Research on Solutions of Security Issues of Ecommerce and Their Comparative Analysis

Shahbaz Khan, Dilina Nair
Department Of MCA,
Mumbai University Maharashtra, India

Abstract: Lots of events till now showed us the security issues in ecommerce system. Electronic commerce and information security are growing areas of concern to user communities. This paper describes the R&D activities in security issues of ecommerce, it has briefly explained about the current security issues and then it has come up with the current security algorithms used by ecommerce for security, later it explains about LUHN algorithm which can help provide security to ecommerce in better way for credit card systems. We also derive a comparison table of all most commonly used security algorithms for ecommerce websites. The paper concludes with a look at the future and discussions on what can be done.

Keywords: Symmetric & Asymmetric algorithm, RSA, MD5, SHA, *LUHN* algorithm, Digital Signatures.

I. INTRODUCTION

With the enhancement of Internet technology, a setup also in existence now that is E-commerce which is based on network and multimedia technology. Ecommerce is an online transactions system wherein security is must to maintain since third party attacks are quite usual in this environment. There are several symmetric as well asymmetric algorithms available for the encryption and security of data in ecommerce. Customers are cautious to take participate in e-commerce due to security problems like hacking customers' information and many other attacks exist in the open network which is dangerous to the customer information. Ecommerce transactions' are mostly carried out using credit cards etc which is hacked by the hackers revealing customer information, bank data, passwords etc. Web ecommerce applications that handle payments (online banking, electronic transactions or using debit cards, credit cards, PayPal or other tokens) have more compliance Issues are at increased risk from being targeted than other websites and there are greater consequences if there is data loss or alteration. Online shopping through shopping websites having certain steps to buy a product with safe and secure.

To solve some of this issues there are number of digital signatures, security algorithms etc used

II. CURRENT STATUS OF ELECTRONIC SECURITY ALGORITHMS

The several algorithms available for security includes asymmetric & symmetric algorithms etc. Digital signatures are also present to verify the identity of the sender and the receiver. Digital Signature is a process that guarantees that the contents of a message have not been altered in transit. When you, the server, digitally sign a document, you add a one-way hash (encryption) of the message content using your public and private key pair. Your client can still read it, but the process creates a "signature" that only the server's public key can decrypt. The client, using the server's public key, can then validate the sender as well as the integrity of message contents. Whether it's

- an email
- an online order
- or a watermarked photograph on eBay.

Below are short descriptions of the current security algorithms used by various sites for providing privacy. RSA, MD5, SHA type algorithms are used for the privacy or security of ecommerce sites in current scenario.

ISSN: 2278 – 1323

A. RSA(Ron Rivest, Adi Shamir and Leonard Adleman) :

RSA is one of the first practical public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem.

RSA algorithm:

RSA involves a public key and private key. The public key can be known to everyone, it is used to encrypt messages. Messages encrypted using the public key can only be decrypted with the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two different large random prime numbers $p$and q
2. Calculate $n = pq$
   - $n$is the modulus for the public key and the private keys
3. Calculate the totient: $\phi(n) = (p-1)(q-1)$.
4. Choose an integer $e$such that 1 < e< $\phi(n)$, and $e$is coprime to $\phi(n)$ **ie:** $e$and $\phi(n)$ share no factors other than 1; gcd(e,$\phi(n)$) = 1.
   - $e$is released as the public key exponent
5. Compute $d$to satisfy the congruence relation $de \equiv 1 \pmod{\phi(n)}$**ie:** $de = 1 + k\phi(n)$for some integer k.
   - $d$is kept as the private key exponent

Encrypting messages

Alice gives her public key (n & e) to Bob and keeps her private key secret. Bob wants to send message M to Alice. First he turns M into a number smaller than $n$by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text $c$corresponding to:

$$c = m^e \mod n$$

Decrypting messages

Alice can recover m from c by using her private key d in the following procedure:

$$m = c^d \mod n$$

Given m, she can recover the original message **M**. The decryption procedure works because first

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$$

B. MD5(message-digest):

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

MD5 Algorithm:

- Step1 Append padding bits

The input message is "padded" (extended) so that its length (in bits) equals to 448 mod 512. Padding is always performed, even if the length of the message is already 448 mod 512. Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448 mod 512. At least one bit and at most 512 bits are appended.

