An Unlinkable Key Update Scheme Based on Bloom Filters for Random Key Pre-distribution

Bin Wang
(Corresponding author: Bin Wang)

Information Engineering College, Yangzhou University
No. 196 West HuaYang Road, Yangzhou City, Jiangsu Province, P. R. China, 225127
(Email: jxbin76@yeah.net)

(Received Apr. 16, 2019; Revised and Accepted Dec. 10, 2019; First Online Jan. 29, 2020)

Abstract

Eschenauer *et al.* presented an efficient random key predistribution scheme for WSNs that assigns symmetric keys to sensor nodes by randomly sampling from a large key pool. Most research in this line assume nodes exchange key identifiers to determine common keys between them. However, an adversary can learn topology information of the underlying random key graph by intercepting exchanged key identifiers. In addition, when key exposure occurs, compromised nodes should be revoked and uncompromised nodes' key rings should be updated securely. In this paper, we design an unlinkable key update mechanism that can revoke compromised nodes while an adversary is infeasible to link key identifiers with a node. A key update node is responsible for distributing a random seed among uncompromised nodes in order to update their key rings securely. The revoked keys are represented by a bloom filter to avoid exchange of key identifiers when checking whether a node is compromised. As a bloom filter has zero false negative rate, we utilize negative answers returned by a bloom filter to identify uncompromised keys and nodes with high probability. Then a local broadcast mechanism is used to speed up update of uncompromised nodes' key ring securely.

Keywords: Bloom Filter; Key Pre-distribution; Unlinkablity

1 Introduction

Wireless sensor networks (WSNs) are networks consisting of battery-powered sensor nodes that are able to perform sensing tasks, data processing and multi-hop wireless communication. With the rapid development in sensor technologies and wireless communication, WSNs have been widely used in applications such as environment monitoring, target tracking, military operations and attracted a lot of attention from research communities [15]. As sensor nodes may be deployed in hostile environments, they must forward data packets to a base station (a sink node) in a secure manner to prevent an adversary from breaking data privacy or mounting a forgery attack [1, 13]. Hence security is an important issue to be addressed for wide deployment of WSNs. To provide security services (*e.g.*, data encryption or identity authentication) for WSNs, it is necessary to establish shared keys between nodes via appropriate key management mechanisms. However, as sensor nodes are resource-constrained equipments with limited storage and computational capability, traditional public key cryptographic schemes(*e.g.*, Diffie-Hellman key agreement protocol [5]) are not applicable for WSNs since computational cost of public key operations are too costly to be implemented for sensor nodes [4]. That is, energy efficiency is an important factor to be considered when handling security challenges for WSNs [8].

As senor nodes can only afford light-weight operations such as hash operations, symmetric encryption/decryption operations, Eschenauer and Gligor [6] suggested a random key pre-distribution scheme for WSNs in which each sensor node is equipped with a fixed-sized key ring comprising symmetric keys randomly sampled from a large key pool before network deployment. Afterwards, two nodes can compute a session key for secure communication if their key rings share at least a common key. Connectivity of the induced random key graph is proven to hold asymptotically under certain choices of system parameters [17]. Several improvements or extensions to this kind of random key predistribution schemes are suggested such as q-composite random key pre-distribution scheme [3], random pairing key pre-distribution scheme [16]. On the other hand, deterministic key pre-distribution schemes based on combinatorial designs [14] are also presented as alternatives to key pre-distribution schemes for WSNs. Deterministic key pre-distribution schemes have the advantage that secure connectivity property can be proven to hold in a deterministic way.

A subtle point inherent in random key pre-distribution schemes for WSNs is how to determine common keys between a pair of nodes. That is, a kind of key confirmation mechanism is a pre-requisite in order to find common keys between nodes. Currently, it is generally assumed that nodes can exchange their own key identifiers directly as a solution for key confirmation. However, the potential security risk of this simple solution is that an adversary may link observed key identifiers with nodes after observing communication between nodes. These information can help an adversary to deduce topology structure of the underlying key graph.

