2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)
Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore

True Random number generator method based on image for key exchange algorithm

P.Murali [1] and R.Palraj [2]

Department of Computer Science and Engineering
Periyar Maniammai University Thanjavur, 613403 India

Abstract. In this paper we propose a new method for generating True random numbers based on image which generates 256 bits key or higher for key exchange algorithm. True random numbers are always secured and good, compared to pseudo random numbers. Diffie-Hellman key exchange algorithm has two weaknesses: Discrete logarithm attack and Man-in-the-Middle attack. Our proposed method can easily overcome the above problems. This method of implementation is very easy, cost effective and convenient for transmission of shared session keys. Our proposed method rapidly increases the security of the key exchange protocols over an unsecured channel and can also be used for Public key cryptosystem.

Keywords: Diffie-Hellman, True random numbers, discrete logarithm attack, Man-in-the-Middle attack.

1. Introduction

The recent growth in Internet and communication technologies have made the privacy and security of data a major concern in fields like e-Banking, e-commerce,business in general, etc. It leads the development of various techniques, enhancement and adaptation of cryptography. Random numbers are investigated with cryptography over the decades. It plays an important role in the use of encryption for various network security applications. For example [1][2][3][4], to prevent replay attacks in key distribution centre, Session key generation and public key encryption algorithm. True random numbers are more secured than pseudo random numbers because it is generated from non-deterministic resources. Random sequences are classified into two categories: true random numbers (TRNGs) and pseudo random numbers (PRNGs). Pseudo random sequences can be generated from deterministic sources BBS methods, ANSI X9.17, FIPS 186 generator etc. [2]. True random numbers are generated from hardware, software and De-skewing methods [2]. It is more secured, compared to pseudo random numbers. But it requires additional devices which makes inconvenient for normal users.

In this paper we propose a method to generate true random numbers based on image which is very easy, cost effective and convenient for all users. It generates 256 bits key or higher bits from small image. Complexity of the pseudo random number generators is based on proper selection of seed values. If the seed values are known to the intruder, the output sequence can be easily generated [7]. Here small image (black and white) can be used to generate random numbers otherwise a user can easily create an image from MS-Paint software.

Diffie-Hellman key exchange algorithm possesses two major weaknesses: Discrete logarithm attack and Man-in-the-Middle attack [1][2]. An intruder easily breaks the algorithm with the help of public elements p and g. Our proposed work makes the p and g as private and secret [1][2][3]. These two elements are derived from small image which is transmitted between sender and receiver. So intruder can not derive the key. The following section is arranged as follows: Section 2 introduces a method to generate the random numbers from image, Section 3 describes the analysis of Diffie-Hellman Key exchange algorithm, and Section 4 proposes a method for Key Exchange Algorithm and Section 5 proposes security analysis.

2. Trng Based on Image

2.1. Need for Image

As stated before, TRNGs lead to a very good security compared to Pseudo random numbers. When TRNGs numbers are used for PC applications, the selections of non-deterministic source for TRNGs are very difficult. There are many alternatives such as thermal noise from a semiconductor diode or resistor, free running oscillator, air turbulence with a sealed disk, sound from a micro phone, video input from a camera, atmospheric noise, radioactive decay, system clock, content of input /output buffers [2], mouse movement [8] etc., when running cryptographic applications based on these methods become either too expensive or too slow for normal users. In some cases, additional devices are required for transferring and converting data to applications, which make the application not global.

Sometimes these methods consume more time to generate output. Here we are generating TRNGs based on small image (black and white). Users can also easily generate image from MS-Paint software. This method of generating TRNGs is very cheap and cost effective. The speed is very high and no additional device is required for generating TRNGs from this method. So TRNGs based on image is a good method for cryptographic applications.

2.2. Image to Binary format

Here small image (black and white) can be used for generating TRNGs and also image from paint software (eg: MS-Paint from Windows). The pixel value of the image can be obtained with the help of simple functions from.NET and it is converted into string value. To convert image into binary format, we are checking the RGB value of the each and every pixel. And then we are comparing those values in the pixel. The corresponding values (0's and 1's) are written in the text file from left to right or in any other format. If there is a small change in the image it leads to a big difference in the generated random numbers.

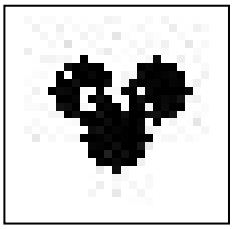

Fig (i) image (25x25)

```
1010101010110010000000000
0100011111010111000000000
1010010111110010000000000
1001011001001001000000000
0110110111111010000000000
1001011011111101000000000
0100001111111101000000000
1011111111011111000000000
0101001110101111110010100
1110101111111111111101110
1000101111111111111000010
1011010001011111111111110
0110101111001111111110000
0111110111111111111001010
0110101111111111111010110
1011011111111111101111010
0011011111111110000000000
1111101111111001000000000
0100101111111111000000000
1110011111110010000000000
0101100001100111000000000
1110111011101101000000000
0101001100011010000000000
1010100010110010000000000
0000000000000000000000000
```

Fig (ii) corresponding binary value from image

2.3. Generation of numbers

Here we are using 25 x 25 pixels (image) for generation of random numbers. It is composed of 625 elements.

- Scan those 625 numbers from top to bottom and left to right.
- Concatenating the value to generate the 625 –bits random number.
- We can apply any rule for deriving random numbers like XOR, mapping, discarding etc.
- p value can be generated by concatenating columns only or rows only or rows and columns.
- g value can be generated by concatenating columns only or rows only or rows and columns.
- Similarly secret keys of Alice and Bob are also chosen from the same image or different image.

- We are transmitting only image to the destination instead of values, the receiver can apply the same rule on the transmitted image to retrieve the values.

