A Systematic Key Management Mechanism for Practical Body Sensor Networks

Xinyu Yang*, Cong Zhao*, Shusen Yang†, Xinwen Fu‡ and Julie McCann†
*Xi'an Jiaotong University, Emails: yxyphd@mail.xjtu.edu.cn, zhaocong@stu.xjtu.edu.cn
†Imperial College London, Email: s.yang09@imperial.ac.uk, j.mccann@imperial.ac.uk
‡University of Massachusetts Lowell, Email: xinwenfu@cs.uml.edu

***Abstract*—Security plays a vital role in promoting the practicality of Wireless Body Sensor Networks (BSNs), which provides a promising solution to precise human physiological status monitoring. A fundamental security issue in BSN is key management, including establishment and maintenance of the key system. However, current BSN key management solutions are either designed for specific phases of a BSN's life-time or restricted to strong assumptions such as homogeneous BSN composition, pre-deployed key materials, and existing secure path, which limits their applications in real-world BSNs. In this paper, we develop the Systematic Key Management (SKM) for practical BSNs, where basic human interactions are conducted for non-predeployed secure BSN initialization, and authenticated key agreement is achieved using lightweight non-pairing certificateless public key cryptography. We construct a BSN prototype consisting of self-designed motes and Android phones to evaluate the real-world performance of SKM. Through extensive simulations and test-bed experiments, we demonstrate that our lightweight SKM scheme manages to provide high security guarantee while outperforming state-of-the-art approaches in terms of both computation and storage efficiency.**

I. Introduction

In the last decade, Wireless Body Sensor Networks (BSNs) draw considerable attentions as a viable solution to human physiological status monitoring [1]. Compared with general Wireless Sensor Networks (WSNs), human physiological data generated by BSNs have more rigorous security and privacy preserving requirements [2]. For instance, the broadcasting nature of wireless communication leads to the vulnerability of BSNs: attackers can breach personal privacy of BSN users by eavesdropping the communication. In addition, false data may be injected to incur detrimental physiological status judgement and may lead to a fatal consequence. Therefore, a practical BSN system must be carefully secured.

For the security and privacy concerns, the wireless communication of a BSN should be encrypted. A key management scheme is often used for the establishment and maintenance of keys in a secure BSN. Existing key management schemes are often designed for specific phases of a BSN's life-time [3]–[10]. These schemes often take unrealistic assumptions, including homogeneous BSN composition [6], [7], pre-deployed key materials [8], [9], and the existence of secure paths [5], [10], to fight against potential threats. However, for practical BSNs, multiple types of keys should be subtly organized to form an interactive system as the foundation of upper level security schemes. Meanwhile, the tradeoff among system security, usability, and resource occupation should be carefully managed. A secure, thorough, and efficient key management mechanism is critical for the practicality of BSN.

In this paper, we design the Systematic Key Management (SKM) scheme to manage an interactive key system for BSNs. Specifically, SKM performs human-interactive non-predeployed network initialization, elliptic curve based non-paring certificateless authenticated key agreement for both wide-area and local BSNs, and key system maintenance during the entire life-time of BSNs. SKM can prevent major security threats in BSNs, including impersonating attack, combinatorial attack, public key replacement attack, and collusion attack. Meanwhile, it outperforms current BSN key management approaches in terms of computation and storage costs.

II. Systematic Key Management

The typical architecture of BSNs is shown in Fig.1. Multiple wearable and implantable wireless sensor nodes are associated by a personal controller to continuously monitor user's physiological and environmental status. All controllers regularly transmit sensed data to the medical data server for profiling and querying by BSN accessors. In this paper, we treat the subsystem of the controller and sensor nodes as a local BSN, and that of the data center, controllers and BSN accessors as the wide-area BSN.

