A Combinatorial Approach for Key-Distribution in Wireless Sensor Networks

H. Shafiei*, A. Mehdizadeh†, A. Khonsari‡ and M. Ould-Khaoua§

Emails: shafiei@ipm.ir, a_mehdizadeh@aut.ac.ir, ak@ipm.ir, mohamed@dcs.gla.ac.uk

*School of Computer Science, IPM, Tehran, Iran
†Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
‡Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
§Department of Computing Sciences, University of Glasgow, UK
¶Department of ECE, University of Soltan Qaboos, Oman

Abstract—Sensor nodes are usually deployed in adversarial environments in which they are subject to compromise and revelation of critical information rendering the entire network useless. Therefore, secure communication of wireless sensor networks (WSNs) necessitates utilization of efficient key distribution schemes. Over the past few years, several works using probabilistic, deterministic and hybrid methods have been conducted to address key distribution among sensor nodes. In this paper we propose a novel method to deterministically distribute key-chains throughout a WSN utilizing expander graphs based on the Zig-Zag graph product. Given a set of constraints such as network size, amount of storage, radio range and key-chain length, we are able to efficiently construct a resilient yet scalable key distribution graph. The main advantage of the obtained method is providing a more user-adjustable and predictable framework compared to the previously proposed approaches. Simulation results demonstrate the efficiency of our proposed scheme and its general applicability to different network paradigms with diverse requirements.

I. INTRODUCTION

Constant monitoring and surveillance of some of today’s applications is well addressed by the widespread use of sensors which are inexpensive, low-power devices with limited computation, wireless communication and storage capabilities [1]. A WSN is comprised of a number of sensors collaborating with each other. Wireless nature of communication, limited amount of resources embedded in sensors, lack of infrastructure and uncontrolled environment make WSNs susceptible against adversarial attacks. Security requirements of WSNs are similar to those of ad-hoc networks [2]-[4]. These requirements could be attained by an efficient communication key distribution mechanism that addresses objectives such as efficient storage, processing and communication capabilities and measures such as scalability, connectivity and resilience against compromise of credentials. Due to the orthogonality of these objectives, there have been numerous works focusing on the design of several key distribution schemes over the past few years.

Sophisticated security primitives such as public key cryptography and elliptic curve cryptography have proven their high resilience against several security attacks in different applications [5]-[7]; however, their high computational and storage intensiveness, make them not applicable in resource limited sensor nodes. Therefore, secure communication in such networks is usually addressed utilizing symmetric keys to be shared between communicating parties. Generally, a key distribution scheme is required in which a list of keys (key-chain) is to be assigned to each of sensors before deployment. Sensors having identical keys can communicate with each other directly. It is desirable to store the keys in sensor nodes in order to the neighboring nodes possess one or more common keys. Nodes not having a key in common have to communicate through a path, called key-path, in which a key is shared between each pair of neighboring nodes. Due to the adversarial nature of deployment environment, sensor nodes are highly subject to compromise. An efficient key distribution algorithm should impede information revelation or disclosure of keys being used in other parts of the WSN.

An extreme solution to this requirement would be the utilization of unique pair-wise keys for each of node pairs in the network. In other words, in a WSN of size \( n \), each node would store a unique pair-wise key with each of the other \( n - 1 \) nodes of the network, resulting in a key-chain of size \( n - 1 \) in each sensor. Perceptibly, compromising any sensor node will not divulge the key-chain except the key of the compromised node, resulting in a desirable yet resource demanding resilience. In this method, the probability of key share success is equal to 1 and average key-path length is identical to 1 as long as the two communicating parties are in each other’s radio range. Nevertheless, having this perfect resilience calls for storing \( n - 1 \) keys in each node in a network of size \( n \), which is potentially beyond the storage limits of a sensor node in most practical circumstances. To utilize the WSNs limited yet priceless resources and preserving resiliency, while achieving scalability, a random pair-wise key distribution scheme based on Erdos and Renyi’s work on random graphs has been proposed in [8]. In a network of size \( n \) each sensor node stores the key of the remaining nodes with probability \( p \) and thus stores a random set of \( np \) pair-wise keys. Each nodes identity (ID) is matched with \( np \) other randomly selected node IDs with probability \( p \). A pair-wise key is generated for each ID-pair, and is stored in both nodes key-chain along with the ID of other party.