To counteract the above mentioned security risk of the simple solution for key confirmation, Marek Klonowski and Piotr Syga [9] presented a novel unlinkable key confirmation solution based on bloom filters to determine common keys between a pair of nodes. Each node should compute a local bloom filter as a compact representation for secret keys hold by itself. Then two nodes can exchange their bloom filters other than key identifiers. A node can check whether a key hold by itself is also an element of the other node's key ring by issuing set membership queries to the other node's bloom filter. As positive answers returned by bloom filters may be faulty with non-zero probability (false positive rate), their key confirmation process should be repeated with fresh randomness for several times between a pair of nodes to ensure that positive answers from bloom filters can be considered as correct with high probability. The additional communication overhead will also consume a large amount of nodes' energy.

As networks structure of WSNs may vary due to factors such as node failure or malfunctioning, single phase key pre-distribution is not able to adapt to dynamic changes in WSNs.To support a flexbile secure infrastructure for new nodes deployments,Albert Levi, and Salim Sarimurat [10] suggested use of multiple generation of dynamic key pools. As nodes should evolve their key rings by iterative hash computations by the end of each generation, it is implicit that their method requires all nodes to refresh their key rings synchronously. On the other hand, when WSNs are deployed in hostile environments, some compromised keys should be revoked since they may have been broken by an adversary. A node is compromised if its key ring is a subset of the compromised keys controlled by an adversary. The scheme in [10] is unable to revoke compromised nodes from WSNs. Moreover, an adversary is still able to intercept key identifiers exchanged between nodes during a specific generation. Generally speaking, revoking nodes from WSNs is more difficult than addition of new nodes.

As a result, how to efficiently exclude compromised keys and nodes from WSNs and evolve uncompromised nodes' key ring in an unlinkable way is also an issue to be addressed for random key pre-distribution. In this paper, we suggest an unlinkable key update mechanism based on bloom filters to ensure that uncompromised nodes can refresh their key rings securely while an adversary is infeasible to link key identifiers with a node.

Given a set of keys to be revoked, a key update node in our solution uses a bloom filter to represent the set of revoked keys and is responsible for evolving key pool and uncompromised nodes' key rings as well as excluding compromised nodes. Then the key update node runs several rounds of unlinkbale key confirmation process based on this bloom filter to determine whether a node is compromised or not. Recall that a node is compromised if its key ring is a subset of the revoked keys. If we use positive answers from the bloom filter to identify compromised keys, efficiency issues due to non-zero false-positive rate of bloom filters will also be encountered. As a bloom filter has zero false negative rate, we suggest that negative answers from the bloom filter can be used as indications of uncompromised keys to identify uncompromised nodes. Analysis shows that an uncompromised node can determine its unrevoked keys shared with the key update node with high probability in specified parameters setting.

Then a random seed that can be used to refresh uncompromised nodes' key rings should be generated and distributed among uncompromised nodes securely.The key update node broadcasts the random seed encrypted under a shared unrevoked key to a selected uncompromised node and its neighboring nodes in the formed key graph. Finally, uncompromised nodes can apply hash operations with the received random seed to update their key rings. Simulation results shows that this local broadcast mechanism help speed up propagation of the random seed. Section 2 decribes concept of bloom filters and CPA security of symmetric encryption. Section 3 decribes a key update scheme for revoking compromised nodes in WSNs and defines unlinkability notion for this kind of key update schemes. The presented key update scheme is proven to be unlinkable when the underlying symmetric encryption scheme is assumed to be CPA secure. Section 4 concludes this paper.

2 Preliminaries

2.1 Bloom Filter

The concept of bloom filter is presented by [2], which is a compact data structure used for answering set membership queries. Given l hash functions $h_1(\cdot), \cdots, h_l(\cdot)$ with range $[1, m]$,a bloom filter $BF_{m,l}$ is a bit vector with length m. To represent a set S consisting of n elements by $BF_{m,l}$, compute l entries $h_1(s), \cdots, h_l(s)$ for each element $s \in S$ and set $BF_{m,l}[h_1(s)] = 1, \cdots, BF_{m,l}[h_l(s)] = 1$. Given a membership query x, $BF_{m,l}$ returns a positive answer $BF_{m,l}(x) = 1$ to indicate that $x \in S$ if $BF_{m,l}[h_1(x)] == 1 \bigcap \cdots \bigcap BF_{m,l}[h_l(x)] == 1$ is true; Otherwise it returns a negative answer $BF_{m,l}(x) = 0$ to indicate that $x \notin S$. It is well known that a bloom filter will probably return false positive answers for membership queries but its false negative rate is always zero.

