Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications

Conference Paper · April 2014
DOI: 10.1109/WCNC.2014.6952860

5 authors, including:

Pawani Porambage
University of Oulu
27 PUBLICATIONS **181** CITATIONS
SEE PROFILE

Corinna Schmitt
Universität der Bundeswehr München
39 PUBLICATIONS **356** CITATIONS
SEE PROFILE

Pardeep Kumar
University of Oxford
43 PUBLICATIONS **623** CITATIONS
SEE PROFILE

Andrei Gurtov
Linköping University
279 PUBLICATIONS **4,634** CITATIONS
SEE PROFILE

Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications

Pawani Porambage*, Corinna Schmitt†, Pardeep Kumar*, Andrei Gurtov‡ and Mika Ylianttila*

*Centre for Wireless Communications, University of Oulu, P.o.Box 4500, FI-90014 Oulu, Finland
{pporamba, pkumar, mika.ylianttila}@ee.oulu.fi
†Institute of Informatics, University of Zurich, Binzmühlestrasse 14, CH-8050 Zurich, Switzerland
schmitt@ifi.uzh.ch
‡Department of Computer Science and Engineering, Aalto University, FI-00076 Aalto, Finland
gurtov@cs.helsinki.fi

***Abstract*—In the centralized Wireless Sensor Network (WSN) architecture there exists a central entity, which acquires, processes and provides information from sensor nodes. Conversely, in the WSN applications in distributed Internet of Things (IoT) architecture, sensor nodes sense data, process, exchange information and perform collaboratively with other sensor nodes and end-users. In order to maintain the trustworthy connectivity and the accessibility of distributed IoT, it is important to establish secure links for end-to-end communication with proper authentication. The authors propose an implicit certificate-based authentication mechanism for WSNs in distributed IoT applications. The developed two-phase authentication protocol allows the sensor nodes and the end-users to authenticate each other and initiate secure connections. The proposed protocol supports the resource scarcity of the sensor nodes, heterogeneity and scalability of the network. The performance and security analysis justify that the proposed scheme is viable to deploy in resource constrained WSNs.**

***Index Terms*—Distributed Internet of Things, Wireless Sensor Networks, implicit certificate, security, authentication**

I. Introduction

Wireless Sensor Network (WSN) is a key technological building block of Internet of Things (IoT), which is considered the future evolution of the Internet. During past decade WSN and its security are well investigated amongst the industry and academia. Although the concept and applications of IoT have no novelty right now IoT security is still in its infancy. However, substantial amount of research work have been done to identify the challenges and possible protection mechanisms for securing IoT as shown throughout references [1], [2], [3], [4], and [5]. In the context of IoT application domains, WSN architectures exist as centralized and distributed approaches [6]. In centralized networks, there is little or no support to access the data sensing network devices directly. The distributed networks allow the end-users and other network entities to obtain raw data straightaway from the sensor nodes. Although provisioning of services is located at the edge of the network, different application platforms and end-users can collaborate dynamically with each other. As a result of the decentralized and distributed nature of the network, it is essential to consider the secure management of identity and authentication of connecting devices. In IoT applications multiple entities (*e.g.*, sensing nodes, service providers, and information processing systems) have to authenticate each other to establish a trusted network. The authentication protocols should be not only resistive to malicious attacks, but also they should be lightweight to be deployed in less performing edge devices of the WSN.

Rather using for generic WSN applications, IoT combined WSN use-cases are currently deployed in smart-home, smart-city, health-care, and industry monitoring applications. In a hospital environment there can be different sensors installed in monitoring health conditions of patients (*e.g.*, blood pressure, heart beat, oxygen concentration). Doctors, who are outside the hospital, might be interested in examining health records of particular patients. Another data access scenario can be that some medical machineries, which maintain the environmental conditions of the ward, need to access similar records. In this scenario, doctors have to access the sensor nodes as an end-user and the machineries have to collaborate as the edge nodes from the same or a distinctive WSN. However, in all three cases, the two communication parties need to prove their authenticity to each other before establishing a secure communication link. In power plant monitoring applications WSNs are deployed inside the factory premises to obtain raw data on machinery vibration, temperature, flow-rate and light intensity. Their collected data are used to identify machine abnormalities and to create safety alarms. There can be instances where the users inside and outside the power plants want to acquire raw data directly from the sensor nodes. The end-users and the sensor nodes have to authenticate each other before transferring raw data.