In another study, a deterministic pair-wise key scheme using Ramanujan expander graphs is introduced [10]. The authors
reported that their approach is superior compared to the probabilistic method in [8] due to its lower storage requirement and shorter average key-path length. Their approach attempts to design and implement a topology with the nearly optimal minimum diameter. However, the unpredictable size of the WSNs which is a natural consequence of random deployment of sensors in the field has been underscored in their study. This has been aroused by the fact that Ramanujan expander graphs are suitable to some particular cases in which the number of nodes in the graph is a prime number or of a prime power [11].

Recently, Zig-Zag product has been proposed by Reingold, Vadhan, and Widgerson [12]. This is the first purely combinatorial approach of constructing expander graphs which is more intuitive compared to the relatively complicated Ramanujan constructions that is based on rigorous mathematics.

In this work, we utilize Zig-Zag product expander construction that is simply applicable to nearly all real world circumstances. The attained Zig-Zag expander graph is more scalable than the Ramanujan ones while preserving an almost optimal diameter which is not far from that of Ramanujan [13]. Moreover, our approach provides a user-adjustable and predictable framework in which the user can specify how many resources i.e. memory she is willing to invest in order to reduce the diameter of the underlying graph. That is due to the fact of eliminating the abovementioned constraints i.e. having a prime or of prime power number of nodes. In this paper we introduce a mechanism for key distribution in WSNs based on such a combinatorial construction. The efficiency of the introduced mechanism is validated by our experimental results.

The rest of the paper is organized as follows. Section II describes the preliminaries of expander graphs and Zig-Zag product concepts. Section III explains our methodology to construct a key distribution graph by the proposed model. Section IV applies simulation to validate the analytical results and presents some sensitivity analysis. Finally, Section V concludes the paper and proposes a future direction for the study.

II. PRELIMINARIES

A. Expander Graphs

Expander graphs are of great importance in mathematics and engineering and have recently been explored in many computer science areas [12], [14], [15].

Informally, a regular undirected graph is an expander if it is highly connected. That is, it is easy to get from any vertex to any other vertex in just a few hops. Therefore, it is a graph \(G(E, V)\) in which every subset \(S\) of vertices expands quickly, in the sense that it is connected to many of the vertices in the set of complementary vertices \((\overline{S})\) . Formally it can be expressed as follows:

**Definition 1: Expansion Parameter-** Consider a graph \(G(E, V)\) with \(n\) vertices. Suppose \(S\) is a subset of \(V\), we define edge boundary of \(S\), \(\partial S\), to be the set of edges connecting \(S\) to its complement, \(\overline{S}\). This is actually the set of outgoing edges from \(S\). The expansion parameter of \(G\), denoted by \(h(G)\), is defined as:

\[
h(G) = \min_{\{S: |S| \leq \frac{n}{2}\}} \frac{|\partial S|}{|S|}
\]

where \(|S|\) denotes the size of \(S\).

According to the definition, if \(h\) is not small, then every such subset \(S\) has many neighbors outside \(S\) and therefore \(G\) is an expander. Thus, an expander is a highly connected sparse graph. It is this apparently contradictory feature of being both highly connected and at the same time being sparse that makes these graphs invaluable; which on one hand makes the existence of such graphs counterintuitive and on the other hand makes them so useful. Thus, explicit constructions of expanders have been considered a very important issue in the literature. Until recently all of the expander graph constructions have been algebraic, usually leveraging the theory of finite fields and the Cayley graphs of certain groups [15]. Only recently has there been some success using combinatorial tools to create families of expander graphs. The Zig-Zag product of Reingold, Vadhan, and Widgerson [12] gives a recursive construction of expander graph families that yields to a pleasingly intuitive analysis. In what follows, we applied the very recently graph theoretic Zig-Zag product [12] approach to systematically study resilient WSNs key distribution graph.