2.2 Semantic Security

Given a symmetric encryption scheme $\Pi = (KG, E, D)$,where E is encryption function and D denotes decryption function, define an experiment CPA^{A}_{Π},where A is a probabilistic polynomial time (PPT) adversary:

1) Challenger S generates a secret key $k = KG(\cdot)$;
2) A is given access to an encryption oracle $O_k(\cdot)$ that outputs a ciphertext $c = E_k(m)$ when taking as input a plaintext m.
3) A outputs two distinct equal-length plaintexts m_0 m_1.
4) S picks a random bit $b \leftarrow \{0, 1\}$ and provides A with $c^* = Enc_k(m_b)$.
5) A outputs a random bit b'.

A symmetric encryption scheme is CPA secure if $|Pr[b' = b] - \frac{1}{2}|$ is negligible for any PPT adversary A in the experiment CPA^{A}_{Π}.

3 An Unlinkable Key Update Scheme for WSNs

3.1 System Setup

Assume there are n sensor nodes $N_j, 1 \leq j \leq n$, distributed in a geographic region. Let ID be the set of key identifiers, K^i be the key pool with constant size P at the start of the i^{th} round,$1 \leq i$. Each node N_j holds a key ring $R^i_j \subseteq K^i$ with fixed size r at the start of i^{th} round.$M^i : K^i \rightarrow ID$ is a one-to-one mapping from the key pool of the i^{th} round to the set of key identifiers. Initially, the key ring R^1_j hold by N_j contains symmetric keys randomly sampled from a large key pool K^1 at the start of first round. Afterwards, key pool K^{i+1} will be derived from key pool K^i by executing the key update process described in subsection 3.2.

In case of key exposure, some compromised keys should be revoked and uncompromised nodes' key rings should be evolved to exclude compromised nodes. Our key update process is divided into several rounds. A key update node V constructs a set RK^i of revoked keys with size q at the start of the i^{th} round. For the sake of simplicity, we assume the full visible communication assumption as in [9] that assumes each pair of nodes can communicate with each other directly. This assumption help speed up propagation of a secure random seed.

The key update node V maintains a table S_j that records key identifiers associated with each node N_j. V also picks a random secret seed $seed_i$ at the start of i^{th} round and must ensure the following hold by the end of i^{th} round:

1) Key pool K^i will be replaced by K^{i+1} by the end of i^{th} round as follows: Each key $k^i \in K^i$ is replaced by a key $k^{i+1} = G(k^i||seed_i) \in K^{i+1}$, where $G(\cdot)$ is a secure hash function.
2) A node N_j is uncompromised at the start of i^{th} round if the set $R^i_j \backslash RK^i$ is not empty. In other words, an uncompromised node must hold at least one unrevoked key in its current key ring. Key ring R^i_j of an uncompromised node N_j at the start of i^{th} round will be replaced by R^{i+1}_j by the end of the i^{th} round as follows: Each key $k^i_j \in R^i_j$ is replaced by a key $k^{i+1}_j = G(k^i_j||seed_i) \in R^{i+1}_j$.
3) A PPT adversary that has knowledge of the revoked keys RK^i at the start of i^{th} round should have not enough knowledge to link key identifiers of the key set $R^i_j \bigcap \{K^i \backslash RK^i\}$ with the corresponding uncompromised node N_j by the end of i^{th} round by intercepting communications between nodes.

Define an experiment $Link^{n,M}_{\Pi}$ as a notion for unlinkbility, where M is a PPT adversary and Π is a symmetric encryption scheme.