3. Diffie-hellman key exchange algorithm

Diffie-Hellman key agreement is the first practical solution for key exchange problem, to establish shared secret key by exchanging messages over an open channel. The security of the protocols lies on problem of computing discrete algorithms [1][2]. In this protocol, two keys are public (p and g). Based on the public key values only, shared secret key is derived for further usage. To make Diffie-Hellman safe from the discrete logarithm attack, p value should be more than 300 decimal digits. But we can easily break those numbers with advent of new technologies. The protocol has another weakness, is called Man-in-the-Middle attack. Again this attack is also based on public values.Our proposed method overcomes the above problems over an unsecured channel.

4. Proposed Method for Key Exchange Algorithm

Our proposed method of key distribution algorithm is based on TRNGs from image. The key values (p and g) and secret values of Alice & Bob also are derived from image which is transmitted between sender and receiver. In this method we are transmitting small image (25x25 pixels, black and white) to the destination. To make key distribution algorithm safe from attacker, the two keys (p and g) are to be private and secret. So the intruders can not break the protocols with Discrete Logarithm, Man-in-the-Middle attack or any other attacks. We can use higher pixel image for generating large random numbers. The remaining rules and regulations are same as Diffie-Hellman Key exchange algorithm.

The proposed method is as follows:

- Generate p and g values from the image. Both p and g values are secret.
- Alice chooses a large random number x from same image or another image such that $0 \leq x \geq p-1$ and calculates $R_1 = g^x \bmod p$
- Bob chooses a large random number y from the same image or another image such that $0 \leq y \geq p-1$ and calculates $R_2 = g^y \bmod p$
- Alice sends R_1 to Bob. Similarly Bob sends R_2 to Alice.
- Alice calculates $K = (R_2)^x \bmod p$.
- Bob also calculates $K = (R_1)^y \bmod p$.

4.1. Example

In our example we use very small numbers but note that in a real situation, the numbers are very large. Here all the values are chosen from the same image. Sender and receiver can choose secret values from the same image or different image.

- Assume p = 85 and g = 4;
- Alice chooses x = 2 calculates $R1 = 4^2 \bmod 85 = 16$
- Bob chooses y=3 calculates $R2 = 4^3 \bmod 85 = 64$
- Alice and Bob exchange R1 and R2
- Alice calculates the shared key $K = 64^2 \bmod 85 = 16$
- Bob calculates the shared key $K = 16^3 \bmod 85 = 16$

5. Security Analysis of Proposed method

This proposed methodology rapidly increases the security of the key exchange algorithm. We can easily generate 256, 512, 1024 or higher bit values. We are generating True Random Numbers from image which is always secured, compared to Pseudo Random numbers. The inner structure of this method is very simple. We can also generate good TRNGs from non-deterministic sources but it requires additional devices [8]. It makes the applications not suitable for all users. In our proposed method all values are derived from the small image and we can also create image from MS-Paint software.

The main drawbacks of Diffie –Hellman algorithm are discrete logarithm and Man-in-the-Middle attack because p and g values are made public. The intruder can easily break the algorithm. Currently many algorithms are available for exchange of symmetric keys but it requires additional steps like public key certificate, encryption etc. Our approach modifies the loop hole in the Diffie-Hellman key exchange algorithm without adding much complex steps. It can be easily implemented and the cost of implementation also is very less. The speed is considerably high, compared to other methods. In our method we are transmitting only small image (25 x 25 pixel values, black and white) to the destination. For areas with low bandwidth or very less memory storage, this method can be used. This method of generating random numbers can also be further extended to other cryptographic applications.

6. Conclusion

In this paper, we propose a method to generate True Random numbers based on image which is very cheap, cost effective, convenient and universal. The attacker could not derive the key from the image. And also it does not require any additional devices. Small change in the image should lead to a significant difference in the generated random number. Further we can also use lossless compression techniques to protect image during transmission.

7. References

[1] Schnier B, Applied cryptography: protocols, algorithms and source code in C. New York: John Wiley and sons, 1996.

[2] Menezes AJ, Oorschot PCV, Vanstone SA, Handbook of applied cryptography. Boca Raton, Florida, USA: CRC Press; 1997.

[3] Johannes A.Buchmann, Introduction to Cryptography. Second Edition, Springer –Verlag NY, LLC, 2001.

[4] Behrouz A. Forouzan, Cryptography and Network Security. Special Indian Edition, The McGraw- Hill companies, New Delhi,2007.

[5] Dhiren R.Patel, Information Security Theory and Practice. First Edition, Prentice-Hall of India Private Limited, 2008.

[6] Keith Harrison, Bill Munro and Tim Spiller, Security through uncertainty. P Laboratories, February, 2007.

[7] William Stallings, Cryptography and Network Security Principles and Practice. Second edition, Pearson Education.

[8] Yue Hu, Xiaofeng Liao, Kwok-wo Wong, Qing Zhou, "A true random number generator based on mouse movement and chaotic cryptography" Choas, solitons and Fractals, Sciencedirect, 2007.

[9] Jaime Gutierrez, Arne Winterhof "Exponential sums of nonlinear Congruential pseudorandom number generators with Rédei functions" Finite Fields and Their Applications 14 (2008) 410–416, Sciencedirect,2007.

[10] Gaston E. Barberis," Non-periodic pseudo-random numbers used in Monte Carlo Calculations" Physica B 398 (2007) 468–471,Sciencedirect.

[11] Kai Wang, Wenjiang Pei, Haishan Xia, Yiu-ming Cheung," Pseudo-random number generator based on asymptotic deterministic randomness" Physics Letters A 372 (2008) 4388–4394

[12] http://en.wikipedia.org/wiki/Random_number_generator.