IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 February 2024, accepted 21 May 2024, date of publication 23 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404816

RESEARCH ARTICLE

Determining Distributions of Security Means for WSNs Based on the Model of a Neighborhood Watch

BENJAMIN FÖRSTER[1], PETER LANGENDÖRFER[1], AND THOMAS HINZE[2]
[1]Wireless Systems, Innovations for High Performance Microelectronics, 15236 Frankfurt an der Oder, Germany
[2]Faculty of Biological Sciences, Friedrich Schiller University Jena, 07743 Jena, Germany

Corresponding author: Benjamin Förster (bfoerster@ihp-microelectronics.com)

ABSTRACT Neighbourhood watch is a concept allowing a community to distribute a complex security task in between all members. Members carry out security tasks in a distributed and cooperative manner ensuring their mutual security and reducing the individual workload while increasing the overall security of the community. Wireless sensor networks (WSNs) are composed of resource-constraint independent battery driven computers as nodes communicating wirelessly. Security in WSNs is essential to prevent attackers from eavesdropping, tampering monitoring results or denying critical nodes from providing their services and potentially cutting off larger network parts. The resource-constraint nature of sensor nodes prevents them from running full-fledged security protocols. Instead, it is necessary to assess the most significant security threats and implement specialised security solutions. A neighbourhood watch inspired distributed security scheme for WSNs has been introduced by Langendörfer aiming to increase the variety of attacks a WSN can fend off. The framework intends to statically distribute requirement-based selections of online security means intended to cooperate in close proximity on large-scale static homogeneous WSNs. A framework of such complexity has to be designed in multiple steps. We determine suitable distributions of security means based on graph partitioning concepts. The partitioning algorithms we provide are NP-hard. To evaluate their computability, we implement them as $0-1$ linear programs (LPs) and test them on WSN models generated with our novel λ-precision unit disk graph (UDG) generator.

INDEX TERMS Cooperative security framework, distributed security means, graph generator, linear programming, neighbourhood watch, unit disk graphs, wireless sensor networks.

I. INTRODUCTION

WSNs are networks consisting of independent low power computing units called sensor nodes running on battery, communicating wirelessly and carrying out monitoring or controlling tasks. Information gathered by sensor nodes is transmitted to base stations (BSs). In large-scale static homogeneous WSNs considered in this work, the communication takes place hop-by-hop. Additionally, only a small subset of nodes is connected to a BS. The term static means nodes in the network are immobile and placed at a fixed position. Homogeneous implies that all nodes in the network have the same hardware capabilities. In large-scale WSNs, information to and from nodes is transmitted via intermediate nodes (hop-by-hop). Especially, when applied to critical infrastructures, WSNs need to ensure certain security attributes regarding transmitted data. In general, WSNs are vulnerable to a multitude of attacks. Therefore, security risks have to be well assessed and covered in the design of the network. The limited computational power and energy supply of nodes constrain the types, complexity and scoop of security means suitable to WSNs. Hence, we have to compromise between security and longevity of a WSN. Such a compromise requires the necessity to identify the most likely and costly threats to a WSN and select security means accordingly. A number of concepts to

The associate editor coordinating the review of this manuscript and approving it for publication was Arun Prakash.

identify and priorities security means based on numerous properties have been proposed [1], [2]. The general risk assessment, independent of the techniques that come along with it, comprises the steps: risk identification, analysis and evaluation [1]. The risk identification assess a system to identify external threats and system vulnerabilities. The risk analysis assess the likeliness of identified risks to be exploited and the resulting consequences. Finally, the risk evaluation derives consequential actions based on the analysis results. One well established concept to do so are attack defence trees [2], [3], [4]. An attack tree based risk assessment approach for location-privacy in WSNs is presented in [2]. In [3] an attack defence tree based risk assessment model for unmanned aerial vehicles model is researched. Attack-defence trees are graphical models illustrating potential attack scenarios and corresponding countermeasures for a system and evaluate how various attacks can be mitigated through countermeasures and how those are interrelated. In [4], Langendörfer proposed an extended concept of attack defence trees considering the resource limitations of the underlying nodes called ''Attack Defence Resource Trees'' (ADRT). It targets a selection of security means based on pre-defined security incentives tailored to the area of application of a WSN and the resulting most likely threat scenarios while taking into account the hardware constraints and longevity of devices (resources). Even with an optimal selection of security means, the coverage of a larger scope of threat scenarios is limited.

Hence, [4] further proposed the concept of a neighbourhood watch-inspired in-network security. It assumes that an optimal selection of security means properly distributed throughout wirelessly communicating resource-constraint embedded devices, suitable to collaborate increases the threat coverage while keeping the individual security tasks load per node manageable. Taking into account the cooperation of security means, ADRTs are further extended by a cooperative component to ''Cooperation-based Attack Defence Resource Trees'' in [4]. Collaborative security schemes coordinate nodes for more advanced threat detection and mitigation. When referring to either of those two terms, we encompass the broader set including both cooperative and collaborative security approaches. Existing collaborative security frameworks for pervasive systems like WSNs are specialised to very specific system constraints, sizes, network topologies, protocols and threat scenarios. While those frameworks [5], [6], [7] are capable to offer a high degree of security and an increased threat coverage, they are not applicable to a wide range of WSNs. Those frameworks go beyond the scope of collaborative distributed intrusion detection systems (IDSs) [8], [9], [10] by integrating concepts of intrusion detection, prevention and complex communication strategies. It follows that each new WSN requires the development of a new framework, and this is preceded by corresponding research work. To achieve timely, cost-efficient, adaptable and reusable cooperative security solutions for a wide array of WSNs a different approach is necessary. Based on system properties and constraints, the area of application, security and lifetime requirements a selection of components has to be determined. Determining a proper security configuration for given requirements and constraints can be set up as a design space exploration (DSE). There has been a lot of research regarding DSE models for embedded systems including WSNs focussing i.a. on aspects like security, safety, longevity and network topology (with regards to the placement of nodes) [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. Those concepts attempt to determine optimal configurations of components and subcomponents taking into account their interplay of a system with regards to a set of requirements and constraints. Research in the domain of WSN security is constrained in terms of exploring the intricate interrelationships within the design space, particularly when it involves diverse forms of interaction and collaboration arising from hierarchical or clustered network structures.

The design of WSNs encompasses a wide array of factors represented by variables. For the practicality of DSE for the configuration of different aspects of WSNs, it is crucial to narrow down the variables to a computationally feasible subset. Resulting design spaces are expected to encompass interdependent parameters. Moreover, exploration models often contain non-linear non-convex constraints and objective functions. Conflicting objectives such as longevity and highest possible security standards can be handled with a multi-objective optimisation resulting in a Pareto-efficient solution space. In order to enhance DSE approaches by incorporating the interplay of security means and various operational scopes (such as clusters or hierarchical structures), the optimisation models often exhibit a multitude of variables and continue to exacerbate the difficulty that comes with non-convex and non-linear characteristics. Therefore, only a limited number of aspects can be considered in the optimisation process to still achieve a satisfying solution.

In this publication, we propose static distribution concepts of fix numbers of security mean types intended to cooperate by partitioning the WSN accordingly. The approach intends to distribute different security means in the WSN with the objective to ensure the availability within the range of each node. It provides a generic solution for distributed/cooperative security configurations in large-scale static homogeneous WSNs. The availability of security means in proximity of each node is a prerequisite for the neighbourhood watch inspired security framework. It ensures short communication paths between nodes contributing with different security means to common security requirements. The partition concepts we are going to introduce offer a high availability of different security mean types in proximity of nodes in the WSN. Such a partitioning facilitates multiple associations per node, well-suited for in-network cooperation and despite a static placement of nodes and static distribution of security means a flexible load balancing. Therefore,

it provides a generic solution suitable for a cooperative approach of a neighbourhood watch inspired security concept achieving timely, reliable, and energy-efficient collaborative threat prevention, detection and handling. To create a DSE approach that allows requirement-based security configurations for WSNs (e.g. by adapting existing concepts [18]), our partitioning concept builds a foundation that improves the computability by limiting the number of model variables.

In order to highlight the limitations of a concept that statically distributes security means in a large-scale WSN, we make a number of assumptions. A fundamental premise for the success of a framework employing static security mean distributions is the assumption that an attacker possesses no insider-level knowledge about the distribution. Further, we assume that a trusted communication between sensor nodes has been established and as already mentioned, that the WSN is static (immobile nodes). There are three scenarios of security means distributions we are going to evaluate: a single security mean per node, a fixed number of security means per node, a load-based distribution of a variable number of security means per node. For the latter one, it is necessary to pinpoint a common resource capacity per node for all nodes available for security tasks and a resource requirements for all considered security means. To ascertain the distributions, we model the WSNs as undirected graphs wherein the nodes symbolise sensor nodes. The edges of the graphs indicate the connectivity between sensor nodes within transmission range of the underlying WSN. In order to distribute the security means, we determine $0-1$ LPs to compute suitable optimal graph partitions. Optimal with regards to our model and the defined objective function. The $0-1$ LPs fall into the complexity class of non-deterministic polynomial time (NP) hard problems [22]. Therefore, it is imperative to empirically assess whether an optimal solution falls within the feasible and efficiently computable limits of our input sizes for numerous WSNs with realistic topologies and node quantities. Network sizes of WSNs with 20 up to 300 nodes have been evaluated. The graphs representing the WSNs for the evaluation process have been generated as random λ-precision unit disk graphs (UDGs). A UDG is an undirected geometric graph in which each node has a fixed position in euclidean space and two nodes have a common edge if their distance is below a fixed threshold (transmission range) common for all nodes. A λ-precision graph is a geometric graph in which all pairs of nodes are at least λ apart. To generate desired graph topologies, we provide a table of generator seeds for combinations of node numbers, desired average node degrees and covered generation plane space. Seeds are the input values for the generator that are likely to result in random graphs with desired properties. The average node degree is the arithmetic mean of edges connected to each node for all nodes in a graph. The generation plane is in our context a unit square in which the nodes of our random graphs are distributed. The generator is written in Python and utilises the NetworkX library [23] to some degree. It allows to create graphs with an even degree distribution and a low variance of the local cluster coefficient controllable via λ. The local cluster coefficient is a measure indicating the connectivity of the neighbourhood of a node. The generator allows further manipulations of graph properties like enabling to enforce a desired average node degree and receiving connected bridge-free graphs. The $0-1$ LPs have been evaluated using Python with Pyomo [24] and Gurobi [25] to model and solve the linear optimisation problems partitioning the graphs for an optimal distribution of security means.

Cooperation of nodes increases their load. In a cooperation a node has to providing different services and handle requests. Such communication overhead increase further if a larger number of nodes direct their requests to a single node. An equal distribution of security means across the WSN balances the load and increases the availability of neighbouring nodes offering specific services. The necessity for load balancing is caused by imbalances as result of topologically conditioned unequal distributions and routing changes due to various reasons. Due to our partitioning scheme a certain likeliness of alternative nodes providing the same service or accomplishing the same task in proximity of a node is given. The association of nodes and their cooperation partners will not be taken into account when determining the distribution. Rather it will be dynamically determined by the nodes themselves.

In Section II, we acquaint the reader with mathematical terminology and definitions essential for comprehending the paper. Section III delves into the current state of distributed security solutions, dominating sets, domatic partitions, and graph generators specifically tailored for large-scale static homogeneous WSNs. Following that, in Sections IV and V, we illustrate the concept of graph partitioning within the context of the neighbourhood watch inspired security scheme introduced in [4]. Subsequently, in Section VI, we introduce a λ-precision UDG generator designed for large-scale static homogeneous WSNs. In Section VII, we familiarise the reader with the experimental setup used to evaluate the feasibility of the proposed graph partitioning concepts, which have been formulated as $0-1$ linear programs (LPs) and computed on the λ-precision UDGs generated by our novel graph generator. Finally, we present and analyse the test results in Section VIII and draw conclusions regarding various accomplishments of our paper in Section X.

II. BACKGROUND

We introduce mathematical terms and definitions related to graph theory and mathematical optimisation as well as terms necessary for the empirical evaluation.

A. CARDINALITY OF SETS

The cardinality of a set indicates the number of elements a set contains notated as follows $|\{\cdot\}|$.

B. UNDIRECTED GRAPH

An undirected irreflexive graph $G = (V, E)$ is defined as a finite set of nodes V and a set of edges:

$$E \subseteq \{\{v, w\} | v, w \in V \wedge v \neq w\} \tag{1}$$

Throughout this work, we exclusively utilise undirected and irreflexive graphs.

C. SUBGRAPH

A subgraph of an undirected graph $G = (V, E)$ is defined as $SG = (V', E')$ with $V' \subseteq V$ and $E' \subseteq E$ with $\forall \{v, w\} \in E' : v, w \in V'$.

D. CONNECTED GRAPH

An undirected graph is connected when there are no two nodes in the graph without a path.

E. CONNECTED COMPONENT

In an undirected graph, a connected component is a connected subgraph that is not part of any larger connected subgraph.

