A More Secure Hashing Scheme for Information Authentication

Nidhi Sharma
(Assistant Professor in TIT&S, Bhiwani)

Alok Sharma
(Lecturer in Balaji College of Engg, Bhiwani)

Monika Sharma
(Assistant Professor in TIT&S, Bhiwani)

ABSTRACT

The protection of information authentication is based on cryptographically secure hash function. A function that compresses an arbitrarily large message into a fixed small size message digest' is known as a *hash function*. The most currently used hash functions are based on block ciphers and the dedicated hash functions e.g. MD5, SHA-0 and SHA-1 etc.

A hash function is said to be broken if an attacker is able to show that the design of the hash function violates at least one of its claimed security property. In present paper we are planning to design and develop an efficient and effective hashing scheme for information authentication. We will try to design a hashing method which has certain features like speed and more security.

1. INTRODUCTION

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, cryptanalysis is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers.

An extremely simple example of conventional cryptography is a substitution cipher. A substitution cipher substitutes one piece of information for another. This is most frequently done by offsetting letters of the alphabet. Two examples are Captain Midnight's Secret Decoder Ring, which you may have owned when you were a kid, and Julius Caesar's cipher. In both cases, the algorithm is to

offset the alphabet and the key is the number of characters to offset it. For example, if we encode the word "SECRET" using Caesar's key value of 3,

we offset the alphabet so that the 3rd letter down (D) begins the alphabet.

So starting with

ABCDEFGHIJKLMNOPQRSTUVWXYZ

and sliding everything up by 3, we get

DEFGHIJKLMNOPQRSTUVWXYZABC

where D=A, E=B, F=C, and so on

Using this scheme, the plaintext, "SECRET" encrypts as "VHFUHW." To allow someone else to read the ciphertext, you tell them that the key is 3

2. TYPES OF CRYPTOGRAPHIC ALGORITHMS

There are several ways of classifying cryptographic algorithms. For purposes of this paper, they will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms that will be discussed are (Figure 1):

- Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption
- Public Key Cryptography (PKC): Uses one key for encryption and another for decryption
- Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information

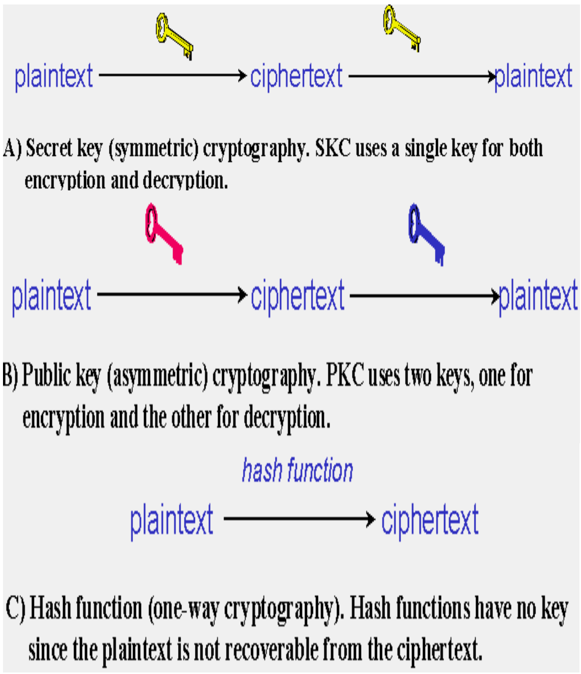

Figure 1: Three Types Of Cryptography: Secret-Key, Public Key, And Hash Function

3.1 Secret Key Cryptography

With secret key cryptography, a single key is used for both encryption and decryption. As shown in Figure 1A, the sender uses the key (or some set of rules) to encrypt the plaintext and sends the cipher text to the receiver. The receiver applies the same key (or rule set) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called symmetric encryption.

With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the distribution of the key.

Secret key cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing. A block cipher is so-called because the scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

Block ciphers can operate in one of several modes; the following four are the most important:

- Electronic Codebook (ECB) mode is the simplest, most obvious application: the secret key is used to encrypt the plaintext block to form a ciphertext block. Two identical plaintext blocks, then, will always generate the same ciphertext block. Although this is the most common mode of block ciphers, it is susceptible to a variety of brute-force attacks.
- Cipher Block Chaining (CBC) mode adds a feedback mechanism to the encryption scheme. In CBC, the plaintext is exclusively-ORed (XORed) with the previous ciphertext block prior to encryption. In this mode, two identical blocks of plaintext never encrypt to the same ciphertext.
- Cipher Feedback (CFB) mode is a block cipher implementation as a self-synchronizing stream cipher. CFB mode allows data to be encrypted in units smaller than the block size, which might be useful in some applications such as encrypting interactive terminal input. If we were using 1-byte CFB mode, for example, each incoming character is placed into a shift register the same size as the block, encrypted, and the block transmitted. At the receiving side, the ciphertext is decrypted and the extra bits in the block (i.e., everything above and beyond the one byte) are discarded.
- Output Feedback (OFB) mode is a block cipher implementation conceptually similar to a synchronous stream cipher. OFB prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that is independent of both the plaintext and ciphertext bitstreams.

