World Applied Sciences Journal 21 (11): 1616-1620, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.21.11.1661

Enhancing Energy Efficiency in Wireless Sensor Networks via Improving Elliptic Curve Digital Signature Algorithm

Behbod Kheradmand

Department of Computer Engineering, Parsabsad Moghan Branch,
Islamic Azad University, Parsabad Moghan, Iran

Abstract: The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA). It is used in wireless sensor networks in order to establish security and allow legitimate users to join in the network and disseminate messages into the networks. The ECDSA needs an addition point and two multi scalar for verifying users' signature in WSNs, that because decrease signature verification speed, in result it has also incurred a series of problems such as high energy consumption and long verification delay. In this paper we propose an efficient technique to accelerate signature verification in WSNs by exploiting cooperation among sensor nodes. In order to verify the effectiveness of the proposed method, simulations were carried out on 4×4 grid-based WSN.

Key words: Digital Signature · ECDSA · Public Key · Wireless Sensor Network

INTRODUCTION

Wireless sensor networks [1-4] consist of small autonomous nodes. Each node has a small microprocessor, a radio chip, some sensors and is usually battery powered which limits network lifetime. Applications of wireless sensor networks run the gamut from environmental monitoring and health-care to industrial automation and military surveillance.

Radio signals [5] are used to exchange information in WSNs. Attackers are likely to use the same signal and pretend themselves as a member of networks and hence gain access to information. Cryptography algorithm in is used to thwart attackers which increase delay and energy consumption. Our purpose in the paper is to notably reduce delay and energy consumption.

The researcher intends to address the issue of speeding up the signature verification for public key in WSNs by exploiting the cooperation among sensor nodes. The impetus behind this research is that some sensor nodes will randomly release the result of their intermediate computation to their neighbors during the signature verification; subsequently, many sensor nodes can use the results of received intermediate computation to accelerate their signature verifications. In order to demonstrate the effectiveness of the proposed technique, the researcher conducted a case study of the broadcast authentication problem in a 4×4 grid-based WSN. The results of the detailed quantitative analysis show that our proposed scheme is significantly better than the traditional signature verification method for WSNs in terms of energy consumption of the whole network.

The Paper Is Organized as Follows: In section 2, ECDSA is described. After that, in section 3, improved ECDSA is explained. Then, in section 4, the simulation model is defined. Finally, in section 5, conclusions of the study are reported.

Elliptic Curve Digital Signature Algorithm (Ecdsa): The ECDSA [6, 7] is categorized in four phases which is described as follows:

- System parameters. Let G be a cyclic subgroup of E (fq) generated by the point P, with the prime order *n* and the identity element *O*. Let H: → {0, 1}, Z^n is a collision-resistant hash function.
- Initial set-up. Signer A randomly selects an integer d ∈ [1, n-1] and publishes its public key Q = d P. The parameter *d* is kept secret to signer A.

Corresponding Author: Behbod Kheradmand, Department of Computer Engineering,
Parsabsad Moghan Branch, Islamic Azad University, Parsabad Moghan, Iran.

- Signature generation. Signer A uses his/her private key *d* to generate a signature (r, s) for a message M$\in$ {0, 1}.
  (3-1) Select a random integer k $\in$ [1, n-1], compute R = k P and set *r* as the x-coordinate of R.
  (3-2) Compute $s = k^{-} (e + dr) \mod n$, where e = H (M).
  (3-3) If r, s $\in$ [1, n- 1], return (r, s); otherwise, go to Step (3-1).
- Signature verification. Upon receiving the message M $\in$ {0, 1} and the signature (r, s) from the signer A, the verifier B verifies the signature using A's public key Q.
  (4-1) Check that r, s $\in$ [1, n-1]. If any verification fails, return "reject signature".
  (4-2) Compute $R = s^{-} (e P + r Q)$, where e = H (M).
  (4-3) Check that the x-coordinate of R is equal to r. If verification succeeds, return 'accept signature"; otherwise, return "reject signature".

Note that if the signature verification in ECDSA is as slow as signature generation, it will be an undesirable property when using ECDSA for multi-user broadcast authentication in WSNs.