F. BRIDGE

In an undirected graph consisting of $c \in \mathbb{N}_{>0}$ connected components, a bridge is an edge, whose absence decomposes it into $c + 1$ connected components.

G. BRIDGE PATH

In an undirected graph $G = (V, E)$, there is a bridge path between nodes u and v iff there is a unique cycle-free path P exclusively composed by a sequence of bridges over a subset of nodes from $V \setminus \{u, v\}$ connecting u with v in which all contained nodes except u and v have a node degree of two and it does not exist any longer path Q with the same properties containing P.

H. GEOMETRIC GRAPH

A geometric graph is an undirected graph in a d-dimensional metric space $[0, 1)^d$ and edges are added based on their pairwise distance r_{tr} (transmission range) determined by a defined distance function. The distance r_{tr} in a geometric graph is fix for all nodes and node pairs of the graph. Throughout this work, we always refer to this distance as r_{tr}.

I. RANDOM GEOMETRIC GRAPH

A random geometric graph (RGG) is a geometric graph in which nodes are placed randomly.

J. UNIT DISK GRAPH

A unit disk graph (UDG) is a geometric graph in a two-dimensional euclidean space with an euclidean distance metric applied to them.

K. λ-PRECISION GRAPH

A λ-precision graph is a geometric graph in which the minimal distance between each pair of nodes is at least λ.

L. NEIGHBOURSHIP FUNCTION

We define the neighbourship of a node v in an undirected graph $G = (V, E)$ with $v, w \in V$ as follows:

$$N[v] := \{w | \{v, w\} \in E\} \cup \{v\} \tag{2}$$

M. NODE DEGREE

A node degree of a node $v \in V$ of an undirected graph $G = (V, E)$ is the number of edges of the graph the node participates in:

$$\deg[v] = |\{e | \forall e \in E : v \in e\}| \tag{3}$$

N. AVERAGE NODE DEGREE

The average node degree of an undirected graph $G = (V, E)$ is the arithmetic mean of the node degree of each node in the graph relative to the number of nodes as follows:

$$\deg_{\text{avg}}[G] = \sum_{v \in V} \frac{\deg[v]}{|V|} \tag{4}$$

O. LOCAL CLUSTER COEFFICIENT

The local cluster coefficient is a measure indicating how well the neighbourhood of a node is connected. Following [26], the local clustering coefficient for undirected graphs is defined as:

$$C[v] = \frac{2 \cdot |\{e | e \in E \wedge e = \{w, u\} \wedge w, u \in N[v] \backslash \{v\}\}|}{|N[v] \backslash \{v\}| \cdot (|N[v] \backslash \{v\}| - 1)} \tag{5}$$

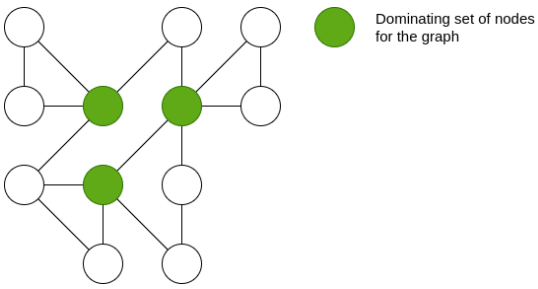

FIGURE 1. The set of green nodes is a dominating set in the given graph.

P. VARIANCE OF THE NODE DEGREE DISTRIBUTION

We define the variance of the node degree distribution for a RGG $G = (V, E)$ as follows:

$$\text{Var}_{\text{deg}}[G] = \sum_{v \in V} \frac{\left(\deg[v] - \deg_{\text{avg}}[G]\right)^2}{|V|} \tag{6}$$

Q. LINEAR PROGRAM

A LP or linear optimisation is a method which tries to optimise a mathematical model based on linear relationships with the following standard form:

$$\begin{aligned} \max \quad & \mathbf{c}^T \cdot \mathbf{x} \} \text{objective function} \\ \text{s.t.} \quad & \left.\begin{aligned} \mathbf{A} \cdot \mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}\right\} \text{constraints} \end{aligned} \tag{7}$$

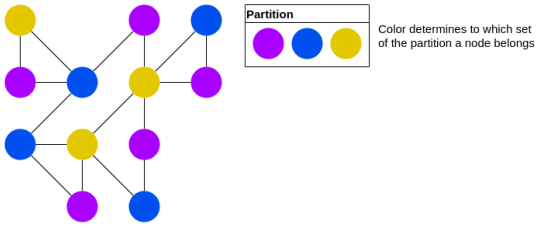

FIGURE 2. The example shows a graph in which the nodes are mapped to a domatic partition consisting of three dominating sets.

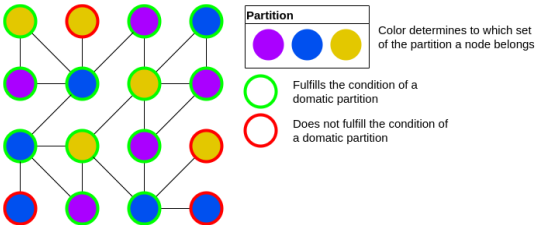

FIGURE 3. The partition of the graph is not a domatic partition because there exist nodes in at least one dominating set of the partition that has no neighbourship with at least one node of each of the other dominating sets of the partition.

with the vectors **b** and **c** and with a matrix **A** that have to be known to the problem. The vector **x** contains the variables whose values have been optimised. Linear programs are called in this way because the objective function as well as the equality and inequality constraints are linear.

In a 0-1 linear program, the components of the vector of variables **x** is bound to $\{0, 1\}$. For integer linear programming as well as 0-1 linear programming without objective function it is known that they belong to the class of NP complete problems [22]. With objective function, their complexity is not bound to an upper limit and the problems are therefore considered to be NP hard. However, experience has shown that 0-1 linear programs perform better than integer linear problems even when they rely on significantly more variables.

R. DOMINATING SET

A dominating set D is a set of nodes of an undirected graph $G = (V, E)$ for which holds:

$$D \subseteq V \text{ whereas } \forall v \in V : \ D \cap N[v] \neq \emptyset \tag{8}$$

In Fig. 1, an example for a dominating set of nodes for a graph is given. As the definition implies, every node in this graph is either part of the dominating set or adjacent to a node from the set.

S. DOMATIC PARTITION

A domatic partition $\mathbb{D}(G)$ is a decomposition of nodes V of a graph $G = (V, E)$ into disjoint dominating sets with:

$$\bigcup_{D \in \mathbb{D}} D = V \ \wedge \bigcup_{\substack{D_1, D_2 \in \mathbb{D} \\ D_1 \neq D_2}} D_1 \cap D_2 = \emptyset \tag{9}$$

A domatic partition can also be defined using the neighbourship term of graphs. Then, a set of dominating sets in G

$$\mathbb{D}(G) = \{D | D \subseteq V, \ \forall v \in V : \ D \cap N[v] \neq \emptyset\} \tag{10}$$

is a domatic partition iff Equation (9) holds. We define a n-domatic partition as a partition of G into n disjoint dominating sets. An example can be seen in Fig. 2. When referring to a node satisfying the properties of a domatic partition, the set consisting of the node itself and its adjacent nodes have to have a non-empty intersection with all sets of the domatic partition:

$$v \in V : \forall D \in \mathbb{D} : N[v] \cap D \neq \emptyset \tag{11}$$

In Fig. 3, we provide an example for a partition in which a number of nodes does not satisfy the definition of a domatic partition. A domatic partition of a WSN ensures that each sensor node has at least one direct neighbour of each dominating set of the partition or is a member of the set. The size of the domatic partition is given by the number of different security means that have been applied to the network. All sensor nodes in the same dominating set of the partition implement the same security mean. In case all nodes in the same dominating set implement the same security mean, we achieve a distribution of security means in which all nodes either implement a security mean or are directly adjacent to a node that does. Therefore, the set of sensor nodes and its neighbours have no empty intersection with any of the sets of the partition. Hence, all security means applied to the WSN are present in the neighbourship of each node.

III. RELATED WORKS

In [4] a neighbourhood watch inspired concept for a cooperative distributed static security framework has been introduced. The objective of distributed security solutions is to cover a wider range of threat scenarios in a large-scale static homogeneous WSN. This section is divided into three parts. The first subsection explores research work towards distributed security solutions for WSNs. The second part evaluates existing research regarding dominating sets and domatic partitions. In the third subsection, we discuss graph generators as model for WSNs.

A. DISTRIBUTED SECURITY SOLUTIONS FOR WSNS

A number of publications propose cooperating security means for WSNs that provide mutual protection. The paper [7] introduces a security framework concept for static heterogeneous WSNs. Each set of nodes is assigned to a cluster head (CH) (a more powerful sensor node). Nodes running IDSs notify their associated CH about identified threats or CH are informed by CHs in close proximity. If a threat is detected and communicated to a CH it will be propagated to other CHs in the WSN. Clusters that consider the threat imminent for their own cluster react by redistributing security means on associated nodes based on the threat scenario. Therefore, the CH holds a set of security means which can be implemented on or revoked from the

sensor nodes. This allows a dynamic threat evaluation and flexible reactions. The proposed security framework for static heterogeneous WSNs [7] has been tested in a simulation including a network with 2000 regular nodes and 10 gateway nodes. The energy consumption was only evaluated for regular nodes, for CHs it was considered unlimited. To test the simulated sensor network, seven abstract attack patterns have been implemented and for each scenario 200 sequential attacks have been executed. The authors of [7] evaluated the simulation based on two metrics, the success rate (number of nodes alive after an attack) and the energy consumption (average percentage of energy of all surviving nodes). For comparison, WSNs implementing one security mean or multiple static security scheme frameworks have been used. The results show that the proposed framework provides the highest success rate while also consuming the highest amount of energy in each simulation. The contribution [6] presents a security framework that has been developed and implemented on a real WSN based on [7]. The test of the resulting security framework has been executed on a rather small WSN with only six nodes. One node acted as the CH which communicated directly to a base station. The authors assumed two kinds of attack scenarios. One in which only a single kind of attack is started on the WSN and one in which two kinds of attack are launched in succession. The results show that the WSN implementing the framework was able to recover from all tested attacks even when they have been executed successively. The energy consumption has not been considered. Both papers propose a security solution with distributed security means for heterogeneous WSNs with rather powerful CHs. The heterogeneity of the WSN is not utilised by the frameworks to which the proposed one is compared. The according statement from [6] has very limited meaningfulness due to their limitation in executed test scenarios, measured parameters and small network size.

Another cooperative security solution is proposed in [5]. The paper proposes a concept to efficiently combine in-network intrusion detection and concealed data aggregation. To do so, it utilises clusters. In each cluster a CH is elected. A CH fulfils multiple roles: it collects the data from the nodes in its cluster, runs intrusion detection on the data, aggregates them and finally forwards them hop-by-hop to a BS.

IDSs are distinguishable by many criteria [27]. Whether the intrusion detection is executed online (in-network), offline (on BSs, external System/Server) or hybrid states if certain tasks are performed on nodes or on a centralised base station affecting whether a timely reaction is possible. The choice of whether to execute intrusion detection online (within the network), offline (on BSs or external systems/servers), or in a hybrid manner depends on whether certain tasks are performed on nodes or BSs, which in turn affects the feasibility of timely response. Based on their detection strategy, IDSs can be classified as anomaly-based, signature-based or hybrid. Reference [28] introduces a distributed neighbour based IDS. Each node monitors a set of neighbouring nodes by storing their attribute vectors sending warnings to other nodes in case a malicious anomaly has been recognised. If a number of nodes communicate the same anomaly, the network acts accordingly. There are distinctions based on the intrusion and intruder type and so on. A comprehensive overview of the classification IDSs is provided in [27]. The publication [28] is built upon [29] which describes a similar distributed approach to detect misbehaving sensor nodes in local areas by comparing their behaviour vector with vectors from other direct neighbours. Another popular concept is LiDeA [30]. Nodes that detect irregularities in the network notify other close-by nodes to establish a vote. Notified nodes decide about the handling of the irregularity as well as the suspicious node. Therefore, nodes provide a number of modules that can be activated on demand and based on received information by broadcasting neighbours. Whether a node is assumed to be an intruder is determined based on a majority vote. In [31] a lightweight, energy-efficient IDS using mobile agents is introduced. These agents are sent through the network as regular messages and are temporally installed on addressed nodes. Therefore, IDSs are dynamically distributed and instead of running on nodes permanently. While agents are run by nodes, they collect information about their energy consumption and initiate warnings to the network if noticeable deviations are recognised. The transmission and installation of changing IDSs on nodes themselves, especially when executed on large-scale static homogeneous WSNs, significantly impacts the energy consumption. Hence, it is assumed inappropriate for our subject of research. Additionally, a IDS is intended not to introduce weaknesses into a WSN. The distribution of IDSs requires by itself a increased level of trust in the communication and sensor nodes. However, many collaborative and distributed IDSs provide a promising basis to design a cooperative security framework integrating further components to collaborate.