Our contribution to this paper is the design and evaluation of a two-phase authentication scheme for WSNs in distributed IoT applications. Since the edge nodes and end-users exploit implicit certificates for mutual authentication, the protocol is lightweight and it supports the heterogeneity of the entities. We have implemented the scheme and taken performance measurements on the high resource restricted sensor nodes along with a security analysis.

This paper is organized as follows: Section II provides a

978-1-4799-3083-8/14/$31.00

brief overview about the related work. Section III describes the system model and the notations used. Section IV and Section V respectively present the proposed authentication protocol, and its assessment for deploying in network devices. Finally, Section VI concludes the paper.

II. Related Work

In centralized WSN, data from the sensor nodes are transmitted to a single central location, which process, combine, and provide information acquisition for customers [1]. Due to the high data availability and massive network size, processing of data on a single location might be inefficient, congested and undertaking a high risk at single entity failure. In distributed networks, the sensor nodes can retrieve, process and provide data for other entities and end-users. Distributed architecture supports the IoT network applications by providing services at local level, and collaborating with all the network devices and users to achieve common goals. Due the network heterogeneity and device mobility, there can be many security threats and issues are encountering with distributed IoT. In reference [1] Roman et al. have identified security challenges in distributed IoT. According to their study, network entity identity, authentication, access control, and secure communication channel establishment are major security concerns in distributed IoT. The proposing mechanisms should be robust to node mobility and network scalability due to the dynamic behavior of nodes and the network needs to scale up after installation.

Substantial works have been done to address device authentication for generic WSNs in centralized and distributed architectures [7]. Nevertheless, WSN applications in distributed IoT need to provide authentication between heterogeneous edge devices (sensor nodes, actuators, etc.) and end-users (*e.g.*, controllers and information consumers) that create and cease connectivity dynamically. In many scenarios, the sensor nodes belonging to the same geographical area (*e.g.*, hospital environment) are granted identities by a local identity provider (*e.g.*, a central server). However, the local entities are not only being able to authenticate to each other within the group, but also they need to communicate with external entities. Therefore, both the internal and external entities need to retrieve temporary identities from a common authority, which enable them to authenticate each other and bring them to a common platform. In WSN inter-domain collaboration is already existing using access tokens [8]. Since lower layer protocols do not provide end-to-end authentication, it is important to address on application layer protocol to prove trusted communication [3]. Host Identity Protocol (HIP) is a network layer authentication protocol that supports secure end-to-end mobility and multihoming [9], [10].

Datagram Transport Layer Security (DTLS) is a mutual authentication scheme proposed for unreliable data transmission networks with User Datagram Protocol (UDP) links [11]. DTLS protocol is an adaptation of well-known TLS protocol, which is used to secure HTTPS in conventional Internet. DTLS is appropriate for low power lossy networks since it inherently supports unreliable datagram transport. However due to the existence of 12 message transfers to complete DTLS handshake it induces a significant overhead to the network traffic. In reference [3] a detailed performance analysis of fully implemented DTLS protocol was presented. The main drawback is the utilization of X.509 certificates and RSA public keys with DTLS handshake, which are too heavy for low performing sensor nodes with less processing power. Therefore, as presented in reference [12] Elliptic Curve Cryptography (ECC) based implicit certificates would bring fewer overheads to constrained networks. They can be extended for pervasive authentication mechanisms in WSN applications in distributed IoT. Therefore, the authors were interested in Elliptic Curve Qu-Vanstone (ECQV) implicit certificate scheme and Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol for developing the proposed solution [13], [14], [15].

III. System Model and Notations

In this section, the authors provide details about the system model, where the protocol is modeled, and the notations used.

