On the Energy Cost of Communication and Cryptography in Wireless Sensor Networks

Giacomo de Meulenaer François Gosset François-Xavier Standaert Luc Vandendorpe
{giacomo.demeulenaer, francois.gosset, fstandae, luc.vandendorpe}@uclouvain.be
UCL/DICE Crypto Group
Place du Levant, 3
Louvain-la-Neuve, Belgium.

Abstract

Energy is a central concern in the deployment of wireless sensor networks. In this paper, we investigate the energy cost of cryptographic protocols, both from a communication and a computation point of view, based on practical measurements on the MICAz and TelosB sensors. We focus on the cost of two key agreement protocols: Kerberos and the Elliptic Curve Diffie-Hellman key exchange with authentication provided by the Elliptic Curve Digital Signature Algorithm (ECDH-ECDSA). We find that, in our context, Kerberos is around one order of magnitude less costly than the ECDH-ECDSA key exchange and confirm that it should be preferred in situations where a trusted third party is available. We also observe that the power dedicated to communications can become a central concern when the nodes need to stay in listen mode, e.g. between the protocol rounds, even when reduced using a Low Power Listening (LPL) protocol. Therefore, listening should be considered when assessing the cost of cryptographic protocols on sensor nodes.

1 Introduction

Wireless Sensor Networks (WSN) are composed of small autonomous devices that process and communicate data acquired from the environment in which they are deployed. Their low cost and rapidity of deployment make them particularly attractive for many applications requiring strong security (health monitoring, pollution detection, etc). However, sensor nodes being powered through batteries, the energy cost of security techniques can be prohibitive and must therefore be minimized. Various techniques can be adopted to perform the cryptographic tasks in WSN. As an example, key exchange can be carried out by relying on methods from symmetric key cryptography (e.g., through Kerberos [15]), or from public-key cryptography (e.g., through SSL/TLS [3]). Besides, different techniques have been proposed, that allow trading between security, communication and computation (e.g., [12] and [4]). In order to appreciate the practical effectiveness of these trading techniques in a specific WSN, the cost of communication and computation must be well understood. Contradictions appear in previous works concerning the importance of the communication energy cost when comparing cryptographic algorithms in WSN (see [16] and [18]). Our goal is to assess and analyze the real cost of cryptography on WSN nodes. This will help choosing directions to optimize the cost of cryptography in low power WSN. For this purpose, we investigate the cost of cryptography through a case study based on measurements on the MICAz [11] and TelosB [11] sensor nodes. We focus on two key agreement protocols, Kerberos and ECDH-ECDSA, the Elliptic Curve Diffie-Hellman key exchange with authentication provided by the Elliptic Curve Digital Signature Algorithm (i.e., the ECC-based SSL/TLS handshake, see [1]). We assess their energy cost using energy models of the sensors based on measurements. Our main contributions are :

1. a methodology to assess the real cost of cryptography on WSN nodes which makes it possible to establish the relative costs of computation and communication.
2. the estimates of the key agreement protocols obtained for the MICAz and TelosB nodes. They allow us to compare symmetric and asymmetric techniques. They point out the importance of idle listening consumption.

This paper is structured as follows. Section 2 presents the previous related works. Then, Section 3 explains how we determined the energy models of the sensors MICAz and TelosB. Next, Section 4 provides an assessment and analysis of energy cost of Kerberos and ECDH-ECDSA, followed by a comparison with related results in Section 5. Finally, the conclusion is given in Section 6.