Many of the concepts of cooperative/collaborative IDSs and security frameworks for WSNs are tailored to distinct properties, targeted application areas and satisfy specific security requirements. One attempt to create a more universally applicable security concept is realisable using DSE. To do so, we can fall back on many existing concepts [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] and adapt them accordingly. Therefore, a design space contains a number of components that can be combined to create a security framework for specific security requirements, lifetime expectancies and hardware constraints. This further necessitates metrics to assess the contribution of components to given requirements. The proposed ‘‘collaborative attack defence resource trees’’ in [4] is a concept displaying countermeasures and the attacks they prevent as well as the resulting resource costs to the WSN. Further, the concept considers the weighting of the frequency of appearances of security means in a WSN based on likeliness and severity

of attacks to the system. For these concepts it is crucial to achieve an optimal distribution ensuring proximity between different types of security means and the corresponding nodes.

B. DOMINATING SETS AND DOMATIC PARTITIONS

To determine suitable static distributions of a fix number of security mean types intended to cooperate, local proximity is a key factor. The concept of dominating sets and domatic partitions (alternatively *f* all *k*-colouring [32][1]) is well suited. In a dominating set, each node is either adjacent to a node of the set or included in it. If sets represent security mean types, such a partition ensures the local proximity in a network. Hence, a security mean type is either available on a selected node or a neighbouring node. A domatic partition of a graph is the partitioning of it into disjoint dominating sets. If for a graph and a given number of security means such a partition exists, local proximity is ensured. Reference [33] states that the domatic partition problem that asks whether the nodes of a graph can be partitioned into $k \in \mathbb{N}_{\geq 3}$ dominating subsets is NP complete. Known applications of dominating sets exist in the field of wakeup scheduling for WSNs [34], [35], [36], [37], [38], [39]. However, in wakeup scheduling applications, dominating sets do not need to be disjoint. The major concern in energy-saving wakeup scheduling schemes is that at least one node in a neighbourhood of each node has to be kept awake to ensure that it can wakeup surrounding nodes. On the contrary, our applications require disjoint partitions into dominating sets. The term fractional domatic partition was introduced in [40]. This algorithm however determines a number of non-disjoint dominating sets. Conversely, we attempt to determine a fixed size partition of disjoint sets approaching dominating sets as far as possible. Therefore, we approach the definition based on desired criteria, as we will introduce in the following sections.

In [34], an approximation algorithm which tries to maximise the number of fractional domatic partitions in a graph to efficiently sleep schedule nodes is shown. Furthermore, there exist a multitude of publications towards the domatic number and domatic partition problem with regards to different approximations and solution for specialised graph types providing lower and upper bound assumptions of their computational complexity. In [41] a polynomial approximation algorithm estimating the lower and upper bounds of the domatic number on general graphs is presented. Additionally, [41] determines a greedy approximation algorithm for domatic partitions of graphs. The algorithm computes as many small disjoint dominating sets as possible to receive a partition of fixed size. Other attempts achieving more precise bounds for the domatic number and domatic partition problem have been executed on general graphs [42], [43] as well as special types of graphs, e.g. interval graphs [44] or RGGs [45]. Reference [46] determines an approximation algorithm for domatic partitions on UDGs. The survey [47] discusses and summarises a large number of research results and solutions towards different dominating set problems and compares the performances and properties of different algorithms proposed.

We intend to calculate our static distributions of security means (partitioning schemes) analytically. The security means are distributed on nodes and not exchanged during runtime. Therefore, an optimal distribution is a key factor for the overall performance of the security framework. Furthermore, we have different requirements towards the partitioning compared to the distributions examined in sleep scheduling applications.

C. GENERATORS FOR GRAPH MODELS OF WSNS

A lot of research is done regarding the generation of graphs as model for different types of networks. One of the first models for generating random graphs as network model is the Erdős-Rényi model [48] expressed by $G(n, p)$. It is a popular way to construct Erdős-Rényi graphs. In this model, n labelled nodes are connected randomly. For all pairs of nodes, an edge is included with the probability p. Other popular models for random graph generators are the Barabási-Albert model [49] and the Watts-Strogatz model [26]. The Barabási-Albert model aims to create scale-free graphs as network models. Therefore, the degree distribution in the resulting graphs follows a power law. The Watts-Strogats model generates graphs with small-world properties which are characterised by a high clustering coefficient and a low average shortest path length between nodes. Reference [50] reasons why RGGs are well suited as graph topology model for WSNs. In [51] the author first mentions similar graph models called "Random Plane Networks" as representation of wireless networks. The resulting graphs are closely related to UDGs. Those type of graphs are most often the model of choice to represent WSNs. In [52] a model to generate WSNs that have a high probability to be connected as model for WSNs and ad hoc networks is introduced. To achieve the property connected with a high probability, the authors rely on a scheme that they call the proximity algorithm (PA). The PA places nodes iteratively on a finite plane. The first node is placed randomly within the generation plane. The following nodes are placed within radius r of the previously placed nodes. Even so, r is usually chosen larger than the distance in which two nodes are connected in a UDG, the likeliness of receiving a connected UDG using the PA increases significantly. A major downside of this approach is the likeliness for nodes in the graph to be highly clustered together. One of the most popular concepts for the generation of random graphs as model for wireless ad hoc, actuator and wireless sensor networks has been published in [53]. The publication introduces two types of algorithms to generate random UDGs. Centre node based algorithms are one type and acceptance/rejection based algorithms are the other. With

[1] A graph colouring problem that determines whether a graph can be coloured with n colours so that in each node's neighbourhood all colours are present.

centre node based algorithms, a node out of the previously placed nodes is chosen (centre) and the new node is placed in reach of the chosen centre. The paper presents four different algorithms. Each of them introduces different centre choosing strategies. The second type, acceptance/rejection based algorithms, works by iteratively choosing random node locations. The selected location is accepted or rejected based on given constraints. The authors propose three different algorithms to apply the acceptance/rejection based concept. The resulting graphs are called constrained connected random UDGs (C-CRUG). The term constrained reflects the circumstance that the placement is not completely random but constrained by the node positions of previously placed nodes. Moreover, the term connected means that the final result will only be accepted if the graph is connected.

In [53] the authors relied on three different constraints. The proximity constraint which is closely related to the PA by [52]. It ensures that each node is placed close to previously placed nodes increasing the likeliness for the resulting graph to be connected. Each node successive to the first node has to be placed within an approximated radius of previously placed nodes. The radius is estimated based on further desired graph properties. As with the PA, the radius constraining the node placement increases the likeliness of receiving islands of strongly clustered nodes. The actual radius used to decide whether two nodes in the graph are connected is determined as the $\frac{N \cdot d_{\text{avg}}}{2}$th shortest edge with N the number of nodes in the graph and d_{avg} the average node degree. Therefore, resulting graphs are not guaranteed to be connected. The second constraint used in [53] is the maximum degree constraint. It accepts the placement of a new node only if it does not increase the degree of the already placed nodes above a given maximum value. The third and last constraint was named the coverage constraint. With the coverage constraint, a new node location is only accepted if it extends the area that will be covered by the nodes of the graph sufficiently. Regarding the proximity constraint [53], a minimal distance in between nodes equal to the λ in λ-precision graphs is considered. But the paper merely employs the distance to avoid that two nodes will be placed on the same coordinate instead of utilising λ for a better spatial node distribution. Hence, proposed centre node based algorithms from [53] are:

Minimum Degree Proximity Algorithm (MIN-DPA): It distributes nodes more uniformly while still maintaining connectivity. The first node is placed completely at random. Succeeding nodes are placed in the range of previously placed nodes with the lowest degree. In case there are multiple equally suitable contenders, all nodes get assigned a weighting scheme based on further criteria.

Clustered Minimum Degree Proximity Algorithm (C-MIN-DPA): Instead of distributing homogeneous nodes, this algorithm starts to distribute access points (APs). They are assumed to be connected first. The nodes will then be placed in close proximity to the APs, so they are connected to them.

Weighted Proximity Algorithm (WPA): This algorithm is similar to MIN-DPA but it considers all previously placed nodes as centres instead of just the ones with the lowest degree. To randomly select nodes, all nodes associated with a weight relative to their node degree. Therefore, nodes with a higher degree receive a smaller weight.

Eligible Proximity Algorithm (EPA): The nodes and their transmission ranges that serve as possible candidates for the location of the next node are selected by a given upper bound of the node degree. If the estimated node degree is larger than a given upper bound the placement of nodes is done according to WPA.

Proposed acceptance/rejection based algorithms are:

Maximum Degree Proximity Algorithm (MAX-DPA): The algorithm sets a maximum degree constraint per node. A random node position is generated uniformly. If the node satisfies the proximity constraint as well as the maximum degree constraint the new position is accepted.

Coverage Algorithm 1 (CA1) The first node is placed completely at random. Subsequent nodes, choose a random coordinate. Their position is validated by a coverage constraint checking if the selected region is already sufficiently covered by previously placed nodes.

Coverage Algorithm 2 (CA2) CA2 works similar to CA1 but with a stricter coverage constraint. The covered portion of the sensing area for a new node location is explicitly computed with regard of the previously placed nodes. If the portion of the sensing area gained by the new node location is below a given threshold, the node location is rejected.

Our graph generator follows a different approach. We distribute nodes uniformly at random only constrained by a generation plane and a minimal distance in between nodes called λ-precision. Instead of using λ to prevent nodes from occupying the same spot as in [53], we apply it to improve their spatial distribution and control a number of graph properties. When distributing a number of sensor nodes with fixed sensing range given by radius r_{sensing}, it is often of interest to maximise the monitored area.

Therefore, λ should usually be set between the radius r_{sensing}, a single sensor nodes sensing range and its transmission range r_{tr} in which a sensor node is able to communicate. Choosing λ larger than the transmission range prevents nodes from communicating. The rings resulting from the two radii λ and r_{tr} limit the maximum node degree of each node. The choice of λ and r_{tr} relative to each other and relative to the generation plane determines the probability that a randomly generated graph is connected.

In general, the proposed generator is also suitable to be further developed into a topology generator allowing distributions of sensor nodes in target environments. To extend our concept, we have to take into account the topological shape of landscapes as well as the varying transmission ranges based on different environmental conditions including various obstacles. Even so, we distribute nodes in a unit square, the generation plane can have any shape. To accommodate diverse landscapes and their environmental conditions, we can establish a connection between the λ-precision and the topological characteristics of specific areas, thereby enabling

a higher concentration of sensor nodes in those regions. Additionally, we demonstrate that even with the flexibility in node placement, it is possible to precisely adjust the local cluster coefficient and average node degree to meet specific requirements.

IV. DISTRIBUTION OF SECURITY MEANS

The neighbourhood watch inspired security concept [4] intends to distribute different security means in a WSN enabling an increased threat coverage while keeping the energy consumption at bay. In order to attain such a distribution, sensor nodes must establish a mutually beneficial cooperation among the applied security means. Moreover, it is evident that nodes cannot continuously operate their security mean for neighbouring nodes as a service while ensuring their own longevity. Instead, they can execute security means in specialised periodic patterns to detect malicious activity and subsequently process security violations. The detection is solvable with a cooperative multilayer IDS approach, while the intrusion prevention pre-emptive and reactive requires further tools. Suitable candidates are lightweight trust-and-reputation systems [54], node isolation schemes [55], resilient recovery techniques for compromised nodes [56], [57], lightweight encryption schemes [58]. For the realisation of the security framework, following three assumptions have to be met:

- trusted communication between sensor nodes has been established
- WSN is static (nodes are immobile)
- attacker has no knowledge regarding the distribution of security means

We consider static distributions of security means. Meaning, sensor node carry pre-installed security means and are incapable to exchange or rotate their security mean. In the considered WSNs, we intend to distribute n different types of security means. Hence, we contemplate it mandatory to ensure the availability of each type of security mean in the neighbourhood of each node if possible. Therefore, nodes have access to all security mean types applied to the network. A distribution of this kind is achievable in case each security mean type is either implemented on the observed node or on one of its neighbours. Therefore, we aim to ensure that the set of all nodes implementing the same security mean type in union with the set of all neighbours of those nodes results in a set containing all nodes of the network. Such a set is called a dominating set in graph theory. Considering the set of nodes implementing the same security mean as a set for all security means, we get n disjoint sets of nodes. Those sets are called dominating sets in graph theory. A partition of n disjoint dominating sets of nodes of a graph is called a domatic partition. The number of applied security mean types distributed in a network implies the number of necessary dominating sets. The maximum number of disjoint dominating sets per graph is called domatic number n. Choosing n larger than the domatic number of a graph, makes a partitioning in to n disjoint dominating sets impossible. Therefore, we introduce the term n-soft domatic partition. An n-soft domatic partition attempts to compute a best possible fit as compromise with regards to the model parameters. Another attempt to achieve an improved distribution of security means is the assumption to soften the neighbourhood term. So far, we are considering direct neighbourhoods (one-hop). Assuming multi-hop neighbourhoods, we are more likely to find an optimal partitioning as we later elaborate. We also discuss fix and workload-based distributions of multiple security means per node. Those approaches have currently limited practical applicability but can become relevant in the future. The partition scheme, introduced in this work, we name maximal/optimal n-soft domatic partition. Primarily, we choose to focus on distributing one security mean per node are the resource limitations and longevity of nodes. For this reason, we also focus on the one-hop neighbourhood in our analysis. A one-hop neighbourhood significantly limits the number of nodes depending on a security mean type and therefore inflicting an increased load to it. Viable alternative strategies are to consider multi-hop neighbourhoods allowing a more flexible rebalancing of node affiliations.