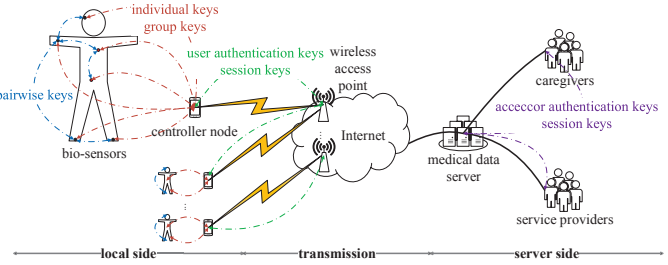


Fig. 1: BSN Key System Architecture

To secure data transmissions among BSN entities, multiple types of keys are implemented. They are mutually related, forming a key system shown in Fig.1. BSN personal controllers and accessors register at the data center for authentication keys. Then, the controller and sensor nodes perform group, pairwise and individual key agreements based on their authentication credentials. Communicating entities use existing

secure path to establish temporary session keys. We design SKM to secure the establishment and maintenance of such a key system.

In this paper, we assume that all sensor nodes are able to correctly measure human physiological and environmental statues, and the BSN operator can be fully trusted. Attackers are not able to obtain physiological data without physical contact with the user, or to physically capture sensor nodes without being noticed.

A. Preliminary

SKM consists of three components: BSN user and accessor registration (Algorithm 1), local BSN network association and authenticated key agreement (Algorithms 2-6), and BSN key system maintenance (Algorithm 7). In SKM, the security of authenticated key agreement is based on commitment schemes [11] and the Computational Diffie-Hellman Problem (CDHP) [12].

1) Commitment Schemes: The commitment scheme allows one to commit to a chosen value, hidden to others, and reveal it later. Generally, it consists of two steps:

- $Commit(m, x) \rightarrow (c, r)$;
- $Reveal(m, c, r) \rightarrow x \in \{0,1\}^n \cup \emptyset$,

where m is public data; x is n-bit private data to be committed; c is the committing value; and r is the revealing value.

Here, given (c, m), x cannot be calculated without r, which is called the hiding property. Meanwhile, given (m, c, r), x must be the only output of the revealing algorithm, which is called the binding property. They guarantee that private data cannot be changed after being committed.

In SKM, the non-malleable hash based commitment scheme in [4] is adopted.

2) Computational Diffie-Hellman Problem(CDHP): CDHP is treated as a basic intractable math problem in asymmetric key agreements [4], [5], [10]. CDHP on elliptic curves (ECDHP) is clarified as follows.

Let G_1 be a cyclic additive group on elliptic curve F_q, the generator is P with a prime order of q.

- *ECDHP*: given P, aP, bP, $a, b \in Z_q^*$, the computation of abP is intractable.

In SKM, we consider ECDHP as intractable.

Notations used in SKM is shown in Table.I.

B. BSN User and Accessor Registration

Before a local BSN is implemented, the controller has to register at the data center to get authenticated initial keys. It is reasonable to assume that the medical data center has already been maintaining pairs of identity and authentication code $(ID_{oi}, CODE_{oi})$ for all legal BSN operators. See Algorithm 1.

In SKM, the registration and session key establishment for data accessors are the same with that of the controller.

TABLE I: Notations

DC :	Data center.
CN_i :	The ith controller.
N_{pi} :	BSN sensor number of CN_i.
$Node_n$:	The nth node.
NID_n :	Node identity of $Node_n$.
x :	Master private key of CN_i.
P_{pub} :	Master public key of CN_i.
d_n :	CN side partial private key of $Node_n$.
T_n :	CN side partial public key of $Node_n$.
x_n :	$Node$ side partial private key of $Node_n$.
P_n :	$Node$ side partial public key of $Node_n$.
K_{indi}^n :	Individual key of $Node_n$.
$K_{pair}^{nn'}$:	Pairwise key between $Node_n$ and $Node_{n'}$.
G_i :	Identity of $Group_i$.
K_{G_i} :	Group key of G_i.
H :	One way hash function.
H_r :	Universal hash function with key r.
$SymEnc$:	Symmetric encryption.
$SymDec$:	Symmetric decryption.
$< F_q, E/E_q, G_q, P >$:	Elliptical curve E on finite field F_q.