2 Previous works

Many recent works investigate the usability of cryptographic algorithms in the context of wireless sensor networks. For instance, symmetric encryption using AES is discussed in e.g., [9] and [13]. For public-key cryptography, implementations of Elliptic Curve Cryptography (ECC [8]) on such sensors are described in e.g., [6] and [14]. Several previous works focused on the energy cost of key agreement protocols for WSN. Based on the first implementations of ECC and RSA on 8-bit microprocessors by Gura et al. [7], Wander et al. [18] quantified the energy costs of ECC and RSA based digital signature and key exchange with mutual authentication for networks composed of Mica2dot sensors [11]. They concluded that these operations are affordable for such sensors. In [16], Piotrowski et al. assessed the energy consumption of most common RSA and ECC operations for other sensor nodes. They based their assessments on the implementation results of [6] and on the datasheets of the sensors. They found that the energy consumed by transmissions was at least one order of magnitude less than the one required for the computation of the cryptographic operations. Therefore, they concluded that it was not an important factor. Hodjat and Verbauwhede [10] compared the cost of the protocols Kerberos and ECDH on 32-bit WINS sensor nodes. The cost of Diffie-Hellman was found between one to two orders of magnitude larger than AES-based Kerberos. Later, Großschädl et al. [5] performed the same comparison but with another version of Diffie-Hellman, ECMQV, on WINS nodes. They found that the cost of ECMQV was only up to twice the cost of Kerberos. To quantify the communication energy costs, these two works used transmission and reception per-bit costs based on measurements. However, this excludes the energy consumption of practical elements such as listening which happens when nodes are waiting for incoming packets of which the exact times of arrival are uncertain. Compared to these previous works, we take more into account the practical aspects of the energy consumption for communication.

3 Energy model of the sensors

In this section, we determine the energy models of the sensors MICAz and TelosB that we later use to estimate the energy consumption of cryptographic protocols. The MICAz is based on the low-power 8-bit microcontroller ATmega128L with a clock frequency of 7.37 MHz. The TelosB features the 16-bit MSP430 microcontroller running at 4 MHz. Both nodes run TinyOS and embed a IEEE 802.15.4 compliant CC2420 transceiver with a claimed data rate of 250 kbps. Table 1 presents the measured consumption of the main operating modes for both platforms. The energy models are established in the following way. For the cost of computation, we make the approximation that the overall power consumption of the node while computing remains constant

Power consumption	MICAz	TelosB
Transmit	65 mW (@ Ptx = -5 dBm)	54 mW (@ Ptx = -5 dBm)
Listen	68 mW	60 mW
Receive	72 mW	61 mW
Compute	26 mW (@ 7.37 MHz)	4.8 mW (@ 4 MHz)
Sleep	25 μW (power down mode)	35 μW (low-power mode 3)

Table 1: Measured power consumption of the MICAz and TelosB in different operating modes.

Energy cost	MICAz	TelosB
Compute for 1 T_{clk}	3.5 nJ (1)	1.2 nJ (1)
Transmit 1 bit	0.60 μJ (170)	0.72 μJ (600)
Receive 1 bit	0.67 μJ (190)	0.81 μJ (680)
Listen for 1 T_{clk}	9.2 nJ (3)	15.0 nJ (13)
Sleep for 1 T_{clk}	3 pJ (10^{-3})	9 pJ (10^{-2})

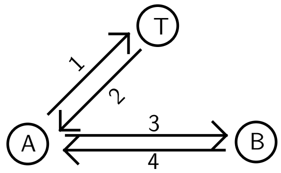
Table 2: Energy costs of common operations on the MICAz running at 7.37 MHz and TelosB at 4 MHz for application data rates of respectively 108 kbps and 75 kbps. The equivalence in number of cycles of computation is indicated in parenthesis.

with the type of microcode operation performed. Therefore, the cost of a particular computation can be assessed knowing the per-cycle mean energy consumption and the total number of cycles of the computation. This simplifying assumption was verified by Law et al. in [13] for the sensor node used in the EYES project [2], which is quite similar to the TelosB. For the communication cost, we measured the effective data rates and the consumption in the transmit, listen and receive modes. The measured data rates, 121 kbps and 94 kbps for the MICAz and TelosB respectively, are far below the claimed rates (250 kbps). The presence of footers and headers and the use of acknowledgment further decrease the rates available for application data to respectively 108 kbps and 75 kbps. Our energy costs of Table 2, based on the measurement results of Table 1, assume these data rates and a typical transmit power of -5 dBm.