1) Challenger C generates a key pool by running the key generation algorithm of Π and selects n node identifiers $id_1, \cdots, id_n$;
2) M is allowed to choose $h \leq n - 2$ compromised nodes from $\{id_1, \cdots, id_n\}$ and keep their corresponding key rings. The compromised nodes' identifiers is kept in RID.
3) M is given access to an oracle $O_C(\cdot)$ that outputs communication transcript between a node id and a key update node when taking as input a node identity id.
4) C outputs two distinct uncompromised nodes' identifiers mid_0 mid_1 from $\{id_1, \cdots, id_n\} \backslash RID$.
5) C picks a random bit $b \leftarrow \{0, 1\}$ and provides A with communication transcript between the chosen node mid_b and a key update node.
6) M outputs a random bit b'.

A key update scheme is unlinkable if $|Pr[b' = b] - \frac{1}{2}|$ is negligible for any PPT adversary M in the experiment $Link^{n,M}_{\Pi}$.

3.2 One Round of Key Update Process

The key update node V first generates a bloom filter $RBF^i_{n_b,l}$ with n_b bits at the start of i^{th} round by choosing l hash functions $h_0(\cdot), \cdots, h_{l-1}(\cdot)$,and initializes all entries of $RBF^i_{n_b,l}$ to zero.V executes the following Algorithm 1 to construct $RBF^i_{n_b,l}$ associated with the revoked key set RK^i with size q.

We use notation $RBF^i_{n_b,l}(k)$ to denote an answer return by the bloom filter $RBF^i_{n_b,l}$ for a membership query

Algorithm 1 Construction of RBF

```
Begin
for each $k \in RK^i$ do
    for $j = 0$ to $l - 1$ do
        $RBF^i_{n_b,l}[h_j(k)] = 1$;
    end for
end for
End
```

k. $RBF^i_{n_b,l}(k) = 1$ is a positive answer to indicate that $k \in RK^i$. $RBF^i_{n_b,l}(k) = 0$ is a negative answer to indicate that $k \notin RK^i$. Define a set $FRK^i = \{k_x | k_x \in K^i \backslash RK^i \bigcap RBF^i_{n_b,l}(k) == 1\}$. That is,$FRK^i$ contains all keys that are not revoked but get positive answers from $RBF^i_{n_b,l}$.

Step 1. In the following, V broadcasts the bloom filter $RBF^i_{n_b,l}$ to all nodes. Having received $RBF^i_{n_b,l}$,a node executes Algorithm 2 to construct a set $USK^i_j \subseteq R^i_j \backslash RK^i$. Recall that R^i_j is the key ring of node N_j with size r at the start of i^{th} round.

Algorithm 2 Construction of unrevoked keys

```
Begin
$USK^i_j = \emptyset$
for $k \in R^i_j$ do
    Initialize the answer $RBF^i_{n_b,l}(k) = 1$;
    for $j = 0$ to $l - 1$ do
        if $RBF^i_{n_b,l}[h_j(k)] == 0$ (a) then
            Set the answer $RBF^i_{n_b,l}(k) = 0$; and break;
        end if
    end for
    if $RBF^i_{n_b,l}(k) == 0$ (b) then
        $USK^i_j = USK^i_j \bigcup \{k\}$;
    end if
end for
End
```

Claim 1. *We have $USK^i_j = R^i_j \backslash \{RK^i \bigcup FRK^i\}$ by the end of Algorithm 2.*

Proof. By construction of the bloom Filter $RBF^i_{n_b,l}$,if $k \in RK^i$, we have $RBF^i_{n_b,l}(k) == 1$ holds. In addition, the set FRK^i enumerates all keys $k \in K^i \backslash RK^i$ with false positive answer $RBF^i_{n_b,l}(k) == 1$.As a result, $RBF^i_{n_b,l}(k) == 0$ holds if and only if $k \in R^i_j \backslash \{RK^i \bigcup FRK^i\}$. As $R^i_j \subset K^i$,$R^i_j \backslash \{RK^i \bigcup FRK^i\}$ is a subset of $K^i \backslash \{RK^i \bigcup FRK^i\}$. When $R^i_j \backslash \{RK^i \bigcup FRK^i\}$ is not empty, $k \in R^i_j \backslash \{RK^i \bigcup FRK^i\}$ implies $RBF^i_{n_b,l}(k) == 0$ holds and we conclude that $k \in USK^i_j$ by condition (b) in Algorithm 2. □

