Public Key Cryptography in Resource-Constrained WSN

Al-Sakib Khan Pathan and Choong Seon Hong
Department of Computer Engineering, Kyung Hee University, South Korea
spathan@networking.khu.ac.kr, cshong@khu.ac.kr

Abstract

In this paper we present a detailed review of the works on public key cryptography (PKC) in wireless sensor networks (WSNs). In the early days of sensor networks, public key cryptography was thought to be completely unfeasible considering its computational complexity and energy requirements. By this time, several works have proved that the lightweight versions of many well-known public key algorithms can be utilized in WSN environment. With the expense of a little energy, public key based schemes could in fact be the best choice for ensuring data security in high-security demanding WSN applications. Here, we talk about the notion of public key cryptography in WSN, its applicability, challenges in its implementation, and present a detailed study of the significant works on PKC in WSN.

1. Introduction

Security in wireless sensor network (WSN) has a great number of challenges, ranging from the nature of wireless communications, constrained resources of the sensors, unknown topologies of the deployed networks, unattended environment where sensors might be susceptible to physical attacks, dense and large networks, etc. Each of these issues in fact leads to different research direction. Nonetheless, considering all the different topics of research in WSN security, Public Key Cryptography (PKC) most probably is the hottest topic in recent times.

This is generally perceived that PKC is complex, slow, resource hungry, and as thus not at all suitable for use in ultra-low power network environments like wireless sensor networks. It is therefore a common practice to emulate the asymmetry of traditional PKC services by using a set of protocols based on symmetric key cryptography (SKC). The main reason for using SKC is that it is comparatively less resource-hungry than PKC. However, with respect to key management, SKC could be inflexible and more difficult than PKC. This is because often the keys for the sensors need to be generated in huge numbers and in many cases they need to be stored in the memories of the sensors prior to their deployment. Such type of pre-distribution of keys could really be cumbersome when a large sensor network is to be deployed.

In contrast to the commonly held belief of the inefficiency and inapplicability of PKC in WSN, some researchers have come forward and proved that the lightweight versions of many well-known PKC schemes can be applied for tiny low powered sensors.

For implementing PKC-based schemes in WSN, there are mainly two approaches:

a) Design and development of customized hardware so that they could be used in the sensor boards for supporting public key based operations and computations.
b) Writing customized software or program that could support PKC operation or using lightweight versions of the PK based schemes.

A combination of hardware support and software optimization also works effectively for such implementations. So far, both from hardware and software perspectives, some works have already been done [1]. This paper presents a detailed survey on the applicable PKC-based schemes in WSN. The main objective of this work is to get a broad picture of the current research trend in applying high-standard cryptographic schemes in wireless sensor networks.

The rest of the paper is organized as follows: Section 2 briefly mentions the pros and cons of implementing PKC in WSN, Section 3 presents the detailed survey, Section 4 discusses the future expectations, and Section 5 concludes the paper.

2. PKC in WSN: Pros and Cons

The sensors that build up the network are usually of inadequate memory, processing, and communication capabilities and their energy sources are also very limited. As an example, Crossbow MICA2 mote [2] is

a well-known sensor node with an ATmega128L 8-bit processor at 8 MHz, 128KB program memory (flash), 512KB additional data flash memory, 433, 868/916, or 310 MHz multi-channel radio transceiver, 38.4 kpbs radio, 500-1000 feet outdoor range (depending on versions) with a size of only 58 x 32 x 7 (mm). Usually it is run by TinyOS operating system and powered by 2 AA sized batteries. A device with this configuration cannot support security mechanisms that require executing a large amount of instructions. In addition, a sensor network usually contains a large number of sensors. The number of sensors in the network might directly affect the use of memory space of nodes, because often they store pre-distributed secret keys, keying information, or the codes to calculate pairwise secret keys between nodes in the network. Node failure is another problem that could also affect the network severely. If a node is busy relatively longer than other nodes (e.g., performing huge calculations related to security), it might lose its energy rapidly and can fail much sooner than other less active nodes.