The consumption in the listening mode is almost as high as for reception (see Table 1) because the transceiver is also active in this mode. It suggests that this mode should be avoided as much as possible to save energy. That is the goal of Low Power Listening (LPL) protocols that save energy at the expense of greater latencies in the communications. They make the time spent in listen mode less important from an energy point of view. In TinyOS, the LPL protocol available is based on B-MAC [17]. In this protocol, the receiving radio modules are periodically turned on to check for activity on the channel and remain active only if a packet is being transmitted. Sending nodes must be kept retransmitting the same packets until the checks of the receivers. The consumption of a listening node can arbitrary be reduced by increasing the sleep interval (i.e., the delay between two checks). However, this is done at the expense of increased synchronization energy costs for senders that have to retransmit during a longer period before the checks of the receivers. After a reception, both the sender and the receiver keep their radio on for a small delay in case of a consecutive packet transmission. This also generates a synchronization cost for the receiver. In this work, we chose the typical values of respectively 10 ms and 100 ms for the delay after reception and the sleep interval while the check duration was a constant of 5 ms. Accordingly, we estimated the energy costs due to LPL as indicated in Table 3. We use the energy costs of Table 2 and Table 3 as an energy model to predict the energy cost of protocols on the MICAz and TelosB platforms. It requires the number of cycles of computation, the

Energy cost	MICAz	TelosB
Listen for 1 T_{clk}	0.4 nJ (0.1)	0.7 nJ (0.6)
Send synchronization	3.77 mJ (1.1 M)	3.17 mJ (2.6 M)
Receive synchronization	0.68 mJ (0.2 M)	0.60 mJ (0.5 M)

Table 3: Energy costs of the LPL protocol for the MICAz and TelosB. The equivalence in number of cycles of computation is indicated in parenthesis.



1: A,B, n_A
2: $\{k_{AB}$, B, t_S, t_E, $n_A\}$ k_{AT}, $\{k_{AB}$, A, t_S, $t_E\}k_{BT}$
3: $\{k_{AB}$, A, t_S, $t_E\}k_{BT}$, $\{$A, $t_A\}k_{AB}$
4: $\{t_A\}k_{AB}$

Figure 1: The simplified Kerberos protocol.

number of bits communicated, the number of synchronizations in the communications (if using LPL) and the time spent in sleep mode.

4 Energy consumption of key agreement protocols

In this section, we use the energy models of Section 3 to assess and analyze the energy cost of cryptographic protocols. As an example, we focus on two key agreement protocols, Kerberos and ECDH-ECDSA. We first describe these protocols, then assess the cost of the cryptographic operations and communications.

4.1 Protocols description

The establishment of shared secret keys between nodes is a first step to provide other security services such as encryption in WSN. This could be achieved by means of pre-deployed shared keys but it raises problems of storage of the keys in large networks and of resiliency to node compromise. Therefore, a solution is to use key distribution or key agreement protocols after the deployment of the nodes. In this work, we compare two of these protocols.

The first protocol is Kerberos [15], a key distribution scheme built on secret-key cryptography, which authenticates the participants. We use its simplified version described in [5]. In this protocol, the two entities A and B wishing to establish a shared secret key k_{AB} already share a secret key (k_{AT} and k_{BT} respectively) with a trusted third party T. In this protocol (see Figure 1), there is first an exchange of messages between A and T. The request of A contains the identities of A and B. In the reply, the key k_{AB} generated by T is encrypted with the keys k_{AT} and k_{BT}. Then, A recovers the key k_{AB} and forwards to B the piece of the message encrypted with k_{BT} together with its identity encrypted with k_{AB}. Finally, B recovers k_{AB} and sends back to A a timestamp encrypted with k_{AB}. Replay attacks are avoided thanks to a timestamp t_A, a nonce n_A and expiration times t_S, t_E.

The second protocol is ECDH [8], the Diffie-Hellman key agreement based on Elliptic Curve Cryptography (ECC [8]) which does not need any trusted third party. In its standard form, ECDH does not provide authentication. Therefore, we use the version known as ECDH - ECDSA in [1]. In this version, authentication is provided through certificates verified using the Elliptic Curve Digital Signature Algorithm (ECDSA [8]).