A. System Model

In Figure 1 we illustrate the assumed network architecture for proposed authentication scheme, where end-users can collaborate with different edge devices in order to obtain a particular information or service. The edge networks may include heterogeneous devices and the end-users can be humans or virtual entities.

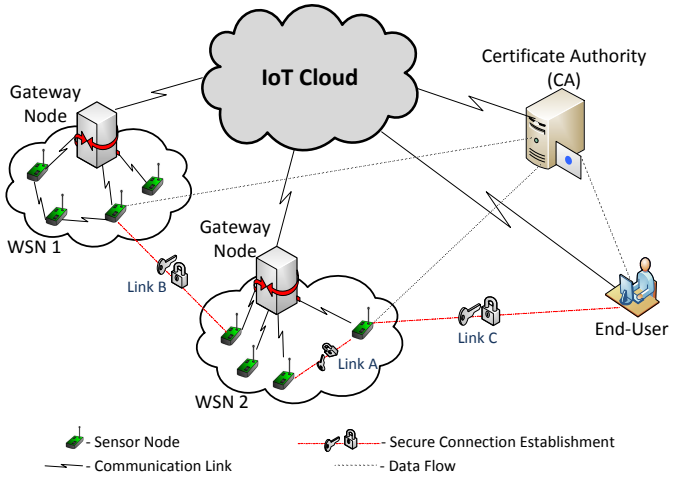


Fig. 1: Assumed network architecture

Based on Figure 1 authentication is considered for three communication scenarios particularly:

1) Two sensor nodes in the same WSN (`Link A`).
2) Two sensor nodes from distinctive WSNs (`Link B`).
3) An end-user and a sensor node (`Link C`).

Before starting the actual authentication protocol between two network entities, it is necessary to undergo a registration process by every communication party, in order to retrieve cryptographic credentials that are used for the authentication phase. Every edge device and end-user have to acquire security

credentials (*e.g.*, cryptographic suites and implicit certificates) from a trusted third party such as a Certificate Authority (CA), which is a highly resource rich server and already known by the edge nodes, during the registration phase. Having a valid certificate allows the two entities for mutual authentication irrespective of their local network. Existing nodes can change their locations dynamically after requesting a new certificate. No matter the size of the network, adding new nodes can easily extend the data acquisition and service providing networks. It is assumed that CA can recognize the valid identities and communicate with the network entities, which are requesting security credentials [7]. As illustrated in Figure 1, network entities first communicate with CA along the already established communication links heading through an IoT cloud. In this paper, an end-to-end authentication in the application layer is proposed, while relying on other security schemes in lower layer communication [16]. Subsequently, the edge devices and end-users can mutually authenticate and establish secure communication channels, due to distribute nature of the entire architecture.

B. Notations

The notations used in this paper are defined in Table I. Elliptic Curve (EC) parameters are denoted by *q, a, b, G, and n*. q is a prime, which indicates finite field F_q. The variables a and b are coefficients of EC $y^2 = x^3 + ax + b$ where $4a^3 + 27b^2 \neq 0$. G is the base point generator with order of n, which is also a prime [14].

TABLE I: Notations used in cryptographic algorithms

Notation	Description
K	Symmetric key for initial message authentication
r_U	Secret random integer value generated by U
R_U	EC point for certificate request sent by node U
$Cert_U$	Implicit certificate of i^{th} node
e	Integer used to keep hash value of $Cert_U$.
s	Integer used to compute private key of the requestor node.
d_U	Node U's private key
Q_U	Node U's public key
N_U	Random cryptographic nonce generated by node U.
K_{UV}	Link key between nodes U and V

IV. Two-phase Authentication Solution

The proposed authentication scheme for WSN applications in distributed IoT encompasses two phases. First phase, called *Registration Phase*, is to obtain security credentials from a trusted party as described Section IV-A. Second phase, called *Authentication Phase*, is to start mutually trusted communication between two network entities, using the obtained security credentials. Details are presented in Section IV-B. The design of the proposed solution is inspired by ECQV implicit certificate scheme [13] and ECDH key exchange mechanism [14].