Algorithm 1 Registration of CN_i at the Data Center

Variables:
ID_{oi}: Identity of $Operator_i$;
$CODE_{oi}$: Secret code of $Operator_i$;
ID_{pi}: Identity of $User_i$;
$NONCE$: Message freshness code;
K_{pi}: Session key between the CN_i and the DC.

1: $Operator_i$ initializes registration of CN_i.
2: CN_i notifies $DC :< ID_{oi}, ID_{pi}, N_{pi} >$.
3: DC checks for $CODE_{oi}$ based on ID_{oi}.
4: DC notifies CN_i :
$M_i = SymEnc(CODE_{oi}, (K_{pi}, SymEnc(K_{pi}, NONCE_{pi})))$.
5: $Operator_i$ enters $CODE_{oi}$ for authentication.
6: CN_i decrypts M_i by $SymDec(CODE_{oi}, M_i)$.
7: CN_i notifies $DC :< ID_{oi}, SymEnc(K_{pi}, NONCE_{pi+1}) >$.
8: DC checks for $NONCE_{pi+1}$ based on K_{pi} :
9: **if** $Match(NONCE) = False$ **then**
Registration fails;
10: **else**
DC stores $< ID_{pi}, N_{pi}, K_{pi}, ID_{oi} >$ as CN_i's index;
DC notifies CN_i : User registration succeed.
11: **end if**

C. Local BSN Network Association and Authenticated Key Agreement

After the registration of the controller, the BSN operator is able to setup local BSNs. Local BSN network association and authenticated key agreement consists of 4 main steps: initialization, node identification, node authentication, and authenticated key agreement.

In the initialization, the operator chooses sensor nodes based on the scale of local BSN. BSN controller then determines and publishes system parameters. See Algorithm 2.

After system parameter publication, the controller notifies nodes in the local group to identify themselves. This process prepares credentials for physical comparison. See Algorithm 3.

After the node identification, nodes have to be authenticated by the BSN operator. See Algorithm 4.

Algorithm 2 Local BSN Initialization

1: $Operator_i$ picks N_{pi} nodes to form group G_i.
2: CN_i determines system parameters:
q : A k-bit prime;
$< F_q, E/E_q, G_q, P >$: Elliptic curve E on prime finite filed F_q;
$x \in Z_q^*$: Master private key;
$P_{pub} = xP$: Master public key;
$H : \{0,1\}^* \rightarrow \{0,1\}^k$: One-way hash function;
$H_r : \{0,1\}^* \rightarrow \{0,1\}^k$: Universal hash function with key r;
3: CN_i publishes $\Omega =< F_q, E/E_q, G_q, P, P_{pub}, H, H_r >$ as the system parameter.

Algorithm 3 Local BSN Node Identification

Variables:
r_n: Revealing value of $Node_n$;
R_n: Group of received r;
C_n: Committing value of $Node_n$;
$NInd_n$: Index of $node_n$;
$Index_n$: Group of received indexes.
Functions:
$Verify(M)$: Verify the validity of M on CN_i.

1: CN_i initializes: $Index_0, R_0 \leftarrow \emptyset$;
2: CN_i broadcasts: Committing begins.
3: $\forall Node_n \in G_i$:
Choose $x_n \in Z_q^*, r_n \in \{0,1\}^*$;
Compute $P_n = x_n P, C_n = H(NID_n | P_n | r_n)$;
Broadcast $NInd_n =< NID_n, P_n, C_n >$.
4: $CN_i, \forall Node_n \in G_i : Index_n \leftarrow NInd_n \cup \{\forall j \neq n, NInd_j\}$.
5: Till committing terminated:
6: **if** $Verify(|Index_0| = N_{pi}) = False$ **then**
abort;
7: **else**
CN_i broadcasts: Revealing begins.
8: **end if**
9: $\forall Node_n \in G_i$ broadcast: $< NID_n, r_n >$.
10: $CN_i, \forall Node_n \in G_i : R_n \leftarrow r_n \cup \{\forall j \neq n, r_j\}$.
11: Till revealing terminated:
12: **if** $Verify(|R_0| = N_{pi}) \cap (\forall j \neq 0, C_j = H(NID_j | P_j | r_j)) = False$ **then**
abort;
13: **else**
Node identification succeed.
14: **end if**

After the node authentication, an interactive key system is established by the controller and all authenticated nodes. See Algorithm 5,6.

D. BSN Key System Maintenance

In BSNs, for newly added network entities, authenticated keys should be agreed upon; for lately exited entities, related keys should be revoked. In SKM, the addition of BSN users can refer to user registration discussed in II-B. For user exits, getting user's exit application, the data center can simply revoke user's authenticated key pair and notifies the entire network. On the other hand, member changes in local BSNs need further discussion.

Algorithm 4 Local BSN Node Authentication

Variables:
SAS: Short authentication string.
Functions:
$trunc(M)$: Truncate the first 20-bits of M;
$PhyCMP(M, G)$: Physically comparison of M among G.

1: $CN_i, \forall Node_n \in G_i$ compute:
$SAS_n = trunc(H_{R_n}(Index_n))$.
2: CN_i broadcasts: Node authentication begins.
3: $\forall Node_n \in G_i$ perform: LED blinking based on SAS_n.
4: **if** $PhyCMP(SAS, G_i) = False$ **then**
abort;
5: **else**
Node authentication succeed.
6: **end if**

Node additions can be divided into single node additions and patch node additions. Patch node additions can be realized by treating new nodes as a group and performing the local BSN network association and authenticated key agreement protocol. Single node additions are basically similar to patch node scenarios except for the counting process. The controller has to announce existing group of the new node and perform key updates. See Algorithm 7.

Algorithm 5 Local BSN Authenticated Key Agreement

1: CN_i extracts partial keys for $\forall Node_n \in G_i$:
Choose $t_n \in Z_q^*$;
Compute $T_n = t_n P, d_n = t_n + xH(NID_n, T_n, P_n) mod q$.
2: CN_i generates the group key of G_i:
$$K_{G_i} = H(\sum_{j=1}^{N_{pi}} t_j mod q).$$
3: CN_i notifies $Node_n : M_n = SymEnc(H(xP_n), d_n | T_n | K_{G_i})$.
4: $\forall Node_n \in G_i$ decrypt M_n by $SymDec(H(x_n P_{pub}), M_n)$.
5: $\forall Node_n \in G_i$ store:
$K_{pn} = (d_n, x_n), K_{bn} = (P_n, T_n), K_{indi}^n = H(x_n P_{pub}), K_{G_i}$.

Algorithm 6 Local BSN Node Pairwise Key Agreement

Functions:
$f(x) = \sum_{i=1}^{N_{pi}} \overline{x \oplus ID_i} d_i P$: Pairwise keying material function.

1: CN_i broadcasts: $M_f = SymEnc(K_{G_i}, f(x))$.
2: $\forall Node_n \in G_i$ decrypt M_f by $SymDec(K_{G_i}, M_f)$.
3: $\forall Node_n, Node_{n'} \in G_i : K_{pair}^{nn'} = H(d_n f(ID_{n'}))$.

Algorithm 7 Single Node Addition into Existing Groups

1: $Node_{add}$ applies for single node addition to CN_i.
2: CN_i authenticates $Node_{add}$ using single SAS comparison.
3: CN_i performs authenticated key agreement with $Node_{add}$.
4: CN_i updates N_{pi}, K_{G_i} and $f(x)$.
5: CN_i broadcasts: $SymEnc(K_{G_i}, N_{pi}^1 | K_{G_i}^1 | f(x)^1)$.
6: $Node_{add}$ performs pairwise key agreement based on $f(x)^1$.

If a node's life-time is expired, or the node is compromised, it has to exit current BSN. The controller needs to broadcast node revocation notification, which clarifies that all keys

related to the exit node are invalid. Then the group key has to be updated and distributed using individual keys of remained nodes.

III. Analysis and Discussions

This section provides theoretical analysis of SKM in terms of security, usability and completeness.

A. Security

We discuss about commonly considered attacks in different phases of BSN key management to demonstrate the security of SKM.

1) BSN User and Accesscor Registration: Registration of the personal controller is based on human interactions. The operator's identity is authenticated using identity and code based Challenge-Response process, after which registration index and individual key are established. Such registration can only be done with an legal BSN operator, mitigating possible impersonating attacks.

2) Local BSN Network association and Authenticated Key Agreement: Identification and authentication of local BSN nodes are based on the commitment scheme. In node identification, commitments are broadcasted and counted before the final revealing. Both the number of node and commitment validity are verified. SKM prevents attackers to access group authentication credentials by combinatorial means in advance. Possible combinatorial attacks are mitigated.

Authenticated key agreement is realized based on Certificateless Public key Cryptography(CL-PKC) [5]. The controller extracts node's public and private key parts. Combining self-generated public and private key parts, nodes are able to establish controller-authenticated public and private key pairs, where the node ID is bound with the public key. Possible public key replacement attacks can be mitigated.

Pairwise keys between legal nodes are established locally based on authenticated keying materials distributed by the controller. The establishment of pairwise keys depends on controller-side private key parts of sensor nodes. Even if multiple nodes are compromised, no keys of legal nodes will be disclosed. Possible polynomial based collusion attacks are mitigated.

3) BSN Key System Maintenance: In SKM, because of the notice of node exit from the controller, keys related to the exit node will not jeopardize the key system. Then, SKM establishes a contributory group key whose composition is determined by all legal group members. It's sensitivity of member changes guarantees forward and backward secrecies of the key system.

B. Usability

In SKM, reasonable human interactions are used to perform wide-area and local BSNs initialization, key system establishment and maintenance.

For wide-area controller registration, system adaptability is only restricted by resource capacity of the data center. The network association of local BSNs does not depend on pre-deployed information. Thus network composition can be flexibly determined considering BSN user's personal conditions. Necessary cryptography operations are transparent to the operator, which provides a high usability.

C. Completeness

SKM is responsible for key system establishment and maintenance during the entire life-time of BSNs. Specifically, authentication key and individual key of BSN users and accessors, as well as individual key, pairwise key, authentication key, and group key of the BSN controller and sensor nodes are established, forming an organic key system.

IV. Evaluation

In this section, computation, storage and communication costs of SKM are discussed. Numerical experiments are conducted to evaluate the performance of SKM. We also built a testbed to verify the correctness and efficiency of SKM.

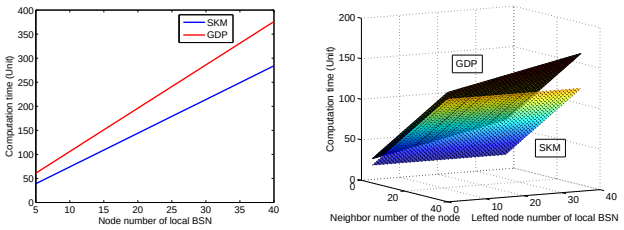
A. Numerical Evaluation

To the best of our knowledge, few of current approaches consider the interconnection among keys in BSN systems. Li *et al.* [4] designed a relatively systematic network initialization and key management scheme. We conduct performance comparison between SKM and GDP proposed by [4]. In numerical experiment, local node number is set to be between 5 and 40.

1) Computation Cost: SKM tends to maintain a reasonable computation cost while guaranteeing its security performance. The experiment compared computation cost of SKM with that of [4]. According to [5], the ratio of computation time, under the same hardware setting, among the exponential operation E on Z_q^*, the point multiplication operation M on G_q, and the $Hash$ operation H on G_q was set to be 4:2:1. Compared with asymmetric operations, computation cost of symmetric operations was negligible. This was treated as the basic computation cost unit in the experiment. Meanwhile, the UDB protocol in [4] was converted to its Elliptic Curve Cryptography(ECC) version in the comparison. The experimental result is shown in Fig.2.