**Definition 2:** A \(d\)-regular graph \(G\) on \(N\) vertices is said to be a \([N, d, \lambda]\)-graph if the second largest eigenvalue of the normalized adjacency matrix \(A\), representing \(G\) is \(\lambda\).

It is now almost common knowledge that for a graph to be a good expander, the second largest eigenvalue of the adjacency matrix \(A\) must be as small as possible compared to the largest eigenvalue [16] [15].

B. Zig-Zag Product

Zig-Zag product is the product of a large graph with a small graph in which, the resulting graph inherits its size from the large one, its degree from the small one and it has good expansion properties as long as the two original graphs have good expansion properties.

Informally, the Zig-Zag product of \(G\) and \(H\), denoted by \(G \boxtimes H\), has vertices \((v, k)\) for every \(v \in V(G)\) and \(k \in V(H)\) (so there is a "cloud" of vertices of \(H\) around every original vertex of \(G\)). Two vertices \((v, k)\) and \((u, l)\) are said to be adjacent, intuitively if one can travel between them in a "Zig-Zag" path of length 3: one step on \(H\) in the \(v\) cloud, then switching to the \(u\) cloud according to an edge of \(G\), and finally step in the \(v\) cloud.

Formally, let \(G_1\) be a \([N_1, d_1, \lambda_1]\)-graph and let \(G_2\) be a \([d_1, d_2, \lambda_2]\)-graph. Randomly number the edges around each vertex of \(G_1\) by \(\{1, 2, \ldots, d_1\}\), and each vertex of \(G_2\) by \(\{1, 2, \ldots, d_2\}\). Then the Zig-Zag product of \(G_1\) and \(G_2\), as introduced in [12], is a graph \(G\) defined as follows:

The vertices of \(G\) are represented as ordered pairs \((v, k)\), where \(v \in \{1, 2, \ldots, N_1\}\) and \(k \in \{1, 2, \ldots, d_1\}\). That is, every vertex in \(G_1\) is replaced by a cloud of vertices of \(G_2\).
The edges of $G$ are formed by making two steps on the small graph and one step on the big graph as follows:

1) A step Zig on the small graph $G_2$ is made from vertex $(v,k)$ to vertex $(v,k[i])$, where $k[i]$ denotes the $i$th neighbor of $k$ in $G_2$, for $i \in \{1, 2, \ldots, d_2\}$.

2) A deterministic step on the large graph $G_1$ is made from vertex $(v,k[i])$ to vertex $(v[k[i]],l)$, where $v[k[i]]$ is the $k[i]$th neighbor of $v$ in $G_1$ and correspondingly, $v$ is the $l$th neighbor of $v[k[i]]$ in $G_1$.

3) A final step Zag on the small graph $G_2$ is made from vertex $(v[k[i]],l)$ to vertex $(v[k[i]],l[j])$, where $l[j]$ is the $j$th neighbor of $l$ in $G_2$, for $j \in \{1, 2, \ldots, d_2\}$.

Therefore, there is an edge between vertices $(v,k)$ and $(v[k[i]],k[i][j])$ for $i,j \in \{1, 2, \ldots, d_2\}$.

It is shown in [12] that the Zig-Zag product graph $G = G_1 \otimes G_2$ is a $[N_1 \times d_1, d_2^2, \lambda]$-graph with $0 < \lambda < \lambda_1 + \lambda_2 + \lambda_2^2$. Further, that $\lambda < 1$ if $\lambda_1 < 1$ and $\lambda_2 < 1$. Therefore, the degree of the Zig-Zag product graph depends only on the smaller component graph whereas the expansion property depends on the expansion of both the component graphs, i.e. it is a good expander if the two component graphs are good expanders.