A. Registration Phase

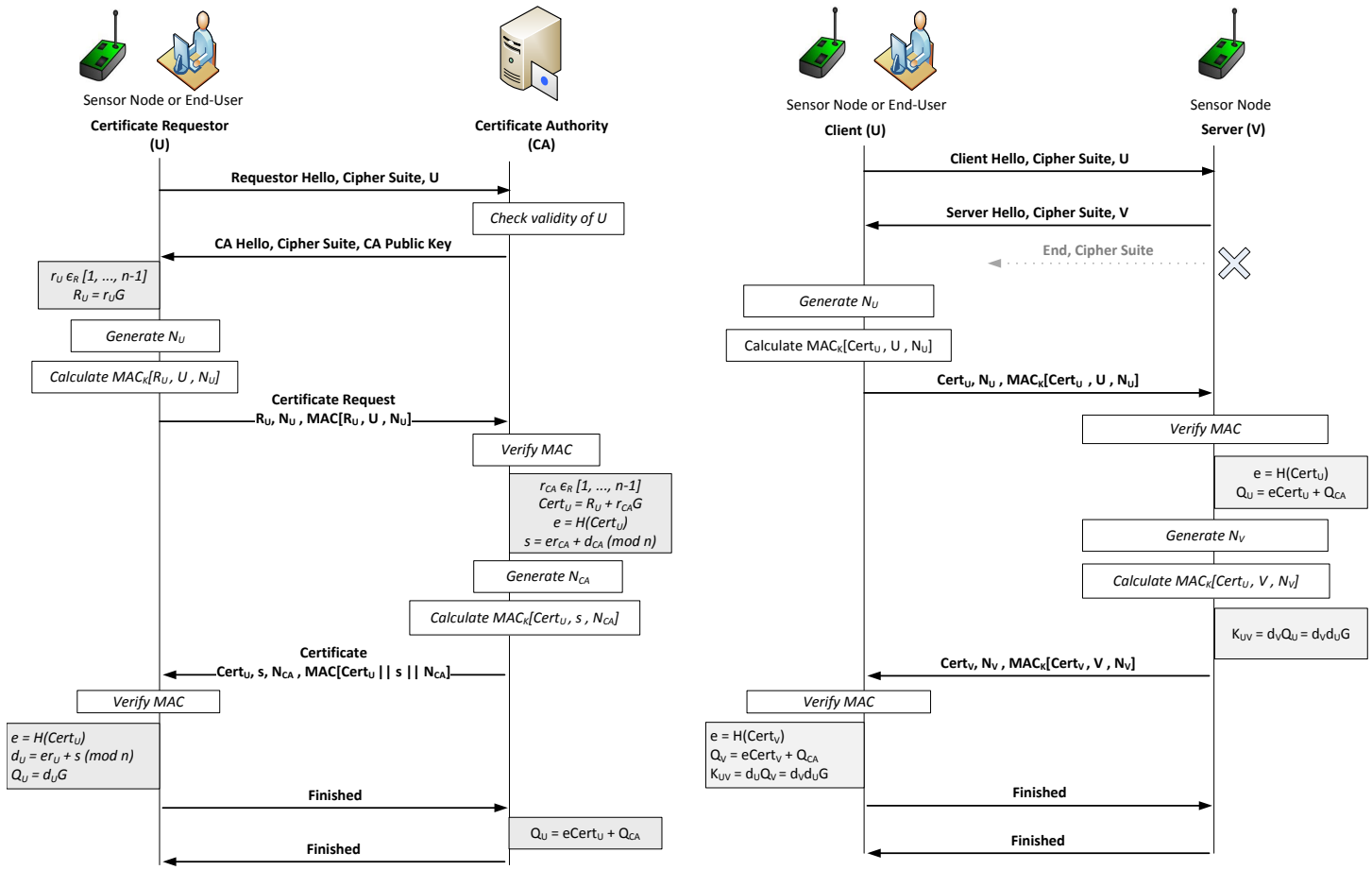
As illustrated in Figure 1 network edge devices (*e.g.*, sensor nodes) and end-users request security credentials and certificates from the certificate authority (CA). Upon the certificate requests and the requestors' identity, CA issues implicit certificates. The flow diagram of the registration phase is shown in Figure 2a, where white boxes indicate performed actions by the entities and grey boxes announce used variables. This information is the same for Figure 2b. The certificate requestor (Node U) can be a sensor node or an end-user. The protocol starts the handshake with a `Requestor Hello` message, node identity (U), and cipher suites that are supported by the requestor. It is assumed that cipher suites are embedded in the sensor nodes and known to end-users at the deployment phase or at the phase of granting access to the particular network. The cipher suites are common to all the edge devices and would include the available cipher options at the requestor side such as EC parameters, message authentication key (K) for MAC, hash function (H) and AES key size for bulk encryption; *e.g.*, *CERT_ECC160_*K_1*_SHA1_AES128* stands for 160-bit EC curves, K_1 message authentication key, SHA1 and 128-bit AES. CA uses node or user identities to verify the legitimacy of the certificate requestors. If the requestor identity verification is successful, CA agrees to one cipher suite combination from the received options, and sends `CA Hello` message with its public key (Q_{CA}) as an unprotected message to approve the initiation of the handshake.

Upon receiving `CA Hello` message, the requestor generates a certificate request EC point (R_U) and a true nonce (N_U), calculates their Message Authentication Code (MAC) value and sends `Certificate Request` message to CA. True nonce and MAC values are included in order to maintain freshness and integrity of the messages. CA first verifies the MAC value to identify the integrity of the request, and then calculates the implicit certificate ($Cert_U$) and private key construction value (s). CA sends `Certificate` message including the two values followed by a nonce (N_{CA}) and MAC value. Upon receiving `Certificate` and after verifying the MAC value, the requestor computes its own private (d_U) and public (Q_U) keys.

The `Finished` message contains an encrypted message digest of previous handshake messages using the requestor's public key (Q_U). According to the EC arithmetic operations that performed for calculating keys [13], CA is also capable of computing Q_U and using it for encrypting previous messages (*e.g.*, to generate `Finished` message). CA answers with the `Finished` message to complete the handshake of the registration phase.

B. Authentication Phase

In order to establish authenticated communication, the edge nodes and end-users should possess implicit certificates for particular cipher suites (*e.g.*, undergoing the registration phase). As shown in Figure 2b several message transfers of the authentication phase between the client node U (*e.g.*, an end-user or a sensor node) and the server node V (*e.g.*, a sensor node) are considered. First the client sends the `Client Hello` message to the server followed by cipher suite options and its identity (U). The client only sends the cipher suites, which its implicit certificates are composed of. If the server possesses certificates,



(a) Flow Diagram of Registration Phase

(b) Authentication Phase (end-to-end authentication)

Fig. 2: Overview of the Protocol

which matches the given list of cipher suites, it agrees to one cipher suite and replies with the `Server Hello` message and its identity. Otherwise, the server abolishes the handshake by sending the `End` message followed by its existing cipher suite options, as shown by X in Figure 2b. Under such scenarios, the clients have to request new certificates, which belong to the particular cipher, and initiate the handshake from the beginning.

Upon receiving the `Server Hello` message, the rest of the protocol can be further proceeded. Then the client sends its certificate accompanied with a random cryptographic nonce and the MAC value. If the MAC verification is successful, the server calculates the client's public key (Q_U), using the received certificate $Cert_U$, and CA's public key (Q_{CA}). The server uses its private key d_V and client's public key Q_U to compute a common secret between two parties: $K_{UV} = d_V Q_U$. Accordingly the server sends its certificate $Cert_V$, nonce N_V and MAC value. Similar to the server side, the client verifies MAC, computes the public key of the server, and derives the common key using own private key and the public key of the server: $K_{UV} = d_U Q_V$.

