A Simple Lightweight Encryption Scheme for Wireless Sensor Networks

Kamanashis Biswas[1], Vallipuram Muthukkumarasamy[1], Elankayer Sithirasenan[1], and Kalvinder Singh[2]

[1] Griffith University, Gold Coast, Australia
kamanashis.biswas@griffithuni.edu.au,
{v.muthu,e.sithirasenan}@griffith.edu.au
[2] IBM, Australia Development Lab and Griffith University
Gold Coast, Australia
kalsingh@au.ibm.com

Abstract. Security is a critical issue in many sensor network applications. A number of security mechanisms are developed for wireless sensor networks based on classical cryptography. AES, RC5, SkipJack and XXTEA are some symmetric-key encryption algorithms that are deployed in sensor network environments. However, these algorithms have their own weakness, such as vulnerable to chosen-plaintext attack, brute force attack and computational complexity. We propose an energy efficient lightweight encryption scheme based on pseudorandom bit sequence generated by elliptic curve operations. We present experimental results of our proposed algorithm employed on real sensor nodes operating in TinyOS. We also discuss the security strength of our algorithm by presenting the security analysis of various tests and cryptanalytic attacks.

Keywords: Data confidentiality, Wireless Sensor Network, Elliptic Curve, Symmetric-key encryption, Cryptanalysis.

1 Introduction

Current research focuses on various properties of Wireless Sensor Networks (WSNs), for example, clustering, routing, resource usage, reliability and security [1–3]. Security is a challenging issue in WSNs, since sensor networks are usually deployed in hostile environments. Moreover, small memories, weak processors, limited energy of sensor nodes introduce a number of problems in implementing traditional cryptographic schemes in sensor networks. Hence, WSNs require efficient encryption schemes in terms of operation speed, storage and power consumption. We propose a lightweight encryption scheme for tiny sensor devices guaranteeing data confidentiality between source and destination nodes.

Our proposed scheme has a number of benefits. First, we present a simple pseudorandom bit sequence generation scheme using elliptic curve points that does not involve any floating point calculation. Thus, it avoids the problem of precision loss and also minimizes the computational costs for sensor nodes. Second, the proposed cryptosystem generates a different pseudorandom bit sequence

M. Chatterjee et al. (Eds.): ICDCN 2014, LNCS 8314, pp. 499–504, 2014.

for every new session and preserves independent behavioural characteristic of the algorithm. Third, the proposed scheme is lightweight compared to RC5 and SkipJack in terms of memory occupation, operation time and energy efficiency.

The organization of this paper is as follows: In *Section 2*, we briefly discuss the suitability of existing security mechanisms in WSN environments. *Section 3* provides details of the proposed pseudorandom sequence generation process and our proposed encryption scheme. *Section 4* and *Section 5* present the security and performance analysis of the algorithm respectively. Finally, *Section 6* concludes the paper.

2 Related Works

The Advanced Encryption System (AES) algorithm operates on a 4×4 array of bytes and has a key size of 128, 192, or 256 bits with 10, 12 or 14 number of rounds respectively. Previously, a chosen-plaintext attack can break up to seven rounds of 128-bits AES and eight rounds of 192-bits and 256-bits AES. Currently, AES running on 10, 12 and 14 rounds for 128, 192 and 256-bits size key respectively is also found vulnerable by researchers [4].

RC5 is a flexible block cipher with a variable block size (32, 64, 128 bits), number of rounds (0-255), and key size (0-2040 bits). Although, RC5 is considered more suitable for WSN applications, it requires the key schedule to be precomputed which uses 104 extra bytes of RAM per key. Moreover, RC5 is designed to take advantage of variable-bit rotation instruction (ROL) which is not supported by most embedded system, for instance, Intel architecture [5].

The SkipJack cipher uses an 80-bits key with 32-rounds to encrypt or decrypt 64-bits data blocks. But, the short key length makes SkipJack susceptible to the exhaustive key search attack [6]. An extended version, SkipJack-X is proposed to make the encryption scheme stronger against security attacks. However, the design strategy is not a proper replacement of SkipJack in WSN.

High Security and Lightweight (HIGHT) encryption algorithm is suitable for low resource devices. The algorithm suggested for 32-rounds is 64-bits block length and 128-bits key length. Although HIGHT is designed for low-cost, low power devices, it takes more memory space and operation time than RC5 [7].

Tiny Encryption Algorithm (TEA) is notable for its simplicity and small memory requirement. But it is vulnerable to related-key attack and chosen-plaintext attacks. To overcome these weaknesses, a corrected block TEA (XXTEA) has been designed with a key size of 128 bits. However, the last reported attack against full-round XXTEA presents a chosen-plaintext attack requiring 2^{59} queries and negligible work [8].

3 The Proposed Encryption Scheme

Our proposed encryption scheme is divided into three phases: i) key establishment phase, ii) pseudorandom bit sequence generation phase and iii) encryption phase. Here, we describe the protocol in detail.