Related Work: When WSNs are deployed in proximity to adversaries, the broadcast authentication can be used as a crucial security mechanism for avoiding adversaries and attackers. The user contacts several sensor nodes in the vicinity and sends a request for the broadcast service. Then, the user and the sensor nodes conduct a mutual authentication procedure which grants the access to the WSN only to a legitimate user.

The ECDSA is employed in WSNs; ECDSA signature verification need two multi scalar and an addition point. Each node needs to calculate Eq. (1) for verifying an ECDSA signature [7].

$$R\in = s^{-1}(e P + r Q), \quad \text{Where } e=H(M) \tag{1}$$

In Fig. 1, the user will sign a query or command and forward it to the sensor nodes (e.g., nodes A, B and C). Nodes A, B and C then verify the user's signature. If the verification succeeds, nodes will send the package to other nodes (within their communication range).

The broadcast and authentication procedures continue until all the reachable nodes receive the user's broadcast package. If any verification fails during the broadcast, sensor nodes will drop the package and report it to the base station.

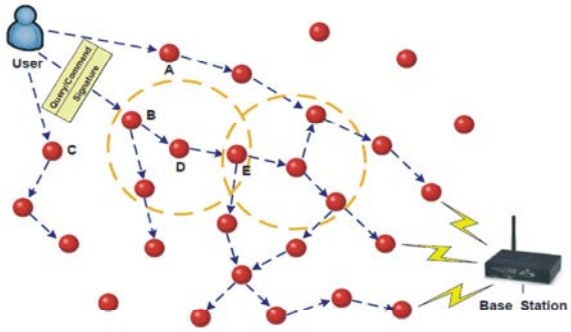

Fig. 1: User broadcast in wireless sensor networks

The Proposed Method : In the broadcast authentication procedure as shown in Fig.1, all the sensor nodes execute the same signature verification procedure after receiving a broadcast package.

According to the basic scheme, each node needs to calculate the followings for verifying an ECDSA signature which causes their energy consumption to increase.

$$R^{1}=s^{-1} (e P + r Q), \text{ Where } e=H (M); N_1= s^{-1} e \text{ and } N_2= s^{-1} r \tag{2}$$

All sensor nodes independently execute the same signature verification procedure during the broadcast authentication. So, if some sensor nodes consume their energy to release some intermediate results, verifying the signature of their neighbors can be accelerated significantly. Moreover, the energy consumption of the whole network will decrease as well.

In Fig. 2, the nodes A, B and C receive broadcast package <M, r and s > from user, where M is a user's query or command and (r, s) is the ECDSA signature for M. When all the three nodes finish the signature verification Zprocess successfully, nodes A (the green node) and B (the yellow node) decide to release (i.e., locally broadcast) their intermediate computation results N_1P and N_2Q. In this way, nodes D and E (the orange nodes), which are the neighbors of node A, can verify the digital signature quickly by performing an elliptic curve point addition $N_1P + N_2Q$, where N_2Q is computed by nodes D and E themselves and N_1P comes from the contribution of node A. Moreover, nodes F and G can also perform the signature verification process in a similar way. Hence, if a node in WSN releases its intermediate computation result, all of its neighbors can quickly verify the digital signature by calculating one scalar multiplication and one elliptic curve point addition which can result in about 50% efficiency improvement as compared to the traditional signature verification procedure.

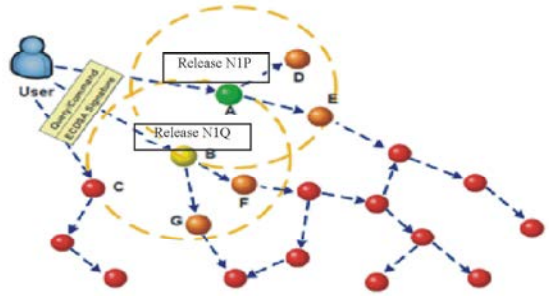

Fig. 2: The proposed scheme for signature verification through nodes' cooperation.

