International Journal on Applications in Basic and Applied Sciences
Volume 8 : Issue 1 : October 2022, pp 32 – 37 www.aetsjournal.com ISSN (Online) : 2395 - 3527

Encryption Techniques in Wireless Sensor Networks- A Review

Dr. J. Senthil Kumar, Professor, Department of Information Technology, Sona College of Technology, Salem
Mr. R. Gowrishankar, Research Scholar, Anna University, Chennai

ABSTRACT

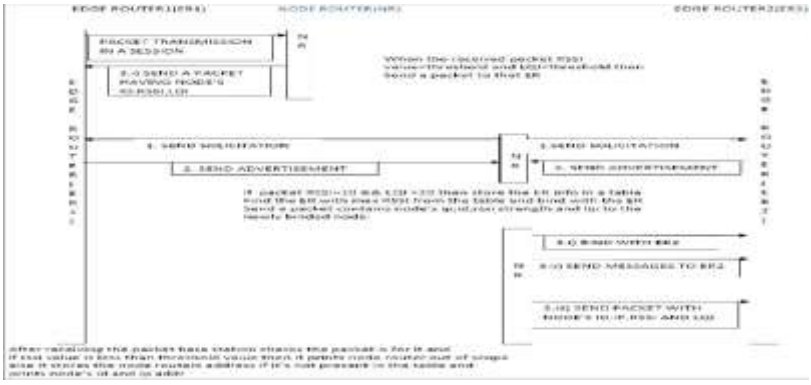
Generally Wireless Sensor Networks occur when sensor nodes are randomly left in an unreliable environment. Sensor node has limited processor, limited memory, and limited radio capacity at low cost. In sensor network applications, security mechanisms must be used, because of unsafe environments, large number of sensor nodes, and wireless communication environments. To ensure confidentiality, the primary goal of security, is one of the most important problems to be solved in order to realize time and vital objectives. While ensuring security, it is also necessary to consider other important criteria such as memory usage, energy and latency of Sensor Networks. In this study, encryption is described in Wireless Sensor Networks and Skipjack, XXTEA and AES encryption algorithms are compared and the results are analysed. The study is considered to be useful to academicians who study security in Wireless Sensor Networks.

INTRODUCTION

Compared to classic computer networks, wireless sensor networks show many unique features. Therefore, security methods that can be used in conventional networks cannot be used in wireless sensor networks [1]. In sensitive wireless sensor networks applications such as monitoring enemy lines or monitoring border zones, security protocols that provide confidential data transmission from the sensor nodes to the base station must be used. However, low processor and radio capacities of sensor nodes do not allow the implementation of traditional security protocols in sensor networks [2]. To make sensor networks practical, many improvements are currently being made in software and hardware. However, the basis of every work done is the reduction of the energy consumption per unit and hence the extension of the life of the sensor nodes. The main reason for this is that replacing or refilling sensor nodes and used energy sources in the working environment is often impossible and costly [3]. Security methods designed for traditional networks and widely used in many applications today cannot be implemented directly in wireless sensor networks because sensor nodes have limited energy resources, insufficient memory capacities and limited processing capabilities. Because of this, the security methods developed for wireless sensor networks take into account safety, energy, memory usage and latency. A security method in which all resources are actively used will be ideal for sensor networks [4]. In sensor networks, encryption algorithms are used to provide data confidentiality. However, it should not be forgotten that sensor networks have limited hardware resources and that the first objective is energy efficiency while providing requirements. Otherwise, a protocol that consumes a lot of energy, even if it provides all security measures, would be useless for sensor networks [5]. This work describes encryption in wireless sensor networks and uses Skipjack [6], XXTEA [7] and AES [8] encryption algorithms to examinememory usage, energy and latency criteria using the TOSSIM simulation program in TinyOS.

TinyOS & TOSSIM

The TinyOS Operating System is installed on the sensor nodes that compose the Wireless Sensor Networks. TinyOS is encoded in the programming language NesC, a variant of the C programming language. With this coding, nodes can be given new features. Designed algorithms or protocols can be seen primarily in the TOSSIM simulation program in TinyOS. TOSSIM is designed to simulate sensor network applications. TinyDB is a small database that is used to collect data and to read the data during simulation. TinyViz is a graphical visualization tool associated with TOSSIM, which is used to view the results of any NesC application. TinyViz interacts with TOSSIM, providing an extensible graphical user interface for debugging and visualization. TOSSIM is a simulation tool designed for TinYOS sensor networks. NesC codes written in TOSSIM also work on sensor node hardware without any change. Then, it can be loaded into the nodes without making any changes. The TinyOS operating system is designed to support the needs of wireless sensor networks. TinyOS is an embedded operating system distributed as completely free and open source for use in wireless sensor networks. TinyOS has a component-based architecture. In TinyOS, sending messages is via the AMStandard component. The encryption can be written as a module in a separate file. After the message package is prepared in the AMStandard component, it is transmitted wirelessly to the receiver via radio. The node that receives the message can still see the contents of the message package in the AMStandard component. This means that users who want to use different encryption algorithms or encryption modes will have to write their own algorithms or modes in this partition. The code written in this way is both modular and easy to use. The diagram showing the message transmission in TinyOS.