Finally, exchanging the `Finished` messages concludes the handshake and is similar to the registration phase. This `Finished` message composes of previous handshake messages which are encrypted by the common key K_{UV}. At the end of six message transfers, the two edge nodes can authenticate each other, and establish a common secret key and a secure communication link that can be used for securing further data acquisitions between the client and the server.

V. Analytical Justification of Protocol

In this section, a detailed analysis of the proposed authentication protocol in terms of performance and security is provided. Furthermore, it is justified why the proposed solution can be deployed on resource-constrained devices in WSNs in distributed IoT applications. At the end of the section, the authors present the limitations that are engaged with the current proposal and should be improved in future.

A. Performance Analysis

Lower layer security protocols do not provide end-to-end communication security [11]. DTLS is the widely used application level security protocol for authentication in IoT networks. Variants of DTLS handshakes are based on ECC and used with

RSA and X.509 certificates [17]. The exploitations of RSA and X.509 certificates with DTLS provide inter-operability. The major drawbacks are, RSA has the large key size which is 2048-bit and standard X.509 certificates are in the order of 1 kB in size.

The developed solution is deployed in a network with TelosB sensor nodes [18] that have IEEE 802.15.4 compliant CC2420 RF transceivers. The hardware includes 8 MHz, 16-bit MCU with 10 Kbyte RAM and 48 Kbyte ROM. CC2420 RF transceiver has a maximum data rate of 250 kbps and frequency band of 2400 MHz. The proposed scheme is developed in NesC on TinyOS 2.1.2 [19]. ECC (*i.e.*, for EC arithmetic operations) and natural number (NN) (*i.e.*, for large natural number operations) interfaces are utilized from TinyECC configurable library [20]. *secp160r1* EC domain parameters are used as defined in [14]. The authors of this paper utilized EC optimization techniques provided in TinyECC such as Barrett Reduction to speed up modulo operations, Hybrid multiplication and squaring for integer multiplication, Projective Coordinate Systems for point addition, and Sliding Window for scalar multiplication. SHA-1 is used as the one way cryptographic function (H). The *check_size.pl* script is used to obtain memory consumption values (*e.g.*, for RAM and ROM) required by each operation and the execution times are measured directly on the sensor nodes. As given in Table II memory utilization values are taken two phases with respect to the communicating node. Since the transmission time depends on the size of the network and the distance between the nodes, only the execution time for the particular operations performed at the edge nodes or CA, as depicted in Table III, are considered. The energy consumptions were then calculated as $V \times I \times t$ based on the voltage (V), the current (I), and the execution time (t) on TelosB sensor nodes [18]. Similar to reference [20] we considered V as 3 v and I as 1.8 mA. For the sake of simplicity and comparison TelosB nodes were used to perform the functions of edge nodes and CA.

TABLE II: Memory utilization

	RAM (bytes)	ROM (bytes)
Registration Phase		
Edge node operations	1410	11774
CA operations	2332	16576
Authentication Phase		
Edge node operations	1530	11650

According to the memory measurements, a TelosB sensor node consumes approximately 2940 bytes of RAM and 23424 bytes of ROM for accommodating both phases of the authentication protocol. The overall implementation of two phases is still below the 10 kB RAM and 48 kB ROM provided by TelosB nodes. Although the memory consumption is higher for CA operations, in the real-time deployment it would be tolerable for a resource rich device. During the registration phase the approximate time utilization at the certificate requestor's side is 8235 ms and CA's side is 10591 ms. Edge node authentication consumes about 8444 ms at each node. A TelosB node consumes nearly 44.47 mJ and 45.6 mJ for registration and authentication phases respectively. These timing, energy, and memory values can be improved by using further optimized basic ECC arithmetic operations. However, experimental results show that proposed authentication mechanism can be easily deployed in low power less performing devices.