Some sensor nodes might receive both intermediate computation results N_1P and N_2Q from their neighboring nodes. However, sensor nodes cannot use both received N_1P and N_2Q in order to quickly verify the signature with one elliptic curve point addition; this is because an adversary can capture a sensor node; hence in order to avoid an attack, we only allow sensor nodes to use at most one intermediate result (i.e., N_1P or N_2Q) from their neighboring nodes for signature verification. Thus, it is further assumed that if some sensor nodes release their intermediate computation results, they will release N_2Q.

In the proposed scheme, each sensor node first waits for ô seconds and then caches some data packages (i.e., <M, r, s> or <M, r, s, N_2Q>) from its neighbors, where ô is selected in a way that the sensor node can receive at least one data package from an honest neighbor. Then, the sensor node checks whether the cached data packages have identical M, r, s and N_2Q. Note that the main goal of adversaries is to broadcast fake data packages into WSN. In contrast to adversaries, all honest nodes forward authentic data packages <M, r, s> or <M, r, s, N_2Q> to their neighboring nodes. So, if the senor node finds out that the received data packages have different M, r, s or N_2Q, it will report the potential attack to the base station immediately. On the other hand, if all the cached data packages have identical M, r, s and N_2Q, the sensor node will further check whether it has received useful data packages <M, r, s, N_2Q> for accelerating signature verification. If the sensor node decides that it has received useful data packages, it will calculate N_1P and then complete the signature verification with 1 SCA + 1 ADD. Otherwise, the sensor node will perform the traditional ECDSA signature verification with 2 SCA + 1 ADD. For the two cases above, if the signature is verified successfully, the sensor node will continue forwarding the broadcast package to its neighbors; otherwise, if the signature verification fails, the sensor node will send a signed report to the base station.

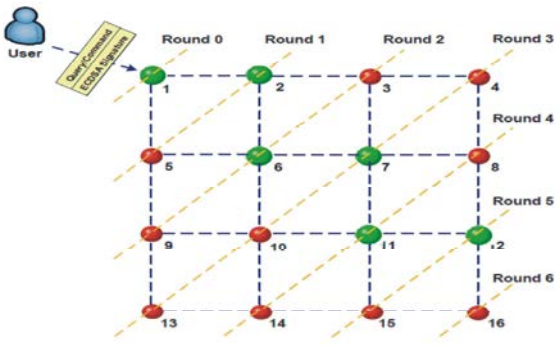

Fig. 3: Broadcast Authentication in a 4 × 4 grid-based WSN.

Analyzing the Proposed Technique: In this section, the efficacy of the proposed acceleration technique for signature verification is analyzed with respect to communication and computation overheads (in terms of energy consumption). The analysis focuses on a simple 4 × 4 grid-based WSN. We also compare our signature verification technique with the traditional ECDSA scheme when applied to the broadcast authentication in the 4 ×4 grid-based WSNs.

Case Study: The efficiency and effectiveness of our significantly fast signature verification technique is closely related to the deployment of WSNs and the distribution of attackers in the network. To analyze the effectiveness of our scheme, a case study is presented for the broadcast authentication problem in a 4×4 grid-based WSN, as illustrated in Fig. 3.

In the presented sensor network, each node can directly communicate only with its one-hop neighbors. A user sends its signed broadcast package to node 1 at Round 0. After six communication rounds, the broadcast package will be received and verified by all sensor nodes. Furthermore, in our significantly fast signature verification scheme, it is assumed that one sensor node will release the intermediate computation result N_2Q in each communication round (see the green nodes 1, 2, 6, 7, 11 and 12 in Fig. 3).

To give a detailed quantitative analysis, the researcher further assumes that MICAz Motes are used in the WSN. Under a typical configuration such as a 3V supply and a 7.37 MHz clock frequency, the MICAz Mote draws a current of 12mA in an active mode (i.e., CPU is operating) [8]. Based on the formula for calculating the energy consumption on MICAz Motes [9, 10], we obtain the following basic results:

- A Chipcon CC2420 radio used in MICAz motes consumes Es = 83.6μJ and Er = 90.4μJ to transmit and receives N_2Q with 40 bytes, respectively;
- An Atmega 128L microcontroller used in MICAz motes consumes about Ever = 22.68mJ and Esca = 11.52mJ to verify an ECDSA signature and computes a scalar multiplication on a 163-bit Koblitz curve.