As $R^i_j \backslash \{RK^i \bigcup FRK^i\}$ is a subset of $R^i_j \backslash RK^i$, it is possible that USK^i_j is empty for some uncompromised node. That is,some uncompromised node will be identified as compromised by algorithm 3.2. Let E_j denotes the event that USK^i_j is empty for some uncompromised node N_j. T_j is a random variable to count the number of revoked keys in $R^i_j \bigcap RK^i$. We compute the probability of E_j as follows:

$$Pr[E_j] = \sum_{0 \le i \le r-1} Pr[E_j | T_j = i] Pr[T_j = i]. \quad (1)$$

As $RBF^i_{n_b,l}$ is used to represent q revoked keys, the probability of a false positive event $RBF^i_{n_b,l}(k) == 1$ for some unrevoked key $k \in R^i_j \backslash \{RK^i\}$ is approximately $(1 - (1 - \frac{1}{n_b})^{l \cdot q})^l$ [12] if we assume the hash functions are modeled by independent random functions. Given $T_j = j$,E_j occurs if and only if $RBF^i_{n_b,l}(k) == 1$ occurs for each of $r - i$ unrevoked keys in $R^i_j \backslash RK^i$. By the independence assumption, we have:

$$Pr[E_j | T_j = i] \approx (1 - (1 - \frac{1}{n_b})^{l \cdot q})^{l \cdot (r-i)}. \quad (2)$$

As key rings assigned to nodes are assumed to be randomly sampled:

$$Pr[T_j = i] \approx \binom{r}{i} (\frac{q}{P})^i (1 - \frac{q}{P})^{r-i}. \quad (3)$$

When taking parameters $P = 1000$, $q = 100$, $l = 2$, $r = 50$, $n_b = 64$, we get $Pr[E_j] \approx 0.0197$ by numerical computation. On the other hand, the optimum false positive rate of bloom filter $RBF^i_{n_b,l}$ in this setting is $\approx 0.6185^{\frac{n_b}{q}} \approx 0.7353$ [12]. Choose larger values for parameters l and n_b can further reduce $Pr[E_j]$ at the cost of additional communication overhead.

Step 2. When USK^i_j is not empty,the corresponding uncompromised node N_j will send a response message to V, which contains node identifier id_j of N_j.

Step 3. Having received response messages from uncompromised nodes with non-empty set USK^i_j, V will randomly picks a node N_j among uncompromised nodes that have sent response messages. Then V transmits a random number r_V to the chosen node N_j.

Step 4. Having received r_V, N_j randomly picks a key $k^*_j \in USK^i_j$ and transmits ciphertext $c^*_j = E_{k^*_j}(id_j || r_V)$ to V, which is encrypted under the chosen symmetric key k^*_j.

Step 5. Having received c^*_j,V performs Algorithm 3 to construct a shared key SK^i_j with the uncompromised node N_j.

Remark: Condition (c) denotes extraction of the matching key k by key identifier kid; Condition (d) denotes decryption of the ciphertext c^*_j under the extracted matching key k.

By the correctness of decryption, we have $k^*_j = k$, where k is the extracted matching key.

Algorithm 3 Key Extraction

```
Begin
Set $SK_j^i$=NULL,$flag = 0$ ;
for $kid \in S_j$ do
    $k = (M^i)^{-1}(kid)$; (c)
    if $D_k(c_j^*) == id_j||r_V$ (d) then
        $flag = 1$,$SK_j^i = k$,break;
    end if
end for
End
```

Step 6. V picks a random *seed* and broadcasts ciphertext $c^* = E_{SK_j^i}(r_V + 1||seed)$ to the chosen node N_j and its neighboring nodes in the key graph.

Having received c^*, N_j decrypts c^* to recover $r_V + 1||seed = D_{SK_j^i}(c^*)$. Then N_j utilizes *seed* to update its key ring as follows:

Algorithm 4 Update Key Ring

```
Begin
for each $k^i \in R_j^i$ do
    $k^{i+1} = G(k^i||seed) \in R_j^{i+1}$
end for
End
```

In addition, each neighboring node of N_j in the random key graph can also try to decrypt the ciphertext c^* by iterating the keys shared with N_j. If their decryption operations are consistent with condition (d), these neighboring nodes of N_j can also use *seed* to update their own key rings.The rest of uncompromised nodes that do not get *seed* continue to interact with V by executing Steps 2-6 repeatedly until their key rings can be successfully updated by the end of *ith* round.