Energy cost	MICAz	TelosB
AES-128 128-bit encrypt	38 μJ (10742)	9 μJ (7483)
ECC-160 point mult	55 mJ (15.6 M)	17 mJ (14.0 M)
ECDSA-160 sign	52 mJ (14.7 M)	15 mJ (12.7 M)
ECDSA-160 verify	63 mJ (18.0 M)	19 mJ (16.2 M)

Table 4: Estimated energy costs of cryptographic operations for the MICAz and TelosB. The number of cycles of computation is indicated in parenthesis.

Thus, the two parties A and B must possess a certificate generated by an authority. They agree to use the same curve parameters and generate in advance their private keys, k_A and k_B and corresponding public keys $Q_A = k_A \cdot G$ and $Q_B = k_B \cdot G$ where G is the generator of the group defined by the elliptic curve. In this protocol, A and B first exchange random nonces. Then, B sends its certificate to A (its public key signed by the authority using ECDSA). After the certificate verification, A uses his private key and B's public key to perform a point multiplication and arrive to a common secret $k_A \cdot k_B \cdot G$, which is used with the exchanged nonces to derive a shared secret key. Then, A sends its certificate to B who performs the same operations to obtain the shared secret ($k_A \cdot k_B \cdot G = k_B \cdot k_A \cdot G$) and derive the shared secret key. The possession of the shared secret key is proved in the ability of both parties to encrypt the exchanged nonces and their identities with the shared secret key. These results, forming the content of $Finished$ messages, are exchanged at the end of the protocol.

4.2 Cost of cryptographic operations

We assess the energy costs of the cryptographic operations playing a part in Kerberos and ECDH-ECDSA using the energy model of the sensors (cf. Section 3) and the number of cycles of computation from known implementations. For the symmetric encryption employed in Kerberos, we use the implementation results of Healy et al. [9]. They implemented AES (128-bit keys) on the microcontrollers of both MICAz and TelosB nodes. We assess the ECC point multiplications and ECDSA verifications involved in ECDH-ECDSA relying on the results of Liu et al. [14]. They implemented ECC and ECDSA (160-bit keys) in TinyOS for many platforms including MICAz and TelosB. Table 4 shows the estimated energy costs of these cryptographic operations. The cost of symmetric encryption is negligible compared to elliptic curve operations. The number of cycles for elliptic curve computations does not diminish much on the TelosB (however based on a 16-bit microcontroller) because the implementation available for this platform is less optimized.

We estimate the cost of the computations for both protocols based on the assessments of Table 4. For Kerberos, the computations consist in the encryption and decryption of 8 blocks of 128 bits (assuming 64-bit timestamps and node IDs and a 32-bit nonce). As a result, the cost of Kerberos is respectively 0.61 mJ and 0.14 mJ on the MICAz and TelosB. For ECDH-ECDSA, each party mainly achieves an ECDSA verification and a point multiplication. The key derivation and symmetric encryption of the nonces and nodes IDs can be neglected considering the relative small cost of AES with respect to ECC operations (see Table 4). It leads to an energy cost for ECDH-ECDSA of respectively 236 mJ and 72 mJ on the MICAz and TelosB. ECDH-ECDSA is more than 2 orders of magnitude more costly than Kerberos on both platforms. This was expected as elliptic curve operations are much more costly than AES-based encryption. The costs of both protocols are around 4 times lower on the more energy-efficient TelosB.

4.3 Communication and total energy assessment

Here we assess the communication energy costs of the protocols. Together with the computation costs of the previous section, they make it possible to obtain the total costs of the protocols. The communication costs are composed of the cost of transmission, reception and listening. For transmission and reception, we make use of the per-bit costs presented in Table 2. The total number of bits communicated in Kerberos is 1568 and 2208 in ECDH-ECDSA (assuming 86-byte certificates, 32-byte nonces and 20-byte *Finished* messages as in [6]). For listening, we use the energy costs (see Table 3) of the LPL protocol of Section 3 and the total listening durations of the protocols. For the MICAz and TelosB, these are respectively 9.1 s and 15.1 s for ECDH-ECDSA and 61 ms and 72 ms for Kerberos. Synchronization costs appear for each transmission except when the nodes answer a previous transmission within the delay after reception of 100 ms (e.g., B immediately answers the first message of A in ECDH-ECDSA). The estimated communication costs for Kerberos and ECDH-ECDSA on the MICAz and TelosB nodes are shown in Table 5. They are higher for ECDH-ECDSA mainly because of the high listening cost due to the long computation delays of this protocol. However, one could save the major part of the LPL listening energy loss by temporarily increasing the sleep interval when waiting for cryptographic results. That would be done at the expense of loosing connectivity during the run of the protocol. By doing this, the communication costs of ECDH-ECDSA would not be much more than those of Kerberos.

