Lightweight Security Framework for IoTs using Identity based Cryptography

Sriram Sankaran
Center for Cybersecurity Systems and Networks
Amrita University
Amritapuri, Kollam-690525
Email: srirams@am.amrita.edu

***Abstract*—Internet of Things (IoTs) is gaining increasing significance due to real-time communication and decision making capabilities of sensors integrated into everyday objects. Securing IoTs is one of the foremost concerns due to the ubiquitous nature of the sensors coupled with the increasing sensitivity of user data. Further, power-constrained nature of the IoTs emphasizes the need for lightweight security that can tailor to the stringent resource requirements of the sensors. In this work, we propose a lighweight security framework for IoTs using Identity based Cryptography. In particular, we develop a hierarchical security architecture for IoTs and further develop protocols for secure communication in IoTs using identity based cryptography. Our proposed mechanism has been evaluated using simulations conducted using Contiki and RELIC. Evaluation shows that our proposed mechanism is lightweight incurring lesser overhead and thus can be applied in IoTs.**

I. Introduction

Advances in sensing, computing and communication have changed the Internet for people to Internet of things. IoTs are composed of sensors and actuators embedded into everyday objects that are capable of real-time communication and decision making. In addition, remote monitoring enables IoTs to be deployed in a multitude of application domains such as Smart home, Industrial Automation, Smart Healthcare, Automotive and transportation. The application-driven nature of IoTs leads to numerous challenges which need to be addressed before IoTs are commercially deployed and widely accepted.

Security is of paramount concern in IoTs due to its ubiquitous nature coupled with the increasing sensitivity of user data. Typically, these sensors are deployed in hostile locations which makes them vulnerable to notorious attacks such as node compromise and false data injection. Further, lightweight mechanisms for security that can tailor to the stringent resource requirements of sensors are necessary due to the power-constrained nature of the IoTs. In addition, energy-security-performance trade-offs need to be analyzed which vary for different applications.

IoTs are characterized by numerous interaction patterns such as periodic and on-demand data transmission which co-exist in different applications. On one hand, sensors can periodically report readings at regular intervals to the gateway node. On the other hand, gateways can query sensors in an on-demand manner and obtain the data. Thus, in order for sensors and gateway nodes to securely communicate with each other, mutual authentication is necessary. Further, mechanisms for securely revoking the sensors in case of compromise or failure need to be devised.

Identity based Cryptography (IBC) has been emerging as a promising public key based cryptographic primitive due to the ability to use identities as public keys. Security mechanisms based on IBC have been shown to incur lesser overhead than traditional public key based cryptography due to reduced key size. In addition, the process of bootstrapping in traditional public key cryptography which involves the distribution of keys for communication can be avoided in IBC since identities are used as public keys.

In this work, we develop a lightweight security framework for IoTs using Identity based Cryptography. In summary, our contributions include:

- Proposing a hierarchical security architecture for IoTs
- Developing protocols for secure communication in IoTs such as intra-domain and inter-domain communication, mutual authentication and revocation and evaluating them using simulations. Evaluation of the proposed mechanism demonstrates lesser overhead and thus can be applied in IoTs.

II. Related Work

The problem of lightweight security has received increased attention from the research community due to resource-constrained nature of the sensors. Mechanisms for providing lightweight security can be classified into symmetric key based cryptography, public key cryptography and hybrid key cryptography. Malan *et al.* [1] demonstrated the feasibility of public key cryptographic mechanisms based on elliptic curve cryptography on resource constrained sensors. This facilitated the development of numerous public key based cryptographic mechanisms for sensors since they provide non-repudiation compared to symmetric key based mechanisms.

Identity based Cryptography has been extensively applied in numerous domains such as Health care [2] [3], delay tolerant networks [4] [5], P2P networks [6], cloud computing [7] and in IoTs [8]. Hengartner *et al.* [9] developed a mechanism for access control based on identity based cryptography. Oliveira *et al.* [10] developed a pairing based cryptographic library for resource-constrained sensor nodes. These mechanisms consist of an initial bootstrapping phase where identities

978-1-5090-2029-4/16/$31.00 @2016 IEEE

are exchanged among communicating entities and the secure communication phase where messages are encrypted using public keys generated using identities.