3.2 Public Key Cryptography

Public-key cryptography is a cryptographic approach, employed by many cryptographic algorithms and cryptosystems, whose distinguishing characteristic is the use of asymmetric key algorithms instead of or in addition to symmetric key algorithms. Using the techniques of public key-private key cryptography, many methods of protecting communications or authenticating messages formerly unknown have become practical. They do not require a secure initial exchange of one or more secret keys as is required when using symmetric key algorithms. It can also be used to create digital signatures.

Public key cryptography is a fundamental and widely used technology around the world, and is the approach which underlies such Internet standards as Transport Layer Security (TLS) (successor to SSL), PGP and GPG.

The distinguishing technique used in public key-private key cryptography is use of asymmetric key algorithms because the key used to encrypt a message is not the same as the key used to decrypt it. Each user has a pair of cryptographic keys — a public key and a private key. The private key is kept secret, whilst the public key may be widely distributed. Messages are encrypted with the recipient's public key and can only be decrypted with the corresponding private key. The keys are related mathematically, but the private key cannot be feasibly (ie, in actual or projected practice) derived from the public key. It was the discovery of such algorithms which revolutionized the practice of cryptography beginning in the middle 1970s.

In contrast, Symmetric-key algorithms, variations of which have been used for some thousands of years, use a single secret key shared by sender and receiver (which must also be kept private, thus accounting for the ambiguity of the common terminology) for both encryption and decryption. To use a symmetric encryption scheme, the sender and receiver must securely share a key in advance.

The two main branches of public key cryptography are:

- **Public key encryption** — a message encrypted with a recipient's public key cannot be decrypted by anyone except a possessor of the matching private key -- presumably, this will be the owner of that key and the person associated with the public key used. This is used for confidentiality.
- Digital signatures — a message signed with a sender's private key can be verified by anyone who has access to the sender's public key, thereby proving that the sender had access to the private key (and therefore is likely to be the person associated with the public key used), and the part of the message that has not been tampered with. On the question of authenticity, see also message digest.

3.3 Hash Functions

Hash functions, also called message digests and one-way encryption, are algorithms that, in some sense, use no key (Figure 1C). Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Hash algorithms are typically used to provide a digital fingerprint of a file's contents, often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions, then, provide a measure of the integrity of a file.

Hash algorithms that are in common use today include:

- Message Digest (MD) algorithms: A series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message.
- MD2 (RFC 1319): Designed for systems with limited memory, such as smart cards.
- MD4 (RFC 1320): Developed by Rivest, similar to MD2 but designed specifically for fast processing in software.
- MD5 (RFC 1321): Also developed by Rivest after potential weaknesses were reported in MD4; this scheme is similar to MD4 but is slower because more manipulation is made to the original data. MD5 has been implemented in a large number of products although several weaknesses in the algorithm were demonstrated by German cryptographer Hans Dobbertin in 1996.
- Secure Hash Algorithm (SHA): Algorithm for NIST's Secure Hash Standard (SHS). SHA-1 produces a 160-bit hash value

4. WHY THREE ENCRYPTION TECHNIQUES

So, why are there so many different types of cryptographic schemes? Why can't we do everything we need with just one?

The answer is that each scheme is optimized for some specific application(s). Hash functions, for example, are well-suited for ensuring data integrity because any change made to the contents of a message will result in the receiver calculating a different hash value than the one placed in the transmission by the sender. Since it is highly unlikely that two different messages will yield the same hash value, data integrity is ensured to a high degree of confidence.

Secret key cryptography, on the other hand, is ideally suited to encrypting messages, thus providing privacy and confidentiality. The sender can generate a session key on a per-message basis to encrypt the message; the receiver, of course, needs the same session key to decrypt the message.

Key exchange, of course, is a key application of public-key cryptography (no pun intended). Asymmetric schemes can also be used for non-repudiation and user authentication; if the receiver can obtain the session key encrypted with the sender's private key, then only this sender could have sent the message.

Public-key cryptography could, theoretically, also be used to encrypt messages although this is rarely done because secret-key cryptography operates about 1000 times faster than public-key cryptography.