3.1 The Key Establishment Process

We assume that prime field, base point and elliptic curve parameters are pre-distributed securely among all sensor nodes (SNs) in WSN. Now, each SN generates a list of elliptic curve points termed as key pool by using point addition and point doubling operation [9]. When a node requires to send data packets, it randomly selects a key from the key pool and converts it into hash code using a hash function. This code is shared with the destination node. The destination node retrieves the shared key by matching the received code with the hash codes generated for each point of its key pool. Upon successful retrieval of the secret key, the destination node acknowledges the source node with a reply message.

3.2 Generation of Pseudorandom Bit Sequence

The security level of many cryptographic systems using Linear Feedback Shift Register (LFSR) or chaotic maps depends on the properties of the random number generation schemes such as unpredictability and unlimited period. But, the security strength of LFSR is poor and cannot meet the demand of unpredictability for secure communication [10]. Again, the chaotic maps require high-precision floating point calculation which is not suitable for resource limited SNs. To avoid these problems, we use elliptic curve over prime field to generate random bit sequence. An elliptic curve (EC) over prime field is a simple algebraic expression that can be defined by the following equation:

$$y^2 \pmod{p} = x^3 + Ax + B \pmod{p} \tag{1}$$

where, A and B are the coefficients and the variables x and y take the values only from the finite field within the range of prime field p. We assume that the values of these parameters are pre-distributed and the participating nodes share a common private key using the key establishment process described in section 3.1. This shared key is used as the base point (G) to generate the random bit sequence in our proposed cryptosystem as shown in the following algorithm.

Algorithm: Pseudorandom binary sequence generation process

Input: Coefficients (A, B); Base Point G (x, y); Prime field p
Output: Binary_Sequence of length N // Initially, N equals to zero
Steps:
1. Generate a new point $\bar{G}$ ($\bar{x}$, $\bar{y}$) using point addition or doubling operation
2. **if** $\bar{x} > \bar{y}$
   Binary_Sequence(N) $\leftarrow$ ($\bar{x}$ mod 2)
   else
   Binary_Sequence(N) $\leftarrow$ ($\bar{y}$ mod 2)
3. $N \leftarrow N + 1$
4. Repeat step 1 to 3 until N $\neq$ desired length

3.3 The Encryption Procedure

The random bit sequence obtained in the previous stage works as a one time password in our proposed encryption scheme. At first step, we convert the plaintext to binary sequence by mapping the characters into their corresponding ASCII codes. Then, the sequence is xor-ed with the pseudorandom bit sequence to generate the ciphertext. We perform the XOR operation because the additive cipher is more secure when the key-stream is random and as long as the plaintext [11]. The decryption process is simply reverse of the encryption procedure.

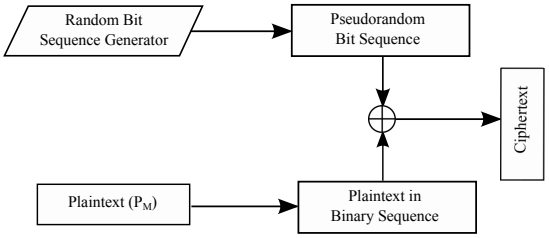


Fig. 1. The general schema of the proposed encryption algorithm

4 Security Analysis

We tested our proposed scheme against various security attacks. The security of elliptic curve cryptography (ECC) relies on the discrete logarithm problem and the best known *brute force attack* requires exponential time to solve the problem. NIST recommends to use 256-bit key for security although the ECC scheme broken to date had a 112-bit key for prime field. *Statistical analysis* is the study of the frequency of letters or common characteristics of words in ciphertext. We used an English article of more than 10,000 words as plaintext to generate the ciphertext. But the ciphertext does not have any statistical features: all of the characters are randomly distributed and do not follow any particular order. Thus, it is too hard to find a co-relation in the ciphertext. *Related-key attack* is based on decrypting ciphertext with various similar keys and analysing the difference in outputs. In our proposed scheme, the EC parameters (A and B), prime number (p), and base point G (x, y) are the primary keys. Our experiment shows that it is hard to generate identical pseudorandom bit sequence if any of the above values is not same. Thus, it is too hard to decrypt the ciphertext without knowing exact value of each parameter used in the encryption process. *Timing Attack* is improbable in our cryptosystem due to data independent behavioural characteristics of the algorithm. Moreover, the binary sequence used in encryption varies each time, hence, it is not possible to derive any statistical co-relation of timing information. Due to frequent re-keying strategy, *chosen plaintext attack* is also not fruitful in our proposed encryption scheme.