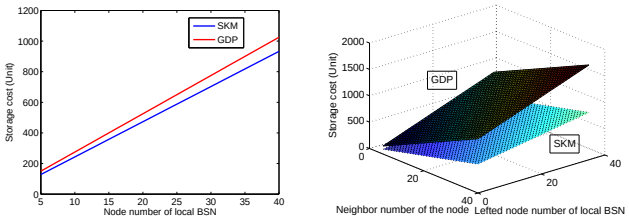
Fig.2a demonstrates the impact of local BSN network scale on computation time of the controller. Under experiment settings, SKM has shorter computation time, as $(2N_{pi}+1)M+(3N_{pi}+2)H$, than that of [4], as $7M+(N_{pi}+2)H+2N_{pi}E$. The reason is that SKM does not perform distributed contributory group key agreement as [4] does. Meanwhile, in SKM, the agreement of individual and pairwise keys only need to perform ECC point multiplication, instead of exponential operation in [4].

Fig.2b demonstrates the impact of local BSN network scale on computation time of the sensor node. For direct observation, the number of node neighbours and the number of left nodes other than node neighbours are chosen to be variables. For SKM, computation time of sensor nodes is only related to the number of node neighbours but the network scale, while that of [4] is in directive proportion to both of them. Under experiment settings, computation time of SKM,



(a) Controller Side (b) Node Side

Fig. 2: Computation Costs at the Controller and Node Sides under Different Network Scales



(a) Controller Side (b) Node Side

Fig. 3: Storage Costs at the Controller and Node Sides under Different Network Scales

as $(N_e + 2)M + (N_e + 3)H$, is less than that of [4], as $6M + (N_e + 1)H + (N_e + 1)E$. The reason is the same as that on the controller. ECC point multiplication in distributed contributory group key agreement and exponential operation in individual and pairwise key agreement enhance computation request at the node side in [4].

2) Storage Cost: Considering the restricted resource of BSNs, SKM manages to reduce storage cost of both the controller and sensor nodes while guaranteeing the security performance. The experiment compared storage cost of SKM with that of [4]. According to [4], for 80-bit key security, asymmetric key A had to be 160-bit. Meanwhile, we set symmetric key S to be 128-bit according to the AES algorithm [13]. For direct observation, storage cost ratio between symmetric keys and asymmetric keys was set to be 3:4. This was treated as the basic storage cost unit in the experiment. Meanwhile, UDB in [4] was converted to its ECC version in the comparison. The experimental result is shown in Fig.3.

Fig.3a demonstrates the impact of local BSN network scale on storage cost of the controller. Under experiment settings, storage cost of SKM, as $(5N_{pi}+3)S+(2N_{pi}+1)A$, is lower than that of [4], as $(3N_{pi}+3)S+(4N_{pi}+4)A$. SKM does not use UDB for group key generation, and huge amount of intermediate asymmetric keying materials are not stored. This reduces storage request of the controller.

Fig.3b demonstrates the impact of local BSN network scale on storage cost of the sensor node. For direct observation, the number of node neighbours and the number of left nodes other than node neighbours are chosen to be variables. The result shows that, for both SKM and [4], storage cost of sensor nodes is in directive proportion to both the number of node neighbours and the network scale. Under experiment settings, storage cost of SKM, as $(2N_{pi}+N_e+5)S+(N_{pi}+3)A$, is lower than that of [4], as $(2N_{pi}+N_e+4)S+(4N_{pi}+4)A$. In SKM, unlike that in [4], centralized contributory group key agreement is conducted only by the controller, and sensor nodes have no need to store group keying materials that are not necessary for successive key management. Besides, in SKM, unlike [4], keys and keying materials for individual and pairwise key agreement are also maintained by the controller. These reduce storage request of nodes significantly.

