Egyptian Informatics Journal (2012) **13**, 185–190

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

Efficient combined security system for wireless sensor network

N.S. Fayed *, E.M. Daydamoni, A. Atwan

Department of Information Technology, Faculty of Computers and Information, Mansoura University, Egypt

Received 11 April 2012; revised 11 September 2012; accepted 16 September 2012
Available online 22 October 2012

KEYWORDS

Security;
Wireless sensor networks;
Energy cost;
Cryptographic key establishment;
Kerberos

Abstract Wireless Sensor Networks (WSNs) need effective security mechanisms because these networks deployed in hostel unattended environments. There are many parameters affect selecting the security mechanism as its speed and energy consumption. This paper presents a combined security system for WSN that enhance the speed of the network and it is energy consumption. This system combines two strong protocols, Lightweight Kerberos and Elliptic Curve Menezes–Qu–Vanstone (ECMQV). The simulation results demonstrate that the combined system can enlarge the life time for wireless sensor networks, enhance its security, and increase its speed.

1. Introduction

Wireless Sensor Networks (WSNs) consist of many small devices each with sensing, processing, and communication capabilities to monitor the real-world environment. They are playing an important role in different areas ranging from critical military surveillance applications to building security [1]. In these networks, a large number of sensor nodes are deployed to monitor a vast field in hostel unattended environments. For that, they should be equipped with security mechanisms to defend against attacks such as node capture, physical tampering, eavesdropping, denial of service, etc. Unfortunately, traditional security mechanisms with high overhead are not suitable for resource constrained sensor nodes due to their lack of processing power, limited memory and energy [2]. This requires to rethink about current effective solutions in terms of speed of calculation and energy consumption, to make wireless sensor networks secure without consuming their energies.

Key management and authentication are the basis for other security services such as encryption. Many key establishment protocols involve trusted third party to set up a shared key between two entities. Examples of such protocols include the Needham–Schroeder protocol [3], the Kerberos key distribution protocol [4], and the SPINS node-to-node key establishment protocol [5]. In these protocols the entities share a pre-distributed long term key with the trusted party (T). The entities use T to prove their identity (authentication) or to generate and

* Corresponding author. Tel.: +20 01001378228.
E-mail addresses: nfayed@mans.edu.eg (N.S. Fayed), emane_daydamoni@mans.edu.eg (E.M. Daydamoni), atwan@mans.edu.eg (A. Atwan).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

1110-8665 © 2012 Faculty of Computers and Information, Cairo University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.eij.2012.09.001

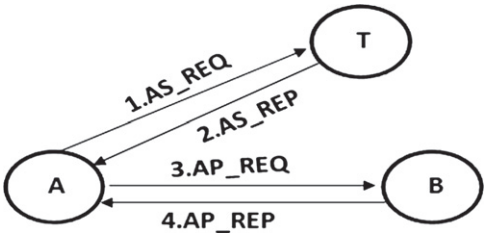
transmit a session key that allows two entities to securely communicate with one another. Depending on the protocol, T either provides the session key by itself or makes a session key generated by one entity available to the other by encrypting it with the long-term key [6]. Other protocols not relay on third party to establish the shared key. They allow two entity to establish the key through exchanging the message between them over an insecure communication channel. So, these protocols have the advantage to authenticate entities that not meet before or not have a key with a third party. Examples of these protocols are Diffie–Hellman protocol [7] and elliptic curve Menezes–Qu–Vanstone (ECMQV) [8]. Some papers talked about key establishment protocols for wireless sensor network [9,10]. In [6] they show that key exchange protocols using elliptic curve systems are feasible for wireless sensor networks. Also, they found that, in large sensor networks, elliptic curve-based key exchange may actually require less energy than Kerberos key distribution which consumes less energy in smaller sensor networks.

This paper suggests a combined system that use the two protocols, Kerberos and ECMQV, to enhance the security of the network and improve the energy consumption in the network. Beside that it increases the network speed due to minimizing the number of communications and calculations. The remaining part of this paper is organized as follows. Section 2, the related work, describes Lightweight Kerberos protocol with short messages and ECMQV. Section 3, explains the Combined Security System and how it is effective for wireless sensor network. Section 4, the discussion of the results compared with the two mentioned protocols. Finally, the paper closes with a conclusion in Section 5.

