IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Experimental Study of Message Digests on Mobile Hand Held Devices

Sachin Upadhye[*1], **Yogadhar Pandey**[2]

[*1,2]Sagar Institute of Research & Technology, Bhopal, India

sachupadhye@gmail.com

Abstract

Due to the continuous advancement in technology, mobile devices are playing important role in everyone's day to day life. Security is the one of the biggest concern in different type of networks. Due to diversify nature of network, security breaching became a common issue in different form of networks. Solutions for network security comes with concepts like cryptography in which distribution of keys have been done. If you want to send data to some other persons through network then if you truly want to keep the information secret, you need to agree on some sort of key that you and he can use to encode/decode messages. But you don't want to keep using the same key, or you will make it easier and easier for others to crack your cipher. So, Cryptographic message digest are a valuable tool in cryptography. They are applied in many areas of information security to provide protection of the authenticity of messages; data integrity verification which prevents modification of data from going undetected, time stamping and digital signature scheme

This paper presents a implementation and evaluation of message digest for mobile devices. This message digest system takes into account the device limitations and thus generates a functional digital signature. Its efficiency is described and justified along the paper. It also presents a comparison between MD5, MD4, MD2, SHA1, SHA256, Tigerdigest , RIPEMD 128, RIPEMD 256 etc. Message Digetst algorithm used in mobile devices.

Keywords: message digest mobile devices; cryptography algorithms; hash function, one way function

Introduction

In the ubiquitous environment, mobile handheld devices have become very popular and have a wide range of applications, including audio-visual, recording of events, surfing the Internet, making phone calls, etc. Among these applications chatting and sending messages is the most indispensable. However, being the internet an open and insecure network, some anxiety has been raised in transmitting sensitive information. The mobile handheld devices are key players in a ubiquitous computing environment. One characteristic of the Ubiquitous computing environment is the limitations of resources [1]. Ubiquitous engineering needs to deal with the inherent limitations of the mobile handheld devices, such as memory space, processing time and battery capacity. With the advent of mobile phones and later the Internet, a revolution took place. Mobility became popular with the introduction of mobile phones. This invention aroused much interest and had, in a few years, an explosion in its use worldwide, transforming the society and causing the greatest moment of mobile communication [5]. The number of mobile phone subscribers worldwide increased from 34 million in 1993 to more than one billion in 2003 and 4.6 billion in 2008 [6].

The many benefits of cell phones are obvious to everyone - anywhere, anytime, unimpeded access to the global telephony equipment via a lightweight and fully portable [7] device. As the use of these devices is still new, there is a methodology for security in the information transaction, which does not give confidence to users. Without an effective security mechanism, a malicious user is able to capture information from other users by means of transmission, and use them as they see fit, committing fraud, causing damage and inconvenience to the owner of the information. In financial transactions, electronic commerce, e-mail sending and other ways to send some information, it is necessary to make sure the transmitter and receiver can sign the document or transaction in a digital way, giving greater reliability to the transaction [8]. Electronic payment is becoming increasingly common nowadays, which makes necessary a high degree of trust between the media (host client, server), which is usually accomplished with the help of security protocols developed for the Internet, as SSL / TLS (Security Socket Layer / Transport Layer Security). Those protocols already give a degree of trust on both sides [9]. When considering mobile devices, they do not have an usual and efficient protocol which ensures trustworthiness in transactions and integrity and reliability of the data, as well as online payment and transfer of sensitive information . The solution lies in using cryptography and secures authentication protocols that guarantee the confidentiality, authentication and integrity of communications. Most of them are based in symmetric key cryptography, asymmetric key cryptography, message digest. A cryptography algorithm performs well in resource constrained platforms and maintains the high security level that one can achieve with the protocols in use today. So experiments have been conducted over various symmetric cryptographic , Asymmetric Cryptographic algorithms to reduce power consumption, Analyses of the time , size and computation cost are performed to offer users information to produce optimal algorithm for sending information.

Mobile Security And Cryptography

One of the most common ways to implement security in a computer system is known as cryptography. In short, encryption can be explained as a set of methods and techniques to encrypt data by using an encryption algorithm parameteri zed by a key, converting an original text, called plain text, in an unreadable text, called cipher text. It is then possible for the receiver to decrypt this ciphertext, that is, to perform the reverse process and retrieve the original information [10] .