It should be noted that the researcher will not compute the energy consumption of sensor nodes when sending and receiving the broadcast package throughout the whole network since it is the same for both faster and traditional signature verification schemes. In this study, the difference of both schemes in terms of communication and computation overhead is compared in the next section.

Simulation Results: In this section, the efficiency of our proposed scheme is evaluated in the presence of two collusive adversaries in the 4×4 grid-based WSN. It is assumed that the existing adversaries or attackers will broadcast identical bogus data packages to their neighbors. To maximize the influence of collusive adversaries, the researcher chose node 2 and node 5. Furthermore, it was assumed that the six green nodes (i.e., Nodes 1, 2, 6, 7, 11, 12) will locally broadcast the intermediate computation results N_2Q to their neighbors. It should be noted that node 6 would be cheated by collusive adversaries because of receiving two identical bogus data packages from node 2 and node 5. Although node 6 continued broadcasting the bogus data package, node 7 and node 10 discarded it and verified the signature itself because they received two different data packages. As a result, bogus data packages from the two collusive adversaries could not be injected into the WSN successfully. Moreover, if node 6 listens to the channel for one more communication round after broadcasting the bogus package, it is also possible for node 6 to find the potential attacks. More specifically, after node 7 and node 10 verify the signature successfully at Round 3, they broadcast the correct data package to their neighbors. Node 6 detects that all the data packages received from its neighbors (i.e., Nodes 2, 5, 7, 10) are different and therefore some attacks have occurred.

MATLAB was used in the study for the evaluation of the proposed scheme. The simulation was carried out on number of nodes and the average results of the simulations were used for the evaluation.

Table 1: Simulation parameters

parameters	value
Number of node	16
Number of green nodes	6
Number of attacker nodes	2
E_{sca}	11.52 mj
E_s	83.6 μj
E_r	90.4 μj
E_{ver}	22.68 mj

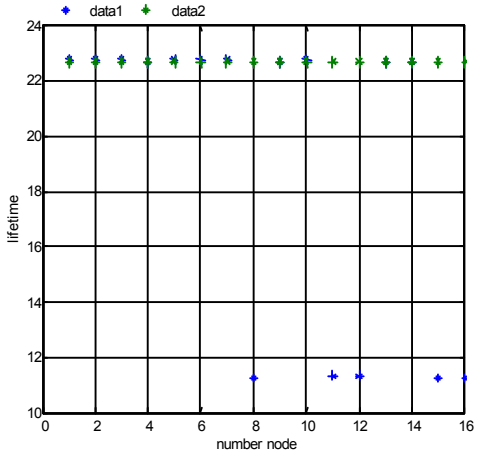

Fig. 4: Network Lifetime

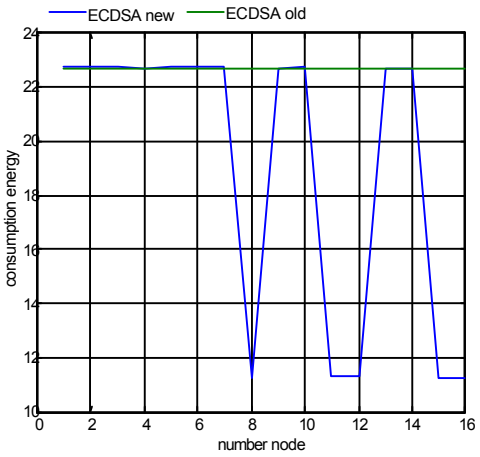

Fig. 5: Energy Consumptin

The required parameters and their values are presented in section 4.1 and table 1. Fig. 4 shows the increased improvement of the network lifetime using the proposed scheme as compared to the basic scheme. This figure depicts the approximate 25% improvement in network lifetime. .

In Figs 5 and 6, energy consumption was compared when the number of sensor nodes will locally broadcast the intermediate computation results N_2Q to their neighbors. This figure demonstrates that network energy consumption has decreased by about 15.5%.