V. OPTIMAL AND MAXIMAL *N*-SOFT DOMATIC PARTITIONS

An n-soft domatic partition describes the partitioning of a graph into n disjoint sets. While a domatic partition of size n is restricted to graphs with a domatic number greater-equal to n, an n-soft domatic partition is computable for graphs with a domatic number lower than n. We define two types of n-soft domatic partitions. Both types use different error terms to define either an optimal or a maximal n-soft domatic partition by minimising its respective error. An n-soft domatic partition of size n with nodes V of a graph $G = (V, E)$ into disjoint sets of nodes $D_1, \ldots, D_n$ is defined as:

$$\mathbb{D}(G) = \{D_i \subseteq V \mid i = 1, \ldots, n \wedge \bigcup_{D \in \mathbb{D}} D = V \wedge \bigcup_{\substack{D_1, D_2 \in \mathbb{D} \\ D_1 \neq D_2}} D_1 \cap D_2 = \emptyset\} \tag{12}$$

The definition of an n-soft domatic partition coincides with the definition of a regular partition of size n. After introduction of the terms optimal and maximal as additional conditions to the n-soft domatic partition, we define more specialised mathematical terms.

A. OPTIMAL N-SOFT DOMATIC PARTITION

An n-soft domatic partition is called optimal iff missing coverages $e_{\text{miss\_cov}}$ from Equation (13) is minimal. In consequence, the optimal n-soft domatic partition minimises the sum of missing coverages over all nodes.

B. MAXIMAL N-SOFT DOMATIC PARTITION

An n-soft domatic partition is maximal iff the number of incompletely covered nodes $e_{\text{inc\_nodes}}$ defined in

Equation (14) is minimal. Therefore, it is irrelevant whether $N[v]$ of a node $v \in V$ of graph $G = (V, E)$ has one or multiple non-empty intersections with any set $D \in \mathbb{D}$.

We use the newly introduced terms to determine a distribution of n security mean types on sensor nodes of a WSN with a domatic number smaller than n. A maximal n-soft domatic partition ensures that the maximum number of nodes and its neighbourhood contains the full set of n security means. The optimal n-soft domatic partition guarantees that the number of missing coverages in a WSN is minimal. Hence, ensuring the sum of the absence of the number of security mean types in the inclusive neighbourhood of all nodes is minimal.

C. ERROR TERMS IN SOFT DOMATIC PARTITIONS

The definition of optimal and maximal n-soft domatic partitions is based on two error terms. Those will be evaluated in our empirical analysis. The missing coverages are defined as the sum of the n security mean types minus the security mean types present in the neighbourhood of a node $N[v]$ in a graph $G = (V, E)$ over all nodes $v \in V$:

$$e_{\text{miss\_cov}} = \sum_{v \in V} (n - |\{D | \forall u \in N[v] : \exists D \in \mathbb{D} : D \cap u \neq \emptyset\}|) \quad (13)$$

with the set of nodes utilising the same security mean type creating a partition D in the set of partitions $\mathbb{D}$.

In Fig. 3, we depict as example a graph with nodes of three colours magenta, blue, yellow. Each of those colours represents a set of nodes D within a partition $\mathbb{D}$ of the given graph. All four nodes marked with a red ring contribute to the number of missing coverages. A node is fully covered if its inclusive neighbourhood contains nodes of all colours. In Fig. 3, the number of missing coverages $e_{\text{miss\_cov}}$ is 6. There are four incompletely covered nodes surrounded by a red ring. The blue node at the lower left corner of the graph lacks the coverage of a yellow and a magenta security mean in its neighbourhood. So, its contribution to the coverage error is 2. The same holds for the blue node at the lower right corner of the graph. Here, two security means (yellow and magenta) are missing. The yellow node directly above has no access to the magenta security mean. Its coverage error is 1. Finally, the yellow node marked with a red ring in the top line of the graph misses the magenta security mean. Resulting in a coverage error of 1. In total, $e_{\text{miss\_cov}}$ results in $2 + 2 + 1 + 1 = 6$.

The second error term is named incompletely covered nodes. Counting the number of nodes $v \in V$ of $G = (V, E)$ for which the number of distinct security means in $N[v]$ is smaller than n:

$$e_{\text{inc\_nodes}} = \sum_{v \in V} f\,(n - |\{D | \forall u \in N[v] : \exists D \in \mathbb{D} : D \cap u \neq \emptyset\}|) \quad (14)$$

$$\text{with } f(x) = \begin{cases} 0, & x < 1 \\ 1, & x \geq 1 \end{cases} \quad (15)$$

Let us again illustrate an example by the graph in Fig. 3. The four nodes marked with a red ring are incompletely covered. Hence, they are missing one or several distinctly coloured nodes in their inclusive neighbourhood. In order to be completely covered by security means, a node needs to have access to all three colours (blue, magenta, yellow) within its direct neighbourhood. The error term $e_{\text{inc\_nodes}}$ identifies these nodes and sums up their occurrences. So, we obtain as result $e_{\text{inc\_nodes}} = 4$.

In the worst case, for a graph $G = (V, E)$ with V the set of nodes and E the set of edges is at most

$$\max_{e_{\text{miss\_cov}}} (G) = |V| \quad (16)$$

incompletely covered nodes and

$$\max_{e_{\text{inc\_nodes}}} (G) = (n - 1) \cdot |V| \quad (17)$$

errors for a partition of size n, since each node has to be in at least one of the sets of the partition. An example for a worst case is the instance in which all nodes of a graph host the same security mean while the total number of required security means is higher ($n > 1$).

1) DOMATIC PARTITION LP

To compute the domatic partition of size n of a given graph $G = (V, E)$, we conceptualise a $0-1$ LP without objective function. The LP returns either a feasible solution or terminate with the response that no feasible solution exists. In case a feasible solution exists, the assignments of the binary variables provide a feasible graph partitioning. Hence, the $0-1$ LP determines a domatic partition of size n.

To construct a $0-1$ LP, we need to define a number of variables and construct a set of constraints representing the properties of a domatic partition. We define the variables $x_i^v \in \{0, 1\}$ of the underlying $0-1$ LP. The upper index provides the identifier for the corresponding node $v \in V$ and the lower index links to the partition $i = 1, \ldots, n$. For each node $v \in V$, there are exactly n variables, one for each partition. A value 1 of a variable x_i^v associates the node v with the set i of the partition. Otherwise, the value 0 indicates that node v is absent from the partition.

The first set of constraints we introduce ensures that each node has to be included in exactly one dominating set of the domatic partition:

$$\forall v \in V : \sum_{i=1}^{n} x_i^v = 1 \quad (18)$$

Moreover, we formalise that each node is either part of a dominating set or adjacent to one:

$$\forall v \in V,\ \forall i \in \{1, \ldots, n\} : \sum_{w \in N[v]} x_i^w \geq 1 \quad (19)$$

Hence, for all dominating sets of a domatic partition the intersection with the set of adjacent neighbours $N[v]$ including the observed node v is not empty.

The final $0-1$ LP without objective function reads as follows:

$$\begin{aligned} &\forall v \in V: && \sum_{i=1}^{n} x_i^v = 1 \\ &\forall v \in V,\ \forall i \in \{1,\dots,n\}: && \sum_{w \in N[v]} x_i^w \geq 1 \\ &\forall v \in V,\ \forall i \in \{1,\dots,n\}: && x_i^v \in \{0,1\} \end{aligned} \tag{20}$$

It determines whether a graph can be partitioned into an n-domatic partition. As a result it provides a domatic partition of the graph as solution. Hence, the LP solves a satisfiability problem stating whether a given graph can be partitioned into n disjoint dominating sets.

We can extend the LP as proposed in the previous section by allowing each node to implement $k \in \mathbb{N}_{>0}$ different security means. To do so, it is only necessary to change the constraint from the Equation (18) to:

$$\forall v \in V: \sum_{i=1}^{n} x_i^v = k \tag{21}$$

In the context of WSNs, the resulting partitioning yields a distribution of n security mean types with k security mean types implemented per node v and all $v \in V: |N[v]| = n$ if one exists.

Furthermore, we can apply a variable number of security means per node based on an estimation of their respective resource costs. To do so, we apply fixed costs $m_i \in (0,1]$ to each security mean $i = 1,\dots,n$, a portion of the total available resources per node which w.l.o.g. is set to 1. As long as the available resources on a node are not exhausted, additional security means can be applied. The constraint from Equation (18) is modified as follows:

$$\forall v \in V: \sum_{i=1}^{n} m_i \cdot x_i^v = 1 \tag{22}$$

2) OPTIMAL/MAXIMAL N-SOFT DOMATIC PARTITION LPS

Based on the LPs for the satisfiability conditions of domatic partitions from the preceding section, we introduce LPs for optimal and maximal n-soft domatic partitions. At first, it is necessary to drop the constraints from Equation (19). The constraints ensure that each set of the partition is a dominating set. For maximal and optimal n-soft domatic partitions of graphs with n greater than their domatic number, no partitioning into n disjoint dominating sets exists. Instead, we introduce an objective function minimising either the number of missing coverages (Equation (13)) or the number of incompletely covered nodes (Equation (14)) for optimal and maximal n-soft domatic partitions.

We start with Equation (13) to minimise the missing coverages. Therefore, we transform the counting of missing coverages into a more applicable form for construction of partitions. The objective function uses the previously defined function f in Equation (15). All identifiers and variables such as x_i^v and n previously introduced in the Equations (18) to (20) of the preceding subsection keep their semantics. The LP to determine an optimal n-soft domatic partition then reads as follows:

$$\begin{aligned} \max\ & \sum_{v \in V} \sum_{i=1}^{n} f\left(\sum_{w \in N[v]} x_i^w\right) \\ \text{s.t. } & \forall v \in V: \sum_{i=1}^{n} x_i^v = 1 \\ & \forall v \in V:\ \forall v \in V,\ \forall i \in \{1,\dots,n\}: x_i^v \in \{0,1\} \end{aligned} \tag{23}$$

At first, we look at $\sum_{w \in N[v]} x_i^w$. The sum iterates over all $w \in N[v]$. It checks for each node v associated with set i whether a node of $N[v]$ is included in the set i of the partition. The result is passed on to the function f from Equation (15). The function indicates whether at least one member of $N[v]$ is linked to set i or no member of $N[v]$ is included in set i with the values 1 and 0 respectively. Hence, the appearance of more than one node in $N[v]$ included in the set i of the partition does not influence the optimisation result. The outer sums $\sum_{v \in V} \sum_{i=1}^{n} f\left(\sum_{w \in N[v]} x_i^w\right)$ ensure that the value is determined for all combinations of nodes $v \in V$ and sets i of the partition. By maximising the resulting value, we are minimising the number of missing coverages (Eq. (13)).

To compute the maximal n-soft domatic partition the objective function is adapted as follows:

$$\max \sum_{v \in V} f\left(n^{-1} \cdot \sum_{i=1}^{n} f\left(\sum_{w \in N[v]} x_i^w\right)\right) \tag{24}$$

The term $n^{-1} \cdot \sum_{i=1}^{n} f\left(\sum_{w \in N[v]} x_i^w\right)$ describes the portion of sets of the partition having at least one common member with the set $N[v]$. For the maximal n-soft domatic partition it only matters whether a node's neighbourhood $N[v]$ has common members with all sets of the partition. Hence, we map the result to 0 or 1 and maximise the sum of those values. The LP applying this objective function minimises the number of incompletely covered nodes (Eq. (14)) by maximising the number of fully covered nodes.

Linear solvers are not able to solve objective functions with case distinctions directly. So, it is necessary to replace them. Therefore, we reformulate the LP to fit the standard form introduced in Equation (7).

To do so, we introduce a set of auxiliary variables and additional constraints:

$$\begin{aligned} \max\ & \sum_{i=1}^{n} \sum_{v \in V} y_i^v \\ \text{s.t. } & \forall v \in V: \sum_{i=1}^{n} x_i^v = 1 \\ & \forall v \in V,\ \forall i \in \{1,\dots,n\}: y_i^v \leq \sum_{w \in N[v]} x_i^w \\ & \forall v \in V,\ \forall i \in \{1,\dots,n\}: x_i^v, y_i^v \in \{0,1\} \end{aligned} \tag{25}$$

The first new set of constraints $\forall v \in V,\ \forall i \in \{1,\dots,n\}: y_i^v \leq \sum_{w \in N[v]} x_i^w$ ensures that the auxiliary variable y_i^v is set to 1 if in $N[v]$ exists a node included in set i of the partition. Therefore, if there are multiple nodes of $N[v]$ in the set i of the partition it does not affect the outcome of our LP because y_i^v is a binary variable and cannot grow larger than 1. The resulting objective function maximises the sum of all y_i^v. Therefore, it replaces our auxiliary function f.