1) Zig-Zag Construction: As mentioned before, this process has been considered to be well described intuitively [12]. It is instructive to illustrate this intuition with the following example. Assume that Graph $G_1$ is a $K_5$ (i.e. 5-Complete graph) and $G_2$ is a $C_4$ (i.e. 4-Cycle graph) as shown in Figures 1.(a) and 1.(b), respectively.

The obtained Zig-Zag product of $G_1$ and $G_2$ is illustrated in Figure 1.(e). In order to clarify the product process, we describe the sequence of producing the obtained graph of Figure 1.(e) by the following two steps. First, we produce the nodes, then the edges of the final graph. In order to produce the nodes of the final graph, it is necessary to substitute every node of $G_1$ with a copy of $G_2$. Since the degree of $G_1$ is 4 and the number of nodes in $G_2$ is 4, it is plausible to do the abovementioned copying process.

The result of the above process is an intermediate graph which is depicted in Figure 1.(c) the next step is to obtain the edges of the final graph using the following iterative process. Let us for the time being focus on node $c[1]$ which is boldfaced in Figure 1.(d). There exists an edge between $c[1]$ and $d[3]$ if the following conditions hold:

1) There is an edge between $c[1]$ and $b[1]$ in a copy of $G_2$
2) There is an edge between $b[1]$ and $a[3]$ in $G_1$
3) There is an edge between $a[3]$ and $d[3]$ in $G_2$

The above process is repeated for all the nodes of the intermediate graph such as $c[1]$ to produce the edges of the final graph.

2) Diameter: Graph’s diameter is the largest number of vertices which must be traversed in order to travel from one vertex to another excluding paths with backtrack, detour, or loops. The lower and upper bound diameter of any connected $k$-regular graph with $n$ vertices is equal to $\log_{k-1} n$ and $2\log_{k-1} n + o(1)$, respectively [17]. Bollobas and Chung [17] proved that the diameter of a $k$-regular expander on $n$ vertices plus a random perfect matching is almost less than

Fig. 1: Zig-Zag product of $K_5$ and $C_4$
log_{k-1} n + \log_{k-1} \log n + o(1) \) as \( n \) tends to infinity. Hence, the growth order of the diameter of the expander graph is \( O(\log n) \), which is near optimal; consequently, makes it a suitable candidate in WSN key distribution graphs which is highly dependent the key-path length.

3) Scalability: Graphs which are constructed using Zig-Zag product iteratively inherit the key feature of having constant degree. In other words, the degree of graph is not dependent on the number of nodes in the graph and consequently, as the number of nodes increases, the degree of graph remains constant. This effectively utilizes the previously determined limited amount of storage while decreasing resilience insignificantly. The noteworthy feature of having constant degree is twofold with respect to the memory space requirements for storing the keys in each WSN nodes. In this way, while keeping the priceless memory resource at the desired size at the same time less power is consumed compared to the case which memory is larger and thus results in a more scalable key distribution paradigm.

III. THE PROPOSED MODEL

This section describes how to construct our key distribution graph. Let \( n \) be the number of nodes in WSN and \( k \) be the keychain length which is dictated by the connectivity requirements of the underlying WSN and the limited amount of storage in every sensor. The key distribution based on the general expander graph can be summarized as the following steps:

1) Constructing an expander graph based on the given values of \( k \) and \( n \) using the method described in Section II.

2) Establishing a pair-wise key \( k_{ij} \) for each connected pair of nodes \((i, j)\) in the expander graph and allocating \( k \times (\text{number of bits for a key}) \) in neighboring nodes connected by a link.

Parameters of expander graph such as degree, expansion and diameter identifies security measures of the key distribution methodology. The expander graph proposed in [9] has used Ramanujan expander graph construction. Explicit construction of such graphs for a fixed \( k \) when \( n \) tends to infinity has only been described in the case where the number of nodes is prime or of a prime power. This is still an open problem in the general case [11]. Moreover, after adding new sensors to a previously established network, the values of \( n \) and \( k \) will change. Thus, it is necessary to render the connectivity graph another time which is a time consuming process. Additionally, redistribution of key-chains calls for extra communication and computation overheads which is not trivial in a WSN.

