Distributed Key Establishment in Disruption Tolerant Location Based Social Wireless Sensor and Actor Network

Jingzhe Du, Evangelos Kranakis
School of Computer Science
Carleton University
Ottawa, Canada
{jdu3, kranakis}@scs.carleton.ca

Amiya Nayak
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada
anayak@site.uottawa.ca

***Abstract*—We describe a novel Distributed Key Establishment (DKE) protocol in Disruption (Delay) Tolerant Location Based Social Wireless Sensor and Actor Networks (DTLBS-WSAN). In DKE, we propose that sensor nodes use neighboring signatures to establish their keys. Pre-distributed keys are used by actor nodes to strengthen communication security. We show that nodes can get guaranteed security when actors are connected and cover the network area and high security confidence can be achieved even without actor nodes when the adversary (malicious node) density is small. In DTLBS-WSANs, key (certificate) establishment, storage and look up are performed in a distributed way. Multiple copies of a certificate can be stored at nodes to improve key security and counter the adverse impact of network disruption.**

***Keywords*-Wireless Security; Wireless Sensor and Actor Network (WSAN); Public Key Cryptography; Disruption Tolerant Network (DTN); Distributed Storage**

I. Introduction

Wireless Sensor and Actor Network (WSAN) improves the robustness of Wireless Sensor Network (WSN) through the use of actor nodes. In the network, wireless sensor nodes are used to gather and relay neighboring environment information to the responsible receiver nodes, while actors are usually resource independent nodes which work according to the input of sensor nodes to take appropriate actions. WSAN works in a distributed way to perform tasks such as environment monitoring, home automation and battlefield surveillance [1].

In a dynamic WSAN, communication delays and disruptions have to be taken into consideration for the proper functioning of the network. Disruption (Delay) Tolerant Networks (DTN) have been widely studied recently, which are also called opportunistic networks or challenged networks. One of the purposes of DTN is to address issues in wireless networks where instantaneous source and destination node connections may not exist.

Location information can be used in WSAN. Wireless nodes could get their location information either by global positioning system (GPS) or localization algorithms [2]. In DTN, there are location based routing protocols (e.g., GLR algorithm [3]), which can fit in a disruption tolerant WSAN.

Wireless communications among nodes may express their social connections. Existing human social networks have a tendency of building relationships and can facilitate network applications [4]. However, these social networks are based on node similarities (e.g., similar origin, interest, etc.) and are not local. They would not work properly when the similarities can not be found in a WSAN.

WSAN needs security provisioning in a hostile environment. However, existing key management protocols [5], [6], [7] mainly work in a network that is densely connected and focus on the use of symmetric keys. Existing protocol [8] that uses symmetric and public keys highly relies on centralized mechanism and thus is hard to scale when some nodes are captured.

A. Contributions and Organization of the Paper

In this paper, we propose a novel distributed key establishment (DKE) scheme in disruption tolerant location based social wireless sensor and actor networks (DTLBS-WSAN). DKE is based on neighbor cooperation, with a node trusting its neighboring nodes with high probability. The working procedure of distributed public key certificate establishment scheme without Certificate Authority (CA) is presented. To further improve security, we can equip actor nodes with pre-distributed symmetric and public/private keys. We show that security is guaranteed when actors are connected and cover the sensor deployment area (*Powerful Model*) and high security confidence level can be achieved in the distributed system when the density of malicious nodes is small, even without central management (*Same as Sensor Model*). We also show that security assurance can be improved when actors increase their transmission powers so that they are connected and cover the entire network area while a sensor node has shorter transmission range than that of actors (*Semi Powerful Model*). We use a mechanism called “safety margin” to counter malicious certificate attack. We propose the use of location based social networks (cells) to facilitate and strengthen the security of distributed certificate storage and look up in wireless sensor and actor networks with disruptions and delays.

The rest of the paper is organized as follows. The related work is presented in Section II. Section III elaborates on our proposed solutions. We theoretically analyze the security strength of the proposed key management scheme in Section IV. Section V describes the details of experiments and analysis. Section VI concludes with possible future work.

II. Related Work

Key management schemes are necessary when security services are required in distributed wireless networks. In symmetric key cryptography systems, the simplest solution is a single key scheme, in which all nodes in a network share a unique key for secure communications. However, once a node is captured, network security is broken. Another extreme scheme is to use pairwise pre-distributed keys. Each node has to store $n-1$ (n is the number of nodes) keys and a total number of $\frac{n\times(n-1)}{2}$ keys are needed for the whole network, which is impractical especially in a dynamic network with new nodes joining in. To solve the above difficulties in key distribution, while still providing reasonable security service level, Eschenauer and Gligor [5] proposed a pre-distributed symmetric key management scheme in Wireless Sensor Networks (WSN). Instead of keeping $n-1$ keys for every node, only a small subset of keys are randomly chosen from a large key pool and the keys stored at every node are significantly reduced. To strengthen the security against malicious nodes, a q-composite key scheme was proposed in [6]. Rather than using one single pre-distributed key as the link key in [5], q keys between two nodes are used to calculate a new link key.

Compared with the symmetric key approach, public key (asymmetric) cryptography provides a security alternative. In Public Key Infrastructure (PKI) [9], a node needs to acquire its own certificate through a Certificate Authority (CA). When another node needs to communicate with this node, it will acquire and verify the certificate by going through the PKI chain until a trusted CA is found. In [8], a new key management protocol was proposed for WSAN, aiming at exploiting the resource abundant actor nodes and reducing the number of keys through a hierarchical approach, which was a combination of symmetric and public key system. High reliance on central management makes these existing works vulnerable to attacks.

Since nodes in PKI need to check public key certificates of others, these look up procedures need to address the adverse impact of long or variable delays introduced by DTN. In [10], a distributed storage mechanism called Cell-based Hash Table (CHT) was proposed to accelerate data item storage and look up, which can be used in facilitating the certificate storage and look up.

Different from PKI, self-generated public/private key pairs without the signature from CA can also be used in communication. GNU Privacy Guard (GPG) [11] and Pretty Good Privacy (PGP) [12] can be classified into this category. Existing algorithms (e.g., RSA [13]) can be used in the key pair generation. In PGP, the author proposed the idea of "Web of Trust", in which a node can collect multiple signatures from multiple third party nodes which are called "trust introducers" for its self-generated public key certificate. A certificate is trusted once the verifier finds it is signed by a trusted "introducer" node. However, once a highly trusted node is broken (i.e., captured), the proper functioning of this scheme is lost. With less social trust relationship, it is difficult for a new node to obtain trust in the network. "Web of Trust" is based on human social networks.

III. Distributed Key Establishment Algorithm

In this section, we will present the basic features of the proposed key management scheme and its security properties, deferring the detailed analysis of the scheme's security strength against malicious attack to the next section.

A. Definitions in Secure DTLBS-WSAN

Definition 1: (**Disruption tolerant location-based social network**) Disruption tolerant location-based social network (DTLBSN) emphasizes the importance of nodes to stay at (around) a specified location during some period of time, which have a tendency of being static during dynamic unrelated movements, but can cooperate with each other. When nodes are composed of wireless sensors and actors, it is called DTLBS-WSAN.

With wireless devices being widely used in recent years, local (and possibly distant) wireless communication without relying on centralized server nodes become possible. These devices may possess diverse sensing capabilities and can cooperate with each other in their local communication range. The use of wireless peer-to-peer emails in an organization is one such example. In [14], the author listed the security threats to this emerging way of communication. The attack to the single network key approach (one key is used by all mobile devices) is likely and the risk is critical. Due to this reason, proper key management has to be proposed. Current network graph partitioning mainly focuses on dense connected static graph. DTLBSNs are focusing on a community of nodes which may even be divided, sparse or evolving. Traditional social networks are about nodes (people) having similarities (familiarities) while DTLBSNs are about nodes with diversities (for some reason, they happen to be in close neighborhood, which may or may not be due to similarities). In a location-based social network, nodes mainly cooperate and communicate with others within their social network.

Definition 2: (**Semi-security**) Communications between nodes with pre-distributed keys are considered secure, while the communication between nodes with self-generated public/private keys without the signatures of trusted nodes is considered as semi-secure.

Semi-secure communication provides a platform for security establishment without relying on centralized nodes or servers. A node generates its public key certificate, asks for signatures from multiple neighboring nodes and stores it in a distributed manner.

B. Adversary Model

We assume the adversary (or a malicious node) can overhear, intercept and manipulate any messages passing through it. However, we assume malicious nodes are randomly deployed with low density. We assume nodes are not allowed to own multiple IDs and malicious nodes with multiple IDs will be detected. This can be achieved through hardware fingerprinting [15], [16] techniques. We differentiate between cooperative malicious nodes and independent malicious nodes (malicious nodes which act independently). The following definition is used in the proposed solution.

***Definition** 3:* (k **cooperative malicious nodes**) k ($k \geq 2$) cooperative malicious nodes means that exactly k malicious nodes cooperate with each other and share the information which they possess, including their own public/private keys as well as intercepted information through covert channels.

C. Key Pre-distribution and Distributed Key Establishment

The use of pre-distributed keys in existing WSAN key establishment protocols assumes the need of sensor to sensor security. In practice, not every sensor node needs to communicate with every other sensor node. In a location-based social network, most network communications exist among wireless sensors in their locality, as well as between wireless sensors and their corresponding actors. Since nodes mainly communicate with other location based social relations (nodes) that they are unable to know in advance, we propose the distributed approach that they establish public/private keys through neighboring cooperation.

We are aiming at a key agreement mechanism with different levels of security (e.g., guaranteed security, semi-security with high confidence, etc.), with possible guaranteed security expansion and negative trust (malicious) node key deletion.

1) Key Pre-distribution: Key pre-distribution can be used to increase network communication security. Before deployment, actors are loaded with symmetric keys, as well as public key certificates. Symmetric keys are used for data confidentiality and private keys are used for digital signatures. We assume these nodes are equipped with tamper resistant devices so that the keys are destroyed once they are captured. Actor nodes are also called trusted nodes. We assume actors have abundant resources for computational needs and they are trusted in their location based social networks (cells).

2) Distributed Key Establishment: We define the distributed key establishment process as follows:

I Setup Phase:

1) When a sensor node, say node A wants to setup its certificate, it generates public/private key pairs K_a, K_a^{-1} and its public key certificate $Cert_a$ (Figure 1, the Multiple Issuer and Extended Certificate Signature fields are left blank at this step).
2) It inquires its neighboring nodes to gather the information of which nodes are willing to sign its certificate.
3) The neighboring nodes which are willing to sign it respond to node A with their IDs. An actor node will reply to node A when it receives the request.
4) Node A picks up s nodes (node N_i has $ID_i, i = 1, 2, ...s$) out of all the replies and sends its signature requests, together with its public key certificate (Figure 1, without contents in the Extended Certificate Signature field at this step). Any replying actor node will be included in the s nodes.
5) These nodes sign the certificate using their self generated (pre-distributed for actors) private keys ($K_{N_i}^{-1}$) and return back their signatures $\{Cert_a\}_{K_{N_i}^{-1}}$to A.
6) When A receives all replies, it attaches the signatures to its certificate (Figure 1).
7) Node A stores its certificate in a distributed manner (i.e., using CHT [10]) and multiple copies of the certificate can be stored simultaneously.
8) All nodes follow the same procedure and certificates of all nodes are established and stored properly.
9) At the same time, some pre-distributed keys are stored at trusted nodes (actors) and communications between nodes with pre-distributed keys are considered secure.

II Verification Phase:

1) Node B acquires the certificate $Cert_a$ for A, either from A or from distributed storage sites.
2) It chooses c out of s signatures and verifies them, depending on its confidence requirement.
3) If a signature is from a trusted node, then $Cert_a$ is immediately valid by this node checking.
4) A node can also check its neighbors to see if there is a trusted neighbor node which has a shared symmetric key with one of the signing node and verifies it accordingly.

3) Properties of Distributed Key Establishment: The distributed key establishment protocol has the following

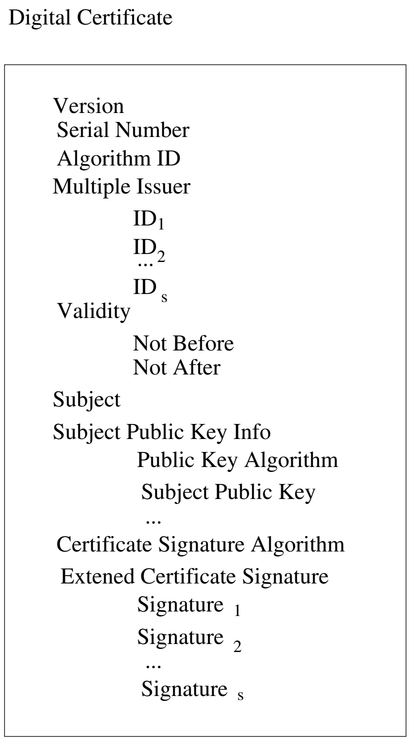

Figure 1. Distributed Certificate

properties:

***Theorem** 1:* The difficulty that an independent malicious node can successfully make changes to a certificate of another node without being discovered (through the verification with signing nodes) is as hard as attacking the key generation algorithm.

Proof. Since the number of signing nodes of a certificate is at least two, any node which wants to forge a certificate should possess at least two pairs of public/private keys. Although a node can generate any number of public/private key pairs as it wants, the key pairs are not verifiable because the node ID and public/private keys at a node should be unique. The malicious node then needs to guess the private keys of other signing nodes in order to successfully make changes to the certificate, which is as difficult as attacking the key generation algorithm.

Similarly, the following result can be obtained.

***Corollary** 1:* When there are s ($s > k$) signatures, the difficulty that k cooperative malicious nodes can successfully make changes to a certificate of another node without being discovered (through the verification with signing nodes) is as hard as attacking the key generation algorithm.

***Theorem** 2:* k cooperative malicious nodes can successfully make changes to a certificate of another node when there are s ($s \leq k$) signing nodes, however undeniable evidence exists once the malicious activity is being discovered.

Proof. To effectively change the content of a certificate without being discovered during verification phase, the malicious nodes have to use their valid private keys and pretend to be the original signing nodes. When k cooperative malicious nodes modify the content of a certificate, their IDs and signatures (which are generated using their private keys) are attached to the certificate. Once their activity is discovered, another node can prove this through their IDs and signatures.

4) Session Key Establishment: After the establishment of public key certificates, communicating nodes can exchange session keys encrypted through the public keys via a challenge/response protocol. Since the number of clock cycles needed by the processor to compute security function in symmetric key cryptography is much smaller than the number of clock cycles in asymmetric cryptography, the combination of symmetric session keys with public keys improves the encryption/decryption speed and reduces energy consumption.

D. Distributed Certificate and Certificate Revocation List Storage

It has been shown in [10] that Cell-based Hash Table (CHT) can be used to facilitate distributed data storage and look up in DTN. We use CHT to store the distributed certificates in the proposed solution when there is high probability that sensor and actor nodes will stay at their deployment location based social network regions (cells). Multiple copies of a certificate can be stored in different mapping regions to strengthen certificate security and speed up the certificate look up process in DTLBS-WSAN.

As with the distributed certificate storage, Certificate Revocation List (CRL) for malicious or captured sensor nodes should also be stored in a distributed manner.

E. Re-Keying and Key Revocation

When a certificate is going to expire, the owner needs to update its certificate accordingly. This node generates new public/private key pairs and the corresponding certificate. It then asks its neighbors to sign the certificate according to the procedures described in key setup phase. It will also sign it using its previous private key so that a distributed storage node can replace the old certificate with the new one simply by verifying its signature.

When a node has been captured, its certificate has to be revoked. The trusted node (i.e., actor) in its location based social network can issue signed certificate revocation message to the distributed CRL storage sites so that the storage nodes can verify the authentication of this message and list this revoked certificate in the CRL following verification. Distributed CRL addresses the issues when nodes are compromised. It is stored at the same mapping site where the corresponding certificates are stored. Once a certificate revocation message is received, the revoked certificate can be deleted.

IV. Analysis of Distributed Key Establishment Security Strategies

The security strengths of the proposed key establishment scheme are different when node transmission power differs. In *Powerful Model*, where actors are connected and cover the network area and sensors possess same transmission range as that of actors, guaranteed security can be achieved. In *Semi Powerful Model*, actors increase their transmission power so that they are connected and cover the network while the sensor communication range may be far shorter than that of the actors. A sensor node can get improved security assurance in this model. When actors can not cover the network (*Same as Sensor Model*), we show that several security mechanisms can be applied to ensure key is secure with high confidence. We present our analysis in the sequel.

A. Guaranteed Security in Powerful Model

***Theorem** 3:* If there are m actors distributed in a unit disk square according to Poisson process with communication range

$$r \geq \sqrt{\frac{\log m + \log\log m + c(m)}{m\pi}} \qquad (1)$$

with $c(m) \to \infty$, when $m \to \infty$ and sensors have the same communication range with actors, then a sensor node will have at least one actor node as its certificate signing node and its certificate can be verified by any other nodes, with high probability.

Proof. It is shown in [17] that with probability 1 a network with m nodes with communication range $r \geq \sqrt{\frac{\log m + c(m)}{m\pi}}$ with $c(m) \to \infty$, when $m \to \infty$ is connected and the network is covered when the communication range satisfies inequality (1). So when (1) holds, each sensor node will have at least one actor node in its distance 1 neighborhood and all the actors are connected. According to DKE, every sensor node will have at least an actor node as its signing node. Any certificate can be verified through this connected actor network. Since a source node can get a genuine key for any destination node, secure communication is guaranteed in this model.

B. Security Assurance in Semi Powerful Model

***Theorem** 4:* If there are m actors distributed in a unit disk square according to Poisson process with communication range satisfying inequality (1), while sensors have much smaller communication range than the communication range of actors, then a sensor node can get security assurance once its certificate is signed by an actor node through routing path in its locality, with high probability.

Proof. As shown in Theorem 3, the network is connected and covered when node communication range satisfies inequality (1). Actor nodes can broadcast their existence together with their public keys to all sensors in their coverage areas. So a sensor node will be either in distance 1 neighborhood of at least one actor node where it can communicate with the actor directly, or not. In the latter case, it encrypts its certificate using the actor's public key and tries to send its certificate to the actor node through local routing. The routing path should be local because the sensor node is close to the actor (i.e., within the actor's communication range). When the actor node signs its certificate and returns back to it (through direct transmission), the sensor node can verify the signed certificate using the already received actor public key. If there are any discrepancies during the routing process, the sensor node can re-initiate the routing process until it successfully gets its certificate signed. A sensor node can get security assurance once it receives the proper confirmation from the actor. Its certificate can be verified by others through the connected actor network.

C. High Confidence in Same as Sensor Model

1) Security Confidence with Key Pre-distribution: We assume nodes with pre-distributed keys are trusted. Assume the probability that a node is trusted is t. Once a copy of the signature from a trusted node is checked, a node will be confident that this certificate is the original which belongs to the owner. The probability that at least a signature belongs to the trusted node is (at least) $1-(1-t)^s$ (s is the number of signatures), in which case the checking node is fully confident with the certificate.

2) Security under k-Cooperative Malicious Attack: There are different possible attacks towards the proposed public key scheme. Malicious nodes can attack the certificate in the certificate setup phase, in distributed storage process or during certificate look up process. The attack in the certificate setup phase is possible when all the signatures are selected from malicious neighbors. Assume nodes are randomly deployed and malicious nodes are generated independently and randomly with probability p, it is difficult (the probability is at most p^s, with s the number of signatures) for malicious nodes to be selected as signing nodes during setup phase so that they can make changes to the certificate later. When k ($k \geq s$) malicious nodes cooperate with each other and possess each other's private keys, they can modify the content of an established certificate during distributed storage or look up process. By doing so, all the original Issuer IDs and authentic signatures in the certificate are replaced by the s malicious IDs and their bogus signatures. There are several ways to counter this attack. One is by checking the IDs to see if there are possibilities that all those signing nodes happen to be in the neighborhood of the certificate owner during the certificate setup phase. The other one is through multiple copies storage and look up approach. With multiple copies approach, if there are discrepancies, a checking node will be alerted. Also non-malicious nodes should cooperate with each other to counter this attack. We further analyze the security strengths of the proposed scheme in the following.

Theorem 5: If there are i nodes on a routing path in a network with n nodes, the probability that there is at least a malicious node inside the i nodes is less than $1-(\frac{n-k-i+1}{n-i+1})^i$. When $i=\frac{n+1}{\alpha k+1}$, this value is at most $1-\frac{1}{\sqrt[\alpha]{e}}$.

Proof. Assume the probability that there is at least a malicious node on routing path is p. Then $p=1-\frac{n-k}{n}\times\frac{n-k-1}{n-1}\times\cdots\times\frac{n-k-i+1}{n-i+1}\leq 1-(\frac{n-k-i+1}{n-i+1})^i$, where $1-(\frac{n-k-i+1}{n-i+1})^i=1-(1-\frac{1}{\frac{n-i+1}{k}})^i\approx 1-\frac{1}{\sqrt[\alpha]{e}}$, when $i=\frac{n+1}{\alpha k+1}$ and $n\rightarrow\infty$.

Theorem 6: When multiple copies (u copies, $u\in N$ and $u\geq 2$) approach is adopted, if the routing paths for these copies are disjoint, then the probability that all the paths have malicious nodes is less than $(1-\frac{1}{\sqrt[\alpha]{e}})^u$, when routing path length is at most $\frac{n+1}{\alpha k+1}$.

Proof. Assume the probability that there is at least a malicious node on routing path is p_j for routing path j, with $j=1,2,\cdots,u$. We start by evaluate p_1. Then $p_1\leq 1-\frac{n-k}{n}\times\frac{n-k-1}{n-1}\times\cdots\times\frac{n-k-i+1}{n-i+1}\leq 1-(\frac{n-k-i+1}{n-i+1})^i$, where $1-(\frac{n-k-i+1}{n-i+1})^i=1-(1-\frac{1}{\frac{n-i+1}{k}})^i=1-\frac{1}{\sqrt[\alpha]{e}}$, when $i=\frac{n+1}{\alpha k+1}$ and $n\rightarrow\infty$.

Assume m_1 nodes in routing path 1 and at least one malicious node in the m_1 nodes. Then $p_2\leq 1-\frac{(n-m_1)-(k-1)}{n-m_1}\times\frac{(n-m_1)-(k-1)-1}{n-m_1-1}\times\cdots\times\frac{(n-m_1)-(k-1)-i+1}{n-m_1-i+1}<1-\frac{n-k}{n}\times\frac{n-k-1}{n-1}\times\cdots\times\frac{n-k-i+1}{n-i+1}\leq 1-(\frac{n-k-i+1}{n-i+1})^i=1-\frac{1}{\sqrt[\alpha]{e}}$, when $i=\frac{n+1}{\alpha k+1}$ and $n\rightarrow\infty$.

Similarly, we can show that $p_j<1-\frac{1}{\sqrt[\alpha]{e}}$ for $j>2$. And we conclude that the probability that all the paths have malicious nodes is less than $(1-\frac{1}{\sqrt[\alpha]{e}})^u$ ($u\geq 2$). It is clear that with multiple copies approach, the security of certificate storage and look up process is greatly improved.

3) Distance k Safety Margin: We can further use distance k (for some small k) safety margin to counter malicious certificate attack. In forwarding a certificate (storage or look up), a non-malicious node will forward it to the next hop node and at the same time, broadcast it to distance k neighbors. Only next hop receiver needs to do the same thing. A node will compare two certificates if they are generated by the same node ID (owner). In Figure 2 (Δ means malicious and circles without labels means unreachable), when node B forwards a copy of a certificate to node M (malicious), it will also forward this certificate to other nodes in its distance k neighborhood (except node A which has the certificate already). This certificate can reach node C through nodes D,E,F when $k=4$. When M tries to modify the certificate, discrepancies will appear at node C.

V. Experimental Evaluation

A. Simulation Environment in NS2

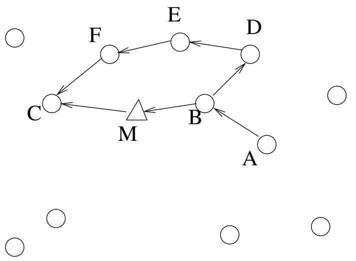

Figure 2. Distance k Safety Margin

We evaluate the performance of the proposed solution using the NS-2 [18] simulator. A random waypoint model is chosen as the motion pattern. The simulation parameters are shown in Table I.

Table I
Parameters of the simulations.

Parameter	Value
Number of mobile nodes	50/200
Mobility	0-20m/s(uniform distribution)
Transmission range	100m
Propagation model	*Two Ray Ground*
Simulation time	300/600 seconds
Topology size	1500m×300m/1500m×600m
Pause time	0/600 seconds
Antenna model	Omni-directional

Through simulation, we show that a node in a low malicious density cell should not select the nodes from high malicious density cells as certificate signing nodes and *Powerful Model* is an ideal security provisioning for a cell when malicious node probability in the cell is high. Our experimentation results on "safety margin" approach show that it can increase certificate security during storage and look up process. A node has a high probability of receiving an original certificate from nodes on the routing path ahead of a malicious node even if it receives a bogus certificate from the malicious node, when the node communication range is not too small. The increase in node communication range can improve the effectiveness of "safety margin" approach. For the simulation results, all points in the figures, as well as numbers in the tables are obtained as an average of 10 different runs with 10 different network topologies and movement patterns. The confidence intervals for the numbers are calculated at 95% confidence level.

B. Effect of Multiple Location-based Regions

We evaluate the effect of multiple location regions with varying malicious probabilities. The network topology area is divided into 9 (1-9) cells with 500m×200m size each and malicious probabilities are 0.2 (5 cells) and 0.8 (4 cells), with different probability cells separate each other. In the simulation, every non-malicious node collects its malicious neighbors probability. As shown in Table II, the collected probabilities for non-malicious nodes in 0.8 cells are consistent with the deployment probability in general. However, the low probability (0.2) cells are greatly affected by their neighbor cells. It is clear that if most of the nodes of a cell are malicious, a neighbor cell node is better not to choose them as signing nodes. With more malicious nodes in a high probability (0.8) cell, *Powerful Model* is a good choice for a few non-malicious nodes in that cell.

Table II
EFFECT OF THE NEIGHBOR CELLS.

Parameter	Column 1	Column 2	Column 3
Row 1	0.351±0.09	0.799±0.06	0.282±0.06
Row 2	0.709±0.11	0.341±0.05	0.724±0.07
Row 3	0.336±0.11	0.797±0.07	0.303±0.06

C. Distance k Safety Margin

We evaluate the distance k safety margin using GLR [3] routing protocol single copy approach. In the simulation, we set $k = 2$. If a node is on the routing path and receives a data packet, we evaluate whether it can receive the data packet through distance k safety margin approach. Figure 3 shows the result. It is clear that the probability that a routing node is also in distance k safety margin increases when node communication range increases.

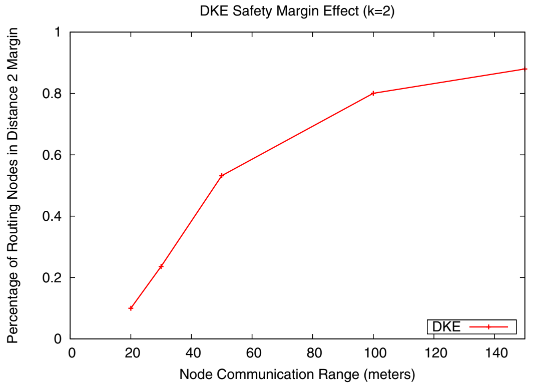

Figure 3. Distance Two Safety Margin Effect

The impact of various maximum nodes' moving speeds to the distance k safety margin effects when the node communication range is 100 meters is also evaluated and the results are shown in Figure 4. Different nodes' moving speeds affect the percentage of nodes which are also in distance k safety margin slightly and higher mobility (≥ 10 m/s) outperforms lower mobility (≤ 5 m/s).

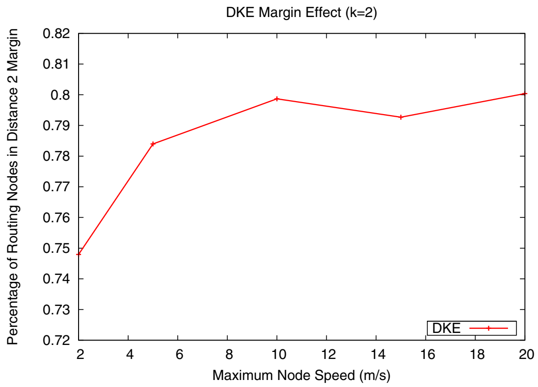

Figure 4. Distance Two Safety Margin Effect under Different Speeds

VI. CONCLUSIONS

We have proposed a novel distributed key establishment mechanism, called DKE. DKE uses a combination of key pre-distribution and neighbor key establishment to set up key pairs at nodes in DTLBS-WSAN. Public key certificates and certificate revocation list of nodes are stored in a distributed way to improve security and counter network disruptions. We have proved that guaranteed security can be achieved when actor nodes are powerful so that they are connected and cover the entire network area. We propose the use of "safety margin" approach to thwart malicious certificate attacks. Through simulation, we have shown the effectiveness of distance k safety margin approach in improving the certificate security.

As future work, we plan to further study security models in DTLBS-WSAN, exploring theoretically distributed trust establishment in a hostile environment.

ACKNOWLEDGMENT

Research supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Mathematics of Information Technology and Complex Systems (MITACS).

REFERENCES

[1] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad hoc networks*, vol. 2, no. 4, pp. 351–367, 2004.

[2] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," *Proceedings of the 7th annual international conference on Mobile computing and networking*, pp. 166–179, 2001.

[3] J. Du, E. Kranakis, and A. Nayak, "A Geometric Routing Protocol in Disruption Tolerant Network," in *Proceedings of 6th Workshop on Wireless Ad hoc and Sensor Networks (WWASN2009), June 22, 2009, (ICDCS Workshops 2009, June 22-26).* Montreal, Canada, pp. 109–116.

[4] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and F. Ferhatosmanoglu, "Crowd-Sourced Sensing and Collaboration Using Twitter," in *Proceedings of 11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM).* Montreal, Canada, 2010.

[5] L. Eschenauer and V. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 41–47.

[6] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *Proceedings of 24th IEEE Symposium on Security and Privacy*, 2003, pp. 197–213.

[7] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 2, pp. 228–258, 2005.

[8] Y. Lee and S. Lee, “A New Efficient Key Management Protocol for Wireless Sensor and Actor Networks,” *International Journal of Computer Science*, vol. 6, no. 2, pp. 15–22, 2009.

[9] ITU-T Recommendation X.509 | ISO/IEC 9594-8: “Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks”, 2001.

[10] J. Du, E. Kranakis, and A. Nayak, “Distributed Storage in Disruption Tolerant Network,” in *Proceedings of 1st International Workshop on Wireless Sensor, Actuator and Robotic Networks (WiSARN), (WoWMoM Workshops 2010)*. Montreal, Canada, 2010.

[11] GNU Privacy Guard, http://www.gnupg.org, Accessed June 28, 2010.

[12] P. Zimmermann, *The official PGP user’s guide*. MIT Press, 1995.

[13] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[14] M. Barbeau, “Assessment of the true risks to the protection of confidential information in the wireless home and office environment,” in *Proceedings of 11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. Montreal, Canada, 2010.

[15] M. Barbeau, J. Hall, and E. Kranakis, “Detection of rogue devices in bluetooth networks using radio frequency fingerprinting,” in *Proceedings of the 3rd IASTED International Conference on Communications and Computer Networks*, 2006, pp. 4–6.

[16] D. A. Knox and T. Kunz, “AGC-based RF Fingerprints in Wireless Sensor Networks for Authentication,” in *Proceedings of 1st International Workshop on Wireless Sensor, Actuator and Robotic Networks (WiSARN), (WoWMoM Workshops 2010)*. Montreal, Canada, 2010.

[17] P. Gupta and P. Kumar, “Critical power for asymptotic connectivity,” in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 1, 1998, pp. 1106–1110.

[18] The Network Simulator, NS-2, http://www.isi.edu/nsnam/ns/, Accessed December 18, 2010.