Most of the sensor network applications need at least a certain level of security for the communications among the sensor nodes and the base station (BS). The minimum requirements for any type of secure communication are: data privacy, integrity, and authenticity. All of these could be provided by using efficient cryptographic mechanisms. SKC is usually viable on sensor nodes but the size and scalability issue sometimes makes its use inefficient. Again, because of the unattended feature of WSNs, often these networks are vulnerable to physical capture attack. If the nodes carry sensitive key information within themselves, the attackers could get that after physically capturing the sensors (if any other preventive mechanisms are not used). Considering this particular issue, it is much more efficient to use PKC as in this case the nodes only carry public key materials of the BS instead of private secret keys.

3. Review of PKC-Based Schemes in WSN

Most of the works related to PKC in WSN are conducted to fit the low-power characteristic of the sensor nodes. Both from software and hardware perspectives, PKC-based schemes have shown reasonable performances. In this section, we present a detailed study on these exclusive research works.

3.1. ECC and RSA on 8-bit CPUs

Gura et al., in their work [3], present a comparative analysis of Elliptic Curve Cryptography (ECC) and RSA on 8-bit CPUs. The authors note that they implemented ECC point multiplication and modular exponentiation on two exemplary 8-bit platforms in assembly code. As the first processor, they chose a Chipcon CC1010 8-bit microcontroller which implements the Intel 8051 instruction set. The CC1010 contains 32KB of FLASH program memory, 2KB of external data memory, and 128 bytes of internal data memory. As the second processor, they took an Atmel ATmega128 processor which is frequently used for sensor network research (for example in Crossbow motes). The ATmega128 is an 8-bit microcontroller based on the AVR architecture and contains 128KB of FLASH program memory and 4KB of data memory. For ECC, they implemented point multiplication for three SECG-standardized elliptic curves with some optimizations. For RSA, they implemented RSA-1024 on both of the chosen processors and RSA-2048 on the ATmega128 with some optimizations. Their experimental findings show that on both platforms, ECC-160 point multiplication outperforms the RSA-1024 private-key operation by an order of magnitude and is within a factor of 2 of the RSA-1024 public-key operation. In summary, Gura et al.'s work for the first time gave the practical idea about the applicability of PKC algorithms in small resource-constrained devices.

3.2. PKI for Key Distribution

[4] presents the first known implementation of elliptic curve cryptography over F_{2p} for sensor networks based on 8-bit, 7.3828 MHz MICA2 mote [2]. In this work the authors first demonstrate that the secret-key cryptography is tractable on MICA2 by instrumentation of TinyOS. Then with their method of implementation of multiplication of points on elliptic curves, they argue that PKI (public key infrastructure) for distribution of secret keys is also tractable. The results show that public key based scheme is viable for the modern-era sensors. Breaking the myth of the inapplicability of public key cryptography in sensor networks, this work shows that public keys can be generated within 34 seconds, and that shared secrets can be distributed among nodes in a sensor network within the same, using just over 1 kilobyte of SRAM and 34 kilobytes of ROM, which could easily be provided by the state-of-the-art sensor nodes.

3.3. Rabin's Scheme and NtruEncrypt

In [5], the authors propose a custom hardware assisted approach which makes PKC feasible in WSN.

In order to validate their claim, they present proof of concept implementations of two different algorithms; Rabin's Scheme and NtruEncrypt.

Rabin's Scheme was proposed in 1979 in [6]. It is based on the factorization problem of large numbers and is therefore similar to the security of RSA with the same sized modulus. The size of the modulus determines the security of the cipher. The disadvantage of RSA is that the algorithm necessary to implement both encryption and decryption is computation intensive. Rabin's scheme has asymmetric computational costs. It is only necessary to perform a simple modular squaring operation to encrypt a message. Therefore encryption can be preformed relatively efficiently, while the computational cost of the decryption algorithm is comparable to RSA. This asymmetrical property of Rabin's scheme is of significant practical importance in applications such as wireless sensor networks where encryption must be done on very low powered devices. On the other hand, NtruEncrypt [7] claims to be highly efficient and particularly suitable for embedded applications such as smart cards or RFID tags, while providing a level of security comparable to that of other established schemes, in particular RSA.

Taking these asymmetric encryption techniques, the authors analyze the viabilities based on chip area, security level, delay, average power, energy per bit, and throughput. Overall, this work shows that it is possible to design PK encryption architectures with power consumption of less than 20 μW using the right selection of algorithms and associated parameters, optimization, and low-power techniques. This work also notes that it might be possible to achieve even better performance for self-powered sensors as this work was based on a regular ASIC (Application-Specific Integrated Circuit) standard cell library that was not specifically optimized for low-power.

3.4. TinyPK

TinyPK system demonstrated in [8] shows that a PK-based protocol is feasible even for an extremely lightweight sensor network. TinyPK is a software-based implementation of public key system tested on UC Berkeley MICA2 motes [2]. Incorporating the use of TinySec [9] or any other symmetric encryption service for sensor networks, TinyPK provides the functionality needed for a sensor and a third-party to mutually authenticate each other and to communicate securely. The mote authentication technique, proposed in this paper can be used either to collect evidence that a mote field is valid or to validate a specific mote and to link future traffic to that mote. The notion of a mote credential makes the task of a spoofing a mote network much more difficult than in an environment without credentials. In a TinyPK protected environment, a single stolen and reverse engineered mote cannot be used to impersonate other motes with different credentials. This level of protection is achieved with very little overhead and has been shown to operate on the most limiting of sensor network platforms.

3.5. Authenticating Public Keys

[10] investigates how to replace the public key authentication with symmetric key operations that are much more efficient. The authors show that due to a unique property of sensor networks, public keys do not need to be authenticated in the same way as it is done in the Internet environment (i.e., using certificates); instead, public keys can be authenticated using one-way hash functions, which are much more efficient than signature verification on certificates. Their scheme uses all sensors' public keys to construct a forest of Merkle trees of different heights. By optimally selecting the height of each tree, they can minimize the computation and communication costs. They also develop a trimming scheme based on sensor deployment knowledge. The results show that their scheme can save up to 86% of the energy for the public key authentication operation in WSN.

3.6. Energy Analysis of PK Algorithms

In [11], the authors present an analysis of energy consumptions of PKC schemes in wireless sensor networks. They also consider the impact of pubic key cryptography on battery life and compare PKC to other factors influencing energy consumption, such as idle listening, data reception and transmission, symmetric cryptography, etc. They quantify the energy costs in an 8-bit microcontroller platform with RSA and Elliptic Curve Cryptography (ECC) (energy costs of digital signature and key exchange computations for RSA-1024, ECDSA-160, RSA-2048, and ECDSA-224). This work shows that: with a given amount of energy, the authors were able to perform 4.2 times the number of key exchange operations (including mutual authentication) with ECC-160 compared to RSA-1024. Overall, this work presents a detailed picture of energy requirements for PKC operations in resource

constrained sensors which could be used as a good reference paper for other similar works.

3.7. Ultra-Low Power PKC for WSNs

[12] shows that special purpose ultra-low power hardware implementations of public key algorithms can be used on sensor nodes. The authors in this work selected three low-complexity PKC schemes (Rabin's scheme, NtruEncrypt, and Elliptic Curve) and for each of the schemes they developed three basic encryption architectures in TSMC 0.13µ CMOS standard cell technology. For comparing the inherently different algorithms and their feasibility for ultra-low power implementations, they chose algorithm specific parameter sets to provide approximately the same level of security. For Rabin' scheme, a modulus of 512 bits was selected, which generally provides a security level of around 60 bits. In ECC architecture, for arithmetic operations, a prime field of 100 bits in size was chosen which could provide a security level between 56 to 60 bits depending on the confidence level one puts into the assumption that no significant cryptanalytic progress has been made. Finally, for NtruEncrypt, system parameters $(N, p, q) = (167, 3, 128)$ were chosen to get a security level of around 57 bits. After analyzing the architectures in conjunction with the full algorithm descriptions, they estimated the overall power and bandwidth requirements of encryption and signature primitives.

Overall, in this work, the authors show that PKC tremendously simplifies the implementation of many typical security services and additionally reduces transmission power due to less protocol overhead.

3.8. Implementations of ECC-based PKCs

Blaß and Zitterbart [13], in their work present efficient and lightweight implementations of PKC algorithms relying on elliptic curves. They checked their codes by running on popular 8-Bit ATMEGA128 microcontroller which is used for MICA2 platform.

For the lightweight implementation of ECC, one of the most important factors is the key size. In case of ECC, key size means the size of underlying finite field that is; if 53 bit keys are to be used, the elliptic curve must be over $F_{2^{53}}$. If the key size is smaller, it provides less security but faster computation. After a detailed primary analysis, the authors chose 113 bit key that means a curve over $F_{2^{113}}$. The argument for choosing this curve was that it could offer about 16 times more security than 109 bit keys which could be considered as enough security for today's hardware. For making the entire implementation easier and lightweight, they also considered some other optimizations for memory savings, point multiplications, handcrafting a source to the target platform, sophisticated loop-unrolling, etc.

This work presents the results of the implementations of various ECC-based algorithms; ECDH (Elliptic Curve Diffie-Hellman), El-Gamal (which relies on traditional discrete logarithm problem and that has been adopted to elliptic curves and ECDLP), and ECDSA (Elliptic Curve Digital Signature Algorithm).

3.9. Sizzle

In [14], the authors show that ECC not only makes public-key cryptography feasible on "mote"-like, embedded devices, but also it allows one to create a complete secure web server stack that runs efficiently within very tight resource constraints. They present their HTTPS stack, named Sizzle which has been implemented on multiple generations of the Berkeley/Crossbow motes where it runs in less than 4KB of RAM, completes a full SSL handshake in 1 second (session reuse takes 0.5 seconds), and transfers 1 KB of application data over SSL in 0.4 seconds.

Sizzle brings the Internet's dominant security protocol (SSL) to devices with significant computational, memory and energy constraints. It uses highly optimized implementations of PKC to offer scalable key management and end-to-end security without sacrificing efficiency. Sizzle running on the Berkeley/Crossbow Mica2dot mote represents the world's smallest secure web server in terms of both physical dimensions and resource utilizations.

3.10. C4W: Identity-Based Public Key Infrastructure for WSN

C4W presented in [15] is basically an identity-based public key infrastructure specially designed for WSNs. Usually any Identity-Based Cryptosystem (IBC) requires heavy computations. However, to reduce the processing burdens, the IBC algorithm of C4W is made lightweight based on ECC optimization techniques in [3] and Combined Public Key (CPK) cryptosystem presented in [16]. The authors in this work show that their identity-based scheme consumes

less energy as it is certificateless and thus it is efficient in terms of computation and communication costs.

3.11. Cooperative Public Key Authentication Protocol in Wireless Sensor Network

A distributed and cooperative PK authentication is proposed in [17]. This work is mainly a theoretical work without practical implementation. To facilitate the proposed protocol, some modifications in the MAC frame are needed. In this cooperative mechanism, each node stores a limited number of hashed keys for other nodes which help in the authentication procedure during public key operation. According to [17], this scheme is free from any cryptographic operations, which is designed to make it fit for the constrained resources of the sensors. However the major drawback of the proposed method is that it is designed only for one hop authentication, which makes it impractical and inefficient for use in usual multi-hop WSNs. However, this scheme could be used for individual clusters with cluster heads, if clustering is used in the network. Hence, there remains enough scope of further investigation and enhancement of the protocol to make it practical for traditional WSNs.

3.12. Implementation of Rabin's Scheme Based Scheme on Tyndall National Institute Mote

Murphy et al. [18] show that it is possible to implement PK algorithms on resource constrained sensor node platforms. Using a hardware/software co-design approach, they successfully map a public key cryptosystem based on Rabin's scheme onto the motes developed by Tyndall National Institute. The implementation was prototyped on 25mm cube modules provided by the Institute [19]. These cubes consist of several different programmable modules that are interchangeable. The cubes were configured so that each node included a low power 8-bit Atmel microcontroller, a Spartan-IIE FPGA, and a RF transceiver. For their implementation, the different modules were synchronized by a shared 4 MHz clock.

Their implementation mainly focused on efficient architectures that execute the public key algorithms using minimal resources. Their finding is that the hardware implementation of the encryption algorithm is much faster than the software implementation. Software implementations of the algorithm are also realizable and have the benefit of low cost and high flexibility. However the time necessary to perform encryption and decryption is significantly increased by using a software-only approach.

3.13. Influence of PKC on Sensor Lifetime

Piotrowski et al. [20] investigate four types of nodes; MICA2DOT, MICA2, MICAz, and TelosB and estimate the power consumptions for RSA and ECC operations. Their work presents detailed results of implementations of RSA-1024, ECC-160, RSA-2048, and ECC-224 on the mentioned platforms.