2. Related work

In sensor networks key establishment, the nodes set up a shared secret key after deployment, either through key transport or key agreement (or key exchange) [11]. In key transport protocol, an entity creates or obtains a secret key and transfers it securely to the other entity(s). On other hand in key agreement, all participating entities contribute a random input to derive a shared secret key. The advantage of key agreement over key transport is that entities cannot predetermine the resulting key because it depends on the input of all participants.

Many key establishment protocols depend on a trusted third party (T) to set up a shared key between two entities. For examples, Needham–Schroeder protocol, the Kerberos key distribution protocol, and the SPINS node-to-node key establishment protocol. Each entity shares a long-term secret key with T. According to the protocol, T either provides the session key or makes a session key generated by one entity available to the other. These protocols can be implemented with secret-key primitives so, they do not require to perform intensive cryptographic computations that may cause battery draining. Key agreement protocols do not rely on a third party to set up a shared secret key. Instead, they allow two entities to directly establish a key by exchanging messages over an insecure communication channel, for example ECMQV. This section explains the participated key establishment protocols: Lightweight Kerberos protocol with short messages and ECMQV.



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

Figure 1 Simplified Kerberos protocol exchange (an expression of the form $\{X\}\, k$ means that message X is encrypted using the key k) [6].

2.1. Lightweight Kerberos protocol with short messages

Kerberos is a distributed authentication service that allows a client to prove its identity to a server without sending data across the network that might allow an attacker to subsequently impersonate the client. The basic Kerberos authentication protocol allows a client with knowledge of the user's password to obtain a ticket and session key to prove its identity to any sever registered with the authentication server [12].

Lightweight Kerberos protocol with short messages [6] can be described as "Basic Kerberos authentication protocol without ticket granting service." To illustrate the idea of Lightweight Kerberos in authenticate two entities (Say A and B) to each other, Fig. 1 illustrates the message transfers between entity A and B and the trusted third party T (authentication server). Assume that A wishes to establish a session key with entity B and Both A and B share a long-term secret key with T. The description of the communication messages is as the following:

- The first message is the Authentication Server Request (AS_REQ) message, which is sent from A to T. This message contains A's identity, B's identity, and a random nonce n_A that will be used to associate reply messages with the matching AS_REQ request and to detect replays.
- After receipt of the AS_REQ message, T looks up entities A and B in its database, verifies that they are authorized to establish a session key, and fetches their long-term keys k_{AT} and k_{BT}. Then, T generates a new random session key k_{AB} to be shared between A and B and embeds it into a ticket. The ticket also contains A's identity, and the ticket's validity lifetime (expiration time t_E and an optional starting time $t_{S)}$. The ticket is encrypted using k_{BT} that only known by T and B. Next, T creates the AS_REP message, consisting of the ticket for A to present to B, k_{AB}, t_E, B's identity, and n_A from the AS_REQ message. All elements except the ticket are encrypted with k_{AT}.
- After receiving of the AS_REP response, A uses k_{AT} to decrypt the non-ticket part of the message. Entity A verifies that the received nonce matches the nonce it supplied in the AS_REQ message and that the current time is within the lifetime of the session key. Also, entity A checks whether the ticket was created for B. In the third message, the

AP_REQ (Application Request) message, entity A transfers the ticket together with an authenticator to B. The authenticator contains A's own identity and a timestamp t_A, both encrypted in k_{AB}. The purpose of the authenticator is to prove that entity A knows k_{AB} and to ensure that every AP_REQ message is unique.

- After receiving of the AP_REQ message, B decrypts the ticket using k_{BT} and extracts k_{AB}, the identity of A, and t_E. Then, B uses k_{AB} to decrypt the authenticator and compares the information in the ticket with that in the authenticator. If all checks pass, B considers A as authenticated. Mutual authentication requires that entity B proves its identity too by sending Application Reply (AP_REP) message, consists of the timestamp encrypted in the session key $k_{AB,}$ back to A. After A received and decrypted the AP_REP message, A verifies that the timestamp is the same one it sent in the AP_REQ message. This ensures A that k_{AB} successfully transmitted to B.

Most of protocols uses third parity, like Kerberos, are three-way communication since two entities wishing to set up a secret key do not only transmit messages to each other but also to the trusted authority. Thus, the communication energy cost of Kerberos-like protocols is much higher than the energy required for calculating cryptographic primitives [13].