Although identity based cryptography has been applied in specific domains, these mechanisms may not be readily applicable in IoTs since IoTs involve communication of sensors across domains. Thus these domains need to interoperate with each other so as to enable the operation of key management mechanisms. Further, many of the currently proposed mechanisms assume flat topologies which cannot meet the scalability requirements of IoTs thus making it vulnerable to single point of failure. In contrast to the existing mechanisms, we envision hierarchical topologies for IoTs that can adapt to deployment at a massive scale. Further, we develop protocols for intra-domain and inter-domain communication, mutual authentication and revocation that are necessary for secure communication in IoTs. Our proposed approach is lightweight incurring lesser overhead and thus can be applied in IoTs.

III. Background

A. Internet of Things

The functionality of IoT is illustrated in Figure 1. IoTs [11] are typically organized into three tiers. Tier I contains multitude of embedded devices monitoring objects and their surrounding areas. Tier II represents gateway nodes which receive data from the embedded devices. These gateway nodes are also referred as Edge nodes and are computationally more powerful than the embedded sensors. Tier III contains servers or datacenters which store data received from gateway nodes for processing. Servers or data centers perform complex analytics by developing models of the application behavior using the data received from the gateway nodes.

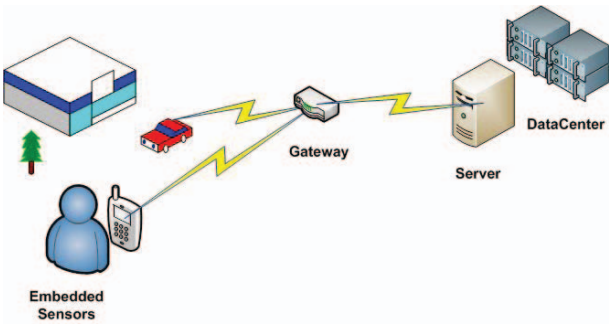

Fig. 1: Internet of Things

Embedded Sensors (Tier I): Typically, sensors in an IoT communicate with each other and the gateway node using the Zigbee protocol which is part of the IEEE 802.15.4 standard commonly referred as Personal Area Networks (PANs). In a Zigbee protocol, sensors are organized in star, ring or mesh based topologies and have considerably lesser transmission power compared to those in Wireless LANs.

Gateway Node (Tier II): Gateway nodes act as an interface between embedded sensors and back-end servers. Typically these nodes are equipped with WLAN (802.11) and WPAN enabled interfaces which have the capability to interact with embedded sensors and back-end servers. In addition, they are computationally powerful and are capable of transmitting data over larger distances.

Back-end Servers (Tier III): Back-end servers are composed of a wide variety of heterogeneous components such as routers and data centers communicating through high-bandwidth wireless links. These servers are capable of housing and processing significant chunks of real-time data from IoT applications. Typically, big data computing infrastructures such as Hadoop and MapReduce are utilized to perform analytics on IoT applications.

IoTs are characterized by numerous interaction patterns. These patterns vary depending on the needs of the applications. However we have considered two of the most generic patterns that are applicable across IoT domains. They are termed as Periodic Monitoring and Request-Response based patterns. Below, we provide a description of each of the patterns.

Periodic Monitoring: In a periodic monitoring scenario, sensors typically report observations to gateway nodes such as smartphones. These scenarios are highly prevalent in health care IoTs where patient data is routinely monitored by hospitals or care providers. In this scenario, performance is dependent on the packet size, sampling rate of the application, security mechanism and data rates of the communication channel.

Request-Response: In contrast to a periodic monitoring scenario, request-response based scenarios typically involve gateway nodes requesting data from a specific set of sensors in an on-demand manner. For instance, in smart home based IoTs, users can query a particular room to determine if any of the lights are turned on. In this scenario, performance is dependent on the request rates of the application, security mechanism and number of sensors that are actively in use.