3) Communication Cost: For BSNs, communication cost of message interactions is critical for system performance. SKM has to minimize the message number and the message length. We compared communication cost of SKM with that of [4]. By analysing experiment settings, message length are basically the same in two schemes. For direct observation, messages were divided into three categories: Broadcasting Parameter Message (BPM), Broadcasting Text Message (BTM), and Unicasting Encryption Message (UEM). They were treated as the basic communication cost unit.

Communication costs of both SKM and [4] are basically the same. For the controller, SKM needs extra two BTM costs, which could be neglected for their lightweight in broadcasting. For the sensor node, communication costs are identical, which leads to no further discussion.

B. Testbed Experiment

In this section, we implemented SKM and evaluated its feasibility on a self-designed BSN testbed.

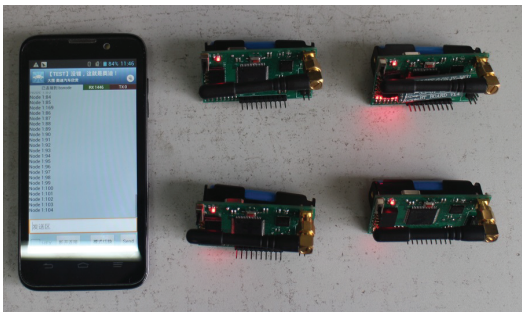
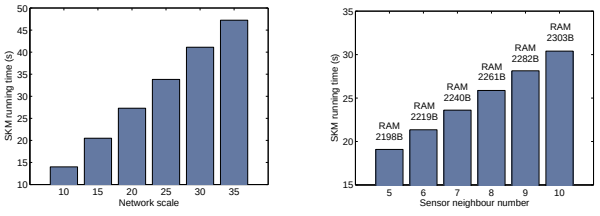


Fig. 4: A BSN Prototype for SKM Evaluation

1) Implementation: As we know, commercially available sensor nodes commonly used by BSN prototypes (like MICAz, TelosB, and Tmote-Sky nodes) had no specific module (like the Bluetooth module) to communicate with the smartphone-based controller. Besides, to the best of our knowledge, there was still no usable RF module that supported the latest IEEE 802.15.6 protocol [14]. In the experiment, we independently developed a sensor prototype with a HC-06 module for communications between the controller and sensor nodes based on Bluetooth 2.0 protocol. On the other hand, mutual communication among sensor nodes was realized based on



(a) Controller Side (b) Node Side

Fig. 5: Results of Testbed Experiments

IEEE 802.15.4/Zigbee protocol, implemented on a CC2420 RF module.

Our experimental testbed consisted of an Android smartphone as the local controller and self-designed motes as sensor nodes. The controller, MI 2S(Aries), possessed a 1741MHz QualComm SnapDragon 600 processor, 2GB of RAM, and 32GB of ROM. Each self-designed sensor node possessed an 8MHz ATmega128L microcontroller, 4KB of RAM, and 128KB of ROM. The network association test is shown in Fig.4.

For preliminary experiments, we implemented Algorithms 1-7 on our testbed. The ECC parameter was adopted from secp160r1 in [15], and the length of symmetric keys was set to be 128-bit according to [13].

The programming of the controller was under Android-4.4.4. Primitive cryptography operations were provided by Bouncy Castle Cryptography [16] and Oracle Java Cryptography APIs [17]. The programming of sensor nodes was under TinyOS-2.1.1. Primitive cryptography operations were provided by TinyECC-2.0 [18] with all optimization switches enabled. The running-time and storage costs of SKM were evaluated.

2) Results: Results of testbed experiments are shown in Fig.5.

Fig.5a demonstrates the relation between running time of SKM on the controller and local BSN network scale. ROM cost of the experimental SKM is about 3.61MB, and RAM cost is no more than 31MB. Storage cost of SKM is practical on the controller considering its 32GB ROM and 2GB RAM capacity. Running time of SKM on the controller is no more than 50s.