5 Performance Analysis

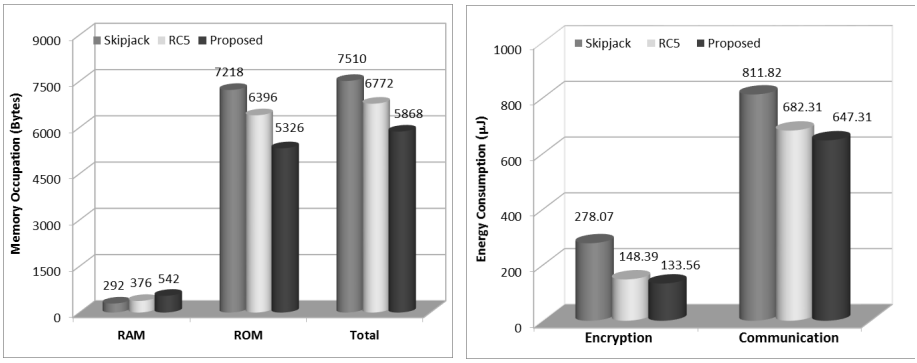
We have implemented our encryption scheme in MICA2 sensor mote operating at 7.3728 MHz (ATmega128L), 128KB program memory and 4KB data memory. The mote supports an event-driven operating system TinyOS and a high level programming language nesC. RC5 and non-optimized SkipJack protocols are also implemented and the results are compared with our proposed scheme.

Operation Time– In this experiment, ATEMU, a high fidelity large scale sensor network emulator, is used to get the total CPU cycles required to encrypt 32 bytes data in MICA2. The results indicate that our algorithm performs better in terms of CPU elapsed time (6.207 ms) using only 45839 CPU cycles. For RC5, the number of CPU cycles and encryption time is little bit higher compared to our scheme and is about double in case of SkipJack.

Table 1. CPU use and elapsed time to encrypt 32 bytes data

Algorithms	CPU Cycles	Time
SkipJack	91224	12.353
RC5	48709	6.595
Proposed	45839	6.207

Memory and Energy Efficiency– Fig. 2(a) shows that our algorithm occupies less memory than that of RC5 and SkipJack. The flash memory (ROM) required by our proposed scheme is lower than RC5 and SkipJack but it occupies more RAM than the other two schemes. However, the total memory required by our protocol is 5868 bytes whereas the amount is 6772 bytes and 7510 bytes for RC5 and SkipJack respectively. Finally, we find the total amount of energy to encrypt 32 bytes of data in our experiments. For this purpose, we use PowerTOSSIM to



(a) Memory usage

(b) Total energy consumption

Fig. 2. Comparison on memory usage and total energy dissipation

measure the total amount of energy required to encrypt and to send the data packets by sensor node. The graph in Fig. 2(b) indicates that our encryption scheme consumes less energy than RC5 and SkipJack.

6 Conclusion

This paper presents a new idea of using different base point of an elliptic curve (i.e., shared key) to generate different pseudorandom bit sequence for two communicating nodes. Due to the ability of producing large bit sequences, our proposed scheme is suitable for large volume data encryption such as image, audio and video. The proposed algorithm uses blocks of plaintext as input and generates corresponding blocks of ciphertext. One of the limitations of block cipher is that it requires to transmit additional bits (padding) when the size of plaintext is smaller than defined block size. However, this situation can be avoided using stream cipher. Since our proposed scheme generates different random bit sequences for every new session, it can also be implemented in the form of stream cipher. The proposed scheme has a few drawbacks. First, the initial parameters need to be pre-distributed using secure channel or a key exchange mechanism. Second, we used 128-bit elliptic curve in our experiments. If we use 256-bit elliptic curve for enhanced level of security then it will result in additional computational cost and memory usage. In our future work we will implement the protocol in combination with other protocols (e.g. TinySec) in a large scale sensor network to evaluate overall message throughput, latency and key set-up costs.

References

1. Biswas, K., Muthukkumarasamy, V., Sithirasenan, E.: Maximal clique based clustering scheme for WSNs. In: 8th IEEE ISSNIP, Melbourne, pp. 237–241 (2013)
2. Biswas, K., Muthukkumarasamy, V., Sithirasenan, E., Usman, M.: An energy efficient clique based clustering and routing mechanism in WSNs. In: 9th IEEE IWCMC, Italy, pp. 171–176 (2013)
3. Shazly, M., Elmallah, E.S., Harms, J., AboElFotoh, H.M.F.: On area coverage reliability of WSNs. In: 36th IEEE LCN Conference, pp. 580–588 (2011)
4. Computerworld Magazine: AES proved vulnerable by Microsoft researchers (2011)
5. Intel Corporation: Intel architecture software developer's manual (1997)
6. Biham, E., Birykov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. J. of Cryptology 18(4), 291–311 (2005)
7. Koo, W.K., Lee, H., Kim, Y.H., Lee, D.H.: Implementation and Analysis of New Lightweight Cryptographic Algorithm for WSNs. In: ICISA, pp. 73–76 (2008)
8. Yarrkov, E.: Cryptanalysis of XXTEA (2010), `http://eprint.iacr.org/2010/254.pdf`
9. Amara, M., Siad, A.: Elliptic Curve Cryptography and its applications. In: IEEE WOSSPA, pp. 247–250 (2011)
10. Canteaut, A.: Linear Feedback Shift Register. In: Encyclopedia of Cryptography and Security, pp. 355–358. Springer
11. Burke, J., McDonald, J., Austin, T.: Architectural support for fast symmetric-key cryptography. In: 9th ICASPLOS, pp. 178–189 (2000)