For the maximal n-soft domatic partition, we repeat the pattern applied to Equation (25) in similar fashion:

$$\begin{aligned} \max \quad & \sum_{v\in V} z^v \\ \text{s.t.} \quad & \forall v \in V: \quad \sum_{i=1}^{n} x_i^v = 1 \\ & \forall v \in V,\ \forall i \in \{1,\dots,n\}: \quad y_i^v \le \sum_{w\in N[v]} x_i^w \\ & \forall v \in V,\ \forall i \in \{1,\dots,n\}: \quad z^v \le y_i^v \\ & \forall v \in V,\ \forall i \in \{1,\dots,n\}: \quad x_i^v, y_i^v \in \{0,1\} \end{aligned} \tag{26}$$

Rather than aggregating the y_i^v as the number of sets of the partition that intersect non-empty with $N[v]$, we establish $z_v = 1$ under the condition that their are no empty intersections. Again, y_i^v is 1 if $N[v]$ incorporates at least one node of the set i of the partition. Additionally, we introduce the set of auxiliary variables $z^v \in \{0,1\}$. The constraint $\forall v \in V,\ \forall i \in \{1,\dots,n\} : z^v \le y_i^v$ and objective function ensure z^v is equal to the largest y_i^v. Hence, the LP minimise the number of incompletely covered nodes from Equation (14) by maximising the number of completely covered nodes.

The LP for the optimal as well as the maximal n-soft domatic partition can also be modified to minimise the number of missing coverages or incompletely covered nodes if a node incorporates more than one security mean. We have discussed two versions of this approach: Either by implementing a fix number of security means per node or by distributing different combinations of security means based on their individual estimated costs. If a node is allowed to implement a fix number of $k \in \mathbb{N}_{>1}$ different security means, the constraint $\forall v \in V : \sum_{i=1}^{n} x_i^v = 1$ changes to:

$$\forall v \in V : \sum_{i=1}^{n} x_i^v = k \tag{27}$$

Next, we apply security means (associated with sets i of the partition) based on the share of resources necessary per security mean m_i, available at each node $v \in V$. The constraint $\forall v \in V : \sum_{i=1}^{n} x_i^v = 1$ has to be updated as follows:

$$\forall v \in V : \sum_{i=1}^{n} m_i \cdot x_i^v = 1 \tag{28}$$

The resource costs over all security means form a vector $\mathbf{m} \in \mathbb{R}^n$ with its components $m_i \in (0,1]$. Without loss of generality, the overall resources available for security means per node have been set to 1. Each value m_i represents the individual portion of costs caused for operating security mean i in relation to the total costs for all security means. The representation of necessary and available resources as a scalar is a simplification showing the feasibility of our LPs to take those into account.

VI. λ-PRECISION UDG GENERATOR

The algorithms we propose to distribute security means in favour of the neighbourhood watch inspired security framework for large-scale static homogeneous WSNs are NP hard. It is necessary to validate the computability of the algorithms on a large number of realistic WSN models. Since, we cannot pinpoint the exact influence of graph properties on the computation time of our partitioning algorithms, we examine it empirically. Computations on a large set of models enable us to study the relation between different graph properties and the computation time. To do so, we need a generator supplying it with a large variety and number of WSN graph models with desired properties.

With the growing demand and sizes of WSNs [59], the attention of potential attackers [60] increases as well. As consequence, more complex security [61] and communication protocols [62] are developed. The application of those protocols leads to an increasing power consumption which affects the available computational and energy resources for the actual tasks of nodes. Since nodes and their distribution are expensive and their failure can lead to the failure of the network, network operators are interested in maximising the potential lifetime of nodes and the networks. An attempt to deal with the higher demand in power are smart sleep scheduling schemes [63] and hop-by-hop communication strategies [64]. Additionally, there are many algorithms whose complexities exceed the deterministic polynomial time bound or are bound to higher polynomial degrees [65], [66]. For researchers to decide whether those algorithms can be solved analytically or bring the need of an approximation, empirical evaluations on WSN graph models with desired properties are necessary. To generate these models, we introduce a graph generator that creates λ-precision UDGs by distributing nodes randomly and uniformly in a unit square.

λ-precision UDGs have several advantages compared to ordinary UDGs. The λ-precision limits the node degree of each node in the graph. The limitation results from the size of the ring given by the radii λ and r_{tr} with $0 < \lambda < r_{\text{tr}}$. Nodes only connect (have common edges) to nodes within this ring, since each node has to have at least λ distance to other nodes in the network. Therefore, there is an upper limit of nodes that are able to connect. Choosing the λ distance and node number $|V|$ so that a large portion of the area of the generation is covered ensures that the nodes are more evenly spaced out on the generation plane. Hence, it allows to control the variance of the local cluster coefficient. With regard to WSNs, evenly distributed nodes improve the area-wide monitoring.

A. NODE DISTRIBUTION

To generate the graphs, we start with randomly and uniformly distributing nodes in a unit square with the constraint that two nodes have to have a minimal distance λ in between them. For an efficient computation, it is necessary to discretise the unit square. We do so, with a uniform grid size of 1000 times 1000. The grid size can be adapted as needed and is often chosen based on the computational limits and the intended graph properties as for example the number of nodes. In the implementation, we distribute the nodes iteratively. Each node occupies the grid coordinate of its centre and all grid

coordinates within λ distance from it. For this purpose, each grid coordinate gets assigned a marker value. The marker indicates whether the coordinate is still available (0) or occupied (1). After a new node has been added, all surrounding marker values in λ distance are updated by setting them to 1. The coordinate for the centre of the succeeding nodes is randomly and uniformly selected from the non-occupied coordinates. The process is repeated until either no grid coordinates are available or the desired amount of nodes has been placed within the unit square.

B. GENERATOR SEEDS

In order to create λ-precision UDGs with specific properties, it is essential to determine the input parameters (generator seeds) resulting in graphs with the desired properties. We use the following input parameters: number of nodes $|V|$, the pairwise minimal distance in between the nodes λ and the distance r_{tr} up to which nodes are connected. For the empirical evaluation of random graphs, we compute for each parameter set a certain amount of graphs. After computing a set of graphs for chosen input parameters, we compare the properties with our target values. Depending on the outcome, we either save the result or adjust the input parameters. To do so, we apply a binary search separately for both parameters λ and r_{tr}, starting with λ. Table 1 shows the resulting generator seeds to create random λ-precision UDGs with desired properties. As shown in Table 1 the target values are the medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$ and the medium average node degree $\overline{\deg_{\text{avg}}}$.

The value of $\overline{A_{\text{coverage}}}$ affects the probability of resulting random λ-precision UDGs to be connected $P_{\text{connected}}$. In addition, it ensures a low variance of the local cluster coefficient and an even coverage of the generation plane as we discuss in Subsection VI-C. Applying a binary search, we first approach the radius λ achieving a medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$ between 85% and 87.5%. The coverage is determined numerically. After distributing nodes as described in the previous subsection, the relation between occupied grid coordinates and the grid size yields the total coverage of the generation plane A_{coverage}. Finally, the medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$ of all generated graphs for the input parameter set is computed.

Next, we determine the transmission range r_{tr} using a binary search until we reach a medium average node degree $\overline{\deg_{\text{avg}}}$ over all graphs obtained for the given input parameters. The final results of the computed generator seeds is shown in Table 1. To generate the graphs for the evaluation of our $0-1$ LPs we will use the results from this table.

TABLE 1. Empirically determined seeds to generate graphs with an expected average node degree in between $\deg_{\text{exp}}$ to $\deg_{\text{exp}}+0.25$ for a given number of nodes $|V|$ and a desired medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$ from 85% to 87.5%. The values have been determined by generating repeatedly sets of 20 graphs for varying values of λ and r_{tr} until approaching the desired properties. The probability $P_{\text{connected}}$ is the empirically determined likeliness of a graph to be connected for the given parameters. The results for $\overline{A_{\text{coverage}}}$, $\overline{\deg_{\text{avg}}}$ and $P_{\text{connected}}$ are the arithmetic mean values of 20 graphs of the determined input parameter combinations.

$\lvert V\rvert$	$\deg_{\text{exp}}$	λ	r_{tr}	$\overline{A_{\text{coverage}}}$	$\overline{\deg_{\text{avg}}}$	$P_{\text{connected}}$
20	3	0.148	0.290	0.855	3.105	0.75
20	4	0.148	0.333	0.855	4.005	0.95
20	5	0.148	0.383	0.855	5.210	1.00
20	6	0.148	0.411	0.855	6.055	1.00
40	3	0.104	0.196	0.869	3.055	0.50
40	4	0.104	0.226	0.869	4.140	0.85
40	5	0.104	0.250	0.869	5.100	1.00
40	6	0.104	0.277	0.869	6.195	1.00
60	3	0.085	0.159	0.862	3.198	0.05
60	4	0.085	0.179	0.862	4.156	0.70
60	5	0.085	0.197	0.862	5.075	0.95
60	6	0.085	0.219	0.862	6.198	0.95
80	3	0.072	0.136	0.854	3.222	0.20
80	4	0.072	0.152	0.854	4.061	0.65
80	5	0.072	0.168	0.854	5.067	0.80
80	6	0.072	0.181	0.854	6.008	0.90
100	3	0.065	0.120	0.861	3.142	0.20
100	4	0.065	0.137	0.861	4.234	0.70
100	5	0.065	0.150	0.861	5.127	0.85
100	6	0.065	0.164	0.861	6.198	1.00
120	3	0.059	0.108	0.855	3.120	0.10
120	4	0.059	0.122	0.855	4.086	0.70
120	5	0.059	0.135	0.855	5.115	1.00
120	6	0.059	0.147	0.855	6.196	1.00
140	3	0.054	0.098	0.866	3.074	0.10
140	4	0.054	0.112	0.866	4.127	0.70
140	5	0.054	0.124	0.866	5.089	0.95
140	6	0.054	0.136	0.866	6.237	1.00
160	3	0.051	0.094	0.857	3.215	0.05
160	4	0.051	0.105	0.857	4.057	0.60
160	5	0.051	0.116	0.857	5.164	0.90
160	6	0.051	0.127	0.857	6.183	1.00
180	3	0.048	0.087	0.853	3.093	0.00
180	4	0.048	0.098	0.853	4.063	0.50
180	5	0.048	0.109	0.853	5.128	0.90
180	6	0.048	0.117	0.853	6.064	0.95
200	3	0.045	0.082	0.866	3.095	0.00
200	4	0.045	0.093	0.866	4.139	0.55
200	5	0.045	0.102	0.866	5.047	0.75
200	6	0.045	0.111	0.866	6.056	0.95
220	3	0.044	0.077	0.867	3.031	0.05
220	4	0.044	0.089	0.867	4.164	0.65
220	5	0.044	0.098	0.867	5.132	0.90
220	6	0.044	0.107	0.867	6.179	1.00
240	3	0.041	0.075	0.863	3.138	0.05
240	4	0.041	0.084	0.863	4.011	0.25
240	5	0.041	0.093	0.863	5.041	0.75
240	6	0.041	0.101	0.863	6.047	0.85
260	3	0.040	0.070	0.855	3.006	0.00
260	4	0.040	0.081	0.855	4.116	0.30
260	5	0.040	0.089	0.855	5.088	0.65
260	6	0.040	0.097	0.855	6.167	0.95
280	3	0.039	0.067	0.867	3.025	0.00
280	4	0.039	0.077	0.867	4.080	0.60
280	5	0.039	0.086	0.867	5.082	0.85
280	6	0.039	0.094	0.867	6.183	0.95
300	3	0.037	0.066	0.874	3.008	0.05
300	4	0.037	0.074	0.874	4.036	0.60
300	5	0.037	0.083	0.874	5.029	0.95
300	6	0.037	0.090	0.874	6.014	1.00

C. CLUSTER COEFFICIENT AND DEGREE DISTRIBUTION

We show that the variance of the local cluster coefficient and the variance of the node degree distribution decreases with an increasing coverage of the generation plane. This allows to generate specific graphs and test the effect of those properties on the computation time of our partitions. We can assume that a graph contains an edge, or a small number of edges, whose removal would disconnect the graph into several larger connected components. In such cases, we can expect a decrease in computation time compared to a graph

TABLE 2. The seeds for our generator to test the behaviour of the variance of the local cluster coefficient and the variance of the node degree distribution subject to the total coverage of the generation plane have been computed as in Table 1. The abbreviation $A_{\text{exp\_cov}}$ stands for the expected coverage area. It represents the coverage range for which we determine λ and r_{tr}. Therefore, the resulting coverage area A_{cov} of graphs generated with those parameters is likely to be within the specified range. $\overline{A_{\text{cov}}}$ is short for $\overline{A_{\text{coverage}}}$ and P_{conn} abbreviates $P_{\text{connected}}$ according to Table 1.