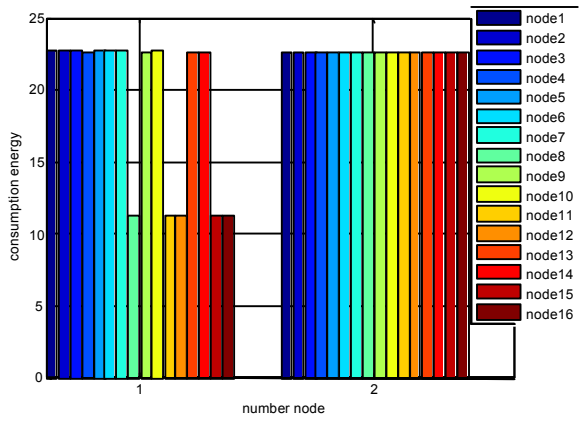

Fig. 6: Energy Consumption

CONCLUSIONS

The relatively slow signature verification process in public-key cryptosystems undesirably leads to high energy consumption in broadcast authentication in WSNs. In this paper, a novel and efficient acceleration technique was proposed for the signature verification process in WSNs. It fully promotes the cooperation among sensor nodes and significantly improves efficient energy consumption for the whole network. As a case study, the researcher applied the technique to the broadcast authentication in a 4×4 grid-based WSN and analyzed the performance and effectiveness of it in the presence of collusive adversaries. Moreover, the quantitative analysis and evaluation as reported above showed that the presented scheme is capable of saving energy consumption by about 15.5 % and also running 50% faster than traditional signature verification method. As a follow-up and further study, the researchers should analyze the effectiveness and efficiency of the proposed technique when it is applied to other WSNs with more complicated topologies and deployments.

REFERENCES

1. Patwardhan, A., 2010. "Energy based Path Planning for WSNs", International Journal on Emerging Technologies, 1: 16-18.
2. Sen, J., 2009. "A Survey on Wireless Sensor Network Security", International Journal of Communication Networks and Information Security(IJCNIS), 1(2).
3. Boyle, D. and T. Newe, 2008. "Security Wireless Sensor Networks: Security Architectures", Journal of Networks, 3(1).
4. Ghaffari, A. and F. Kaviani, 2012. "Surface Coverage Improvement in Wireless Sensor Network", World Applied Science Journal, 16(1): 67-72.
5. Ren, K., S. Yu, W. Lou and Y. Zhang, 2007. "On Broadcast Authentication in Wireless Sensor Networks", IEEE Transactions on Wireless Communications, Chicago,
6. Hankerson, D., A. Menezes and S. Vanstone, 2004. "Guide to Elliptic Curve Cryptography", New York, USA: Springer-Verlag,
7. Khalique, A., K. Singh and S. Sood, 2010. "Implementation of Elliptic Curve Digital Signature Algorithm ", International Journal of Computer Aplication (0975-8887), 2(2).
8. Crossbow Technology Inc, "MICAz-Wireless Measurement System". Available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet. pdf.
9. Driessen, B., A. Poschmann and C. Paar, 2008. "Comparison of Innovative Signature Algorithms for WSNs", Proceedings of the First ACM Conference on Wireless Network Security (WiSec'08), pp: 30-35.
10. Wang, H. and Q. Li, 2006."Efficient Implementation of Public Key Cryptosystems on MICAz Motes", The 8th International Conference on Information and Communications Security-ICICS, Berlin, Germany,
11. Liu, A. and P. Ning, 2008. "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks", Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN), Raleigh, NC,
12. Singh, S. and H.K. Verma, 2011. "Security for Wireless Sensor Networks", International Journal on Computer Science and Engineering (IJCSE) 3(6).
13. Antipa, A., D. Brown, R. Gallant, R. Lambert, R. Struik and S. Vanstone, 2008. " Accelerated Verification of ECDSA Signatures", Berlin, Germany,
14. Engels, D., X. Fan, G. Gong, H. Hu and E.M. Smith, 2009. "Ultra-Lightweight cryptography for Low-Cost RFID Tags: Hummingbird Algorithm and Protocol", Centre for Applied Cryptographic Research (CACR) Technical Reports, Berlin, Germany,