Based on the results, the authors conclude that: transmission power is not an important factor when comparing cryptographic algorithms. Even sending a 2048 bit RSA signature by a transceiver that requires 1.0 μWs/bit, needs not more than 2 mWs for one signature. This is at least one order of magnitude less than the energy consumption required for computation of the cryptographic operations. However, for large multi–hop networks, it might become a factor. In that case a large signature increases the overall transmission power consumption in the network.

3.14 Implementing Minimized Multivariate PKC on Low-Resource Embedded Systems

In [21], the authors implement minimized multivariate PKC on Low-Resource Embedded Systems. They illustrate most of their minimization techniques on a current variant in the family of multivariate PKC called the Enhanced TTS (enTTS). TTS [22] is a consequence of the public-key cryptosystem TTM (Tame Transformation Method) and shares many of its superior properties, resulting in low signature delays, fast verification, and high complexity. To evaluate the performance of enTTS on modern sensor nodes, they benchmark enTTS on the Tmote Sky mote. Tmote Sky is equipped with an 8 MHz Texas Instrument MSP430 microcontroller and a Chipcon CC2420 2.4 GHz radio that supports IEEE 802.15.4 wireless low-power medium access control standard. A typical sensor node like the Tmote Sky mote has a small working RAM (10 KBytes in this case), a slightly larger read-only program memory (48 KBytes), and a relatively large flash memory (1 MBytes) for storing collected raw data and other auxiliary information for its operations. The results from this work show that multivariate schemes could be better contenders against the established PKC schemes if they are a lot better customized and optimized for use in wireless sensor networks.

3.15. Low-cost ECC for WSNs

In [23], the authors present a low-cost PKC-based solution for security services such as key-distribution and authentication for WSNs. They propose a custom hardware assisted approach to implement ECC with the goal of obtaining stronger cryptography and minimizing the power consumption.

They also present the details of their designed Elliptic Curve Processor (ECP). Their ECP has the operational blocks: a Control Unit (CU), an Arithmetic Unit (AU), and Memory (RAM and ROM). In ROM, the ECC parameters and some constants could be stored. The RAM contains all input and output variables and it communicates with both the ROM and the ALU. The CU controls the scalar multiplication and the point operations. In addition, the controller commands the ALU which performs field multiplication, addition, and squaring.

With this hardware assisted approach, the authors present their analysis of ECC algorithm and some positive results to allow the use of PKC in WSNs.

3.16. Power Aware Design of an Elliptic Curve Coprocessor for 8-bit Platforms

A work on hardware implementation of PKC for elliptic curve over binary extension fields is proposed in [24]. The goal of this work is to evaluate the energy costs of different ECC implementations for low-end systems, and to propose a new hardware coprocessor architecture. In their implementation of the dedicated coprocessor, the operands had a size of 163 bits, while the existing data path was of 8 bits only. They designed two different hardware devices in VHDL, and synthesized them using the 0.18 μm CMOS technology library by ST Microelectronics. Using the Synopsys tools Design Compiler and PrimePower, they obtained the area occupancy, the critical path, and the power consumption for each implementation.

By using their coprocessor, they show that achieving the goals of low energy consumption, reduced silicon area requirements, and significant speed-up (compared to software solutions) are possible. This could be a feasible hardware assisted solution to get improved performance of ECC implementations in WSNs without degrading other performance parameters.

3.17. Self-Certified Public Key Generation on the Intel Mote 2 Sensor Network Platform

[25] presents an efficient ECC-based method for self-certified key generation in resource-constrained sensor nodes. In particular, the authors provide implementation results on the Intel Mote 2 sensor network platform [26] which demonstrate that such key generation can be established in the order of 60 msec while consuming less than 30 mJ.

The methodologies developed by the authors were implemented on Intel Mote 2 platform [26] which has an Intel PXA271 XScale processor running at a clock frequency ranging from 13 MHz to 416 MHz. The core frequency could be dynamically set in software, allowing the designer to carefully adjust the timing/power trade-off to optimize performance of a specific application.