$A_{\text{exp\_cov}}$	$\lvert V\rvert$	$\deg_{\text{exp}}$	λ	r_{tr}	$\overline{A_{\text{cov}}}$	$\overline{\deg_{\text{avg}}}$	P_{conn}
[0.850, 0.875]	100	4	0.065	0.136	0.857	4.193	0.575
[0.850, 0.875]	100	5	0.065	0.148	0.857	5.073	0.825
[0.850, 0.875]	200	4	0.045	0.094	0.864	4.174	0.525
[0.850, 0.875]	200	5	0.045	0.103	0.864	5.093	0.850
[0.875, 0.900]	100	5	0.066	0.148	0.890	5.044	0.975
[0.875, 0.900]	100	4	0.066	0.136	0.890	4.161	0.800
[0.875, 0.900]	200	4	0.047	0.094	0.884	4.163	0.650
[0.875, 0.900]	200	5	0.047	0.103	0.884	5.070	0.900
[0.900, 0.925]	100	4	0.068	0.136	0.915	4.143	0.800
[0.900, 0.925]	100	5	0.068	0.152	0.915	5.249	0.925
[0.900, 0.925]	200	4	0.048	0.094	0.918	4.125	0.600
[0.900, 0.925]	200	5	0.048	0.105	0.918	5.224	0.900
[0.925, 0.950]	100	4	0.070	0.136	0.932	4.094	0.950
[0.925, 0.950]	100	5	0.070	0.153	0.932	5.231	1.000
[0.925, 0.950]	200	5	0.051	0.103	0.947	5.028	0.975
[0.925, 0.950]	200	4	0.051	0.094	0.947	4.109	0.700
[0.950, 0.975]	100	5	0.074	0.153	0.968	5.179	1.000
[0.950, 0.975]	100	4	0.074	0.136	0.968	4.083	0.900
[0.950, 0.975]	200	5	0.051	0.103	0.959	5.006	1.000
[0.950, 0.975]	200	4	0.051	0.094	0.959	4.114	0.875
[0.975, 1.000]	100	4	0.078	0.136	0.987	4.114	0.950
[0.975, 1.000]	100	5	0.078	0.153	0.987	5.104	1.000
[0.975, 1.000]	200	4	0.059	0.094	0.999	4.157	0.900
[0.975, 1.000]	200	5	0.059	0.105	0.999	5.074	1.000

without such breaking points. Therefore, we expect that a low variance of the distribution of the node degrees provides information about an upper bound of the computation time. The portion of the covered area of the generation plane is directly linked to the combined choice of the number of nodes $|V|$ and their minimal pairwise distance λ within the graph. To evaluate the behaviour of the interplay between the portion of the covered area of the generation plane and the variance of the node degree distribution as well as the variance of the local cluster coefficient we determine additional generator seeds. To do so, we first determine generator seeds for selected target values as in the previous section. We have chosen an expected average node degree $\deg_{\text{exp}}$ of 4 and 5. The observed node numbers $|V|$ are 100 and 200. For each of those combinations the expected covered area of the generation plane is set to the intervals {[0.850, 0.875], [0.875, 0.900], [0.900, 0.925], [0.925, 0.950], [0.950, 0.975], [0.975, 1.000]}. We exhibit the determined generator seeds in Table 2. The results indicate that the determined seeds maintain a high probability to create connected graphs even with a decreasing medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$.

For our empirical analysis of the relation between the medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$ and the variance of the node degree distribution as well as the variance of the local cluster coefficient, we determine the parameters with the same binary search utilised in the previous subsection. By means of these parameters, we compute 40 sample graphs for each of the discussed target parameter combinations. The target parameters are number of nodes $|V|$, medium average node degree $\overline{\deg_{\text{avg}}}$ and medium

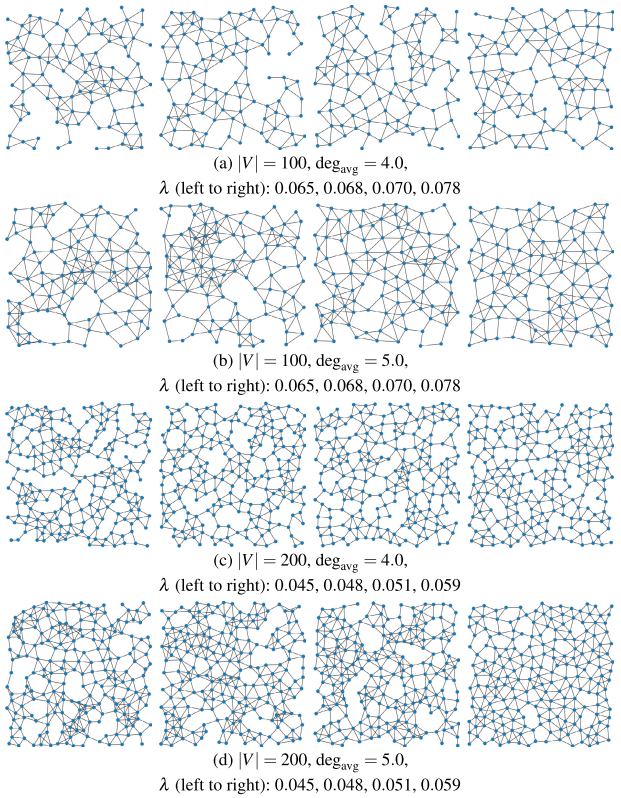

FIGURE 4. As example for the resulting λ-precision UDGs shown per row are from left to right generated for A_{coverage} of the ranges [0.850, 0.875], [0.900, 0.925], [0.925, 0.950] and [0.975, 1.000] respectively.

total coverage of the generation plane $\overline{A_{\text{coverage}}}$. A selection of 16 of the resulting uniformly and randomly determined λ-precision UDGs is displayed in Fig. 4.

The results of our evaluation are depicted in Fig. 5 and 6. The x coordinate for each data point is located at the lower value of the respective range representing the medium total coverage of the generation plane $\overline{A_{\text{coverage}}}$ in both diagrams. Each data point represents the arithmetic mean over the variance of the local cluster coefficients and the list of node degrees per graph for a sample size of 40 graphs.

D. METHODS FOR ADAPTATION OF GENERATED UDGS

Use case dependent adaptations of generated λ-precision UDGs can be necessary to satisfy certain requirements. Therefore, to specify the accuracy of the graph properties, we have implemented methods to adapt the graphs resulting from our graph generator. Connectivity, occurrences of bridges or average node degree are properties, we have considered for adaptation. It is unlikely to receive a large randomly generated graph that meets exactly a set of desired properties based on selected input parameters. To improve quality, validity and precision of an evaluation using graphs, it is desirable that those graphs meet exact criteria. Certain properties are perhaps achievable solely by repeatedly generating graphs. However, such a process is tedious and time consuming, especially for large numbers of graphs. **Connectivity:** Our approach to connect a graph consisting

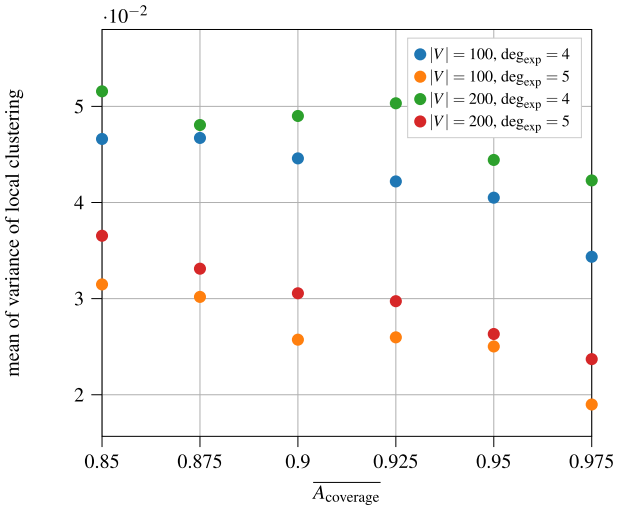

FIGURE 5. The mean of the variance of the local cluster coefficients tends to decrease along with increasing $\overline{A_{\text{coverage}}}$ leading to more homogeneously distributed nodes with larger pairwise distances. A sample size of 40 graphs per data point has been utilised. This sample size balances the expressivity of the decrease trend with the computational effort for parameterised graph generation.

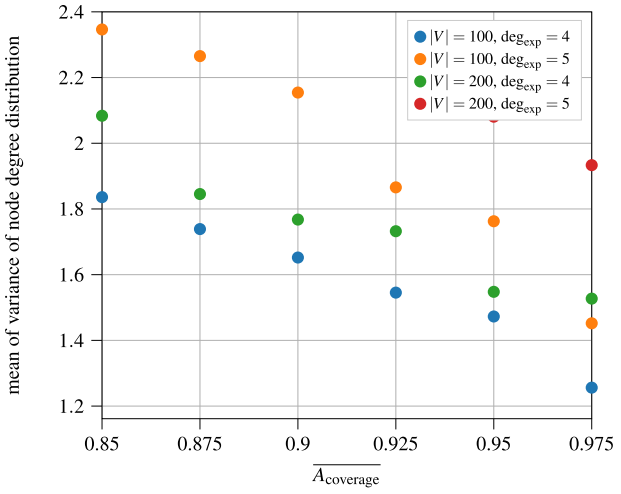

FIGURE 6. The mean of the variance of the node degree distribution mostly diminishes subject to a growing $\overline{A_{\text{coverage}}}$. This behaviour results in graphs that can be better employed for n-soft domatic partitions in a certain range for n. Again, a sample size of 40 graphs per data point has been utilised which implies some minor local fluctuations.

of several connected components uses the nearest neighbour attempt. The algorithm determines all pairs of nearest neighbour nodes between distinct connected components. For each iteration, the nearest neighbour pair with the shortest edge length (euclidean distance) is chosen and added as edge to the graph. Subsequently, all nearest neighbour pairs which in turn contain nodes from a single connected component inside the resulting graph are removed from the set. The last two steps are repeated until there is only one connected component in the graph.

Further, we assume that the occurrence of bridges in a graph model significantly affects the empirical test results and influences the evaluation of the computability of complex algorithms. To validate this hypothesis, we provide an algorithm allowing to identify and to remove bridges. The result is a connected, bridge-free graph that serves as representation of large-scale static homogeneous WSNs. To identify possible bridges in our λ-precision UDGs, our generator utilises the NetworkX library. Our general algorithm selects one of the identified bridges. Each node in the bridge indicates one of the bridge connected components. Thus, we determine a new edge that connects both components which in turn do not include either of the nodes of the bridge. Subsequently, we start over with the next bridge connecting two remaining bridge-connected components. The algorithm repeats the process until there are no more bridge-connected components left. A special case that needs to be treated before running the general algorithm is the appearance of bridge paths. We treat those first to prevent the general algorithm from infinitely looping. In such a case, we start at one end of the bridge path P incorporating the nodes v_s, v_{s+1}, …, v_{s+k} and the edges $\{\{v_s, v_{s+1}\}, \{v_{s+1}, v_{s+2}\}, \ldots, \{v_{s+k-1}, v_{s+k}\}\}$ of the graph $G = (V, E)$. Starting at v_s of the bridge path, we add an edge to the graph from v_s to the next but one node v_{s+2}. This procedure has to be repeated for each node except the nodes v_{k-1} and v_k. After applying this procedure, all bridge paths have been eliminated from the graph and the general algorithm to remove the bridges of the graph can be executed.

Average Node Degree: The algorithm will remove edges from the graph until a desired average node degree has been achieved. We have chosen the edge length as decisive property to select the edges to be removed, since in WSNs a connection between nodes that are further apart is less likely. To accomplish this, the algorithm selects edges by different criteria. The selection can be done randomly, with the probability of an edge being removed weighted by its length and a given exponent or in order by edge length. As additional conditions, we can exclude edges that, if removed, would cause the graph to become disconnected or introduce bridges within it.