B. Identity-Based Cryptography

Identity based Cryptography is an emerging public key based cryptographic technique which can be developed using elliptic curves and pairings. The term Identity based Cryptography was coined by Adi Shamir in 1984 when he envisioned the need for using identities as public keys. It was not until 2001 when Dan Boneh devised the fully practical Identity based Encryption from the Weil Pairing [12]. Identity based Encryption incurs lesser overhead than traditional public key cryptography due to lesser key size while providing better security. In addition, the problem of bootstrapping in public key cryptography can be avoided since public keys can be generated using identities.

Many variations have been developed since the realization of Identity based Encryption using Weil Pairing. These mechanisms can be sub-divided into identity based encryption and identity based signatures. Identity based Encryption involves sender to encrypt the message using the identity of the receiver. The receiver authenticates to the Private Key Generator (PKG) using its identity and obtains the private key to decrypt the message. In addition, hierarchical versions of identity based

cryptography [13] [14] capable of scaling to nodes deployed at multiple levels have been developed. Many challenges exist in Identity based Encryption such as PKG being a single point of failure and designing mechanisms for revocation.

IV. Security Architecture

Figure 2 shows the pictorial representation of a hierarchical architecture for IoTs. We envision IoTs to be organized in a hierarchical manner where a central node called the Root manages n Gateway nodes which in turn communicate with their group of sensors. We claim that the hierarchical architecture is feasible and scalable within the context of IoTs due to their massive deployment foreseen in the next decade.

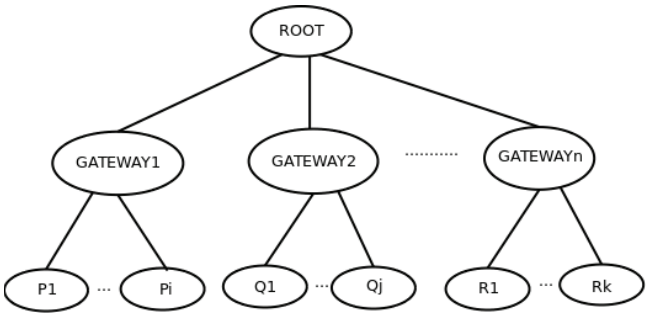

Fig. 2: Security Architecture for IoT

We define the identities of nodes starting from root node to the sensor nodes also known as the leaf nodes. Since IoTs are organized in a hierarchical manner, identity of a sensor node contains the identity of corresponding parent nodes until the root of the tree. In particular, we denote the identity of root node, gateway node and sensor nodes in the following manner.

$$ID_{Root} = ID_R$$

$$ID_{Gateway} = ID_G||ID_R$$

$$ID_{Sensor} = ID_S||ID_G||ID_R$$

where ID_{Root}, $ID_{Gateway}$ and ID_{Sensor} refer to identity of root, gateway and sensor nodes in the IoT.

Root node set-up:

Root node performs the following:

- Generate the groups G1, G2 of prime order q and an admissible pairing e such that

$$e : G_1 X G_1 -> G2$$

- Chooses cryptographic hash functions H_1, H_2 which are used to generate public keys corresponding to identities. These hash functions are further mapped to groups G_1 and G_2 respectively
- Selects a random secret $s \in Z_q^*$ such that

$$P_R = H_1(ID_R)$$

$$Q_R = sP_R$$

where s refers to the master key of the root node. System parameters are $< G_1, G_2, e, H_1, H_2, P_R, Q_R >$

Gateway node set-up:

Since there are n gateway nodes, we focus on the system parameters for $Gateway_1$. Similar procedure should be followed for the other gateway nodes. The root node performs the following:

- Computes the public key for the gateway node

$$P_G = H_1(ID_{Gateway})$$

where

$$ID_{Gateway} = ID_G||ID_R$$

- Generates the secret key for the gateway node

$$S_G = sP_G$$

- Picks the secret element $\rho_G \in$ for gateway node. ρ_G is only known to the gateway and root node.
- Computes the public parameter Q in the following manner

$$Q_G = \rho_G P_R$$

The public key and Q_G are made public while private keys are securely retained.