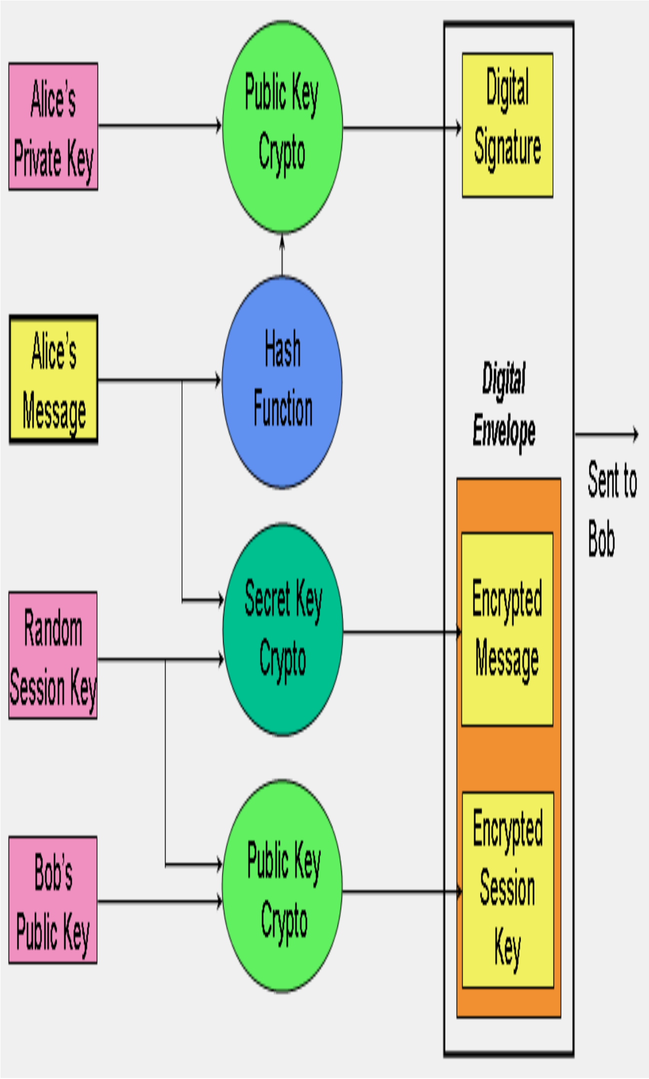

Figure 2: Sample Application of the Three Cryptographic Techniques for Secure Communication.

Figure 2 puts all of this together and shows how a hybrid cryptographic scheme combines all of these functions to form a secure transmission comprising digital signature and digital envelope. In this example, the sender of the message is Alice and the receiver is Bob.

A digital envelope comprises an encrypted message and an encrypted session key. Alice uses secret key cryptography to encrypt her message using the session key, which she generates at random with each session. Alice then encrypts the session key using Bob's public key. The encrypted message and encrypted session key together form the digital envelope. Upon receipt, Bob recovers the session secret key using his private key and then decrypts the encrypted message.

The digital signature is formed in two steps. First, Alice computes the hash value of her message; next, she encrypts the hash value with her private key. Upon receipt of the digital signature, Bob recovers the hash value calculated by Alice by decrypting the digital signature with Alice's public key. Bob can then apply the hash function to Alice's original message, which he has already decrypted (see previous paragraph).

If the resultant hash value is not the same as the value supplied by Alice, then Bob knows that the message has been altered; if the hash values are the same, Bob should believe that the message he received is identical to the one that Alice sent.

This scheme also provides nonrepudiation since it proves that Alice sent the message; if the hash value recovered by Bob using Alice's public key proves that the message has not been altered, then only Alice could have created the digital signature. Bob also has proof that he is the intended receiver; if he can correctly decrypt the message, then he must have correctly decrypted the session key meaning that his is the correct private key.

5. CONCLUSION

This paper discusses various cryptographic techniques that are used to encrypt data to prevent it from being violated during transit. For this purpose we plan and study:

- To analyze the existing schemes
- To analyze the existing hashing scheme in order to improve their performances and security.
- reLiterature survey of related work.
- Analysis of Schemes.

6. REFERENCES AND FURTHER READING

[1] Bamford, J. *Body of Secrets : Anatomy of the Ultra-Secret National Security Agency from the Cold War Through the Dawn of a New Century.* New York: Doubleday, 2001.

[2] _____. *The Puzzle Palace: Inside the National Security Agency, America's most secret intelligence organization.* New York: Penguin Books, 1983.

[3] Barr, T.H. *Invitation to Cryptology.* Upper Saddle River (NJ): Prentice Hall, 2002.

[4] Bauer, F.L. *Decrypted Secrets: Methods and Maxims of Cryptology,* 2nd ed. New York: Springer Verlag, 2002.

[5] Denning, D.E. *Cryptography and Data Security.* Reading (MA): Addison-Wesley, 1982.

[6] Diffie, W. & Landau, S. *Privacy on the Line.* Boston: MIT Press, 1998.

[7] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design.* Sebastopol (CA): O'Reilly & Associates, 1998.

[8] C. Besnard and J. Martin, “DABO: proposed additional message authentication algorithms for ISO 8731,” preprint, 1992.

[9] E. Biham and A. Shamir, “Differential cryptanalysis of Feal and N-hash,” Advances in Cryptology, Proc. Eurocrypt’91, LNCS 547, D.W. Davies, Ed., Springer-Verlag, 2005, pp. 1–16.