When taking parameters $P = 1000$, $q = 100$, $l = 2$, $r = 50$, $n_b = 64$, $n = 100$, our simulation results shows that Steps 2-6 should be looped by 35 times on average to finish one round of the presented key update process in this parameters setting.

Claim 2. *A PPT adversary that has knowledge of revoked keys in RK^i at the start of ith round is computationally infeasible to link key identifiers with any uncompromised node N_j by intercepting communications between nodes, if the underlying symmetric encryption scheme is CPA secure.*

Proof. Note that the bloom filter $RBF^i_{n_b,l}$ contains no information with respect to the unrevoked keys in $R_j^i \bigcap \{K^i \backslash RK^i\}$ associated with an uncompromised node N_j. The ciphertexts $c_j^* = E_{k_j^*}(id_j||r_V)$, $c^* = E_{k_j^*}(r_V + 1||seed)$ encrypted under the symmetric key k_j^* are the only sources that an adversary can gain information about unrevoked keys in $R_j^i \bigcap \{K^i \backslash RK^i\}$. Intuitively, by semantic security of symmetric encryption schemes [11],it is computationally infeasible for a PPT adversary to learn information of the plaintext and encryption key when he can only intercept ciphertexts. As a result, a PPT adversary is not able to link key identifiers with any uncompromised node N_j.

Assume there is an adversary M can break unlinkability of our key uodate scheme with non-negligible probability. We construct an adversary A against a symmetric encryption scheme $\Pi = (KG, E, D)$ in CPA_Π^A.

A simulates $Link_\Pi^{n,M}$ for M in one round as follows:

A generates $n > 1$ node identifiers $id_1, \cdots, id_n$;

When M chooses $h \leq n - 2$ compromised nodes in the simulated $Link_\Pi^{n,M}$, we assume without loss of generality that they are $(id_1, \cdots, id_h)$.

A picks two distinct uncompromised identifiers mid_0 and mid_1 and we implicitly assume the corresponding two nodes share a common key k that is the challenge secret key used by the challenger in CPA_Π^A. Then A generates key rings for nodes in $\{id_1, \cdots, id_n\} \backslash \{mid_0, mid_1\}$. and M is provides with key rings of compromised nodes chosen by him.

Oracle $O_C(\cdot)$ in $Link_\Pi^{n,M}$ is simulated by A as follows:

Given a node identifier id as input, if $id \notin \{mid_0, mid_1\}$, A can simply simulate communication transcript between id and a key update node. Otherwise, A uses oracle access to $O_k(\cdot)$ in CPA_Π^A to generate ciphertexts for simulating communication transcript between $id \in \{mid_0, mid_1\}$ and a key update node. It is implicitly assumed that the challenge secret key k in CPA_Π^A is chosen as the shared key to generate communication transcript in the simulated $Link_\Pi^{n,M}$.

A outputs mid_0 and mid_1 in the simulated $Link_\Pi^{n,M}$.

A submits two distinct equal-length plaintexts $m_0 = mid_0||r_V$ $m_1 = mid_1||r_V$ to challenger S in experiment CPA_Π^A. S picks a random bit $b \leftarrow \{0, 1\}$ and provides A with $c^* = Enc_k(mid_b||r_V)$.

A concatenates c^* with the rest of communication transcript between node mid_b and a key update node.that can be generated by access to $O_k(\cdot)$ in CPA_Π^A and provided M with the correctly generated full communication transcript.

When M outputs a random bit b' in the simulated $Link_\Pi^{n,M}$, A also outputs a random bit b' in CPA_Π^A.

By the above construction, A succeeds in CPA_Π^A with non-negligible probability by the assumption M can break unlinkability of our key uodate scheme with non-negligible probability. This contradicts the assumption that the underlying symmetric encryption scheme is CPA secure. Hence the presented key update scheme is unlinkable.

□

Furthermore, it is relatively straightforward to see that the ciphertexts $c_j^* = E_{k_j^*}(id_j||r_V)$, $c^* = E_{k_j^*}(r_V + 1||seed)$ also provides authentication functionality between an uncompromised node and a key update node to prevent an adversary mount a forgery attack.