Sensor set-up:

We focus on the system parameters for a group of sensors managed by a single gateway node. Let P_1 be a sensor node managed by gateway $Gateway_1$. For each node in the group of sensors, the gateway node performs the following

- Computes the public key for the sensor node

$$P_S = H_1(ID_{Sensor})$$

where

$$ID_{Sensor} = ID_S||ID_G||ID_R$$

- Generates the secret key for the sensor node

$$S_S = S_G + \rho_G P_S$$

- Pick the secret point ρ_S for sensor node S. ρ_S is known only by sensor and gateway node.
- Computes the public parameter Q in the following manner

$$Q_S = \rho_S P_R$$

The secret keys are securely retained while public key and the parameter Q_S are made public.

V. Security Protocol

We develop security protocols for IoTs which leverage the hierarchical architecture composed of root node, Gateways and the corresponding sensors. Since IoTs are composed of numerous domains organized at multiple levels, we focus our attention on securing communications between sensors inside and across domains which we denote as intra-domain and inter-domain respectively. Further, mutual authentication between gateway nodes and sensors are necessary to ensure that senders are communicating with the intended recipients and vice versa. Finally, nodes and their corresponding identities need to be revoked in case of failure/malfunctioning. Below, we provide a description of secure protocols for intra-domain and inter-domain communication, mutual authentication and revocation in IoTs.

Intra-domain Communication:

In intra-domain communication, we assume sensors to know the identities of each other. In such a scenario, we adapt the Sakai-Ohgishi-Kasahara (SOK) scheme for non-interactive key agreement [15]. SOK scheme computes a shared secret on the fly using the identity of the other entity in the following manner.

Let us assume that two nodes A and B in the same domain and their corresponding private keys are $S_A = sP_A$ and $S_B = sP_B$ respectively. By bilinearity, we have the following

$$e(S_A, P_B) = e(S_B, P_A)$$

where e denotes the pairing.

Thus, using the secret key S_A and public key of B, $P_B = H(ID_B)$, node A can compute the shared secret key $k_{A,B}$. Similarly, node B can compute the shared secret using its secret key S_A and the public key of A $P_A = H(ID_A)$.

The main advantage of SOK scheme is that it avoids the need for bootstrapping thus resulting in reduced overhead. Depending on the number of entities willing to communicate, more advanced mechanisms such as Joux's tripartite key agreement [16] protocol which facilitates three communicating entities to compute a shared secret can be applied towards intra-domain communication.

Inter-domain Communication:

We propose an identity based encryption based mechanism to facilitate secure inter-domain communication in IoTs. Our mechanism serves as a foundation for the mutual authentication scheme that we describe next. Let us assume that gateway node G_1, queries the sensor node P_1 for data. Given that the identity of gateway node and sensor node are $ID_G||ID_R$ and $ID_S||ID_G||ID_R$ respectively, gateway node encrypts the message in the following manner.

- Computes
$$P_G = H_1(ID_G||ID_R)$$
and
$$P_S = H_1(ID_S||ID_G||ID_R)$$
- Chooses a random r $\in Z_q^*$
- Outputs the ciphertext
$$C =< rP_R, rP_S, H_2(g^r) \oplus m >$$

After receiving the ciphertext, $C =< C_0, C_1, V >$, Sensor can decrypt C using its secret key as follows:

$$S_S = sP_G + \rho_G P_S$$

- Computes
$$d = (e(C_0, S_S))/(e(Q_S, C_1))$$
- Outputs the message
$$m = H_2(d) \oplus V$$

Mutual Authentication:

Our mutual authentication scheme is a hybrid key management mechanism that uses identity based encryption to set-up pairwise symmetric keys between sensors and gateway nodes. It operates in Bootstrapping, Operational and Post-operational phases. Figure 3 contains the details for the proposed mutual authentication scheme.

Bootstrapping phase:

We assume that each gateway is pre-distributed with the private key K_g and public key $K_{gateway}$ in addition to a function that takes the ID of the sensor and outputs its corresponding public key. The public key of the gateway, $K_{gateway}$ is programmed in the memory of the sensors.