TABLE III: Time and energy consumption

Operation	Time (ms)	Energy (mJ)
Registration Phase		
Initialization	2709	14.63
Cert Req generation	2764	14.93
Cert generation	5728	30.93
Cert verification at U	2758	14.89
`Finished` msg at U	4	0.02
`Finished` msg at CA	2154	11.63
Authentication Phase		
Initialization	2672	14.43
Key Computation (Server/Client)	5768	31.15
`Finished` msg	4	0.02

In the proposed two-phase authentication protocol implicit certificates are used as 160 bits EC points and, therefore, the size of the certificate is only 44 bytes. The utilization of optimally designed EC curves can reduce the certificate size and the exploitation of compression techniques can further decrease the overall message size. Retransmission clocks can be used at both communicating parties, for identifying time outs and retransmitting when there is a message loss. Furthermore, the authentication protocol supports to scale up the network, since the newly adding nodes can authenticate themselves after undergoing the registration phase. As the certificates are not based on the physical locations of the edge devices, they do not have to be alternated according to nodes' mobility.

B. Security Analysis

The proposed authentication is based on ECC and inherently secured due to the PKC nature. While using EC scalar-point multiplication, the scheme is provably secured under the random oracle model that the discrete logarithm problem over the subgroup is intractable. The advantage of using ECC is that it provides an equal security for RSA, however with less overhead (*e.g.*, 160 bit ECC equals RSA with 1024 key size). It is very common that Denial of service (DoS) attacks can be launched against distributed IoT. During the registration phase, the first `Hello` message contains the certificate requestor's identity, which is analyzed by CA. If the unauthorized requestors are trying to access, the CA can identify them at the beginning of identity verification and protect itself from DoS attacks. Similarly, during the authentication phase the cryptographic credentials are exchanged only after the successful `Hello` message exchange, which provides the protection of DoS attacks. In order to overcome illegal message alternations by malicious users and DoS attacks, the subsequent messages contain MAC

with the common authentication key K for preserving data integrity. Cryptographic random nonce is used for keeping the message freshness during the handshake.

With the above performance and security analysis we have shown that the proposed authentication scheme can be easily deployed in the resource constrained devices, along with reasonably high security. Due to the small size of the certificates, they consume less amount of memory at each sensor node. Since the protocol is based on standard ECC operations, which are supportive for all the sensor nodes irrespective of their manufacturer and can be performed at end-users, it is feasible to deploy in IoT enabled heterogeneous WSN. Additionally, the proposed authentication scheme supports the new node (or end-user) addition, and mobility of edge devices and end-users.

C. Limitations of the Proposed Scheme

Unlikely to the utilization of commercialized explicit certificates, it has not yet been matured and standardized to customize implicit certificates in WSN security applications. At the initial phase of the authentication phase, since the nodes are verified based on their identities, the network should have a very good node identification mechanism, which is strongly secured. At the current stage of the protocol, node capture attacks can create very limited harm to the entire network. If an adversary captures a sensor node, it can reveal the node's certificate, public-private keys and CA's public key. Only by having these values, the attacker cannot generate a new certificate and keys. However, they can use those values for communicating with another legitimate user. It might be possible to avoid the losses due to the node capturing attacks of this sort by using beacon message technique, as explained in references [21] and [22]. However, under this stage resistivity for node capturing attacks were not addressed in the design of the proposed solution.

VI. Conclusion

In this paper, the authors have introduced and analyzed an authentication mechanism for WSNs in distributed IoT applications. The proposed authentication comprises two phases; for obtaining cryptographic credentials to the edge devices and end-users, and authenticating mutual communication. Using this authentication protocols the end-users can authenticate themselves to the sensor nodes directly and acquire sensed data and services. With the experimental results, it was shown that the authentication protocol is feasible to deploy in the low performing resource constrained network devices in WSNs. The protocol supports the distributed IoT applications, since the certificates are lightweight and can be handled by the high resource constrained devices irrespective of their originality. According to our security analysis, the proposed scheme is secured under certain types of attacks. However, the authors have shown the possible methods for improving its efficiency and security in further expansions. In future, the authors intend to extend the utilization of implicit certificates for access control and multicasting in the massive scale distributed IoT network applications.