For implementation of the algorithms, some of the functions used in TinyECC [27] were taken. This package targeted the MICAz platform and provided a basic library of ECC-based functions, including scalar multiplication and exponentiation operations. Also some customizations for the XScale processor (including 32-bit operation optimizations) were carried out. In addition, supplementary functions, e.g., Montgomery arithmetic, were added. All codes were written in NesC running on TinyOS operating system.

3.18. PKC-Based Security Architecture

[28] proposes an efficient PKC based security architecture for WSNs with relatively less resource requirements. The proposed security architecture comprises basically of two parts; a key handshaking scheme based on simple linear operations and the derivation of decryption key by a receiver node. The architecture allows both base-station-to-node or node-to-base-station secure communications, and node-to-node secure communications. Analysis and simulation results show that the proposed architecture ensures a good level of security for communications in the network and could effectively be implemented using the limited computation, memory, and energy budgets of the current generation sensor nodes.

3.19. Implementation of Elliptic Curve ElGamal (EC-ElGamal) Cryptosystem

In [29], Ugus et al. implement elliptic curve and finite field arithmetic operations on MICAz mote. The experimental results presented in this work show that scalar point multiplication with a random base takes 1.03s, while it takes only 0.57s in the case of fixed

point multiplication, when 2 pre-computed points are employed. Moreover, they claim to achieve at least 44% faster operation than the best previous result for fixed point multiplication. The implementation of the elliptic curve EC-ElGamal encryption shows that the encryption operation only takes 1.19s, when in total 4 pre-computed points are utilized.

3.20. Energy Cost of Cryptographic Key Establishment

In [30], the authors implement two protocols. The first protocol employs a lightweight variant of the Kerberos key transport mechanism with 128-bit AES encryption. The second protocol is based on ECMQV, an authenticated version of the elliptic curve Diffie-Hellman key exchange, and uses a 256-bit prime field GF(p) as underlying algebraic structure. They evaluate the energy costs of both protocols on a Rockwell WINS node equipped with a 133 MHz StrongARM processor and a 100 kbit/s radio module. The evaluation considers both the processor's energy consumption for calculating cryptographic primitives and the energy cost of radio communication for different transmit power levels.

Their experiments show that the communication energy cost of Kerberos is between 39.5 mJ and 47.5 mJ (which basically depends on the transmit power level). In their architecture, three of the four Kerberos messages are encrypted. When using 128-bit AES, the overall energy required for encryption and decryption of the messages is less than 0.1 mJ. As a whole, the Kerberos protocol is characterized by high communication energy cost, while the energy needed for encryption or decryption is almost negligible.

They implement the ECDH/ECMQV key exchange using a 256-bit prime field as underlying algebraic structure in order to match the security level of Kerberos key transport with 128-bit AES encryption. According to their findings, a point multiplication over a 256-bit prime field takes approximately $4.25 \cdot 10^6$ clock cycles on a StrongARM processor when implemented as their architecture. The results show that the overall energy cost of Kerberos key establishment is between 39.6 mJ and 47.6 mJ, while ECMQV key exchange requires an energy of between 79.0 mJ and 84.6 mJ. In other words, the energy consumption of ECMQV and Kerberos differs merely by a factor of between 1.78 (for high transmit power) and 1.99 (for low transmit power).

4. Other Works and Future Expectations

Other than the mentioned works, [31] looks at several additive homomorphic public key encryption schemes and their applicability to WSNs. The authors in this work provide recommendations for selecting the most suitable PK schemes based on the topologies and the scenarios of WSNs. Also, in a recent work, Roman and Alcaraz [32] talk about the applicability of public key infrastructures in wireless sensor networks.

Although PKC in WSN is one of the relatively new areas, a lot of researchers have tried to deal with this issue within a short period of time. Considering all the works done so far, it can be realized that the abundant use of PKC in WSN is just a matter of time. Moreover, next generation sensor nodes are expected to have more energies. They can have ultra-low power circuitry with so-called power scavengers such as Heliomote [33], which allows continuous energy supply to the nodes. At least 8-20μW of power can be generated using MEMS-based power scavengers [34]. Also some other solar-based systems have been developed by this time which could even be able to deliver power up to 100mW for MICA Motes [35].

All these positive signs and achieved results indicate that: with the advancements of fast growing technology for sensors, PKC will no longer be impractical for WSNs, though still now it is expensive for the current generation sensor nodes.