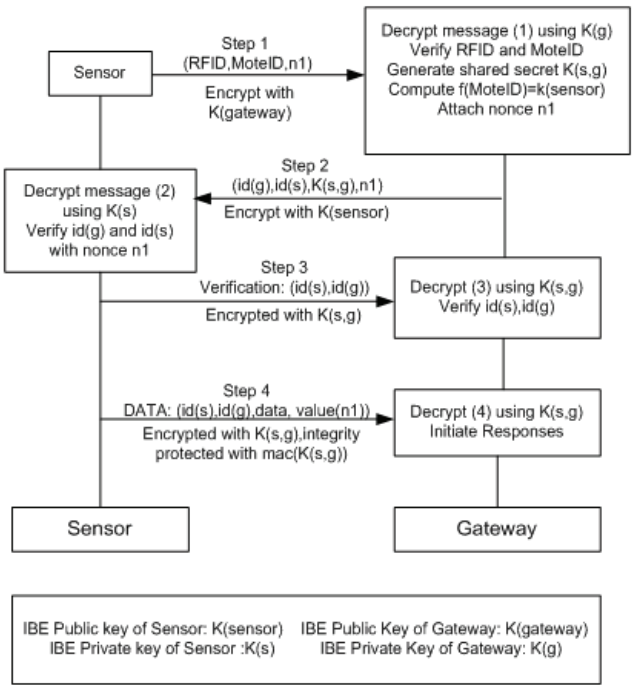

Fig. 3: Mutual authentication for IoT

Operational Phase: Step 1: REQUEST: Initially, when the sensor attached to an object is powered on, the mote obtains the identification information PID from the RFID tag attached

to the object queried by RFID reader. Once the sensor's information PID is obtained, the mote collects PID, its id MoteID, generates a nonce n1, encrypts message using public key of the gateway $K_{gateway}$ and sends it securely to the gateway. The main objective of using $K_{gateway}$ is to encrypt the sensor identification information making it impossible for adversaries to spoof PID. Nonce n1 is included in the message to prevent replay attacks.

Step 2: REPLY: The gateway decrypts the received message using its private key K_g and verifies the authenticity of the sensor using PID and MoteID. Then it uses received MoteID to derive public key K_s for the corresponding sensor, generates pair-wise secret keys, encrypts message using K_{sensor} and sends it securely to the sensor. This message contains the ids of both sensor and gateway, id_s and id_g, pair-wise secret key $K_{s,g}$ along with the nonce that the sensor sent.

Step 3: VERIFY: The critical part of our scheme is the confirmation from gateway that sensor has received the correct pair-wise keys before initiation of data transmission takes place. After decrypting the message using K_s and obtaining the pair-wise secret keys, sensor sends a message containing its ID and its corresponding gateway's ID encrypted using the pair-wise secret key $K_{s,g}$ which is decrypted by gateway and verified.

Step 4: DATA: After the verification, data transmission takes place by encrypting data using pair-wise secret key $K_{s,g}$ along with the identities of sensor and gateway node and a new value for nonce computed using existing nonce.

A MAC $mac_{K(s,g)}$ derived from pair-wise secret key $K_{s,g}$ is used to protect the message from unauthorized message tampering by adversaries. The doctor decrypts it using the pair-wise secret key $K_{s,g}$, verifies the integrity of the received message and accordingly initiates responses.

Post-operational Phase: The pair-wise secret keys are used as session keys for future communications. To update the pair-wise secret keys, the sensor and gateway exchange new values of nonces and gateway computes a new pair-wise key for communicating with the sensors.

Revocation:

Revocation is one of the primary challenges in identity based cryptography. For instance, if identities such as user's names are used for generating public keys, revoking public keys would involve users to change their identities which is difficult to implement in practice. To overcome this, we propose to utilize pseudonyms and further concatenate pseudonyms with date and time to identify secure transactions. Thus, when a node needs to be revoked in case of failure/compromise, its pseudonyms along with the other attributes can be removed. Further, if the same node rejoins the network, pseudonyms can be generated again and passed onto the neighboring nodes for secure communication.