4 Conclusions

Random key pre-distribution is an important security technique for WSNs. In this paper, we consider an unlinkable key update mechanism for WSNs to revoke a subset of compromised nodes and evolve uncompromised nodes' key rings. Our scheme uses a bloom filter as a compact representation for the set of revoked keys. The use of a bloom filter make it unnecessary to exchange key identifiers for nodes to identify revoked keys. As a bloom filter has zero false positive rate, we suggest using negative answers returned by the bloom filter to identify uncompromised keys. Our analysis shows that uncompromised nodes can recover unrevoked keys with high probability. Then a key update node broadcasts a random seed encrypted by a shared unrevoked key to a specified uncompromised node and its neighboring nodes in the key graph. Then these nodes can use the recovered random seed to update their local key rings. In the future, the key update mechanism may be considered in other communication interference model such as protocol interference model [7].

Acknowledgments

The research is supported by Natural Science Foundation in JiangSu province,P.R.China (Grant No. BK20170512) and Natural Science Foundation of Colleges and Universities in Jiangsu Province, P.R.China (Grant No. 17KJB413003), Natural Science Foundation in Yangzhou City, JiangSu province, P.R. China (Grant No. YZ2016113),Yangzhou University Science and Technology Innovation Foundation Project?(Grant No. 2016CXJ025).

References

[1] S. Akleylek, A. Karakaya, "A survey on security threats and authentication approaches in wireless sensor networks," in *International Symposium on Digital Forensic and Security (ISDFS'18)*, pp. 1–4, Mar. 2018.

[2] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[3] H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," in *Symposium on Security and Privacy*, pp. 197–213, 2003.

[4] A. Diaz and P. Sanchez, "Simulation of attacks for security in wireless sensor network," *Sensors*, vol. 16, no. 11, p. 1932, 2016.

[5] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[6] V. Gligor, L. Eschenauer, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and Communications Security (CCS'02)*, pp. 41–47, 2002.

[7] P. Gupta, P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[8] M. S. Hwang, C. T. Li, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks," *Information Sciences*, vol. 181, no. 23, p. 5333-5347, 2011.

[9] M. Klonowski and P. Syga, "Enhancing privacy for ad hoc systems with predeployment key distribution," *Ad Hoc Networks*, vol. 59, pp. 35–47, 2017.

[10] A. Levi and S. Sarimurat, "Utilizing hash graphs for key distribution for mobile and replaceable interconnected sensors in the IoT context," *Ad Hoc Networks*, vol. 57, pp. 3–18, 2017.

[11] Y. Lindell, J. Katz, *Introduction to Modern Cryptography*, 2014. (`https://repo.zenk-security.com/Cryptographie\%20.\%20Algorithmes\%20.\%20Steganographie/Introduction\%20to\%20Modern\%20Cryptography.pdf`)

[12] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probalistic Analysis*, 2005. ISBN 13: 978-0521835404.

[13] J. Newsome, E. Shi, D. Song, A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *The Third International Symposium on Information Processing in Sensor Networks (IPSN'04)*, pp. 259–268, 2004.

[14] M. B. Paterson and D. R. Stinson, "A unified approach to combinatorial key predistribution schemes for sensor networks," *Design, Codes and Cryptography*, vol. 71, no. 3, pp. 433–457, 2014.

[15] K. D. Wong, Y. H. Hu, D. Li, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 17–29, 2002.

[16] O. Yagan and A. M. Makowski, "On the connectivity of sensor networks under random pairwise key predistribution," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5754–5762, 2012.

[17] O. Yagan and A. M. Makowski, "Zero-one laws for connectivity in random key graphs," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 2983–2999, 2012.

Biography

Bin Wang biography. Bin Wang received his Ph. D. degree of communication and information system in Shanghai Jiaotong University, P. R. China. His research interests include cryptography and network security. Dr. Wang is now an associate professor of Department of Electronics and Communication Engineering, Information Engineering College, Yangzhou University, located in No.196 West HuaYang Road, Yangzhou city, Jiangsu province, P. R. China.