5. Conclusions

PKC is often ruled out considering the limitations of resources of sensors. The reason is that the data security is not the only requirement for WSNs. There are also many other aspects which require sharing of the same resources in the sensors. Hence, the main focus is to use cryptographic methods that can ensure the best level of security for sensor data with the least costs. Various works presented in this paper show good signs of the practicality of PKC. Based on our findings and current research trend, we can conclude this article with the comment that: PKC is feasible for the current generation sensors with a little expense of energy. Advancements of sensor capabilities might allow abundant use of PKC in WSN in near future.

6. References

[1] Arazi, B., Elhanany, I., Arazi, O., and Qi, H., "Revisiting Public-Key Cryptography for Wireless Sensor Networks", IEEE Computer, Volume 38, Issue 11, pp. 103- 105.
[2] http://www.xbow.com/

[3] Gura, N., Patel, A., Wander, A., Eberle, H., and Shantz, S. C., "Comparing Elliptic Curve Cryptography and RSA on 8-Bit CPUs", LNCS 3156, Springer-Verlag 04, pp. 119-132.
[4] Malan, D.J., Welsh, M., and Smith, M.D., "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography", Proc. of IEEE SECON'04, pp. 71–80.
[5] Gaubatz, G., Kaps, J.-P., and Sunar, B., "Public Key Cryptography in Sensor Networks-Revisited," ESAS 2004, LNCS 3313, Springer-Verlag 2005, pp. 2-18.
[6] Rabin, M.O., "Digitalized signatures and public key functions as intractable as factorization", Mit/lcs/tr-212, Massachusetts Institute of Technology, 1979.
[7] Hoffstein, J. and Silverman, J.H., "Optimizations for NTRU", Proc. of Public Key Cryptography and Computational Number Theory, de Gruyter, Warsaw, 2000.
[8] Watro, R., Kong, D., Cuti, S.-f., Gardiner, C., Lynn, C. and Kruus, P., "TinyPK: Securing Sensor Networks with Public Key Technology", Proceedings of ACM SASN'04, Washington, DC, USA, 2004, pp. 59-64.
[9] Karlof, C., Sastry, N., and Wagner, D., "TinySec: A Link Layer Security architecture for Wireless Sensor Networks," Proc. of ACM SenSys'04, 2004, pp.162-175.
[10] Du, W., Wang, R., and Ning, P., "An Efficient Scheme for Authenticating Public Keys in Sensor Networks", Proc. of ACM MobiHoc'05, Illinois, USA, 2005, pp. 58-67.
[11] Wander, A.S., Gura, N., Eberle, H., Gupta, V., and Shantz, S.C., "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks", Proceedings of PerCom'05, 2005, pp. 324-328.
[12] Gaubatz, G., Kaps, J.-P., Ozturk, E., and Sunar, B., "State of the Art in Ultra-Low Power Public Key Crytography for Wireless Sensor Networks", Proceedings of the Third IEEE PERCOMW 2005, pp. 146-150.
[13] Blaß, E. O. and Zitterbart, M., "Towards Acceptable Public-Key Encryption in Sensor Networks", Proc. of ACM 2nd Int. Workshop on Ubiq. Computing, 2005, pp. 88-93.
[14] Gupta, V., Wurm, M., Zhu, Y., Millard, M., Fung, S., Gura, N., Eberle, H., and Shantz, S. C., "Sizzle: A Standards-based End-to-End Security Architecture for the Embedded Internet", SMLI TR-2005-145, Sun Micro., June 2005.
[15] Jing, Q., Hu, J., and Chen, Z., "C4W: An Energy Efficient Public Key Cryptosystem for Large-Scale Wireless Sensor Networks", Proc. of IEEE MASS 2006, pp. 827-832.
[16] Tang, W., Xiang-hao, N, and Zhong, C., "Combined Public Key Systems", Proc. of SoftCOM 2004.
[17] Nyang D. and Mohaisen A., "Cooperative Public Key Authentication Protocol in Wireless Sensor Network", UIC 2006, LNCS 4159, Springer-Verlag 2006, pp. 864-873.