Fig.5b demonstrates the relation between running time of SKM on sensor nodes and neighbour number of the single node. ROM cost of the experimental SKM is about 23.6KB, and RAM cost is up to 2.24KB under experiment settings. Storage cost of SKM is practical on sensor nodes considering its 128KB ROM and 4KB RAM capacity. Running time of SKM on sensor nodes is no more than 30.4s.

It is feasible for SKM to accomplish local BSN association and settle the entire key system in less than two minutes.

V. Conclusion

In this paper, we design a lightweight key management scheme, SKM, to establish and maintain an interactive key system for practical BSNs. Based on reasonable human interactions, SKM manages to associate both wide-area and local BSNs with no predeployed information. Different from traditional schemes, SKM does not need to make any existing path assumption. Furthermore, by using ECC based non-paring CL-PKC, SKM manages to guarantee the lightweight authenticated key agreement. Both analytical and experimental evaluation indicate that SKM managing the key system in a secure and efficient way, which demonstrate the great potential of applying SKM in practical BSNs.

Acknowledgment

This work is supported in part by the National Science Foundation of China (NSFC) under Grant 61373115 and Grant 61402356. This work is also supported by the China Scholarship Council.

References

[1] H. Cao, V. Leung, C. Chow, and H. Chan, "Enabling technologies for wireless body area networks: A survey and outlook," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 84–93, 2009.

[2] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *ELSEVIER COMPUT NETW*, vol. 54, no. 15, pp. 2688 – 2710, 2010.

[3] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "Sage: a strong privacy-preserving scheme against global eavesdropping for ehealth systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 365–378, 2009.

[4] M. Li, S. Yu, J. D. Guttman, W. Lou, and K. Ren, "Secure ad hoc trust initialization and key management in wireless body area networks," *ACM TOSN*, vol. 9, no. 2, p. 18, 2013.

[5] J. Liu, Z. Zhang, X. Chen, and K. S. K. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332 – 342, 2014.

[6] H. Chunqiang, C. Xiuzhen, Z. Fan, W. Dengyuan, L. Xiaofeng, and C. Dechang, "Opfka: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *Proc. IEEE INFOCOM*, 2013, pp. 2274–2282.

[7] C. Hu, N. Zhang, H. Li, X. Cheng, and X. Liao, "Body area network security: A fuzzy attribute-based signcryption scheme," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 37–46, 2013.

[8] W. Drira, E. Renault, and D. Zeghlache, "A hybrid authentication and key establishment scheme for wban," in *Proc. IEEE TrustCom*, 2012, pp. 78–83.

[9] D. He, C. Chen, S. Chan, J. Bu, and P. Zhang, "Secure and lightweight network admission and transmission protocol for body sensor networks," *IEEE J. Biomed. Health Inform.*, vol. 17, no. 3, pp. 664–674, 2013.

[10] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "Ibe-lite: a lightweight identity-based cryptography for body sensor networks," *IEEE Trans. Inform. Technol. Biomed.*, vol. 13, no. 6, pp. 926–932, 2009.

[11] M. Cagalj, S. Capkun, and J.-P. Hubaux, "Key agreement in peer-to-peer wireless networks," *Proc. IEEE*, vol. 94, no. 2, pp. 467–478, 2006.

[12] A. Joux and K. Nguyen, "Separating decision diffie–hellman from computational diffie–hellman in cryptographic groups," *Springer J CRYPTOL*, vol. 16, no. 4, pp. 239–247, 2003.

[13] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer, 2002.

[14] I. S. Association *et al.*, "802.15. 6-2012 ieee standards for local and metropolitan area networks–part 15.6: Wireless body area networks."

[15] C. Research, *SEC 2: Recommended Elliptic Curve Domain Parameters*. Standards for Efficient Cryptography Version 1.0, 2000.

[16] B. Castle, "The legion of the bouncy castle java cryptography apis," http://www.bouncycastle.org/java.html.

[17] Oracle, "Cryptoprimitive(java plateform ed.7)," http://docs.oracle.com/javase/7/docs/api/.

[18] A. Liu and P. Ning, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. IEEE IPSN*, 2008, pp. 245–256.