TinyOs- Message Passing

SKIPJACK

A symmetric cryptographic algorithm developed by the U.S. National Security Agency (NSA). It is used in the Department of Commerce's Escrowed Encryption Standard (EES), which was embodied in the CLIPPER chip.

The key to the encrypted message is itself encrypted with a key combined from two escrowed keys. The encrypted key and an identifier of the chip that sent it is encrypted again with a "family key." In this way, a law enforcement agency can use the family key to decrypt the outer layer and glean the chip ID, which is used to obtain the two escrowed keys that are combined to decrypt the key that decrypts the message. Skipjack uses an 80-bit key to encrypt 64-bit blocks, but algorithm details are classified.

XXTEA

In cryptography, **Corrected Block TEA** (often referred to as **XXTEA**) is a block cipher designed to correct weaknesses in the original Block TEA.

XXTEA is vulnerable to a chosen-plaintext attack requiring 2^{59} queries and negligible work.

The cipher's designers were Roger Needham and David Wheeler of the Cambridge Computer Laboratory, and the algorithm was presented in an unpublished technical report in October 1998 (Wheeler and Needham, 1998). It is not subject to any patents.

Formally speaking, XXTEA is a consistent incomplete source-heavy heterogeneous UFN (unbalanced Feistel network) block cipher. XXTEA operates on variable-length blocks that are some arbitrary multiple of 32 bits in size (minimum 64 bits). The number of full cycles depends on the block size, but there are at least six (rising to 32 for small block sizes). The original Block TEA applies the XTEA round function to each word in the block and combines it additively with its leftmost neighbour. Slow diffusion rate of the decryption process was immediately exploited to break the cipher. Corrected Block TEA uses a more involved round function which makes use of both immediate neighbours in processing each word in the block.

XXTEA is likely to be more efficient than XTEA for longer messages.

AES

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.
Points to remember

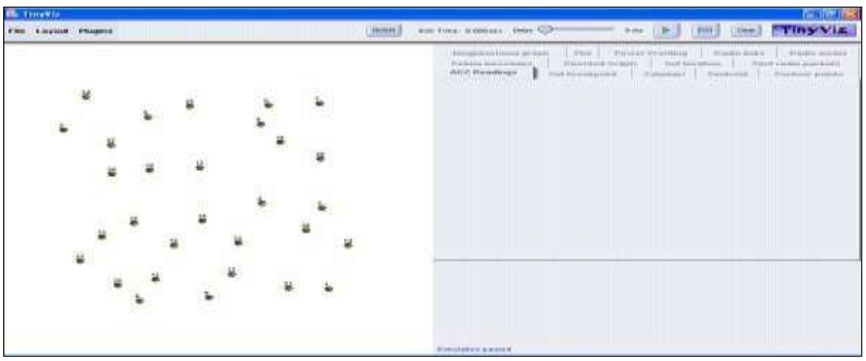
- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

Simulation Works

First, Skipjack, XXTEA and AES encryption algorithms are written in C language. The encryption of the message values used in the future TOSSIM simulation results and the calculation of the MAC value are performed in this program. Then, after verifying the codes of the encryption algorithms, this written code is translated into NesC code. Simulated results were obtained on 30 nodes with the help of the TOSSIM simulation program to test whether the NesC code written after this operation was performed correctly before loading the node. In this chapter, The message generated by the node, Message encryption and MAC calculation, Sending a message, Receiving message, Decryption of the password, Computation and comparison of MAC, If the MACs are equal, the process of accepting the message as shown below.

Fig. 2 Simulation – Tiny Os



According to the simulation program, node 19 will send message to node 15. First, node 19 generates a message after performing node detection. Then, it performs the encryption process before sending the message. The message to be sent on the MAC calculated with the encrypted message is added to the package. The encrypted message sent by node 19 is received by node 15. So even if the node is seized on the road, it will not get a meaning for the aggressor. The decryption of the message received by node 15 is resolved and the MAC value is calculated. When the calculated MAC value is the same as the incoming MAC value, the message is processed.

It is seen that the NesC code for the encryption algorithm written by TinyViz runs smoothly on the nodes. After this, the same code can be loaded in the sensor nodes without making any changes.