Communication cost Kerberos (mJ)	MICAz	TelosB
Send	0.9	1.1
Receive	1.1	1.3
LPL listen	0.2	0.2
LPL synchro	8.9	7.5
Total	11.1	10.1

Communication cost ECDH-ECDSA (mJ)	MICAz	TelosB
Send	1.3	1.6
Receive	1.5	1.8
LPL listen	29.5	43
LPL synchro	14	11.9
Total	46.3	58.3

Table 5: Estimated communication energy costs of Kerberos and ECDH-ECDSA for the MICAz and TelosB.

Gathering the computation and communication costs found above provides the total costs for the protocols shown in Table 6. ECDH-ECDSA is close to respectively 25 times and 13 times more costly than Kerberos on MICAz and TelosB. Communications compose almost exclusively the cost of Kerberos as opposed to ECDH-ECDSA. For both protocols, the relative importance of communications grows for the TelosB which has a lower computational cost.

Kerberos cost (mJ)	MICAz	TelosB
Comp.	0.6 (5%)	0.14 (1%)
Comm.	11.1 (95%)	10.1 (99%)
Total	11.7	10.24

ECDH-ECDSA cost (mJ)	MICAz	TelosB
Comp.	236 (84%)	72 (55%)
Comm.	46.3 (16%)	58.3 (45%)
Total	282.3	130.3

Table 6: Estimated total energy costs of Kerberos and ECDH-ECDSA for the MICAz and TelosB.

5 Comparison with related results

As described in Section 2, two previous works already compared the energy cost of Kerberos and the Diffie-Hellman key exchange on sensor nodes. First, there is the work by Hodjat and Verbauwhede. They used the standard version of ECDH, which does not provide any authentication. They found that ECDH was between one to two orders of magnitude larger than Kerberos on WINS nodes. This is similar to our results of preceding section on the MICAz and TelosB. However, for the same amount of energy (140 mJ), WINs nodes can run Kerberos while TelosB nodes can perform an ECDH-ECDSA key exchange. This illustrates the important impact of the hardware. The WINS node, which contains a more powerful microprocessor (32-bit, 133 MHz), consumes much more energy than the TelosB. The authors obtained the energy cost of computations by implementing the cryptographic algorithms on the WINs node. For the cost of communications, they used the measurement results from a previous work. They did not include the cost of listening in their estimates which therefore should be higher.

Second, Großschädl et al. also compared AES-based Kerberos with ECMQV, a variant of ECDH that provides authentication, on WINS nodes. They found that the ECMQV was only up to twice as costly as Kerberos on the WINS node. ECMQV assumes that both participants have already exchanged their long-term public keys. For large networks, this means a large number of stored keys per node, which may not be desirable. Therefore, the exchange and verification of the long-term public keys could be included in the cost of this protocol. The authors estimated the cost of computations and communications as Hodjat and Verbauwhede. Similarly, they did not take the listening cost into account. Including it in their estimates is likely to have a more important impact for both protocols as the relative cost of communications is higher than in the results of Hodjat and Verbauwhede.

6 Conclusion

Our work provides a methodology to assess the real cost of cryptography on WSN nodes. Our estimates of the energy costs of Kerberos and ECDH-ECDSA on the MICAz and TelosB nodes confirm the advantage of Kerberos, what was noted in previous works. We find that Kerberos is around respectively 25 times and 13 times less costly than ECDH-ECDSA on the MICAz and TelosB. Therefore, it should be preferred in situations where a trusted third party is available. As opposed to previous works, the energy cost of listening is included in our assessments, resulting in higher communication costs. It can remain significant even when minimized using a LPL protocol. Therefore, it should be considered when assessing the cost of cryptographic protocols on WSN nodes. Our work also provides practical insights on the relative costs of computation and communication in WSN. It could therefore be useful to study the interest of techniques trading the cost of computations for communications. A thorough analysis of the energy gain of such techniques could be part of a future work.