Typically, new algorithms are opened to the community as they are developed and confidentiality of information is ensured by the key, which must be kept secret and be offered only to relevant entities. The key size in this case is very important, since i t determines the encryption level [1]. Furthermore, based on the type of key, one can classify the cryptography in symmetric key or public-key. The symmetric key has this name because the processes of encryption and decryption are performed using a single key, that is, both the sender and receiver have the same key and it should be kept secret in order to ensure an acceptable level of safety. The main advantage of symmetric key encryption is that the algorithms of this type are fast and can operate on messages of arbitrary sizes [11]. On the other hand, the disadvantage of this kind of encryption is the difficulty to manage the shared key, which must be sent to all authorized users before messages can be exchanged and must still be kept secret [10].

The asymmetric encryption, also known as public key cryptography, uses a pair of keys called public-key and private-key. Any key can be used to encrypt the data, but it cannot be used to decrypt it, that is, if the encryption was done with the public -key, only the private key may perform decryption, or vice versa. In order to make this type of encryption successful, it is essential that the private key be kept secret while the public-key should be disseminated to other users who want to communicate [11]. This work uses an implementation of an asymmetric algorithm.

Message Digest or Hash Function

Cryptographic hash functions are a valuable tool in cryptography. They are applied in many areas of information security to provide protection of the authenticity of messages; data integrity verification which prevents modification of data from going undetected, time stamping and digital signature scheme.

Hashing

Hashing is a process by which one turns a string of characters with variable length into a fixed length value which represents the original string.

Hashing versus Encrypting

Sometimes, hashing is being referred to in situations where encryption is the most appropriate term and vice versa. We clarify this common confusion once and for all. Hashing, in cryptography, is a one-way operation which transforms a stream of data into a more compressed form called a message digest. The operation is not be invertible, meaning that recovering the original data stream from the message digest should not be possible. All the message digests or hash values generated by a given hash function have the same size no matter what the size of the input value is. Encryption on the other hand, can be thought of as a two-way operation which transforms a plaintext into a cipher text and allows for the process to be inverted by transforming the cipher text back into its original plaintext via a mechanism called decryption. Both operations depend on a key.

Cryptographic Hash Functions

A hash function is a function that takes some message of any length as input and transforms it into a fixed-length output called a hash value, a message digest, a checksum, or a digital fingerprint.

A hash function is a function $f : D \rightarrow R$, where the domain $D = \{0, 1\}^*$, which means that the elements of the domain consist of binary string of variable length; and the range $R = \{0, 1\}n$ for some $n \geq 1$, which means that the elements of the range are binary string of fixed-length. So, f is a function which takes as input a message M of any size and produces a fixed-length hash result h of size n. A hash function f is referred to as compression function when its domain D is finite, in other word, when the function f takes as input a fixed-length message and produces a shorter fixed-length output.

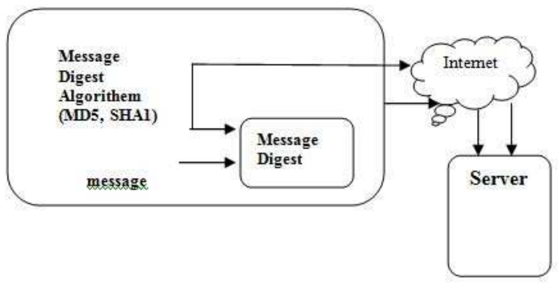

Figure: - Protecting a message with Message Digest

A cryptographic hash function H is a hash function with additional security properties:

1. H should accept a block of data of any size as input.
2. H should produce a fixed-length output no matter what the length of the input data is.
3. H should behave like random function while being deterministic and efficiently reproducible. H should accept an input of any length, and outputs a random string of fixed length. H should be deterministic and efficiently reproducible in that whenever the same input is given, H should always produce the same output.
4. Given a message M, it is easy to compute its corresponding digest h; meaning that h can be computed in polynomial time O(n) where n is the length of the input message, this makes hardware and software implementations cheap and practical.
5. Given a message digest h, it is computationally difficult to find M such that H(M) = h. This is called the one-way or pre-image resistance property. It simply means that one should not be capable of recovering the original message from its hash value.
6. Given a message M1, it is computationally infeasible to find another message M2 ≠ M1 with H(M1) = H(M2). This is called the weak collision resistance or second pre image resistance property.
7. It is computationally infeasible to find any pair of distinct messages (M1, M2) such that H(M1) = H(M2). This is referred to as the strong collision resistance property.

These properties are required in order to prevent or withstand certain types of attacks which may render a cryptographic hash function useless and insecure. In addition to producing a “digital fingerprint" of a message M that is unique and to providing strong collision resistance, a cryptographic hash function should also be highly sensitive to the smallest change in the input message. Such that a change, as small as a single digit, in the input message should produce a large change in the hash value of the message. Note that a message in this context can be a binary text file, audio file, or executable program.