VII. EMPIRICAL TEST SETUP

To evaluate the computability of optimal and maximal n-soft domatic partitions for reasonably sized large-scale static homogeneous WSNs, we outline the details of our empirical test setup. The corresponding graphs are created by the proposed λ-precision UDG generator using the seeds depicted in Subsection VI-B and the associated Table 1. We have chosen graphs with a number of nodes $|V|$ starting from 20 to 300 in steps of 20. Only connected λ-precision UDGs created by our graph generator are accepted in our test setup. In case a generated graph is not connected, we discard it and repeat the generation process for the given parameters until the desired number of connected λ-precision UDGs has been reached. After successfully generating 20 connected graphs for each row of parameter combinations in Table 1, we duplicate the complete set of graphs once for a second test setup. The original set of graphs SG_1 is then adapted to approach the expected average node degree $\deg_{\text{exp}}$ by

successive removal of edges. The algorithm used to adapt the graphs and to reach the desired average node degree is described in Subsection VI-D. To adjust the average node degree, we take the squared edge length of each edge that does not disconnect the graph as weight. The squaring gives longer edges a higher priority to be selected in the process. Then, edges are removed iteratively and by chance based on their respective given weight until the average node degree $\deg_{\text{avg}}$ reaches the desired expected average node degree $\deg_{\text{exp}}$. The graphs in the duplicated set SG_2 are modified by removing all bridges as described in Subsection VI-D. Afterwards, we ensure that the $\deg_{\text{exp}}$ in the table row associated with the graph is reached as described for SG_1 but without the risk of creating new bridges. The set of graphs SG_2 is created to evaluate whether small topological properties like for a graph to be bridge-free in our given set of graphs directly affects the computability or quality of results of our partitioning schemes. Finally, we compute for all graphs the optimal and maximal n-soft domatic partitions for $n \in \{3, 4, 5\}$. For this purpose, the $0-1$ LPs have been implemented using Pyomo [24] and they are computed using the mathematical programming solver Gurobi [25]. In the last step, we evaluate the results via Python. Therefore, we track the wallclock times given by Gurobi. In addition, we count the number of missing coverages $e_{\text{miss\_cov}}$ introduced in Equation (13) as well as the incompletely covered nodes $e_{\text{inc\_nodes}}$ expressed in Equation (14). The time limit for Gurobi to solve a given LP on a given graph is set to 1200 seconds on a system with two Intel® Xeon® Gold 6248R as central processing units and 256 GB of random access memory.

VIII. RESULTS AND EVALUATION

Here, we evaluate the computation results of the optimal and maximal n-soft domatic partitions with $n \in \{3, 4, 5\}$ and for 2400 different λ-precision UDGs divided into two test sets SG_1 and SG_2 as described in the previous section.

First, we start with solely discussing the results computed on SG_1. In Fig. 7, we evaluate the mean of the computation time of the optimal 3, 4 and 5-soft domatic partitions in dependence on the number of nodes $|V|$ of the given graphs. The colours of the respective curves represent the expected average node degree of the given graphs. The dotted lines in between the drawn data points are added exclusively to improve the readability of the plots.

All plots in Fig. 7 show a similar behaviour. The computation time seems to increase for graphs with an average node degree lower or equal to the partition size of a graph. The increase is potentially caused by the implicit increase of a number of missing coverages $\overline{e_{\text{miss\_cov}}}$ in those graphs as illustrated in Tables 3, 4, 5 and 6. For all graphs the n-soft domatic partitions could be solved to optimality. The required computation time remained below 13 seconds for each graph and determined partition. Therefore, even larger graphs should be solvable in reasonable time. Deviations and jumps between node numbers can potentially be attributed to the limited number of test cases, which influence the test result through individual outliers. Such jumps can be observed, for example, in the figure for the optimal 5-soft domatic partitioning in Figure 7. There, the curve for graphs

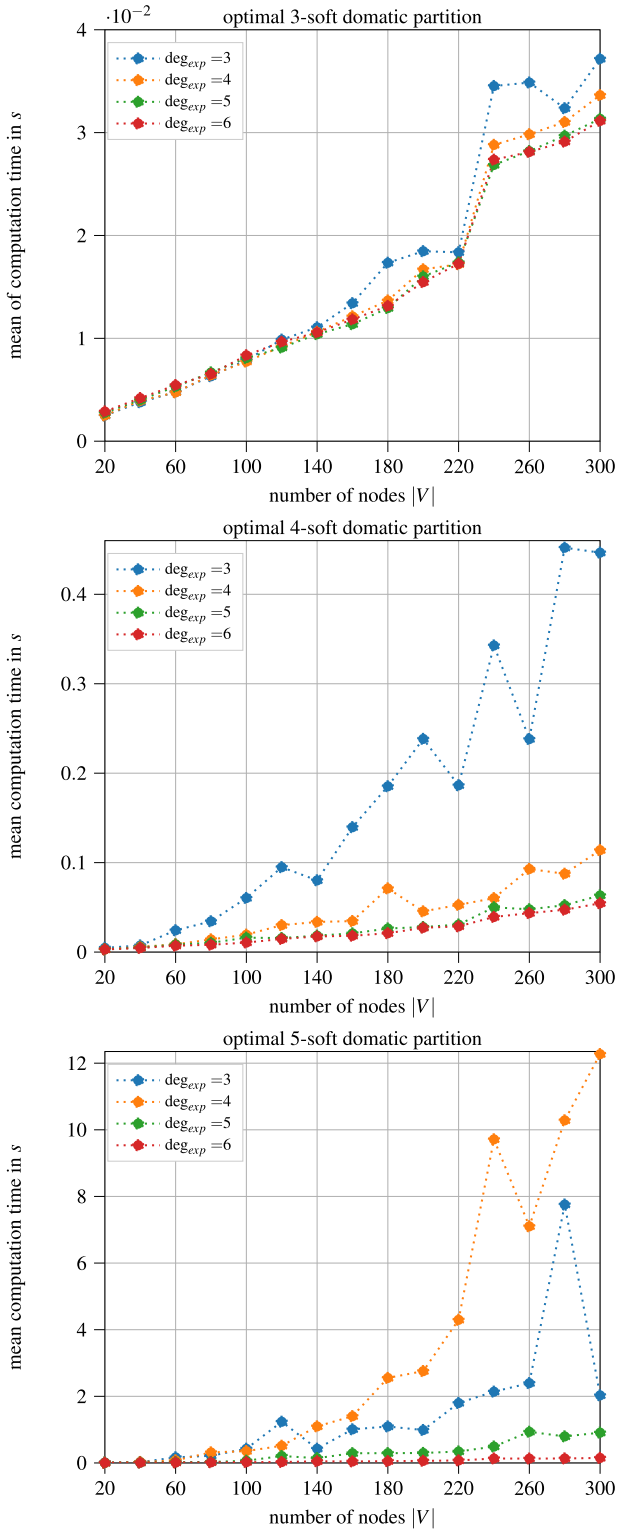

FIGURE 7. Test results of the mean of the computation time in seconds s subject to the number of nodes $|V|$ of given λ-precision UDGs necessary to determine optimal n-soft domatic partitions within a time limit of 1200 s.

with average node degree 5 between 260 and 300 nodes includes an unexpected increase in computation time for graphs with 280 nodes.

Overall, the number of test cases seems to be sufficient to derive a trend towards the required computation time. Moreover, the results show that the computability of the optimal n-soft domatic partitions for $n \in \{3, 4, 5\}$ is given for the set of graphs SG_1.

Fig. 8 shows the results for the computation of the maximal 3, 4 and 5-soft domatic partitions of SG_1. The diagrams are structured in the same way as the diagrams from Fig. 7. The results in this figure are also subject to empirical fluctuations, particularly due to the limited number of test cases.

The topmost diagram of Fig. 8 displays the results of the maximal 3-soft domatic partition. It shows that the mean of the computation time increases with an increasing number of nodes in the graph. Furthermore, we can see that the increase of the average node degree comes with a decrease of the mean of the computation time. This can be a consequence of the decreasing number of nodes that do not cause any coverage errors and therefore contribute to the optimality result of the LPs. A similar effect becomes visible in the results of the computation of the maximal 4-soft domatic partition in Fig. 8 as well. The observed patterns in computation time for maximal 3 and 4-soft domatic partitions align with the trends observed in the optimal 3 and 4-soft domatic partitions, as depicted in Fig. 7. These trends indicate an increase in computation time for graphs where the average node degrees are equal to or lower than the computed partition size. The trend continues in general for the computation time of the optimal and maximal 5-soft domatic partitions while the order of plots for graphs with average node degrees greater and equal to the partition changes compared to the other graphs. In a broader sense, the consistent trend persists regarding the computation time for both the optimal and maximal 5-soft domatic partitions. However, there is a variation in the arrangement of plots concerning graphs with average node degrees equal to or greater than the partition size, distinct from the other graph scenarios. Overall, an observable pattern emerges where an increasing average node degree relative to the partition size notably rises the mean computation time. All maximal 3, 4 and 5-soft domatic partitions have been computed optimally within the given time limit of 1200 seconds.

In Tables 3, 4, 5 and 6, we compare and evaluate the cases in which for the given test setup and the set of graphs SG_1 and for the given set of parameters at least one solution has been computed optimally and one non-optimally within the given time limit for either optimal or maximal n-soft domatic partitions. Even so, we compare results which have been computed for different graphs and for each parameter combination, we set up only a set of 20 graphs. For each row in the Tables 3, 4, 5 and 6 and its respective parameter combinations, we generated 20 graphs and determined the optimal and maximal 3, 4 and 5-soft domatic partitions. As the data used in the plots mirror the information

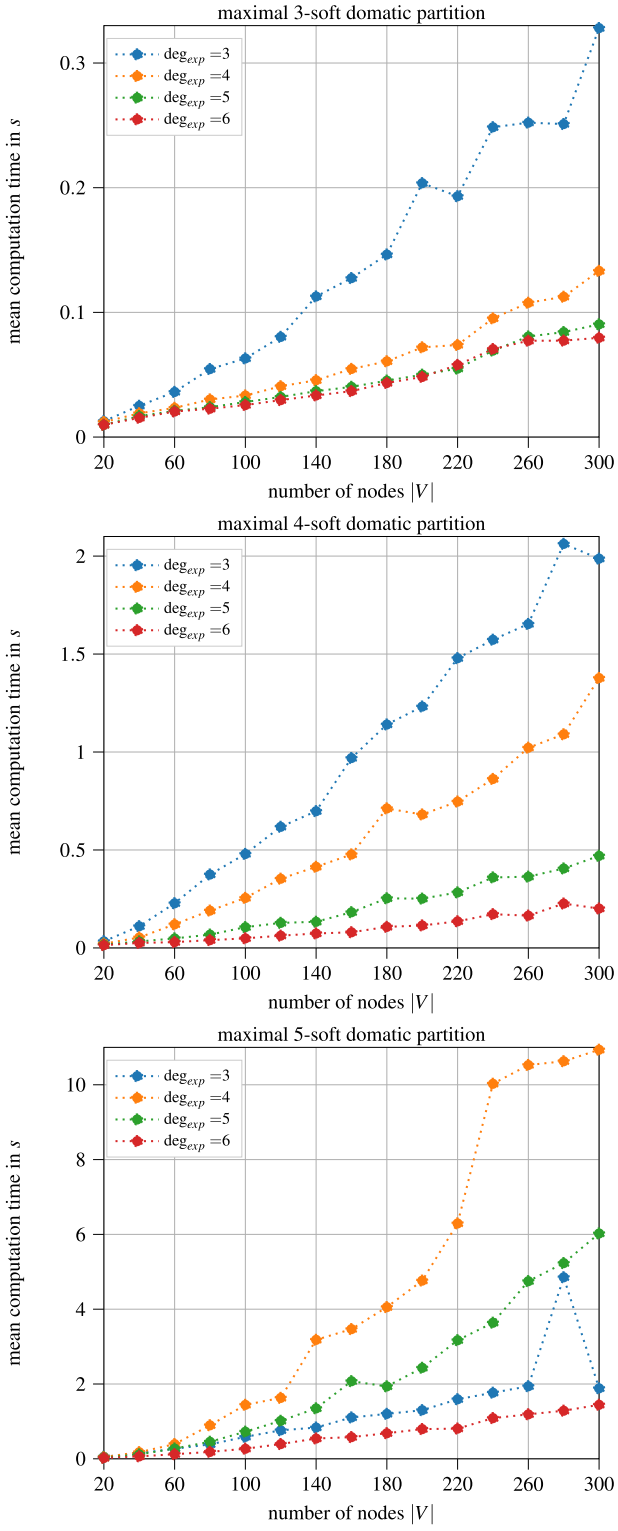

FIGURE 8. Test results of the mean of the computation time in seconds s necessary to determine maximal n-soft domatic partitions in dependence of the number of nodes $|V|$ of λ-precision UDGs within a time limit of 1200 s.

showcased in the tables, all presented data signify optimal solutions identified through Gurobi within the time limit of 1200 seconds. Small fluctuations in the results can be caused