- Step2. Append length

A 64-bit representation of the length of the message is appended to the result of step1. If the length of the message is greater than 2^64, only the low-order 64 bits will be used. The resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. The input message will have a length that is an exact multiple of 16 (32-bit) words.

- Step3. Initialize MD buffer

A four-word buffer (A, B, C, D) is used to compute the message digest. Each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first):

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

- Step4. Process message in 16-word blocks

Four functions will be defined such that each function takes an input of three 32-bit words and produces a 32-bit word output.

F (X, Y, Z) = XY or not (X) Z

G (X, Y, Z) = XZ or Y not (Z)

H (X, Y, Z) = X xor Y xor Z

I (X, Y, Z) = Y xor (X or not (Z))

C. **SHA(**Secure Hash):

The Secure Hash Algorithm is a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS), including:

- **SHA-0**: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA". It was withdrawn shortly after publication due to an undisclosed "significant flaw" and replaced by the slightly revised version SHA-1.

- **SHA-1**: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. Cryptographic weaknesses were discovered in SHA-1.

- **SHA-2**: A family of two similar hash functions, with different block sizes, known as *SHA-256* and *SHA-512*. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words.
- **SHA-3**: It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family.

To add an extra benefit or to contrast with all of the above algorithm, Luhn Algorithm can be used for finding the authenticity of the credit cards in a better way.

III. LUHN ALGORITHM

The Luhn algorithm or Luhn formula, also known as the "modulus 10" or "mod 10" algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers, IMEI numbers, National Provider Identifier numbers in US and Canadian Social Insurance Numbers.

The formula verifies a number against its included check digit, which is usually appended to a partial account number to generate the full account number. This number must pass the following test:

1. From the rightmost digit, which is the check digit, moving left, double the value of every second digit; if the product of this doubling operation is greater than 9 (e.g., 8 × 2 = 16), then sum the digits of the products (e.g., 16: 1 + 6 = 7, 18: 1 + 8 = 9).
2. Take the sum of all the digits.
3. If the total modulo 10 is equal to 0 (if the total ends in zero) then the number is valid according to the Luhn formula; else it is not valid.

Currently luhn algorithm are used in calculators and it is possible to build or implement it in a ecommerce website.

But this can be implemented in ecommerce system to provide security to the credit card numbers.

In today’s scenario luhn algorithm is used to find out the authenticity of credit card or identification number. For example, cards issued by hotels for their guests, frequent flyer cards issued by airlines.

But according to our research it can also be used in an ecommerce system to check out the authenticity of the credit cards during the transaction carried out in an online shopping or etc.

Luhn algorithm can be used in an ecommerce system when e-commerce merchants are required to verify the validity of all bank cards submitted for payment on their websites, just as their brick-and-mortar counterparts are required to do in their physical stores. However, they lack the advantage that store-front retailers have in being able to physically examine the card’s features, in order to determine whether or not it has been tampered with:

Following functions are defined:

```
 int CodePointFromCharacter(char character) {...}
 char CharacterFromCodePoint(int codePoint) {...}
 int NumberOfValidInputCharacters() {...}
```

The function to generate a check character is:

```
char GenerateCheckCharacter(string input) {

        int factor = 2;
        int sum = 0;
        int n = NumberOfValidInputCharacters();
        for (int i = input.Length - 1; i >= 0; i--)
 {
                int codePoint =
CodePointFromCharacter(input[i]);
int addend = factor * codePoint;
                factor = (factor == 2) ? 1 : 2;
                addend = (addend / n) + (addend %
n);
                sum += addend;
        }

        int remainder = sum % n;
        int checkCodePoint = (n - remainder) % n;

        return
CharacterFromCodePoint(checkCodePoint);
}
```

Function to validate a string (with the check character as the last character) is:

```
bool ValidateCheckCharacter(string input)
 {
        int factor = 1;
        int sum = 0;
        int n = NumberOfValidInputCharacters();

        for (int i = input.Length - 1; i >= 0; i--)
{
                int codePoint =
CodePointFromCharacter(input[i]);
                int addend = factor * codePoint;

                factor = (factor == 2) ? 1 : 2;

                addend = (addend / n) + (addend %
n);
                sum += addend;
        }

        int remainder = sum % n;

        return (remainder == 0);
}
```

Luhn algorithm in all is not a malicious attack survivor but can check out the authenticity of user or customer in the process.