Implementation And Analysis of Message Digest

A fundamental primitive in modern cryptography is the cryptographic hash function. To be of cryptographic use, a hash function h is typically chosen such that it is computationally infeasible to find two distinct inputs which hash to a common value (i.e., two colliding inputs x and y such that h(x) = h(y)), and that given a specific hash-value y, it is computationally infeasible to find an input (pre image) x such that h(x) = y [1]. Several hash functions techniques are available.

This includes SHA-1, SHA-256, SHA-384, SHA-512, MD2, MD4,MD5, RIPEMD-128, RIPEMD-160, RIPEMD-256, RIPEMD-320, TigerDigest, below Table presents a comparison among such techniques.

The message digest algorithem implemented in J2ME wireless tool kit 3.1 and Netbeans 6.5 with mobility pack. The details of the Sun Java Wireless Toolkit can be had from [14] and the toolkit can be downloaded from [15]. With the growing diversity of mobile devices to which the protocol targeted for, its portability was a major concern since the beginning. Therefore it was developed using the J2ME, whose features meet this requirement. For implmentation of message digest algoritehm used Bouncy castel API.

Message digest algorithem	Output size	Block size	Word size	Security	Speed
MD2	128	128	8	Medium	Medium
MD4	128	512	32	Low	Medium
MD5	128	512	32	High	Fast
SHA-1	160	512	32	High	Fast

SHA 256/224	256/224	512	32	High	Slow
SHA 384/512	384/512	1024	64	High/Extreme	Low
RIPEMD 128/256	128/256	512	64	Low	Fast
RIPEMD 160/320	160/320	512	64	high	Slow
TigerDigest	128	128	32	Medium	Medium

Table : Different hash algorithm comparison based on emulator

Conclusion

In this paper, we discuss an efficient and secure different message digest (hash function) algorithm for providing security to mobile nodes. Since we have tested different message digest algorithm with different scenarios and it is providing better response time, less network delay and best throughput. These parameters have been shown in above table. We got better results comparsion and we also identify application based hash function in mobile environment Also our research shows that it is helping in efficient routing of packet with much less load on servers.

References

[1] Chu-Hsing Lin, Jung-Chun Liu, Chun-Wei Liao, Energy analysis of multimedia video decoding on mobile handheld devices, Computer Standards & Interfaces Volume 32 Issue 1-2,2009.

[2] Arvinderpal S. Wander, Nils Gura, Hans Eberle, Vipul Gupta, Sheueling Chang Shantz, Energy Analysis of Public-Key Cryptography on Small Wireless Devices, The IEEE PerCom 2005.

[3] Helena Rif`a-Pous and Jordi Herrera-Joancomarti, Computational and Energy Costs of Cryptographic Algorithms on Handheld Devices, Future Internet 3(1): 31-48, 2011.

[4] Shanmugalakshmi.R and M.Prabu, Research Issues on Elliptic Curve Cryptography and Its applications,International Journal of Computer Science and Network Security, VOL.9 No.6,2009.

[5] TANEMBAUM, A. S., "Redes de Computadores". 4º Edição, 2003.

[6] TAURION, Cezar, "Internet Móvel Tecnologias, Aplicações e Modelos". Rio de Janeiro – RJ. Editora Campus 2002

[7] Kurose, J. F.; Ross K. W. "Computer Networking: A Top-Down Approach". Addison Wesley Publishing Company.

[8] FERREIRA, Lucas C.; DAHAB, Ricardo. Blinded-Key Signatures: securing private keys embedded in mobile agents. Março, 2002.

[9] CLAESSENS, Joris. PRENEEL, Bart. VANDEWALLE, Joos. (How) Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of the Security Issues and the Current SolutionsACM Transactions on Internet Technology (TOIT)Vol3Fevereiro2003

[10] MORENO, Edward David; PEREIRA, Fábio Dacêncio; CHIARAMONTE, Rodolfo Barros. Criptografia em Software e Hardware. São Paulo: Novatec, 2005.

[11] ROSENBERG, Jothy; REMY, David. Securing Web Services with WS-Security. Canada: Sams Publishing, 2004.

[12] ROUSAN, Mohammad AL; RJOUB, A.; AHMAD, Baset. A Low-Energy Security Algorithm for Exchanging Information in Wireless Sensor Networks, 2008.

[13] M. Hassinen, SafeSMS - end-to-end encryption for SMS, Proceedings of the 8th International Conference on Telecommunications Volume 2, ConTEL 2005, ISBN: 953-184-081-4

[14] http://java.sun.com/javame/reference/apis.jsp

[15] http://java.sun.com/products/sjwtoolkit/download-3_1.html