

Poster Abstract: A public key technology platform for wireless sensor networks

Wen Chan Shih Wen Hu Peter Corke Leslie Overs
Autonomous Systems Laboratory, CSIRO ICT Centre, Australia
{teddy.wen-chan, wen.hu, peter.corke,leslie.overs}@csiro.au

Abstract

Communication security for wireless sensor networks (WSN) is a challenge due to the limited computation and energy resources available at nodes. We describe the design and implementation of a public-key (PK) platform based on a standard Trusted Platform Module (TPM) chip that extends the capability of a standard node. The result facilitates message security services such as confidentiality, authenticity and integrity. We present results including computation time, energy consumption and cost.

Categories and Subject Descriptors

C.2.0 [Computer Communication Networks]: Security and Protection

General Terms

Security, Rivest Shamir Adelman (RSA), TPM

Keywords

wireless sensor networks, public key (PK)

1 Introduction

Wireless sensor network (WSN) applications are growing but security and privacy is still largely ignored, since they are hard to achieve given the limited computation and energy resources available at node level. However secure communication will be a requirement for many applications in the future to ensure privacy and authenticity of transmitted data and also to ensure the authenticity of commands program downloads.

Symmetric (shared) key algorithms are tractable on mote-class hardware and can achieve message confidentiality however key distribution and management remains a challenge. However these algorithms poorly support message authenticity and integrity. In the Internet, Public Key (PK) technology is widely used to support symmetric key management, as well as message authenticity and integrity. Researchers have investigated methods to support PK technology in WSN [3, 2]. Such approaches have focused on software-based PK technologies, such as Rivest Shamir Adelman (RSA) and Elliptic Curve Cryptography (ECC) but the performance has been poor given the low clock rate and memory availability. Consequently, a smaller RSA public exponent (e) and a shorter key

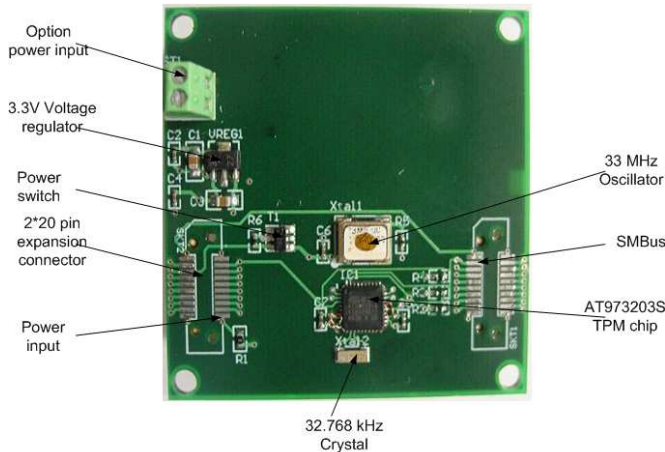


Figure 1. secFleck TPM module (upper side).

size are chosen, which compromises the security level of asymmetric encryption.

In this paper we design and implement secFleck, a PK platform that uses Trusted Platform Module (TPM) hardware to augment the node. Our evaluation shows that secFleck provides Internet-level PK services with reasonable energy consumption and financial overhead.

2 Platform Architecture

The core of secFleck is an Atmel AT97SC3203S TPM chip (see Fig. 1) mounted on a Fleck expansion board (see Fig. 2). The Fleck is a wireless sensor node that features an Atmega 128 micro controller (8 MHz clock rate and 8 KB memory) and a Nordic nRF905 radio [1]. The TPM module has a 100 kHz SMBus which is very similar to the I2C and the TPM is connected to the Fleck's I2C interface.

The TPM chip follows version 1.2 of Trusted Computing Group (TCG) specification for TPM. We have implemented a set of RSA



Figure 2. secFleck (Fleck3 and TPM module).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SenSys'08, Raleigh, NC, Nov. 5 - Nov. 7, 2008.
Copyright 2008 ACM 1-59593-763-6/07/0011 ...\$5.00

```

uint8_t fos_tpm_startup(void);
uint8_t fos_tpm_turnoff(void);

uint8_t fos_tpm_getpubkey(uint8_t *pubkey);

uint8_t fos_tpm_encryption(uint8_t *msg, uint16_t len, uint8_t *pubkey, uint8_t *cipher);
uint8_t fos_tpm_decryption(uint8_t *cipher, uint8_t *msg, uint16_t *len);

uint8_t fos_tpm_sign(uint8_t *digest, uint8_t *signature);
uint8_t fos_tpm_verifysign(uint8_t *signature, uint8_t *pubkey, uint16_t *digest);

```

Figure 3. FOS TPM application interface for control of public key infrastructure.

Table 1. Comparison of RSA encryption times.

Public Exponent (e)	Software 1024 bit	Software 2048 bit	Hardware 2048 bit
3	0.45s	65s	N/A
65,537	4.185s	450s	0.055s

public key cryptographic primitives as a Fleck OS (FOS) [1] module, which include encryption, decryption, sign, and signature verification etc. (see Fig. 3). FOS is a C-based cooperative multi threaded operating system for WSN.

Primitives allow an application to turn the TPM on or off which is important since its current consumption is around 50 mA.

Each TPM has a unique 2048-bit private key established during manufacture which cannot be read. However an application can acquire the corresponding public key from the TPM which it can share with other nodes for encryption and signature verification purposes. An application encrypts a message by providing the plain text, the length of the plain text, and a public key — the ciphertext is returned. Similarly, an application can decrypt ciphertext. The FOS encryption and decryption facilitates message confidentiality.

A base station typically has more computation, memory and energy resources and can be treated as a Certificate Authority (CA). All the nodes store the CA's public key in their permanent memories such as EEPROM before deployment, and the base station has the public keys of all nodes¹. Therefore, message authenticity can be facilitated. To verify or sign messages secFleck provides two additional functions. Space limits preclude detailed discussions of these functions.

3 Performance Evaluation

In this section, we discuss the performance of the secFleck platform in terms of computation time, energy consumption, and financial cost.

As part of our benchmarking we also implemented the RSA encryption algorithm in software for comparison purposes. Table 1 shows the encryption time for different key sizes and RSA public exponents (e) in both software and hardware implementation. The results show that the TPM chip can reduce the computation time of RSA encryption by a factor of 8,000, when $e = 65,537$ and key size is 2048 bits. Table 1 also shows that software RSA implementation is impractical in embedded micro controllers such as Atmega 128 when $e > 3$ and for key size is larger than 1024 bits. A small e will make RSA less secure, and a key size of 1024 bits will no longer be considered secure in a few years time.

¹This approach might not scale. However, multiple base stations and clustering network structure can be used to make this approach more scalable.

Table 2. RSA computation time in secFleck for $e = 65,537$ and 2048 bit key.

Encryption	Decryption	Sign	Verification
55ms	750ms	787ms	59ms

Table 3. secFleck current consumption

Module	Current (mA)
Fleck3 Receive	18.4
Fleck3 Transmit	36.8
Fleck3 + TPM encryption	50.4
Fleck3 + TPM decryption	60.8
Fleck3 + TPM signature	60.8
Fleck3 + TPM signature verification	50.4

Table 4. RSA encryption energy consumption. ($e = 65,537$, 2048 bit key)

Platform	Current (mA)	Time (s)	Energy (mJ)
Software	8	450	14,400
Hardware	50.4	0.055	11

We have not implemented the RSA decryption algorithm in software because it is significantly more computationally intensive than the RSA encryption algorithm (see Table 2). Table 2 also shows RSA encryption, decryption, sign and signature verification computation time in secFleck.

Table 3 shows the current consumption of different secFleck operations. It shows that RSA operations consume 37% to 65% more current than transmitting in secFleck. Table 4 shows the energy consumption of 2048-bit RSA encryption operation when $e = 65,537$. It shows that the software-based approach consumes around 1,300 times more energy compared to secFleck for an RSA encryption. Table 4 also shows that the software-based approach RSA encryption in WSN is indeed impractical in terms of both computation time and energy consumption for reasonable RSA exponent and key size. On the other hand, secFleck makes it feasible to support PK technology for WSN.

An Atmel AT97SC3203S TPM chip costs \$4.5 when ordered in quantities², which is less than 5% of the cost of popular sensor devices such as Telosb, Iris mote, and Fleck (about \$100). The TPM chip is also small (Figure 1) measuring just 6.1×9.7 mm and is less than 2% of the area of the Fleck and could be integrated onto a future version rather than the cumbersome expansion board used in this prototype.

4 Conclusion and future work

We have presented secFleck, a TPM-based PK platform for sensor networks that facilitates message security services such as confidentiality, authenticity and integrity. Our evaluation shows that secFleck provides Internet-level public-key services quickly, with low energy consumption and at low cost in terms of parts and board real estate. Our next step is to design and implement a secure code dissemination protocol based on secFleck.

5 References

- [1] P. Corke and P. Sikka. Demo abstract: FOS — a new operating system for sensor networks. In *Fifth European conference on wireless sensor networks (EWSN 2008)*, Bologna, Italy, Jan, 2008.
- [2] A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. In *IPSN '08: Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 245–256, Washington, DC, USA, 2008. IEEE Computer Society.
- [3] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus. TinyPk: securing sensor networks with public key technology. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 59–64, New York, NY, USA, 2004. ACM.

²http://www.atmel.com/dyn/products/view_detail.asp?ref=&FileName=embedded10_18.html&Family_id=620 (accessed on 18th June, 2008)