2.2. Elliptic Curve Menezes–Qu–Vanstone (ECMQV) protocol

ECMQV protocol is based on Diffie–Hellman key agreement and modified to work in an arbitrary finite group and, in particular, elliptic curve groups. It is an example of key exchange protocols with implicit authentication [8].

In the ECMQV protocol each entity has both a static (i.e. long-term) public/private key pair and an ephemeral (i.e. short-term) key pair. A shared secret is derived using the static keys and the ephemeral keys, which guarantees that each protocol run between two entities A and B produces a different shared secret. Formally, an elliptic curve over a prime field $GF(p)$ can be defined by a Weierstraß Eq. (1), where α, β $GF(p)$ and $4\alpha^3 + 27\beta^2 \neq 0 \bmod p$ [14].

$$y^2 = x^3 + \alpha x + \beta \tag{1}$$

In what follows, let E be an elliptic curve group of order n, and G shall be a point on the curve. Assume that the order n is prime, which means that E is cyclic and G is a generator of E. Also, assume the domain parameters p, α, β, n, and G are publicly known to every entity of the network. Let A and B be two entities wishing to establish a shared key. First, entity A chooses a random secret number a with $2 \leqslant a \leqslant n-2$, calculates $S = a \cdot G$. Entity B also chooses a random secret number b in the range of $[2, n-2]$, calculates $T = b \cdot G$. Entity A has the static key pair (a, S) which consists of a secret part (a) and a public part (S). Entity B has the static key pair (b, T) consisting of the secret key b and the public key $T = b \cdot G$. The entities first exchange the public part of their static keys. After that, entity A and B perform the following steps to agree on a shared secret: First, entity A generates the ephemeral key pair (c, U), whereby $U = c \cdot G$, and entity B generates the ephemeral key pair (d, V) with $V = d \cdot G$. They exchange the public parts of these ephemeral keys. After that, entity A knows its own secret keys a, c, and the public keys S, T, U, and V. Also, B knows b, d, S, T, U, and V. The shared secret K is determined by entity A as in Algorithm 1. B also compute the same value of K by swapping (a, c, T, U, V) in Algorithm 1 with (b, d, S, V, U) [15].

Algorithm 1: ECMQV key derivation for entity A
Input: Elliptic curve domain parameters p, α, β, n, G, the secret keys a, c, and the public keys S, T, U, V
Output: A secret point K E shared with the entity with public static key T
1: $m \leftarrow \lceil \log_2 \rceil (n)/2$ {m is the half bit length of n}
2: $u_A \leftarrow (u_x \bmod 2^m) + 2^m$ {u_x is the x-coordinate of U}
3: $s_A \leftarrow (c + u_A\, a) \bmod n$ {implicit signature}
4: $v_A \leftarrow (v_x \bmod 2^m) + 2^m$ {v_x is the x-coordinate of V}
5: $z_A \leftarrow s_A v_B \bmod n$
6: $K \leftarrow s_A \bullet V + z_A \bullet T$

In order to derive the shared secret K, entity A and entity B have to accomplish an operation of the form $k \cdot P + l \cdot Q$ (step 6 in Algorithm 1). This operation, which is called multiple point multiplication, has an impact on the overall computational cost of the ECMQV key exchange. This operation can be performed much faster when the doublings are combined as shown in Algorithm 2.

Algorithm 2: Multiple point multiplication
Input: The points P, Q E, scalar $k = (k_{m-1}, \ldots, k_1, k_0)_2$ and scalar $l = (l_{m-1}, \ldots, l_1, l_0)_2$
Output: $R = k \bullet P + l \bullet Q$
1: $Z \leftarrow P + Q$
2: $R \leftarrow \mathcal{O}$
3: **for** i **from** $m-1$ down to 0 **do**
4: $R \leftarrow R + R$ {point doubling}
5: **if** $(k_i = 1)$ and $(l_i = 0)$ **then** $R \leftarrow R + P$ **end if**
6: **if** $(k_i = 0)$ and $(l_i = 1)$ **then** $R \leftarrow R + Q$ **end if**
7: **if** $(k_i = 1)$ and $(l_i = 1)$ **then** $R \leftarrow R + Z$ **end if**
8: end **for**
9: return R