VI. Security Analysis

In this section, we analyze our proposed scheme for resistance against different kinds of attacks that can be mounted in IoTs. Due to the privacy critical nature of the data, we focus our attention on identity and data tampering attacks. Impersonation attacks are not possible since only legitimate nodes have access to the public key of the gateway nodes which is generated using identities. Even without impersonation attacks, it is possible that rogue gateways replay old data that may be appropriate for the sensors. In that case, we require gateways to attach the nonces sent by sensors to prevent replay attacks. Sensors typically buffer their nonces to compare with those received from gateways to check for consistency. If any kind of inconsistency is observed, the received packet is discarded.

Our scheme preserves the data integrity required for IoTs apart from confidentiality by computing a MAC on the pair-wise symmetric key to provide increased level of security. Thus data tampering or false data injection attacks can be detected. In addition, our scheme preserves forward secrecy by requiring the generation of session keys for subsequent communications through exchanging new values of nonces and old keys are erased from memory to prevent key compromise. Thus if a malicious node obtains the shared keys by compromising the devices of the communicating entities, it cannot recover the session keys.

VII. Evaluation

We implement our proposed identity based cryptographic scheme using Contiki [17] and RELIC [18]. Contiki is a networked embedded operating system written in C language which can simulate embedded devices developed for various platforms such as MicaZ, TelosB, AVR, Z1 etc. RELIC is a cryptographic tool-kit which contains support for primitives such as symmetric key cryptography, elliptic curve cryptography and pairing based cryptography. We built RELIC toolkit for Contiki so as to be able to profile cryptographic primitives and understand their impact on execution time and energy consumption. We chose AES block cipher for symmetric encryption and HMAC-MD5 for integrity protection.

We wrote the source code in C and invoked RELIC cryptographic primitives from Contiki towards implementing our proposed intra-domain, inter-domain and mutual authentication protocols for secure communication in IoTs. We profiled the source code towards computing the execution time of time consuming operations such as encryption and integrity protection. All simulations were ran for 100 iterations and unique values for execution times were selected from the resulting set of values. These unique values were then averaged to find the mean execution time of cryptographic protocols. Table I lists the CPU execution time of cryptographic protocols used in proposed scheme. It is evident from the table that the integrity protection is compute intensive compared to the other operations since it requires to encrypt and compute a message authentication code (MAC) on the key.

In addition, we estimate energy consumption E of cryptographic operations in our proposed scheme using the following equation.

$$E = P * T_{execute}$$

TABLE I: CPU Execution Time and Energy Consumption

Cryptographic Protocol	Execution Time	Energy Consumption
Identity Based Encryption	179.6 ms	4.81 mJ
Identity Based Decryption	238.4 ms	6.38 mJ
AES Encryption	111 ms	2.97 mJ
AES Decryption	91.9 ms	2.46 mJ
HMAC-MD5	4.547 s	121.8 mJ

TABLE II: Packet Execution Time and Energy Consumption

Packet type	CPU Execution Time	CPU Energy Consumption
REQUEST	0.418s	11.19mJ
REPLY	0.418s	11.19mJ
VERIFY	0.2029s	5.43mJ
VERIFYACK	0.2029s	5.43mJ
DATA	9.2969s	249.03mJ

where P and $T_{execute}$ refer to power consumption and execution time respectively.

Power Consumption P was estimated using the following equation.

$$P = V * I$$

where V and I refer to voltage and current respectively.

We obtained the values for voltage and current from the MicaZ energy model [19] which are $3V$ and $8.93mA$ respectively. Thus energy consumption of identity based encryption E_{IBE} can be computed as follows.

$$E_{IBE} = 3 * 8.93 * 10^{-3} * 179.6 * 10^{-3} = 4.81mJ$$

Similarly, energy consumption of other cryptographic protocols was computed. Table I lists the energy consumption of cryptographic protocols used in our proposed scheme. Since energy consumption is directly proportional to execution time, HMAC-MD5 is the most energy consuming protocol in our proposed scheme.

