CRYPTOGRAPHIC ALGORITHMS FOR WIRELESS SENSOR NETWORK

[1] Pinak M. Popat, [2] Pooja A. Vaishnav, [3]Ankita M. Parmar, [4]Bhumi K. Padodara

[1] PG Student (Department of Computer Engineering) Marwadi College, GTU, Rajkot, Gujarat, India.
[2] PG Student (Department of Computer Engineering) VVP Engineering College, GTU, Rajkot, Gujarat.
[3]PG Student (Department of Computer Engineering) VVP Engineering College, GTU, Rajkot, Gujarat.
[4]PG Student (Department of Computer Engineering) VVP Engineering College, GTU, Rajkot, Gujarat.

[1]pinak.popat@ymail.com, [2] pooja.vaishnav@ymail.com,[3]ankita.parmar2188@gmail.com, [4]bhoomi.padodara@gmail.com

***ABSTRACT*:** *A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on. Many algorithms are already developed for security in wireless sensor network but with many limitation .For instance key maintenance is a great problem faced in private key encryption methods and less security level is a problem of public key encryption methods even though key maintenance is easy .*

Keywords—Wireless Sensor Network (WSN), ultra-low power, security, cryptographic, primitives, Message Authentication Code (MAC), UMAC, OCB.

I: INTRODUCTION

A wireless sensor network (WSN) is one in which tiny devices (sensor nodes), positioned fairly close to each other, are used to sense and gather data from their environment and to exchange information through wireless connections between these nodes. Apart from the built-in sensors, these sensor nodes also have wireless transceivers and power sources built-in allowing them to work autonomously. Sensor networks have been undergoing extensive research and studies in recent years because of their various potential applications, such as monitoring the safety and security of buildings or homes (intelligent buildings and homes), measuring traffic flows, tracking environmental pollutants, monitoring factory instrumentations, monitoring temperature and lightings on a farm or in a greenhouse. Sensor nodes can even be distributed throughout a bridge allowing them to continuously sense and monitor the mechanical stress level of the bridge. In order to easily deploy a relatively large number of sensor nodes, the sensor nodes are typically designed for low price, small size and long operation life, which causes them to have very limited resources available (e.g. energy, processing power and memory size).When speaking of the security of any system, it can be categorized into three main concerns: Confidentiality, Integrity and Authenticity (or sometimes Availability) (C.I.A.).Security primitives are used to achieve these basic concerns. For example, encryption algorithms are used to achieve confidentiality, while cryptographic hash functions or message authentication codes (MAC) are used for achieving integrity and authenticity. Over the years, different security primitives have been proposed and refined aiming at utilizing modern processing power e.g. 32-bit or 64-bit systems, SIMD (Single Instruction Multiple Data) architecture such as MMX (Multi Media Extension) etc. In other words, security primitives have targeted the high-end systems (e.g. desktop or server) in software implementations. Several hardware-oriented security primitives have also been proposed. However, most of them have been designed aiming only at large messages and high-speed processing, with no power consumption or other resources (such as memory space) taken into consideration. As a result, security mechanisms for ultra-low power devices such as wireless sensor nodes must be carefully selected or designed with their limited resources in mind. Ultra-low power at the moment is typically referring to power consumption less than 500μW.

II:SECURITY IN WIRELESS SENSOR NETWORK

Security in any network system does not simply involve only one or two layers, but rather needs to be viewed across all layers as a whole. The security issues for a conventional network differ greatly to the security issues in WSNs because of the extremely limited resources available in sensor nodes. This chapter provides an overview of security considerations in the context of the WSN.

Trust Models

One or more base stations often exist in WSN. Base stations are more powerful nodes with rich computational, memory, energy and radio resources. By radio resources it means that they have more powerful transceivers for a wider communication range and higher bandwidth links for communication amongst other base stations. A base station may exist in the form of a PC or server, where the sensor data flows to and is stored. Therefore they are also known as sink nodes. Base stations may act as a gateway between WSN and another network; therefore may be connected to an outside TCP/IP network. These resourceful nodes are sometimes also known as rich uncles [5]. Base stations are more expensive nodes, and are often assumed to be physically protected or have tamper-proof hardware.

As a result, in a WSN environment, a base station usually plays the role of a central trusted authority (point of trust). A point of trust base station is what the other standard sensor nodes trust for its authenticity and accepts the keys managed by the base station. In a base station trust model, for two nodes to communicate directly with each other, they need to first rely on the base station to establish a shared secret key between them before communication can take place. However, scalability may become a problem for base stations. If every sensor node in the network has a unique secret key, then for two nodes to communicate with each other they need to first go through the trusted base station to establish a shared secret key. If every node needs to communicate with its neighboring nodes, then the base station becomes a scalability bottleneck. This paper also assumes the base station as the trusted authority in the trust model.

III: CRYPTOGRAPHIC CIPHERS

Cryptographic ciphers often provide the most basic security requirements such as confidentiality ,authenticity and integrity checking in any system. However, not all cryptographic ciphers that are suitable for conventional networks will also be suitable for WSNs. This chapter discusses security primitives through the use of cryptographic ciphers and their applicability to the ultra-low power WSN environment. The background of block ciphers as well as modes of operation are investigated and discussed here. The only stream cipher implemented in this paper, RC4, is also discussed here.

TEA

TEA (Tiny Encryption Algorithm) [3] and its related variants (XTEA, Block TEA, XXTEA) are symmetric key block ciphers designed for modern 32-bit word architecture.The emphasis of TEA is on small code size and easy implementation with typically few lines of codes. It uses a large number of iterations rather than a complicated algorithm. All TEA and its variants are based on the Feistel structure, every TEA cycle consists of two Feistel rounds (Figure 4.1).TEA and XTEA operate on two 32-bit words as a 64-bit data blocks with a 128-bit key, therefore all operations are done in 32-bit words. Block TEA and XXTEA operate on variable-length blocks of arbitrary multiples of 32 bits size. The advantage of Block and XXTEA is that it eliminates the need for using a mode of operation (CBC, OFB, CFB, OCB etc.) on messages larger than one block. i.e. they can be applied directly to a complete message.

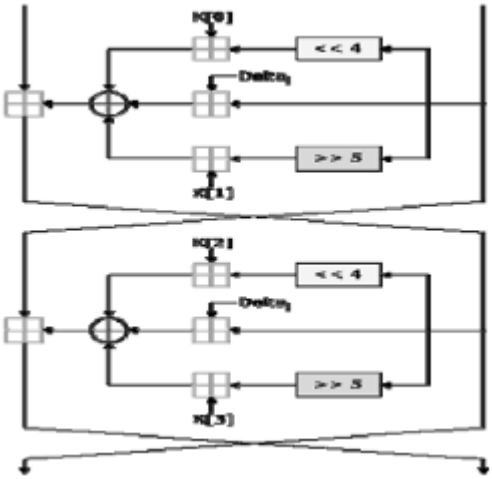

Fig.1 One TEA cycle (two Feistel rounds) [3].

Cryptanalysis of TEA

TEA suffers from two types of cryptanalysis, the related-key [9] and equivalent-key [8] attacks. The equivalent-key attack is targeted at TEA's extremely simple key-schedule.This results in the problem that when flipping the most significant bits of the first two 32-bit words of the key, the encryption will not be affected. This attack has allowed hackers to successfully run Linux operating system on the Microsoft's Xbox gaming console. The best related-key attack on TEA requires 223 chosen plaintexts and 232 computation time to recover the key. XTEA is proposed by TEA designers to prevent weaknesses found in TEA. The best attack so far on XTEA is a related-key differential attack on 27 rounds [10].This attack requires 220.5 chosen plaintexts and has a time complexity of 27-round XTEA encryptions.for minimum of drag. Here inlet velocity is taken as 40 m/s and k-e turbulence model is selected for capturing turbulent motion of vortices [5].

SAFER K-64

SAFER K-64 [4] (stands for Secure And Fast Encryption Routine with a Key of length 64 bits) is a non-proprietary secret (symmetric) key block cipher. The block length is 64 bits (8 bytes) and only byte operations are used for key scheduling, encryption and decryption.The encryption structure of SAFER K-64 is shown in the following figure

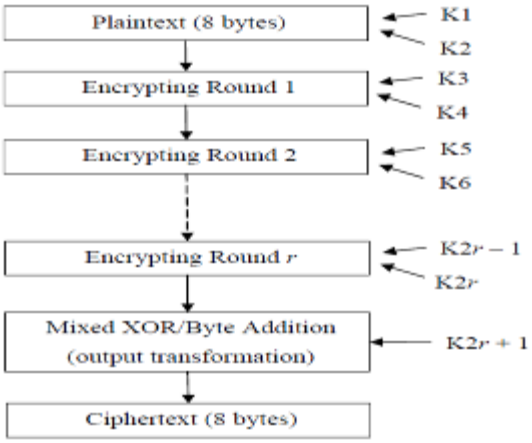

Fig 3. Encryption structure of SAFER K-64.

The encryption/decryption algorithm consists of *r* rounds, typically 6 rounds are recommended. Each round (shown in Figure 4.4) requires two 64-bit (8 bytes) subkeys and the output transformation needs one 64-bit subkey. In total 2*r* + 1 subkeys are needed, which is derived from the user-selected secret key "K1". The output transformation involves byte XOR and byte addition (modulo 256) of the last subkey (K2*r* + 1) with output from the *r*-th round. The decryption structure is similar to the encryption structure except that the output transformation now becomes the input transformation and is executed first. The subkeys in the decryption structure are also used in a reversed order.

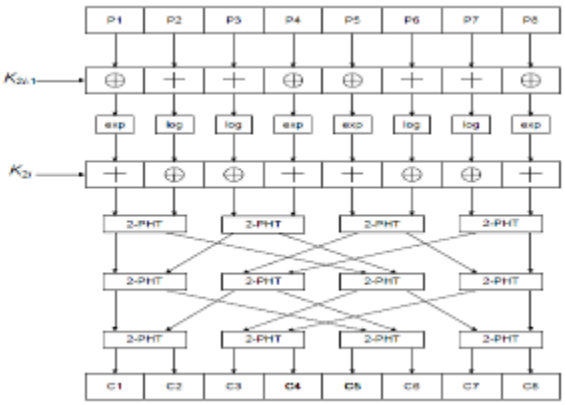

Fig 4 One encryption round structure of SAFER K-64.

TREYFER

TREYFER is a 64-bit block cipher with 64-bit symmetric key and is proposed by Yuval [4]. It is aimed at applications with extremely limited resources, e.g. smart card and is designed to be very compact (less than 50 bytes of code on an 8051 microcontroller with assembler language). It can be executed on a very constrained architecture, for example an 8051 microcontroller with typically 1 KB flash EPROM, 64 bytes RAM, 128 bytes EPROM and a peak instruction rate of 1 MHz. TREYFER is designed to use only byte operations and requires fixed bit rotations and modulo 256 additions. The algorithm is as

```
for (r = 0; r < NumRounds; r++){
text[8] = text[0];
for(i = 0; i < 8; i++)
text[i+1] = (text[i+1] +
Sbox[(key[i]+text[i])%256]) <<< 1;
//rotate 1 left
Text[0] = text[8];
}
```

In the above pseudo code, "text" represents the 8-byte plaintext, "Sbox" is the 256×8-bit (256 bytes) S-box chosen at random, and "NumRounds" is the number of rounds executed in TREYFER, which is typically 32. One of the motivations of the TREYFER design is the use of a large number of rounds (32) to thwart any possible practical attacks in spite of the simple round function design. The S-box was suggested by the author to be taken from another place in the memory running non-cryptographic codes. In this way there is no need to explicitly define a 256-byte S-box and thus code space is saved.

AES

AES (Advanced Encryption Standard) was published by NIST (National Institute of Standards and Technology) to replace DES (Data Encryption Standard). Out of the many candidates for AES, the Rijndael cipher was eventually selected to become the new AES [8]. AES is a symmetric key block cipher with a block size of 128 bits and three key size alternatives of 128, 192, or 256 bits. Unlike many conventional symmetric key block ciphers, AES does not use the Feistel structure, where typically half of the data block is used to modify the other half of the data block before the two halves are swapped in the next round. AES processes the entire data block (128 bits) in parallel during each round. AES typically has 10 rounds; each round has four different stages, one of permutation and three of substitution. The encryption and decryption functions in AES differ. The encryption and decryption speed does not vary significantly, however, the key setup performance is slower for decryption and requires more memory than for encryption. All AES operations can be byte operations allowing it to be efficiently implemented on 8-bit processors. Its operations can also be defined in 32-bit words for efficient implementation on 32-bit processors [2].Although AES has been well studied over the years and proven to be secure, it does not seem to be suitable for the platform which this paper is based on, or in many other WSN environments. One of the main reasons is that although AES has been designed for lowend 8-bit microcontroller, its baseline version still uses over 800 bytes of look-up tables. A speed optimized AES version, which runs about 100 times faster, uses over 10 KB of lookup tables. This memory requirement is not acceptable to many sensor node platforms. For example, the microcontroller MSP430F1232 used in the sensor

nodes (TinyMote) of this paper has only 8KB of flash code memory in total. Apart from the large code size, AES also requires large RAM space to store expanded subkeys, typically larger than 156 bytes. Furthermore, because of the small packet size of WSN, a cipher with 128-bit (16 bytes) block size may not be very efficient. For example the last cipher call may only need to encrypt the last two bytes of the data packet, since the cipher uses 16-byte block.

RC5

RC5 is a symmetric encryption algorithm with a block size of 32, 64, or 128 bits [2]. The key length ranges from 0 to 2040 bits. RC5 encrypts two-word blocks, for example a 32-bit block has a word size of 16-bit. The maximum number of RC5 rounds is 255, but typically 12 rounds encryption/decryption algorithm is suggested.RC5 has a simple structure similar to a Feistel structure. Instead of half of a block being updated as in the classic Feistel structure, both halves are updated in each RC5 round [6]. RC5 uses only three primitive operations: modulo 2naddition/subtraction (n is the word size), XOR, and circular rotation. The encryption/decryption algorithm is very simple and can be implemented in few lines of codes. These characteristics make RC5 suitable for both hardware and software implementations. RC5 requires complex key expansion operations on user-selected secret keys. The number of sub keys that are needed is 2r + 2, where r is the total number of rounds .RC5 has also been around for some years and appears to be secure. Although it was designed to be of small size for efficient software and hardware implementation, its smallest word size is still 16-bit. The key setup operations have been shown to be very time consuming [5] and also require a relatively large amount of RAM space to store the expanded subkeys [11]. Furthermore, RC5 rotation operations are data-dependent, meaning that it has to rotate variable number of bits and often requires a large number of bit rotations. This large number of bit rotations is especially time consuming for processors with a word size smaller than that of the RC5 word size (e.g. 16-bit RC5 word on an 8-bit processor).Law et. al. [5] have compared RC5, AES and several other block ciphers on the same family of microcontrollers (TI MSP430) as the one used in this paper. These comparisons have shown that RC5 is not the most efficient cipher nor does it have the smallest code size..

III. Conclusion

The well known AES and the WSN-popular RC5 block ciphers have been shown to be not very suitable for WSN. The block cipher SAFER K-64 has been investigated for the first time for its applicability in WSN. Compared to other block ciphers investigated for WSN environment, SAFER K-64 achieves the best performance in CPU usage known to the author. It, however, requires slightly more RAM. XTEA requires a fairly small amount of flash/ROM memory and no RAM is needed for the subkeys setup. Even though XTEA is designed for a 32-bit architecture, it performed well on the 16-bit MSP430 platform and outperformed both AES and RC5 on the same MSP430 platform. Although TREYFER requires the least flash memory and also does not need RAM for the subkey setup, it requires a considerable number of CPU cycles.

REFRENCES

[1] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J.D.Tygar, "SPINS: Security Protocols for Sensor Networks", *Proceedings of 7th Annual International Conference on Mobile Computing and Networks (Mobicom)*, pp. 189-199, Rome, Italy, 2001.

[2] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, *Handbook of Applied Cryptography*, 5th ed., CRC Press, 1996.[3] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", *Proceedings of the 1st IEEE International Workshop on SensorNetwork Protocols and Applications (SPNA)*, pp. 113-127, Anchorage, USA, May 2003.

[4] P. Ganesan, R. Venugopalan, P. Peddabachagari et. al., "Analyzing and Modeling Encryption Overhead for Sensor Network Nodes", *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, San Diego, USA, 2003.[5] Y. Law, S. Dulman, S. Etalle et. al., "Assessing Security-Critical Energy-Efficient Sensor Networks", Department of Computer Science, University of Twente, Tech.Rep. TR-CTIT-02-18, 2002.

[6] C. Karlof, N. Sastry and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks", *Proceedings of the 2nd ACM Conference onEmbedded Networked Sensor System (SenSys)*, vol. 47 issue 6, Baltimore, USA, November 2004.

[7] S. Mahlknecht, "Energy-Self-Sufficient Wireless Sensor Networks for the Home and Building Environment", Doctor's thesis, Technical University of Vienna, 2004.[8] H. Y. Yang, H. Luo, F. Ye et. al., "Security in Mobile Ad Hoc Networks: Challenges and Solutions", *IEEE Wireless Communications Magazine*, pp. 38-47, February 2004.

[9] E. Shi and A. Perrig, "Designing Secure Sensor Networks", *IEEE WirelessCommunications Magazine*, pp. 38-43, December 2004.

[10] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, pp. 102-114, August 2002.

[11] Y. Chun Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks", *Proceedings of 8th Annual International Conference on Mobile Computing and Networks (Mobicom)*, Atlanta, USA, September 2002.