3. Efficient combined security system

A wireless sensor network can be divided into several clusters. Each cluster has a number of sensors nodes and one of the nodes is elected as the coordinator (head). The head is responsible for the general mission and collecting the sensed data of other nodes and routing to the sink. For that, the head energy-consumption is higher than other nodes [16]. The energy analysis of the Kerberos protocol shown in [6] is based on the assumption that entity A can directly send/receive messages to/from the third party T. This is reasonable for small sensor networks, but not for large networks where the sensor nodes may be located apart from the base station. The communication energy cost of Kerberos depends on the transmit power level and on the number of intermediary nodes between A and T. Multi-hop communication between A and T increases overall energy consumption since any intermediary node has to forward the message to its neighbor

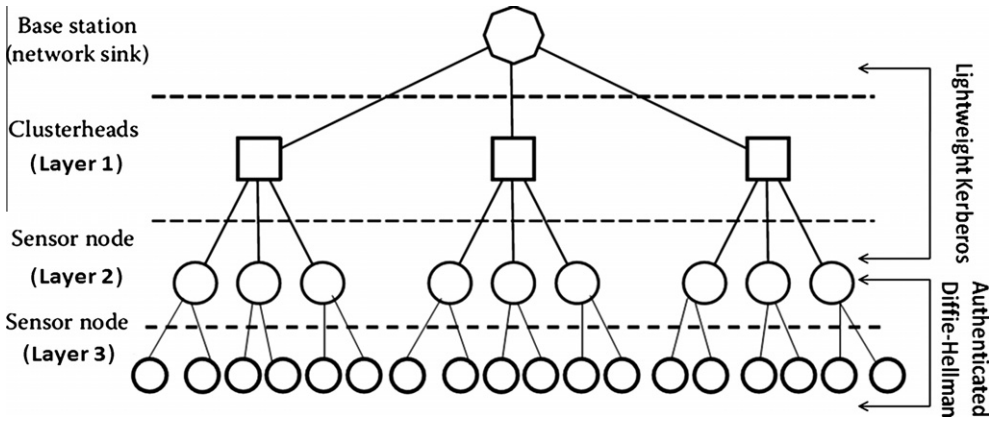


Figure 2 Hierarchical architecture for the combined system.

located on the route to the final destination. The Lightweight Kerberos protocol is more energy efficient than ECMQV when A can directly communicate with T or when at most one intermediary node lies between them. On the other hand, ECMQV requires less energy than Kerberos if there is more than one hop between A and T, which is always the case in large sensor networks. So, There is a need for system that compromise between the two protocol. That system is supposed to take the advantages of the two protocols and limits their shortening.

The suggested system in this paper combines the using of the protocols in the same network in the following way: the network is divided into three layers. The first layer is 1-hop layer, means the nodes in this layer can communicate directly with the base station, it contains the base station (the sink) and clusters heads. The second layer is 2-hop layer and the third one is 3-hop layer, these two layers contain the ordinary sensors that belong to clusters. The idea of this system is looking at the network as two networks: small network (contain layer 1) and large one (contain layer 2 and layer 3). Lightweight Kerberos protocol with short messages is applied on the small network and ECMQV protocol on the large one. When sensors in layer 2 want to communicate with layer 1 they will use the Lightweight Kerberos protocol with short messages. The architecture of the combined system will be as in Fig. 2.

The benefits of combining the two protocols in this system are as the following:

- Benefits of using Lightweight Kerberos protocol with short messages on layer 1 and for communication between layer 1 and layer 2:
  - The Lightweight Kerberos protocol is more energy efficient when the node is within direct communication to T (in most cases the base station) which is the case in layer 1 or when at most one intermediary node lies between them which is the case in layer 2.
  - Kerberos does not need extensive computation so, it save the energy on the heads which is critical to these nodes because they are responsible for the general mission, collecting the sensed data of other nodes and routing to the sink.
  - The number of heads and their neighbors is relatively small, so the total number of Kerberos communication messages will be relatively small. So, conserving the total energy of the network. For that, Kerberos is preferable in the small networks.

- Benefits of using ECMQV protocol among sensor nodes in layers 2 and 3:
  - ECMQV requires less energy than Kerberos if the communication between the node and T passes through more than one hop, which is the case in layer 3.
  - The sensor nodes do not do additional tasks as heads, so they have some energy to do the computation of ECMQV protocol.
  - The number of nodes in the two layers is relatively large and ECMQV is reasonable for large networks.
  - The number of communication messages needed for this protocol is small so improve the power consumption of the network.