[18] Murphy, G., Keeshan, A., Agarwal, R., and Popovici, E., "Hardware-Software Implementation of Public-Key Cryptography for Wireless Sensor Networks", Proc. of Irish Signals and Systems Conference, 2006, pp. 463-468.
[19] O'Flynn, B., Barroso, A., Bellis, S., Benson, J., Roedig, U., Delaney, K., Barton, J., Sreenan, C., and O'Mathuna, C., "The Development of A Novel Miniaturized Modular Platform for Wireless Sensor Networks", Proc. of the fourth IPSN'05, LA, CA, USA, April 24-27, 2005, pp 370-375.
[20] Piotrowski, K., Langendoerfer, P., and Peter, S., "How Public Key Cryptography Influences Wireless Sensor Node Lifetime", Proc. of ACM SASN'06, VA, USA, pp. 169-176.
[21] Yang, B.-Y., Cheng, C.-M., Chen, B.-R., and Chen, J.-M., "Implementing Minimized Multivariate PKC on Low-Resource Embedded Systems", SPC 2006, LNCS 3934, Springer-Verlag 2006, pp. 73–88.
[22] Chen, J.-M., Yang, B.-Y., and Peng, B.-Y., "Tame Transformation Signatures With Topy-Turvy Hashes", Proc. of the 2nd IWAP'02, Oct. 30-Nov. 1, 2002.
[23] Batina, L., Mentens, N., Sakiyama, K., Preneel, B., and Verbauwhede, I., "Low-Cost Elliptic Curve Cryptography for Wireless Sensor Networks", LNCS 4357, pp. 6-17.
[24] Bertoni, G., Breveglieri, L., and Venturi, M., "Power Aware Design of an Elliptic Curve Coprocessor for 8 bit Platforms", Proc. of PERCOMW'06, 2006, p. 337.
[25] Arazi, O., Elhanany, I., Rose, D., Qi, H., and Arazi, B., "Self-Certified Public Key Generation on the Intel Mote 2 Sensor Network Platform", 2nd IEEE Workshop on Wireless Mesh Networks 2006 (WiMesh 2006), pp. 118-120.
[26] Adler, R., Flanigan, M., Huang, J., Kling, R., Kushalnagar, N., Nachman, L., Wan, C.-Y., and Yarvis, M., "Intel Mote 2: An Advanced Platform for Demanding Sensor Network Applications", Proc. of ACM SenSys, 2005, p. 298.
[27] Liu, A. and Ning, P., TinyECC: Elliptic Curve Cryptography for Sensor Networks (Version 0.1), Technical Report, September 2005, available at: http://discovery.csc.ncsu.edu/software/TinyECC
[28] Haque, M. M., Pathan, A.-S. K., Choi, B. G., and Hong, C. S., "An Efficient PKC-Based Security Architecture for Wireless Sensor Networks", Proc. of IEEE MILCOM 2007, October 29-31, Orlando, Florida, USA, 2007.
[29] Ugus, O, Hessler, A., and Westhoff, D., "Performance of Additive Homomorphic EC-ElGamal Encryption for TinyPEDS", Tech. Rep., 2007, available at: http://www.ist-ubisecsens.org/publications/EcElgamal-UgHesWest.pdf
[30] Großschädl, J., Szekely, A., and Tillich, S., "The Energy Cost of Cryptographic Key Establishment in Wireless Sensor Networks", Proc. of ASIACCS 2007, pp. 380–382.
[31] Mykletun, E., Girao, J., and Westhoff, D., "Public Key Based Cryptoschemes for Data Concealment in Wireless Sensor Networks", Proc. of IEEE ICC, 2006, pp. 2288-2295.
[32] Roman, R. and Alcaraz, C., "Applicability of Public Key Infrastructures in Wireless Sensor Networks", EuroPKI 2007, LNCS 4582, Springer-Verlag 2007, pp. 313-320.
[33] Kansal, A., Potter, D., and Srivastava, M., "Performance Aware Tasking for Environmentally Powered Sensor Networks", In Proc. of ACM SIGMETRICS'04, 2004.
[34] Amirtharajah, R. and Chandrakasan, A., "Self-powered signal processing using vibration-based power generation", IEEE Jnl. of Solid-State Circuits, Vol. 33, pp. 687-695, 1998.
[35] Kansal, A. and Srivastava, M., "An Environmental Energy Harvesting Framework for Sensor Networks", Proc. of ACM/IEEE ISLPED, 2003.