Acknowledgement

This work has been supported by Tekes under Massive Scale Machine-to-Machine Service (MAMMotH) project and Academy of Finland project SEMOHealth.

References

[1] R. Roman, J. Zhou, and J. Lopez, "On the Features and Challenges of Security and Privacy in Distributed Internet of Things," *Computer Networks*, vol. 57, no. 10, pp. 2266 – 2279, 2013.

[2] S. Raza, D. Trabalza, and T. Voigt, "6LoWPAN Compressed DTLS for CoAP," in *Proceedings of 8th IEEE Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2012, pp. 287–289.

[3] T. Kothmayr, C. Schmitt, W. Hu, M. Brnig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Networks, ELSEVIER*, 2013.

[4] M. Brachmann, S. L. Keoh, O. Morchon, and S. Kumar, "End-to-End Transport Security in the IP-Based Internet of Things," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, 2012, pp. 1–5.

[5] R. H. Weber, "Internet of Things New Security and Privacy Challenges," *Computer Law and Security Review*, vol. 26, no. 1, pp. 23–30, 2010.

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[7] W. Hu, H. Tan, P. Corke, W. C. Shih, and S. Jha, "Toward Trusted Wireless Sensor Networks," *ACM Transaction on Sensor Networks*, vol. 7, no. 1, pp. 5:1–5:25, 2010.

[8] G. Bai, L. Yan, L. Gu, Y. Guo, and X. Chen, "Context-aware Usage Control for Web of Things," *Security and Communication Networks*, 2012.

[9] A. Gurtov, M. Komu, and R. Moskowitz, "Host Identity Protocol (HIP): Identifier/Locator Split for Host Mobility and Multihoming," *Internet Protocol Journal*, vol. 12, no. 1, pp. 27–32, 2009.

[10] J. Pellikka, Z. Faigl, and A. Gurtov, "Lightweight Host and User Authentication Protocol for All-IP Telecom Networks," in *Proceedings of 3rd IEEE Workshop on Data Security and PrivAcy in wireless Networks(D-SPAN)*, 2012.

[11] Z. Shelby, K. Hartke, and C. Bormann, "Constrained Application Protocol (CoAP)," IETF draft, RFC editor, June 2013. [Online]. Available: http://tools.ietf.org/pdf/draft-ietf-core-coap-18.pdf

[12] P. Kotzanikolaou and E. Magkos, "Hybrid Key Establishment for Multi-phase Self-Organized Sensor Networks," in *Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2005, pp. 581–587.

[13] "SEC4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV), version 0.97," www.secg.org, August 2013.

[14] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to Elliptic Curve Cryptography*. Springer, 2004.

[15] P. Porambage, P. Kumar, C. Schmitt, A. Gurtov, and M. Ylianttila, "Certificate Based Pairwise Key Establishment Protocol for Wireless Sensor Networks," in *Proceedings of IEEE 16th International Conference on Computational Science and Engineering*, 2013, pp. 667–674.

[16] "IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15.4., 2011.

[17] V. Gupta, M. Wurm, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle, and S. C. Shantz, "Sizzle: A standards-based end-to-end security architecture for the embedded internet," *Pervasive and Mobile Computing*, vol. 1, no. 4, pp. 425 – 445, 2005.

[18] "TelosB Datasheet," Crossbow Inc., Tech. Rep., 2013. [Online]. Available: http://www.datasheetarchive.com/4--Crossbow*-datasheet.html

[19] "TinyOS Documentation," www.tinyos.net, August 2013.

[20] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," in *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)*, 2008, pp. 245–256.

[21] R. Lu, X. Li, X. Liang, X. Shen, and X. Lin, "GRS: The Green, Reliability, and Security of Emerging Machine to Machine Communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 28–35, 2011.

[22] S. H. Jokhio, I. A. Jokhio, and A. H. Kemp, "Node Capture Attack Detection and Defence in Wireless Sensor Networks," *Wireless Sensor Systems, IET*, vol. 2, no. 3, pp. 161–169, 2012.