- Using two strong protocols as Lightweight Kerberos and ECMQV will improve the network security.
- Using the two protocols increase the speed of the network. This speed is drawn from:
  - Using Kerberos in layer 1 and for communication between layer 1 and layer 2 reduce the number of calculation related to using ECMQV instead.
  - Using ECMQV among sensor nodes in layers 2 and 3 reduce the number of communication related to using Kerberos on this large number of sensor nodes.

All these benefits will be gained by using the combined system and the results in experimental results section support that. Unfortunately, switching between the two protocols in layer 2, using Kerberos for communication with layer 1 and ECMQV for communication among nodes in layer 2 and for communication between layer 2 and layer 3, cause some load in this layer. But comparing to the saving in the power and enhancing the security it can be used.

4. Experimental results

This section analyzes and compares the energy demands of Lightweight Kerberos key distribution, ECMQV and the combined system. The evaluation of the energy cost of cryptographic key establishment was conducted on a WINS sensor

Table 1 Total energy of combined system.

Pair layer	Energy consumption of combined system (mJ)
1	39.6–47.6
2	79.0–84.6
3	79.0–84.6
1 and 2	39.7–47.7
2 and 3	79.0–84.6

Table 2 Comparing energy consumption of Lightweight Kerberos, ECMQV and combined system.

Pattern no.	# Com. Pairs	Pairs layer	Energy consumption		
			Lightweight Kerberos (mJ)	ECMQV (mJ)	Combined system (mJ)
1	1	1	39.6–47.6	79.0–84.6	39.6–47.6
2	1	2	61.5–73.9	79.0–84.6	79.0–84.6
3	1	3	83.4–100.3	79.0–84.6	79.0–84.6
4	2	3	166.8–200.6	158–169.2	158–169.2
5	2	2,3	144.9–174.2	158–169.2	158–169.2
6	3	2	184.5–221.7	237–253.8	237–253.8
7	3	3	250.2–300.9	237–253.8	237–253.8
8	3	1,2,3	184.5–221.8	237–253.8	197.6–216.8
		Sum	1341	1353.6	1025.8

node from Rockwell Scientific [17]. The motivation for using this specific sensor node is to use the same node as the authors of [6] so that can directly compare the results and also, ARM processors have a considerable market share in the embedded systems field.

The evaluation of key establishment protocols considers both the energy that the Strong ARM consumes during the execution of cryptographic algorithms and the energy cost of radio communication. The energy characteristics of the WINS node reported in [18] were used. The energy required for the calculation of cryptographic primitives is simply the product of the average power consumption and the execution time. The execution time of the cryptographic primitives was determined through simulations with SimIt-ARM, a cycle-accurate instruction set simulator for the Strong ARM [19]. The communication energy depends on the distance between sending and receiving node and the time required for sending the message, which it is proportional to the message length and to the transmission rate. Also, the transmission of messages consumes energy on the sending and the receiving node.

4.1. Energy consumption of the combined system

The combined system, as described in Section 3, takes the advantages of the two protocols. The results show that the energy consumption for combined system will be less than using one of the two protocols alone. Table 1 includes the energy consumption for one pair authentication. The energy consumption changes according to the layer of the two nodes, because the applied protocol differs according to the layer. For example, if the two nodes in layer 1 the energy will be 39.6–47.6 mJ (Kerberos energy consumption) and if they in layers 2, 3 or one node in layer 2 and the other in 3 the energy will

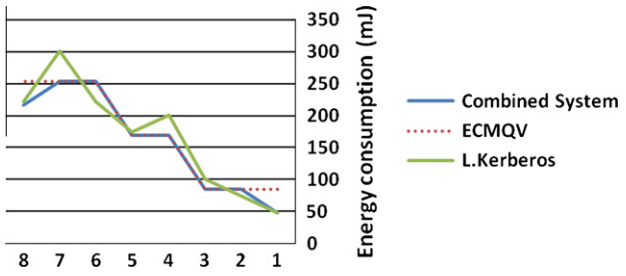


Figure 3 Energy consumption of Lightweight Kerberos, ECMQV and combined system.

be 79.0–84.6 mJ. But if one node in layer 1 and other in 2 the energy will be 39.7–47.7 mJ, the applied protocol will be Kerberos but the energy increased because of switching between the two protocols.