EVALUATION RESULTS

The performance of a cryptographic algorithm can be determined according to the length of the system breakable period, the time spent in ciphering and deciphering processes, and the amount of memory required for encryption and decryption operations [9]. For a 40-bit key $n = 2^{40}$ or n = 1 099 511 627 776 (one trillion ninety nine billion five hundred ten ten million six hundred twenty seven thousand seven hundred seventy six) is a possible key word. In a competition, a credit card transaction on the Internet, encrypted with a 40-bit key with the RC4 algorithm, was solved in 3 and a half hours by a student who have just a modest computer lab in 1995 [10]. Therefore, the key length used must be carefully selected.

It is concluded that a 64-bit algorithm can be used up to 1994, an 80-bit algorithm up to 2013, and a 128-bit algorithm up to 2076 can be used in the proportions made by knowing that the 56-bit DES algorithm introduced in 1977 is broken in 1982 [11]. The comparison of the encryption algorithms used in the study according to the block length, key length and number of cycles are given in Table 1.

Table 1 .Energy comparison of the encryption algorithms used

	Increased amount of memory usage compared to TinyOS (%)		Increased amount of latency compared to TinyOS (%)	Increased amount of energy compared to TinyOS (%)
	ROM	RAM		
Skipjack	4.32	14.23	4.7	5
XXTEA	4.34	14.32	5.1	6
AES	4.64	45.22	8.4	13

Although the key length of the XXTEA encryption algorithm is 128 bits, there is a negligible increase according to the Skipjack encryption algorithm according to the memory usage, energy and delay criteria. It would be better to use the XXTEA encryption algorithm instead of the Skipjack encryption algorithm, which has a key length of 80 bits, according to studies done in the literature. The AES encryption algorithm is more costly than other algorithms because of the specific methods it uses for encryption. However, the AES algorithm can be preferred in high-level applications that require much security.

CONCLUSIONS

When developing a security approach, it is necessary to consider the capacities of wireless sensor node resources (memory, processor, power supply). Sensor networks are expected to increase node energy consumption quantities and average end-to-end latency of encryption mechanisms added to improve security in applications. It is very important here that the needs of the applications are well established. In a simple large-scale environment or industrial sensor networks application, safety is of utmost importance, while energy consumption is a major factor. On the other hand, while security is of great importance in military and health applications, node energy consumption can be relatively neglected. Therefore, it is very important to select the encryption algorithm, encryption mode appropriate to the security solutions developed for military and health applications. Developed security solutions must be modular. That is, the newly introduced encryption algorithm and encryption modes of the literature should be integrated directly into the developed security solution if security, energy, memory usage, latency are better.

In this work, encryption is described in wireless sensor networks and Skipjack, XXTEA and AES encryption algorithms are compared using TOSSIM simulation program in TinyOS operating system considering memory usage, energy and delay criteria. XXTEA encryption algorithm is a negligible increase according to Skipjack encryption algorithm, but the result is that XXTEA encryption algorithm should be used instead of using Skipjack encryption algorithm with 80 bit key length. At the same time, it is concluded that the more costly AES encryption algorithm can be used in applications that require high security.

REFERENCES

1. C-Y. Chong, S.P. Kumar, "*Sensor Networks : Evolution, opportunities, and challenges*",Proc IEEE, **91(8)**, 1247-1256, (2003).
2. M. Dener, "*Security Analysis in Wireless Sensor Networks*", International Journal ofDistrubuted Sensor Networks, **2014**, Article ID 303501, 1-9, (2014).
3. R. Lin, Z. Wang, Y. Sun, "*Energy Efficient Medium Access Control Protocols forWireless Sensor Networks and Its State-of-Art*", IEEE, pp 669-674, (2004).
4. I.F. Akyıldız, W. Su, Y. Sankarasubramaniam, E. Çayırcı, "*A survey on sensor networks", IEEE Communications Magazine*, **40(8)**, 102-114, (2002).
5. T. Kavitha, D. Sridharan, "*Security Vulnerabilities in Wireless Sensor Networks: A Survey*", Journal of Information Assurance and Security, **5**, 31-44, (2010).
6. C. Karlof, N. Sastry, D. Wagner, "*Tinysec: a link layer security architecture for wireless sensor networks*", In n: SenSys '04: Proceedings of the 2nd international conference on Embedded Networked Sensor Systems, **1008**, 162– 175, ACM, New York, USA, (2004).
7. D. Wheeler, R.M. Needham, "*XXTEA: Corrections to XTEA*", Technical report,Computer Laboratory, University of Cambridge, (1998).
8. D. Joan, R. Vincent, "*The Design of Rijndael*", Springer-Verlag New York, Inc., Secaucus, NJ, USA, (2002).