In addition to estimating execution time and energy consumption of cryptographic algorithms, we estimated the execution time and energy consumption by packet type in the mutual authentication process. Table II contains the results for execution time and energy consumption by packet type. It is evident from Table 3 that "DATA" packet transmission and reception incurs higher execution time and energy consumption compared to the other packets since it involves compute-intensive operations such as encryption and MAC integrity protection.

The proposed approach is lightweight since it uses identity based cryptography to create pairwise-symmetric keys for secure communication between sensors and gateways in the IoTs. This incurs lesser overhead than traditional public key based cryptography since identities are used to generate public keys.

VIII. Conclusion

In this work, we developed a lightweight framework for securing IoTs using Identity based Cryptography. In particular, we proposed a hierarchical security architecture for IoTs and further developed protocols for secure communication. Our protocols for secure communication focused on aspects such as intra-domain communication, inter-domain communication, mutual authentication and revocation. In addition, we analyzed the security of the proposed protocol and evaluated them through simulations conducted using Contiki and RELIC. Our proposed protocol is scalable and that it incurs lesser overhead than traditional public key based cryptography thus making it applicable for IoTs.

References

[1] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, Oct 2004, pp. 71–80.

[2] K. Malasri and L. Wang, "Addressing security in medical sensor networks," in *Proceedings of the 1st ACM SIGMOBILE International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*, ser. HealthNet '07. New York, NY, USA: ACM, 2007, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/1248054.1248058

[3] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Body sensor network security: An identity-based cryptography approach," in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec '08. New York, NY, USA: ACM, 2008, pp. 148–153. [Online]. Available: http://doi.acm.org/10.1145/1352533.1352557

[4] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo, "Applicability of identity-based cryptography for disruption-tolerant networking," in *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking*, ser. MobiOpp '07. New York, NY, USA: ACM, 2007, pp. 52–56. [Online]. Available: http://doi.acm.org/10.1145/1247694.1247705

[5] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *Proceedings of the First International Conference on Secure Network Protocols*, ser. NPSEC'05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 31–36.

[6] S. Ryu, K. Butler, P. Traynor, and P. McDaniel, "Leveraging identity-based cryptography for node id assignment in structured p2p systems," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, vol. 1, May 2007, pp. 519–524.

[7] B. Y. Hongwei Li, Yuanshun Dai, "Identity-based cryptography for cloud security," Cryptology ePrint Archive, Report 2011/169, 2011, http://eprint.iacr.org/2011/169.

[8] T. Markmann, T. C. Schmidt, and M. Wählisch, "Federated end-to-end authentication for the constrained internet of things using ibc and ecc," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. ACM, 2015, pp. 603–604.

[9] U. Hengartner and P. Steenkiste, "Exploiting hierarchical identity-based encryption for access control to pervasive computing information," in *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, Sept 2005, pp. 384–396.

[10] L. B. Oliveira, M. Scott, J. Lopez, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," in *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, June 2008, pp. 173–180.

[11] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[12] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '01. London, UK, UK: Springer-Verlag, 2001, pp. 213–229.

[13] C. Gentry and A. Silverberg, "Hierarchical id-based cryptography," Cryptology ePrint Archive, Report 2002/056, 2002, http://eprint.iacr.org/2002/056.

[14] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, ser. EUROCRYPT '02. London, UK, UK: Springer-Verlag, 2002, pp. 466–481.

[15] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing, scis 2000-c20, jan. 2000," *Okinawa, Japan*.

[16] A. Joux, "A one round protocol for tripartite diffie-hellman," in *Proceedings of the 4th International Symposium on Algorithmic Number Theory*, ser. ANTS-IV. London, UK, UK: Springer-Verlag, 2000, pp. 385–394.

[17] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 455–462.

[18] D. F. Aranha and C. P. L. Gouvêa, "RELIC is an Efficient LIbrary for Cryptography," https://github.com/relic-toolkit/relic.

[19] (2008) Micaz datasheet. [Online]. Available: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf