An Online Threshold Key Distribution Scheme for Symmetric Key Management

Alireza T. Boloorchi, M. H. Samadzadeh, and T. Chen
Computer Science Department, Oklahoma State University
Stillwater, OK, USA

Abstract— **The threshold secret sharing technique has been used extensively in cryptography. This technique is used for splitting secrets into shares and distributing the shares in the network to provide protection against attacks and to reduce the possibility of loss of information. In this paper, a new approach is introduced to enhance communication security among the nodes in a network based on the threshold secret sharing technique and traditional symmetric key management. The proposed scheme aims to enhance the security of the symmetric key distribution in the network. In the proposed scheme, key distribution is online which means key management is conducted whenever a message needs to be communicated. The basic idea is encrypting a message with a key (the secret) at the sender, then splitting the key into shares and sending the shares from different paths to the destination. Furthermore, a pre-distributed shared key scheme is utilized for more secure transmissions of the secret's shares. The proposed scheme, with the exception of some offline management by the network controller, is distributed, i.e., the symmetric key setups and the determination of the communication paths is preformed in the nodes. This approach enhances communication security among the nodes in a network that operates in hostile environments. The security analyses of the proposed scheme are provided.**

Keywords-component; Threshold Secret Sharing; Symmetric Key Management;

I. Introduction

In the cryptography literature, there are three basic techniques that are used by all protocols and key management schemes. The three are: symmetric shared keys, asymmetric public keys, and threshold secret sharing. Each technique has its advantages and disadvantages. For example, the symmetric approaches typically need a secure key distribution technique while the asymmetric ones provide this security implicitly [Diffie and Hellman 76]. Also, the symmetric algorithms are typically less computationally complex than the asymmetric algorithms [Stallings 10].

Threshold secret sharing is a widely-used technique in key management literature that was introduced in 1979 [Shamir 79]. This technique is used either by itself [Ogata and Kurosawa 96] [Ren et al. 08] or combined with other techniques [Deng et al. 04] [Wu et al. 07].

More specifically, the threshold secret sharing technique is used to protect vulnerable data (secret) by splitting the data into shares and distributing the shares among a number of nodes [Shamir 79] [Deng et al. 04] [Ogata and Kurosawa 96] [Wu et al. 07]. The most important property of the threshold secret sharing technique is that a specific number of the shares are needed for reconstructing the original secret, in other words, access to fewer than that specific number will not disclose any information about the original secret.

A. Related Work

Symmetric shared keys, asymmetric public keys, and threshold secret sharing are salient examples among a number of different techniques have been introduced in the cryptography literature to tackle the problem of security in network communication [Diffie and Hellman 76] [Shamir 79]. Along side of the symmetric and asymmetric key management techniques, many proposed schemes have used the notion of the threshold secret sharing technique in order to add more security to the symmetric schemes or in order to reduce the computational overhead of the asymmetric schemes [Deng et al. 04] [Wu et al. 07]. The threshold secret sharing technique is used by its own to enhanced data security in a number of studies such as [Ogata and Kurosawa 96] and [Ren et al. 08].

In a recent study [Lu et al. 09], the threshold secret sharing technique was used to improve Certificate-Based Encryption (CBE) [Gentry 03]. CBE is an asymmetric approach that uses Identity-Based Encryption (IBE) [Shamir 84] in public key encryption. Lu et al. utilized the threshold secret sharing technique to split the system master key (secret) into shares and distribute the shares among the nodes, which in CBE are stored in a single location. In a relatively similar work, Deng et al. [Deng et al. 04] used the threshold secret sharing technique and identity-based key management for authentication in Wireless Sensor Networks (WSN) where entities have less computational and communicational capabilities compared to other types of networks.

Ren et al. [Ren et al. 08] proposed a new scheme called HybridS to securely store and retrieve data in a Wireless Sensor Network in a distributed manner. In their scheme, they used the threshold secret sharing technique and Reed-Solomon coding [Reed and Solomon 60] to split keys and data, respectively. Each node encrypts its sensed data with a key, splits the key (secret) into shares by using the threshold secret sharing technique. It also uses the Reed Solomon coding technique to split the data shares.

In all of the above-mentioned schemes, the security of transmitting the shares after splitting the secret is generally overlooked. While it is true that access to fewer than a specific number of shares will not disclose any information about the secret, making the procedure of distributing the shares more secure is desirable when the transmitting data are critical. A method to provide this security is making the obtaining of *m* or more shares harder for adversaries so that they cannot reconstruct the secret easily. In the proposed scheme, a fully distributed secure scheme is provided with a goal of improving the shares' confidentiality.

A pre-distributed shared key scheme that is based on the EG scheme [Eschenauer and Gligor 02] is also used in this paper. The EG scheme is a well known and highly referenced symmetric based key management technique and several studies have improved its security and efficiency [Chan et al. 04] [Du et al. 07]. EG scheme is a probabilistic scheme in which a key pool exists, the nodes draw keys from that pool, and are subsequently put in their respective key rings [Eschenauer and Gligor 02]. Du et al. provided a new key management scheme that supports two tier heterogeneous sensor networks using a clustering approach. The authors proposed using EG scheme [Eschenauer and Gligor 02] in heterogeneous networks. They have compared their results with EG scheme. It is claimed that their new scheme outperforms EG scheme based on the number of messages that could be decrypted by adversaries where the same number of nodes are compromised for both schemes [Du et al. 07].

B. Summary of Contribution

In the EG scheme, the symmetric keys should be distributed among all the nodes before the nodes can securly communicate. In the proposed scheme, a new key is generated for encrypting each message. The generated key is sent through a fairly secure path to a receiver who wants to decrypt the message. In this work, an idea analogous to the EG scheme is used to secure the distribution paths. The number of pre-distributed keys in the proposed scheme is fewer than the EG scheme since nodes need to be connected only to a limited number of other nodes and certainly not to all the nodes in the network like the EG scheme. The strength of the proposed schemes is in securing the distribution of the symmetric keys that are used to encrypt messages in a light-weight manner.

In the analysis sections (Sections 5 and 6), it is shown that the proposed scheme is remarkably more secure than two similar schemes, i.e., the EG scheme [Eschenauer and Gligor 02] and Du et al.'s scheme [Du et al. 07]. The analysis results show that the messages in the network are communicated fairly securely even in the case that 90 percent of the nodes are compromised.

II. THE PROPOSED SCHEME

The proposed scheme is divided into two schemes with one completing the other: basic and enhanced schemes. Figure 1 presents an abstract architecture of the proposed scheme. For the basic scheme, it is assumed that each node, intending to send a message through the network to a given destination, encrypts the message with a key. Function *Th(n, m)* represents applying the threshold secret sharing technique to provide *n* shares where more than or equal to *m* of these *n* shares are needed to decrypt the secret with $m \leq n$ [Shamir 79]. The sender uses *Th(n, m)* to generate *n* shares from the key, i.e., the secret, that is used to encrypt the message. Then the sender sends the shares to the destination on different paths. The destination, using polynomial interpolation [Shamir 79], can reconstruct the key from *m* shares. Obtaining fewer than *m* shares does not reveal any information about the key [Shamir 79].

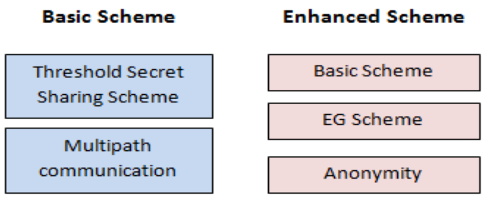

Figure 1. Architecture of the proposed scheme

In the enhanced scheme, more security is added to the basic scheme. The EG pre-distributed shared key scheme [Eschenauer and Gligor 02] is utilized to encrypt the shares that are being transferred by the links between the sender and the holders of the shares, to be referred to as shareholders hereafter. The shareholders decrypt the shares that are passing through them, discard information about the sender, and send the clean shares, i.e., the decrypted shares, to the destination. As a result of discarding information about the sender, the links between the shareholders and the destination do not reveal any information about the sender and consequently about the encrypted main message as to which shares are related.

A secure hash function, e.g., a suitable hash function from Secure Hash Algorithm family [NIST 08], is utilized to enhance the integrity of the communication, in other words, if the data is modified, it would be detectable.

The rest of this section is divided into two subsections. The first subsection describes the detail of the basic scheme that is subsequently improved in the second subsection where it is converted to the enhanced scheme.

A. Basic Scheme

The symmetric key management techniques have generally simpler and more light weight algorithms compared to the asymmetric techniques so, in spite of the increased security that the asymmetric methods usually provide, symmetric methods are still being used widely [Stallings 10]. Here security means resistance to any type of attack and not just cryptanalysis. A number of different protocols have utilized the symmetric approach, their common denominator is that the same key is used for both encryption and decryption. One of the most important challenges in symmetric techniques is the mechanism and detail of the distribution of the keys.

In the proposed scheme, a key is used as a symmetric key, and the threshold secret sharing technique is utilized to split the key as a secret. The shares of the secret are sent through multiple paths to the destination.

The advantage of using the threshold secret sharing technique is that access to fewer than *m* of the shares, out of a total of *n* shares, with $m \leq n$, does not disclose any information. In this technique, at least *m* paths must be compromised for a breach of security.

The proposed basic scheme exposes the shares to being compromised since the shares are not securely transmitted. A second component is required to be added to the infrastructure of the system as a distribution method with the goal of providing a more secure scheme. This component is described in the following subsection as a part of the enhanced scheme.

B. Enhanced Scheme

To improve the security of the basic notion of symmetric key management, the enhanced scheme is provided in this section. The enhanced scheme is divided into two parts: Pre-

Distributed Shared Keys and Symmetric Threshold Multipath Scheme that complements the first part, as explained below.

1) *Pre-Distributed Shared Keys*

This section tailors the EG scheme, which is described in section 1.2, to be used to enhance the security of the links between the sender and the shareholders. In the proposed scheme, the existence of a key pool at the network controller is assumed where a unique ID is associated with each key. The analysis of finding an appropriate pool size for different networks is provided in the analysis section, i.e., the first subsection of Section 4. Before a network is deployed, each node picks *k* distinct keys from the pool and put them in its key ring (*k* is a network parameter and is the number of distinct keys each node should draw from the pool to provide a given expected number of connections for the nodes in the network). The keys will be put back in the pool after a node draws *k* keys. After all of the nodes drew their *k* keys, they broadcast the list of their keys' IDs. Each node then compares the received IDs with its own keys' IDs to find shared keys with the other nodes.

In the proposed scheme, the keys are pre-distributed in a way that on average each node has shared keys with at least *n* other nodes, where *n* is the number of distributed shares in the threshold secret sharing technique.

The nodes in the network are considered as the vertices of a graph, and having at least one shared key between two nodes as a link between the two respective vertices. Each node has a degree of connections, which is the number of nodes with which a specific node has at least one shared key [Eschenauer and Gligor 02].

If *Pr* is calculated to be the probability that two nodes have at least one shared key in their key ring (based on Eschenauer and Gligor paper) then the expected degree of connections for each node *i* is:

$$d_i = \sum_{j=1}^{b} Pr_{ij} = Pr* (b - 1) = \left[1 - \frac{(1-\frac{k}{x})^{2(x-k+\frac{1}{2})}}{(1-\frac{2k}{x})^{(x-2k+\frac{1}{2})}}\right] * (b-1) \qquad (1)$$

where *b* is the number of nodes in the network [Eschenauer and Gligor 02]. Pr_{ij} is equal to the probability of node *i* having at least one shared key with node *j* and Pr_{ii} is considered to be 0, and each network node draws *k* distinct keys from the pool and the number of keys in the pool is *x*, $0 < k \leq x$.

After distributing the keys in the network, some of the nodes might not obtain the minimum number of required shared keys with distinct nodes in the network since *d* is an expected value. Several methods are proposed to solve this issue [Eschenauer and Gligor 02]. In the proposed scheme, the minimum number of required shared keys for each node is *n*, which is the first parameter of threshold secret sharing technique.

2) *Symmetric Threshold Multipath Scheme*

As mentioned in the previous section, a classic pre-distributed symmetric key management scheme is used in the proposed scheme that is called EG scheme [Eschenauer and Gligor 02]. EG scheme is deployed in a way that each node has shared keys with at least *n* other nodes in the network, where *n* is a parameter in the *Th(n, m)* threshold secret sharing technique and shows the number of distributed shares. These *n* nodes are referred to as neighbors hereafter.

A communication session starts every time a sender sends a message to a destination. The sender generates a random polynomial of degree *m-1* (*m* minus one), *S(x)*, at the beginning of each communication session where *m* is the second threshold secret sharing parameter. *m also* shows the minimum number of shares needed to reconstruct a secret. In the following equation, a_0 is the secret that is used as the key to encrypt the main message which is going to be sent from the sender to the destination.

$$S(x) = a_0 + a_1x + a_2x^2 + \dots + a_{m-1}x^{m-1}$$

The sender generates *n* shares as follows.

$$S_i = a_0 + a_1i + a_2i^2 + \dots + a_{m-1}i^{m-1} \bmod p$$

where *n* is the first threshold secret sharing parameter and indicates the number of distributed shares for each communication session. S_i is the share between the sender and the *i*th shareholder, $0 \leq i < n$, and *p* is a large prime number in finite field.

Then the sender sends the shares to the *n* neighbors. The shared keys are used to enhance the security of the links between the sender and the shareholders. Because of the properties of the threshold secret sharing techniques, it is guaranteed that adversaries cannot find any information about the secret with attacking less than *m* shareholders, and with at least shared keys from *m* shareholders the destination can rebuild the secret and decrypt the encrypted main message [Shamir 79].

Then the sender sends the following message to the shareholders.

$$H\,((share_i||address\ of\ the\ destination)_{Ski})$$

In this message, *H* is a secure hash function to provide integrity of the message, $share_i$ is the *i*th share, $0 \leq i < n$. The notation || shows concatenation, the subscript SK_i indicates the key with which the share is encrypted. The same notations are used hereafter.

Then the sender sends the following message to the destination:

$$H\,((Data)_K)$$

where *Data* is the clean main message and $(Data)_K$ is the main message that is encrypted with the secret *K*.

Receiving a message from the sender, the shareholders use shared keys to decrypt the shares. The shareholders then send the following message to the destination.

$$H\,(share_i)$$

where $share_i$ is the *i*th share and $0 \leq i < n$.

Before sending the clean shares to the destination. Each shareholder sends a message to the sender including the amount of time it takes for the clean share message to be transferred from that shareholder to the destination, which is referred to as the time distance between these nodes hereafter. The sender sends the shares' IDs list to the destination with a delay equal to the maximum amount of the times that it receives from the shareholders. The sender also sends share discard-messages to the shareholders that have not yet sent their shares and asks them to discard their shares. This procedure ensures that there would not be any information about the ciphertext and its secret's related shares in the network. Furthermore, the destination has already received all the needed shares in the case the clean shares are not being modified by the adversaries during their transmission from the shareholders to the destination. To deal with the possible modifications of the clean shares by adversaries, the amount of time the sender waits before sending the shareholders list and the discard-messages can be increased for environment's with more harshness. In this case, since the discard-messages will be received later, it is obvious that the probability of receiving redundant shares at the destination increases as well. Note that the modified clean shares could be recognized using hashing techniques. Even if adversaries have obtained all the main messages and the shares, as would be shown in Section 5.2, it is not easy to find the relation of the shares and main messages since there are many other messages and shares in the network.

Finally, receiving the ciphertext and enough shares, i.e., *m* shares, the destination can rebuild the secret, which is the key that is used to encrypt the main message, and use it to decrypt the ciphertext.

III. Theoretical Analysis

This analysis section contains the investigation of the security of the proposed scheme.

In the proposed scheme, adversaries might attack the following six component types to obtain information about the main messages. To analyze the security of the proposed scheme, the six component types can be considered as two groups where based on the data they contain or transfer, different types of attacks should be considered for each group.

1) Senders: The sender type nodes contain clean main messages and they are vulnerable against any type of attack.

2) Destinations: The nodes of this type only contain ciphertexts before receiving all required *m* shares, where *m* indicates the minimum number of a ciphertext's related shares required to reconstruct its secret.

3) Shareholders: Nodes of this type contain shared keys permanently. They also contain shares from the time they finish the decryption of shares until sending out clean shares, i.e., unencrypted shares, to the destination.

4) Sender-Destination links: The links of this type only transfer ciphertexts.

5) Sender-Shareholder links: Any link of this type transfer encrypted shares and the information about a share's corresponding sender which might disclose the relation between the share and the related ciphertext.

6) Shareholder-Destination links: Eavesdroping a link of this type, an attacker will acquire a clean share but no information about the share's related ciphertext.

Considering the six components listed above and the information they contain, adversaries could follow one of the following scenarios to obtain a clean main message.

Scenario 1: Attack component 1, a sender, and obtain a clean main message. This vulnerability is a common problem in security schemes unless a scheme provides a mechanism to discard the sensitive data as soon as they have been sent or encrypted. In the proposed scheme, the clean main messages are the sensitive data and are discarded as soon as they are encrypted. Therefore, adversaries do not have any chance to obtain a clean main message by compromising a sender and adversaries can only deal with a ciphertext after a message is encrypted.

Scenario 2: Attack a destination (component 2). Before the arrival of all shares related to a ciphertext if adversaries attack a destination, they do not obtain any data but the ciphertext which is useless without its corresponding secret (the main key). After the arrival of enough shares to rebuild the secret, adversaries might be able to decrypt the corresponding ciphertext that does not affect any other ciphertext's privacy since each main message is encrypted with a different secret.

Scenario 3: Brute force attack after adversaries eavesdrop a link between a sender and a destination and obtain a ciphertext. This scenario fails the adversaries since a secure symmetric key algorithm with a sufficiently large key is used to secure the main message [Barker and Roginsky 10]. By average the attackers should check $2^{key\ size\ -1}$ different keys and as Barker and Roginsky recommended, *112*-bit long keys are safe considering currently available machines. It shows that from this aspect, the proposed scheme is computationally secure.

Scenario 4: Acquiring a ciphertext and its secret's related shares, rebuilding the related secret, and decrypting the ciphertext. To obtain shares adversaries could attack component 3, 5. Different possible scenarios are discussed below.

a) An attack to components 3 (shareholders): Using this attack, adversaries obtain the shared keys of a shareholder and the shares arriving to that shareholder. Adversaries need to obtain *m* related shares to be able to decrypt the corresponding ciphertext. The shared keys in the network are shared among several nodes, and attacking a node, adversaries also obtain several other nodes' shared keys. However, obtaining the shared keys are not enough to decrypt the other encrypted shares, and adversaries also need the related counter for each share, which is different for each node. Therefore, even if through attacking a component 1, 2, or 4, adversaries gain the encrypted shares and the shares' related ciphertext, they require to either attack all the related shareholders or check all the available keys in the key pool to decrypt the shares. By average adversaries find the key for decrypting an obtained share after trying half of the keys in the pool size. This is correct if the adversary knows all the keys in the pool, otherwise it needs to check by average half of all the keys in the key space. For example, if the key size is *112* bits, adversaries need to try 2^{111} different cases. Note that all the above computations are only for one obtained share, and adversaries need to obtain more than *m-1* clean shares and also the ciphertext.

Considering a node being compromised as a success and not being compromised as a failure where the attack to a node is independent of the attacks to any other nodes. And also considering that the attacks to the nodes occur with the same probability, the number of compromised nodes in a network follows a binomial distribution.

Lemma 1: In the case of attacking shareholders to acquire shares, the probability of p_d adversaries decrypting a main message can be computed as follows.

$$p_d = \frac{1}{p_s * b} * \frac{c(p_{cn} * b * n * p_s , m)}{c(p_s * b * n , m)} \tag{2}$$

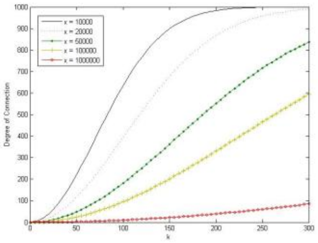

Figure 2. Expected degree of connection for 10000 nodes in the network and different pool sizes

where b is the total number of nodes and the probability of a node being compromised is p_{cn}, n and m are the threshold secret sharing technique's first and second parameters, and p_s, which is a network parameter, the probability that a node sends out a message.

Proof: The expected value of the number of compromised nodes, N_{cn}, is:

$$N_{cn} = p_{cn} * b \qquad (3)$$

N_{link}, the average number of links emanating from each node, which is the average number the links to which adversaries can gain access by attacking each node, can be stated as follows.

$$N_{link} = \frac{n * p_s * b}{b} = n * p_s \qquad (4)$$

where n is the threshold secret sharing technique's first parameter and p_s is the probability that a node sends out a message.

The probability of adversaries acquiring a specific ciphertext i, p_{c_i}, is:

$$p_{c_i} = \frac{1}{p_s * b} \qquad (5)$$

The average number of third type links in the network, N_{3rd}, which are the links between shareholders and destinations, is:

$$N_{3rd} = p_s * b * n \qquad (6)$$

The probability of adversaries acquiring all the shares related to a specific ciphertext i, p_{sh}, is calculated as follows:

$$p_{sh} = \frac{c(N_{cn} * N_{link}, m)}{c(N_{3rd}, m)} \qquad (7)$$

where m is the second parameter in the threshold secret sharing technique.

The probability of adversaries decrypting an encrypted main message, p_d, can be computed as follows.

$$p_d = p_{c_i} * \; p_{sh} = \frac{1}{p_s * b} * \frac{c(p_{cn} * b * n * p_s, m)}{c(p_s * b * n, m)} \qquad (8)$$

In Section 5.1, the probability of decrypting a ciphertext is numerically analyzed based on Lemma 1.

b) An attack to component 5 (sender-shareholder links):

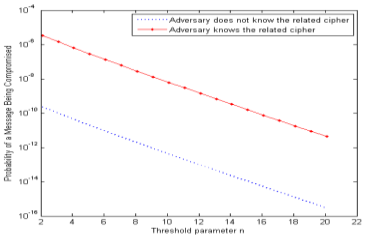

Figure 4. Comparing the probability of a message being compromised based on the threshold parameter in the case that adversary knows and does not know the related cipher

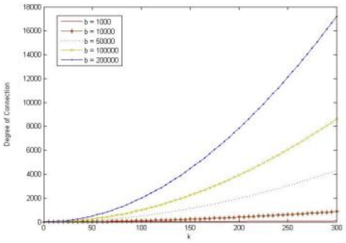

Figure 3. Expected degree of connection for different number of nodes and a pool with size 1000000

Using this attack, adversaries can obtain an encrypted share from each link they attack. As it has been mentioned for scenario 3, a large computational overhead is incurred by adversaries who intend to decrypt an encrypted message without having its key. This leads the attack to failure. Because of the same reason as in scenario 3, the link is computationally secure since the computational overhead makes it infeasible for adversaries to decrypt the encrypted messages they obtain [Stallings 10].

IV. NUMERICAL ANALYSIS

A. Pre-distributed Shared Keys

This sub-section provides numerical analysis for the pre-distributed shared key component of the proposed scheme. Figure 2 depicts the number of keys that each node should drag from the pool to satisfy a certain expected degree of connections for the nodes in the network. In this figure, several pool sizes are assumed, where *10000* nodes exist in the network. As Figure 2 shows, decreasing the pool size, the expected degree of connections for a certain key ring size increases. Figure 3 shows the effect of network growth on expected degree of connections for different ring sizes. These two diagrams illustrate the fact that the scalability of the proposed system is high, and a certain expected degree of connections can be obtained by choosing an appropriate pool size for different network sizes. However, the required expected degree of connections depends on parameter *n*, where *n* in *Th(n, m)* defines the redundancy and for security sake should not be very larger than *m*. On the other hand, *m* should not be very large because of the computational overheads in secret's reconstructing phase. Accordingly, it is not necessary that the actual required expected degree of connections to be very large.

B. Symmetric-Threshold Technique

1) Attack to components 3 (shareholders):

Based on Lemma 1, the probability of adversaries decrypting a main message, p_d, can be computed as follows.

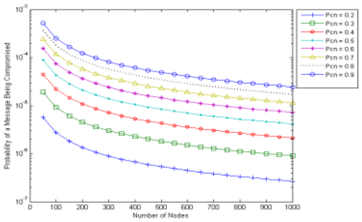

Figure 5. Comparing the probability of a message being compromised based on the probability of a node being compromised for different number of nodes in the network

$$p_d = \frac{1}{p_s * b} * \frac{c\left(p_{cn} * b * n * p_s,\ m\right)}{c\left(p_s * b * n,\ m\right)}$$

For several threshold secret sharing parameters, Figure 4 shows the probability of compromising a single main message in a network with 500 nodes and the probability of compromising a node equal to 0.5. As Figure 4 presents, the order of the probability of a node being compromised changes between 10^{-9} and 10^{-16}, in the case that adversaries do not know the relations between ciphertexts and their shares. Even if adversaries obtain the information about the relation between a ciphertext and its related shares, i.e., where p_d equals to one, the probability of a node being compromised is between 10^{-5} to 10^{-10}. The comparison of the probability of adversaries acquiring a clean main message for different number of nodes in the network and different probability of attacks to the nodes is provided in Figure 5. In this Figure, the first parameter of the threshold secret sharing technique, which is equal to the minimum degree of connections of each node in the network, is assumed to be equal to 3. The same assumption for the degree of connections of the nodes in the network has been made by Du et al. [Du et al. 07]. Figure 5 shows that even if 90% of the nodes in the network are attacked when 200 nodes are existing in the network, the probability of a message being compromised is near 10^{-4}. This amount is much better than EG scheme and Du et al.'s results based on the results provided in [Du et al. 07] where with compromising 180 nodes in the network the adversaries can obtain messages with the probability of 0.4 in EG scheme and 0.05 in Du et al.'s scheme [Du et al. 07] [Eschenauer and Gligor 02].

V. CONCLUSION

In this paper, a new scheme for highly secure communication in a network is introduced. The proposed scheme is a symmetric key management scheme with secure online key distribution. The strength of the proposed schemes is in securing the distribution of the symmetric keys that are used to encrypt messages. The symmetric keys are generated for each message and, to provide the security, the keys/secrets are split using the threshold secret sharing technique. A multipath approach along with a pre-distributed symmetric key management scheme is utilized to enhance the security of transferring the splits of the secrets to their respective destinations. It can be claimed that the proposed scheme provides enhanced network security requirements. Based on the analysis provided in this paper, confidentiality (i.e., access to the confidential data being restricted to authenticated entities) is assured to an acceptable level compared to the related studies. A level of dependability (i.e., being resilient to compromise and fault tolerant) is also provided by the proposed scheme. The shares of the secrets are distributed through different paths and reconstructing the secrets depends on the reception of the shares. The level of dependability can be determined based on the threshold secret sharing technique's parameters with the redundancy they provide for the shares.

REFERENCES

[Barker and Roginsky 10] E. Barker and A. Roginsky, *Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths*, Draft NIST Special Publication 800-131, U.S. Department of Commerce, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, June 2010.

[Chan et al. 04] H. Chan, A. Perrig, and D. Song, "Key Distribution Techniques for Sensor Networks", a chapter in a book titled *Wireless Sensor Networks*, pp. 277-303, Kluwer Academic Publishers, Norwell, MA, 2004.

[Crossbow 07] Crossbow Technology Inc., *imote2 Hardware Refernce Manual*, Crossbow Technology Inc., September 2007.

[Deng et al. 04] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and Identity-Based Key Management and Authentication for Wireless ad hoc Networks", *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, Vol. 2, pp. 107-111, Las Vegas, NV, April 2004 .

[Diffie and Hellman 76] W. Diffie and M. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, November 1976.

[Du et al. 07] X. Du, Y. Xiao, M. Guizani, and H. Chen, "An Effective Key Management Scheme for Heterogeneous Sensor Networks", *The Journal of ad hoc Networks*, Vol. 5, No. 1, pp. 24-34, January 2007.

[Eschenauer and Gligor 02] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks", *Proceedings of the Nineth ACM Conference on Computer and Communications Security (CCS'02)*, pp. 41-47, Washington, DC, November 2002.

[Gentry 03] C. Gentry, "Certificate-Based Encryption and the Certificate Revocation Problem", *Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'03)*, pp. 272-293, Warsaw, Poland, May 2003.

[Lu et al. 09] Y. Lu, J. Li, and J. Xiao, "Threshold Certificate-Based Encryption", *The Journal of Software*, Vol. 4, No. 3, pp. 210-217, May 2009.

[NIST 08] NIST - National Institute of Standards and Technology, Information Technology Laboratory, "Secure Hash Standard (SHS)", *Federal Information Processing Standards, FIPS PUB 180-3*, Gaithersburg, MD, October 2008.

[Ogata and Kurosawa 96] W. Ogata and K. Kurosawa, "Optimum Secret Sharing Secure Against Cheating", *Lecture Notes in Computer Sience: Advances in Cryptography (Proceedings of EUROCRYPT'96)*, pp. 200-211, Vol. 1070, Springer-Berlin, Heidelberg, Germany, 1996.

[Reed and Solomon 60] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields", *The Journal of the Society for Industrial and Applied Mathematics (SIAM)*, Vol. 8, No. 2, pp. 300-304, June 1960.

[Ren et al. 08] W. Ren, Y. Ren, and H. Zhang, "HybridS: A Scheme for Secure Distributed Data Storage in WSNs", *Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '08)*,Vol. 2, pp. 318-323, Shanghai, China, December 2008.

[Shamir 79] A. Shamir, "How to Share a Secret", *Commununications of the ACM*, Vol. 22, No. 11, pp. 612-613, November 1979.

[Shamir 84] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", *Proceedings of Conference on Advances in Cryptology (CRYPTO'84)*, pp. 47-53, Santa Barbara, CA, August 1984.

[Stallings 10] W. Stallings, *Network Security Essentials: Applications and Standards*, Prentice Hall PTR, Upper Saddle River, NJ, 2010.

[Wu et al. 07] B. Wu, J. Wu, E. B. Fernandez, M. Ilyas, and S. Magliveras, "Secure and Efficient Key Management in Mobile ad hoc Networks", *Journal of Network and Computer Applications*,Vol. 30, No. 3, pp. 937-954, August 2007.