Table 2 compares the energy of the two protocols with the combined system. The results in this table show the efficiency of our combined system especially when the number of communicating nodes increases. Column 1 is the number of the pattern, Column 2 is the number of communicating pairs, Column 3 is layer where the nodes exist. At the last row, the summation of the energy of all pairs, show that the energy consumption for combined system in the network is less than ECMQV and Light Kerberos.

Fig. 3 shows energy consumption of Lightweight Kerberos, ECMQV and combined system. As shown, better efficiency is achieved for the combined system. The *x*-axis represents the number of the pattern and the *y*-axis represents the energy consumption.

5. Conclusion

This paper presented combined security system combines Lightweight Kerberos and ECMQV Protocols. The combining system takes the benefits of the two protocols. One of system benefits is enhancing the energy consumption. Saving energy means decreasing number of communications and computations, and this improve the speed of the network. Another benefit is, using two strong protocols as Lightweight Kerberos and ECMQV improves the network security. The experimental results of the system compared with energy cost of Lightweight Kerberos and ECMQV Protocols showed that, the overall energy cost of using the combined system is less that using of Lightweight Kerberos or ECMQV alone. These results are based on the energy characteristics of the WINS sensor node.

References

[1] Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E. A survey on sensor networks. IEEE Commun Mag 2002;40(8):102–14.
[2] Sen J. A survey on wireless sensor network security. Int J Commun Netw Inform Secur (IJCNIS) 2009;1(2).
[3] Needham R, Schroeder M. Using encryption for authentication in large networks of computers. Commun ACM 1978;21(12):993–9.
[4] Kohl J, Neuman B. The Kerberos network authentication service (Version 5). Internet Engineering Task Force, Networking Group, Internet Draft RFC 1510; September 1993.
[5] Perrig A, Szewczyk R, Wen V, Culler D, Tygar J. SPINS: security protocols for sensor networks. In: Proceedings of the 7th annual

international conference on mobile computing and networking. ACM Press; 2001. p. 189–99.
[6] Großsch J, Szekely A, Tillich S. The energy cost of cryptographic key establishment in wireless sensor networks. In: Proceedings of the 2nd ACM symposium on information, computer and communications security (ASIACCS 2007); 2007. p. 380–2.
[7] Diffie W, Hellman M. New directions in cryptography. IEEE Trans Inform Theory 1976;22(6):644–54.
[8] Law L, Menezes A, Qu M, Solinas J, Vanstone S. An efficient protocol for authenticated key agreement. Des, Codes Cryptogr 2003;28(2):119–34.
[9] Raghavendra C, Sivalingam K, Znati T. Wireless sensor networks. Kluwer Academic Publishers; 2004.
[10] Singh K, Muthukkumarasamy V. Analysis of proposed key establishment protocols in multi-tiered sensor networks. J Netw 2008;3(6).
[11] Menezes A, van Oorschot P, Vanstone S. Handbook of applied cryptography. CRC Press; 1996.
[12] Neuman B, Theodore Ts'o. Kerberos: an authentication service for computer networks. <http://gost.isi.edu/publications/kerberos-neuman-tso.html>; 30 March 2012.
[13] Carman D, Kruus P, Matt B. Constraints and approaches for distributed sensor network security. Technical report #00-010, NAI Labs, Network Associates, Inc., Glenwood, MD, USA; September 2000.
[14] Blake I, Seroussi G, Smart N. Elliptic curves in cryptography. Cambridge University Press; 1999.
[15] Blake I, Seroussi G, Smart N. Advances in elliptic curve cryptography. Cambridge University Press; 2005.
[16] Chen L, Lyu Z, Hong Z. An efficient cluster head selection strategy for wireless sensor network. In: Proceedings of the 5th international conference on genetic and evolutionary computing. IEEE; 2011.
[17] Agre J, Clare L, Pottie G, Romanov N. Development platform for self-organizing wireless sensor networks. In: Unattended ground sensor technologies and applications of Proceedings of SPIE, vol. 3713. SPIE; 1999. p. 257–68.
[18] Raghunathan V, Schurgers C, Park S, Srivastava M. Energy-aware wireless microsensor networks. IEEE Signal Process Mag 2002;19(2):40–50.
[19] Qin W. SimIt-ARM (Release 3.0). <https://sourceforge.net/projects/simit-arm/>; 30 March 2012.