The Zig-Zag method is a straightforward algorithmic expander graph construction scheme which can be applied to general expander graphs with arbitrary \( k \) and \( n \) without worrying about any correlation constraints between \( k \) and \( n \). These flexibilities makes Zig-Zag product graphs a candidate of paramount value compared to other expander graphs in WSNs. As described in [15], the diameter of Zig-Zag expander graphs are not as good as the diameter of Ramanujan expander graphs which are optimum, however as shown in [13] the diameter of expander graphs of both Ramanujan and Zig-Zag methods have an order of \( O(\log n) \) which their differences are not significant in almost all practical applications.

IV. EXPERIMENTS AND SIMULATION RESULTS

The obtained results have been validated by means of a discrete-event simulator. Each simulation experiment was run until the model reached its steady state that is until a further increase in simulated model cycles does not change the collected statistics appreciably. For each simulation experiment, 10 batches were run to collect the statistics of interest; each batch consisted of 1,000,000 cycles. Statistics gathering was inhibited for the first batch to avoid distortions due to the initial warm-up conditions.

Numerous validation experiments have been performed for several combinations of important parameters, such as different key distribution schemes with arbitrary node degrees and different deployment distribution of sensors. However, for the sake of specific illustration, we only presented very few cases. For each scenario we firstly, construct an expander graph, using the Zig-Zag product as described in section II-B, with the number of nodes \( n = 1000 \) and we change the node degrees, \( k \), as described bellow. Then we distribute keys such that \( k \) keys are assigned to each node. Secondly, we generate a Ramanujan graph with an approximately same number of nodes and degrees and then we distribute keys to each node, subsequently. Finally, we generate a random geometric graph \( G^{\mathbb{Z}}(n, r) \) for a network of size \( n = 1000 \) and radio range \( r \) to simulate the underlying physical network.

Figure 2 depicts the probability of connectivity of different physical networks with \( n = 1000 \) number of nodes and node degrees \( k = 50, k = 100, k = 200, k = 400 \), for both, Ramanujan graph depicted in solid lines and Zig-Zag expander graph, illustrated as dotted lines in the figure. The X axis in the figure represent the probability of connectivity of physical network and the Y axis shows the radio transmission range in meters. Figure 2 reveals that the key distribution scheme which is based on Zig-Zag graph has comparable probability of connectivity compared to that of Ramanujan graph. To sum up, we could preserve resilience while our method is not restricted to the graphs with prime or of prime power number of nodes. Such method is considered a viable key distribution scheme in resource limited WSNs of today’s technology.

V. CONCLUSION

Security is an indispensable concern for sensor networks to work safely in real world circumstances, in particular over adverse or hostile environments. There have been a great deal of research studying various aspects of WSNs security. One of the phenomenal topics explored in WSN security that requires special consideration is establishment of cryptographic keys for later use. Researchers have proposed a variety of protocols over several decades for this well-studied problem. Considering scarce and priceless resources of WSN in terms of processing, communication and storage, proposing an efficient key distribution scheme which considers objectives
such as generality, connectivity, resilience and scalability is mandatory. In this paper, we introduced Zig-Zag product as a novel solution to construct efficient and scalable expander graphs. The obtained expander graphs are used to distribute pair-wise key-chains among sensor nodes. Simulation results revealed the efficiency of the proposed scheme in order to achieve a WSN with high degree of connectivity and resiliency. The proposed method is applicable to a wide range of WSNs environments with different requirements. As a future research, parameters such as degree, diameter and connectivity are yet to be explored within a solid mathematical framework using wellknown probabilistic models.

ACKNOWLEDGEMENT

The authors would like to thank anonymous conference reviewers for their valuable and constructive opinions enriching this final manuscript. We also thank persistent efforts of Mr. Arash Tavakol and Ms. Foozhan Hosseini who helped us delivering this work.

REFERENCES