References

[1] S. Blake-Wilson, T. Dierks, C. Hawk, "ECC cipher suites for TLS," Transport Layer Security Working Group, Internet draft available from http://tools.ietf.org/id/draft-ietf-tls-ecc-01.txt, Sept. 2001.

[2] EYES, "European research project on self-organizing and collaborative energy-efficient sensor networks," http://www.eyes.eu.org/.

[3] A. Freier, P. Karlton, P. Kocher, “The SSL protocol version 3.0,” Transport Layer Security Working Group, Internet draft available from http://wp.netscape.com/eng/ssl3/draft302.txt, Nov. 1996.

[4] M. Girault, D. Lefranc, “Server-aided verification: theory and practice,” Asiacrypt’05, LNCS 3788, pages 605-623. Springer, 2005.

[5] J. Großschädl, A. Szekely, S. Tillich, “The Energy Cost of Cryptographic Key Establishment in Wireless Sensor Networks,” Proc. of 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS 2007), pages 380-382, ACM Press, 2007.

[6] V. Gupta, M. Millard, S. Fung, Y. Zhu, N. Gura, H. Eberle, S. C. Shantz, “Sizzle: A standards-based end-to-end security architecture for the embedded Internet,” Proc. of Third IEEE International Conference on Pervasive Computing and Communications (PERCOM ’05), pages 247-256, IEEE Computer Society, 2005.

[7] N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz, “ Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs,” Proc. of CHES, pages 119-132, 2004.

[8] D. Hankerson, A. Menezes, S. Vanstone,“ Guide to Elliptic Curve Cryptography,” Springer-Verlag, 2003.

[9] M. Healy, T. Newe, E. Lewis, “ Efficiently securing data on a wireless sensor network,”Journal of Physics Conference Series, vol. 76, Issue 1, 2007.

[10] A. Hodjat, I. Verbauwhede,“The energy cost of secrets in ad-hoc networks,” Proc. IEEE Circuits and Systems Workshop on Wireless Communications and Networking, page 4, 2002.

[11] Crossbow Technology Inc., “Crossbow product information,” available at http://www.xbow.com/Products/productdetails.aspx?sid=156.

[12] B.C. Lai, D. Hwang, S. Kim, I. Verbauwhede, “Reducing radio energy consumption of key management protocols for wireless sensor networks,” Proc. ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED 2004), pages 351-356. ACM Press, 2004.

[13] Y. Law, J. Doumen, P. Hartel,“Survey and benchmark of block ciphers for wireless sensor networks,” ACM Trans. Sen. Netw., vol. 2, pages 65-93, 2006.

[14] A. Liu, P. Ning, “TinyECC: A configurable library for Elliptic Curve Cryptography in Wireless Sensor Networks,” Technical Report TR-2007-36, North Carolina State University, Department of Computer Science, 2007.

[15] B. Neuman, T. Ts’o, “Kerberos: An authentication service for computer networks,” IEEE Communications, volume 32(9), pages 33-38, Sept. 1994.

[16] K. Piotrowski, P. Langendoerfer, S. Peter, “How public key cryptography influences wireless sensor node lifetime,” Proc. of fourth ACM workshop on Security of ad hoc and sensor networks (SASN ’06), pages 169-176, 2006. ACM.

[17] J. Polastre, J. Hill, D. Culler,“ Versatile low power media access for wireless sensor networks,” Proc. of 2nd international conference on Embedded networked sensor systems (SenSys ’04), pages 95-107, 2004. ACM Press.

[18] A. Wander, N. Gura, H. Eberle, V. Gupta, S. C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks” Proc. of Third IEEE International Conference on Pervasive Computing and Communications (PerCom ’05), pages 324-328, March 2005.