost effective manner to suit the fast growing Internet applications. A nested hash function with lower computational and storage demands is designed with a view to providing authentication as also to encrypt the message as well as the hash code using a fast stream cipher MAJE4 with a variable key size of 128-bit or 256-bit for achieving confidentiality. Both nested Hash function and MAJE4 stream cipher algorithm use primitive computational operators commonly found in microprocessors; this makes the method simple and fast to implement both in hardware and software. Since the memory requirement is less, it can be used for handheld devices for security purposes.

1. INTRODUCTION

Due to the prospering use of Internet applications like e-commerce, ensuring confidentiality, integrity and authenticity of information have acquired increased importance [1]. When two parties are communicating over an insecure channel, they need a method by which the information sent by the sender can be accepted as confidential, unmodified and authentic by the receiver. The confidentiality of the message can be achieved by encrypting the message using the symmetric key algorithms in cryptography, which are faster and efficient than asymmetric key algorithms. The integrity of the message can be verified by hash functions. Hash function is a function which involves all the bits of the message. It accepts a variable size message as input and produces a fixed size output as the hash code. A change in any bit or bits in the message results in change in the hash code [2] thus providing an indication of message tampering. When A sends a message to B, it appends the hash code to the message, which is computed using the hash function. After receiving the message B re-computes the hash code using the same hash function and compares with the original hash code. If both are same then B can assure that the message has started off from the intended sender and it has not been tampered with, during the transmission.

Common uses of Hash functions include authorization of financial transactions, mobile communications (GSM and GSPP) and authentication of Internet communications with SSL/TLS and IPSec. It is also used as a pseudo random function. Hash function provides a deterministic mechanism for generating random seeming bit streams from some input source without disclosing any information about the input. It can also be used to give protection against viruses. Viruses typically modify the host files that they infect, and hence one way of virus detection involves checking files for signs of unauthorized modification by computing the authentication tags from each file [3].

As shown in Figure 1, sender A uses the nested hash function to compute the hash code $H(M)$ of the message $M$ and appends it to the message $M$. Using the 128-bit key $K$ and the fast stream cipher algorithm MAJE4, the message and the hash code are encrypted as $E_K[M || H(M)]$ and sent to receiver B. Using the same key $K$ and the fast stream cipher algorithm MAJE4, the cipher text is decrypted back to produce the message and the hash code. Now B re-computes the hash code of the received message using the
upto about 135 kilobytes then the time taken for
producing the hash code is negligible. For large
messages the additional time requirement is very
less as shown in Figure 3. The memory size re-
quired for executable code for nested hash code is
5899 bytes and for MAJE4 it is 6254 bytes. Hence
a total of nearly 12 Kilobytes memory is enough
for providing both authentication and confidential-
ty of messages.

7. CONCLUSION

From the analysis of results it is concluded that
message authentication can be achieved with con-
fidentiality of message by using a very small in-
crease in time even for too large messages. The
time required for messages with a memory of upto
135 Kbytes is found negligible. Also the addi-
tional memory size needed for implementing the
hash function is only 5899 bytes. Because of the
low memory requirement, this hash code can be
very well used in handheld devices like mobile
phones, personal digital assistants (PDA), etc.
for authentication purposes. Since it is faster,
it can be used for applications that require mes-
sage integrity as well as encryption / decryption
of stream of data sent through the Internet. As
advanced cryptography becomes easier to imple-
ment and manage, more companies and organi-
zations can take advantage of these benefits.

REFERENCES

1. Mihir Bellare, Ran Canetti, Hugo Krawczyk.
Message Authentication Using Hash Func-
tions. The MAC Construction, RSA Labora-
tories CryptoBytes, 2(1), 1996.
2. Mihir Bellare, Ran Canetti, Hugo Krawczyk.
Keying Hash Functions for Message Authentic-
tion, Advances in Cryptology. Proceedings
of Crypto 96, 1109(1996).
considerations for Incremental Hash Func-
tions based on Pair Block Chaining. Journal
4. William Stallings. Cryptography and Net-
work Security: Principles and Practices. Prentice Hall,
2003.
5. Ivan Damgard. A Design Principle for
Hash Functions. Advances in Cryptology -
6. R. Merkle. One Way Hash Functions and
DES. Advances in Cryptology - CRYPTO 89,
7. Mihir Bellare, Roch Guerin, Phillip Rog-
away. XOR MACs: New Methods for Message
Authentication using Finite Pseudorandom
Functions. Advances in Cryptology - Crypto
8. Sheena Mathew, K. Paulose Jacob. A New
Fast Stream Cipher: MAJE4. Proceedings of
9. D. E. Knuth. The Art of Computer Program-
mimg: Seminumerical Algorithms. Addison
Wesley, 1969.

Sheena Mathew, Reader in Cochin University
of Science and Technology has 14 years of teach-
ing experience in Computer Science. She had her
graduation from Madurai Kamaraj University
and post graduation from the Indian Institute of
Science, Bangalore. She is presently a research
scholar, her area of interest being Cryptography
and Network Security.
Dr. K. Poulouse Jacob, a National Merit Scholar all through, got his degree in Electrical Engineering in 1976 from University of Kerala, followed by his M.Tech. in Digital Electronics and Ph.D. in Computer Engineering, from the Cochin University of Science and Technology (CUSAT). He has been teaching at CUSAT since 1980 and currently occupies the position of Professor of Computer Science and Dean of Engineering. He has served as a Member of the Standing Committee of the UGC on Computer Education and Development. His research interests are in Information Systems Engineering, Intelligent Architectures and Networks.