IV. COMPARE AND CONTRAST ALL MOST COMMONLY USED SECURITY ALGORITHMS

	Limitations	**Strength**
AES	Sharing the key. More damaged if compromised. High Computational overhead is involved with RSA.	Extremely Secure, Fast and Flexible.
DES	Key size is too short for proper security (56 effective bits; this can be brute-forced, as has been demonstrated more than ten years ago Inadequate For security. It is not flexible. The structure of DES doesn't support any modifications.	DES was not designed for software and hence runs relatively slowly. A $1 million DES cracking machine can search the entire key space in about 7hours
BLOWFISH	Blowfish should not be used to encrypt files that are larger than 4Gb because of its small 64-bit block size Susceptible to attacks on reflectively weak keys	Takes less execution time and is comparatively smaller than other algorithms.
IDEA	The structure of IDEA doesn't support any modifications. It is not flexible. IDEA contains 8 rounds in which first 3 rounds appears to highly expose to key attacks such as key-schedule attacks and related-key differential timing attacks.	Its strength can be measured against differential cryptanalysis and concluded that it is immune under certain assumptions. No successful linear or algebraic weaknesses have been reported.
RC4	Sensitive to cryptanalytic attacks.	Enhanced speed of RC4, high interference level.
RSA	It lacks in encryption speed. RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm. Weak keys might be generated.	Enhanced security and convenience.
MD5	Lack of security. Use MD5 only as checksum hash like CRC. Slower than CRC.	Speed- Fast cryptographic hash function. Convenience.

V. LITERATURE REVIEW

It was shown in [8] that RSA consume longest encryption time and memory usage is also very high but output byte is least in case of RSA algorithm. It was concluded in [5] that MD5 algorithm is convenient and is referred as Speed- Fast cryptographic hash function. It was concluded in [7] that AES is faster and more efficient algorithms. When the transmission of data is considered there is insignificant difference in performance of different symmetric key schemes. A study in [8] is conducted for different popular secret key algorithms as DES, AES, RSA, RC4, IDEA, blowfish and their comparative analysis. A study in [9] is conducted to conclude *luhn* algorithm and its technique to validate credit cards.

VI. CONCLUSION

Luhn algorithm will provide a beneficial point for credit card authenticity. It can be further developed more and can be merged up with other highly security providing algorithms to validate the numbers entered before finalizing the transactions.

This paper reflects about how the importance for the authenticity of credit card, master visa card etc type numbers to ensure proper security. It also enlighten a comparison about the most used algorithm for credit card system in ecommerce system.

A further research can be done to validate the performance of *luhn* algorithm to check authenticity of ID card numbers especially for websites that analysis ID cards to ensure non-repudiation of users or customers.

REFERENCES

[1] http://www.ijarcce.com/upload/2013/july/69-o-Niranjanamurth
[2] http://www.etsi.org/deliver
[3] https://www.instantssl.com/"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”- Rivest, R.; A. Shamir; L. Adleman (1978)
[4] http://en.wikipedia.org/wiki/MD5
[5] http://blog.unibulmerchantservices.com
[6] http://simple.wikipedia.org/wiki/RSA_algorithm.
[7] Comparative Analysis Of Encryption Algorithms For Data Communication-Shashi Mehrotra Seth, Rajan Mishra
[8] http://www.academia.edu/5320811/Study_of_Various_Cryptographic_Algorithms.
[9] http://www.academia.edu/4093051/Enhance_Luhn_Algorithm_for_Validation_of_Credit_Cards_Numbers_

AUTHORS PROFILE

Dilina Nair currently pursuing MCA from IMCOST Thane affiliated by Mumbai University.

Shahbaz Khan currently pursuing MCA from IMCOST Thane affiliated by Mumbai University.