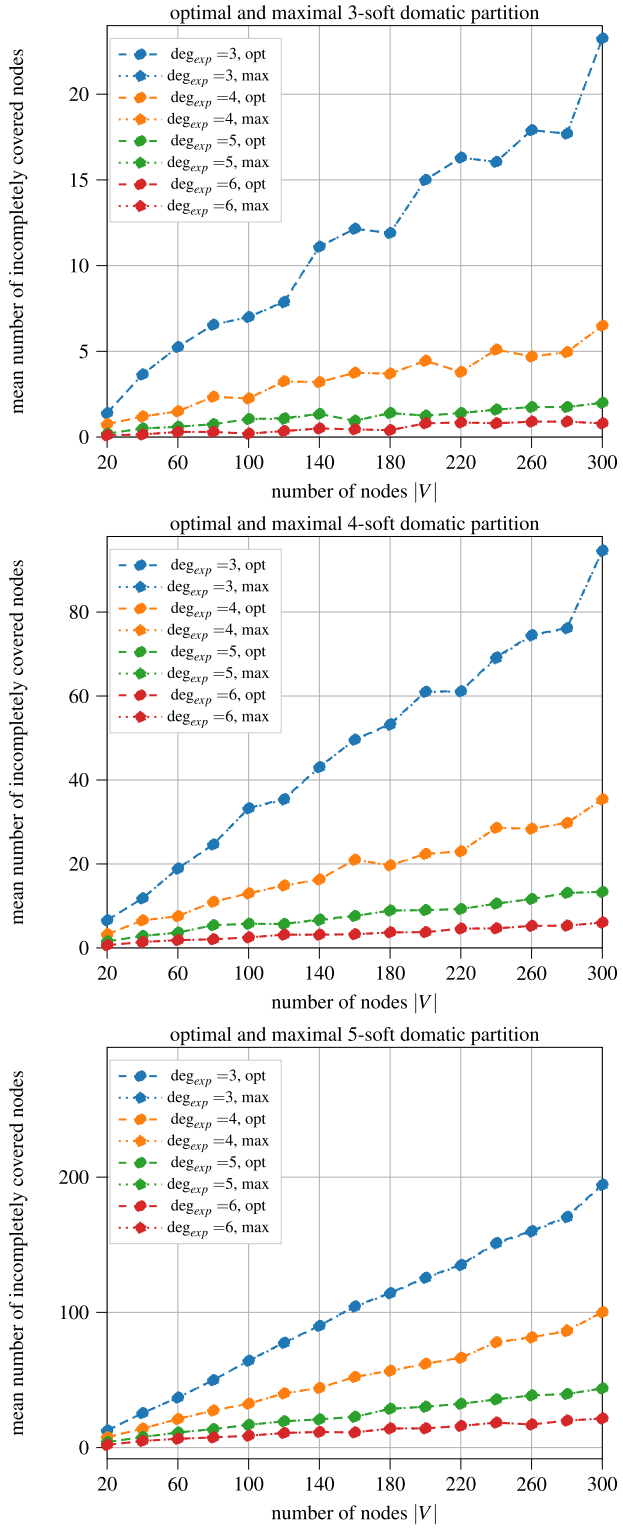

FIGURE 9. Arithmetic mean of the number of incompletely covered nodes in optimal and maximal n-soft domatic partitions subject to the number of nodes $|V|$ of the given λ-precision UDGs of optimal n-soft domatic partitions within a time limit of 1200 s.

by Gurobi even without a given MIPGap. By default, Gurobi aims to prove optimality within certain numerical tolerances without the user explicitly setting the MIPGap parameter. Gurobi's MIPGap represents the allowable gap between

TABLE 3. Results for average node degree deg = 3, partition sizes $n \in \{3, 4, 5\}$ and node numbers $|V| = \{20, 40, \ldots, 300\}$ for optimal and maximal n-soft domatic partitions. Showing the mean of the number of missing coverages $\overline{e_{\text{miss\_cov}}}$ and number of incompletely covered nodes $\overline{e_{\text{inc\_nodes}}}$.

n	$\lvert V\rvert$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$
		opt n-soft domatic partition		max n-soft domatic partition	
3	20	1.40	1.40	4.20	1.40
	40	3.65	3.65	10.95	3.65
	60	5.25	5.25	15.75	5.25
	80	6.55	6.55	19.65	6.55
	100	7.00	7.00	21.00	7.00
	120	7.90	7.90	23.55	7.85
	140	11.10	11.10	33.30	11.10
	160	12.15	12.15	36.45	12.15
	180	11.90	11.90	35.70	11.90
	200	15.00	15.00	45.00	15.00
	220	16.30	16.30	48.90	16.30
	240	16.05	16.05	48.15	16.05
	260	17.90	17.90	53.70	17.90
	280	17.70	17.70	53.10	17.70
	300	23.25	23.25	69.75	23.25
4	20	8.00	6.60	26.40	6.60
	40	15.45	11.80	47.20	11.80
	60	24.20	18.95	75.40	18.85
	80	31.15	24.65	98.40	24.60
	100	40.30	33.30	132.80	33.20
	120	43.35	35.50	141.20	35.30
	140	54.20	43.10	172.00	43.00
	160	61.70	49.60	198.40	49.60
	180	65.35	53.35	212.40	53.10
	200	76.05	61.05	243.80	60.95
	220	77.45	61.15	244.40	61.10
	240	85.40	69.25	275.80	68.95
	260	92.50	74.55	297.60	74.40
	280	93.90	76.20	304.40	76.10
	300	117.85	94.70	378.00	94.50
5	20	20.65	12.70	62.75	12.55
	40	40.95	25.50	127.50	25.50
	60	61.15	37.05	184.50	36.90
	80	81.10	49.90	249.00	49.80
	100	104.40	64.30	320.75	64.15
	120	120.50	77.70	386.25	77.25
	140	143.85	90.05	449.75	89.95
	160	165.80	104.60	520.50	104.10
	180	178.95	114.35	569.75	113.95
	200	201.10	125.75	627.00	125.40
	220	212.20	135.25	673.75	134.75
	240	235.55	151.50	754.50	150.90
	260	252.45	160.30	798.00	159.60
	280	264.00	170.85	851.00	170.20
	300	311.40	194.60	970.50	194.10

the best-known solution and the proven optimal solution. Therefore, in some cases, we can observe those numerical tolerances in the table. To compare the results of the maximal and optimal n-soft domatic partitions, we evaluate Fig. 9 in which we reflect the number of incompletely covered nodes as result of the computation of the maximal and optimal n-soft domatic partition. The dashed and dotted lines in the diagrams serve solely to enhance readability and do not correspond to computed data or specific interpretations.

For a final comparison of the performance of the solutions for maximal and optimal n-soft domatic partitions, we determine the relative mean of the results for the graphs in SG_1. On average, the number of missing coverages $e_{\text{miss\_cov}}$ of the optimal n-soft domatic partition for our test setup and for the set of graphs SG_1 is 10.52% lower compared to the maximal n-soft domatic partition. In contrast, the same comparison for the number of incompletely covered nodes yields only an improvement of $\overline{P_{\text{inc\_nodes}}} = 0.04\%$ for the

TABLE 4. Results for average node degree deg = 4, partition sizes $n \in \{3, 4, 5\}$ and node numbers $|V| = \{20, 40, \ldots, 300\}$ for optimal and maximal n-soft domatic partitions. Showing the mean of the number of missing coverages $\overline{e_{\text{miss\_cov}}}$ and number of incompletely covered nodes $\overline{e_{\text{inc\_nodes}}}$.

n	$\lvert V\rvert$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$
		opt n-soft domatic partition		max n-soft domatic partition	
3	20	0.75	0.75	2.25	0.75
	40	1.20	1.20	3.60	1.20
	60	1.50	1.50	4.50	1.50
	80	2.35	2.35	7.05	2.35
	100	2.25	2.25	6.75	2.25
	120	3.25	3.25	9.75	3.25
	140	3.20	3.20	9.60	3.20
	160	3.75	3.75	11.25	3.75
	180	3.70	3.70	11.10	3.70
	200	4.45	4.45	13.35	4.45
	220	3.80	3.80	11.40	3.80
	240	5.10	5.10	15.30	5.10
	260	4.70	4.70	14.10	4.70
	280	4.95	4.95	14.85	4.95
	300	6.50	6.50	19.50	6.50
4	20	4.00	3.25	13.00	3.25
	40	7.75	6.55	26.20	6.55
	60	9.05	7.55	30.20	7.55
	80	13.35	11.00	43.80	10.95
	100	15.25	13.00	51.80	12.95
	120	18.25	14.95	59.20	14.80
	140	19.50	16.30	65.20	16.30
	160	24.75	21.00	84.00	21.00
	180	23.40	19.70	78.60	19.65
	200	26.85	22.40	89.40	22.35
	220	26.85	23.00	92.00	23.00
	240	33.70	28.65	114.60	28.65
	260	33.10	28.40	113.60	28.40
	280	34.75	29.80	118.80	29.70
	300	41.90	35.40	141.40	35.35
5	20	11.70	7.65	38.25	7.65
	40	21.85	14.05	70.25	14.05
	60	30.25	21.15	105.25	21.05
	80	40.85	27.50	136.50	27.30
	100	47.75	32.50	162.25	32.45
	120	58.15	40.15	199.75	39.95
	140	63.45	44.15	220.25	44.05
	160	77.00	52.30	260.75	52.15
	180	80.15	56.95	283.00	56.60
	200	88.90	62.05	309.25	61.85
	220	93.15	66.50	331.25	66.25
	240	111.75	77.95	388.75	77.75
	260	114.65	81.65	406.25	81.25
	280	120.25	85.95	434.25	86.85
	300	141.90	100.35	499.50	99.90

TABLE 5. Results for average node degree deg = 5, partition sizes $n \in \{3, 4, 5\}$ and node numbers $|V| = \{20, 40, \ldots, 300\}$ for optimal and maximal n-soft domatic partitions. Showing the mean of the number of missing coverages $\overline{e_{\text{miss\_cov}}}$ and number of incompletely covered nodes $\overline{e_{\text{inc\_nodes}}}$.

n	$\lvert V\rvert$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$
		opt n-soft domatic partition		max n-soft domatic partition	
3	20	0.20	0.20	0.60	0.20
	40	0.50	0.50	1.50	0.50
	60	0.60	0.60	1.80	0.60
	80	0.75	0.75	2.25	0.75
	100	1.05	1.05	3.15	1.05
	120	1.11	1.11	3.15	1.05
	140	1.35	1.35	4.05	1.35
	160	0.95	0.95	2.85	0.95
	180	1.40	1.40	4.20	1.40
	200	1.25	1.25	3.75	1.25
	220	1.40	1.40	4.20	1.40
	240	1.60	1.60	4.80	1.60
	260	1.75	1.75	5.25	1.75
	280	1.75	1.75	5.25	1.75
	300	2.00	2.00	6.00	2.00
4	20	1.75	1.55	6.20	1.55
	40	3.35	2.85	11.40	2.85
	60	4.25	3.65	14.60	3.65
	80	6.15	5.40	21.60	5.40
	100	6.80	5.75	23.00	5.75
	120	6.75	5.70	22.80	5.70
	140	8.15	6.80	26.31	6.58
	160	8.55	7.60	30.40	7.60
	180	10.30	8.90	35.60	8.90
	200	10.25	9.00	36.00	9.00
	220	10.65	9.25	37.00	9.25
	240	12.15	10.55	42.20	10.55
	260	13.40	11.65	46.60	11.65
	280	14.85	13.10	52.40	13.10
	300	15.35	13.35	53.40	13.35
5	20	5.80	4.05	20.25	4.05
	40	11.15	7.80	39.00	7.80
	60	15.40	11.15	55.75	11.15
	80	19.80	13.65	68.25	13.65
	100	23.60	16.85	84.25	16.85
	120	26.20	19.45	97.25	19.45
	140	29.10	20.90	104.25	20.85
	160	31.40	22.75	113.50	22.70
	180	38.90	28.70	143.50	28.70
	200	40.45	30.20	151.00	30.20
	220	43.05	32.40	161.50	32.30
	240	47.80	35.65	178.00	35.60
	260	52.10	38.60	193.00	38.60
	280	54.55	39.60	198.00	39.60
	300	59.25	43.80	219.00	43.80

maximal compared to the optimal n-soft domatic partition. Meaning that on average the number of incompletely covered nodes for the maximal n-soft domatic partition is 0.04% lower compared to the optimal n-soft domatic partition. In Tables 3, 4, 5 and 6 absolute values are shown that give an impression on the behaviour of the number of incompletely covered nodes and the number of missing coverages resulting from the maximal and optimal n-soft domatic partitions.

For graphs in SG_2, we adapted the graphs from SG_1 to be bridge-free. Our expectation was that this property has a significant impact on the computation time and on the quality of results regarding the number of missing coverages and incompletely covered nodes. Despite our expectations, the results yield that there exists no notable difference between the quality of results and the computation time. Our simulation case studies demonstrate that the elimination of bridges does not imply a measurable effect on the computation time necessary to obtain optimal and maximal n-soft domatic partitions.

For all graphs, we have been able to compute the optimal and maximal n-soft domatic partitions to optimality. Moreover, our results illustrated that the optimal n-soft domatic partitions exhibit an almost similar number of incompletely covered nodes as the maximal n-soft domatic partition. The number of missing coverages on the other hand increases significantly for the maximal n-soft domatic partitions in contrast to the optimal n-soft domatic partitions. The computation time seems to increase significantly with an rising average node degree. All together, the tests revealed that for all considered large-scale static homogeneous WSNs, the computation of maximal and optimal n-soft domatic partitions is possible and yields an optimal solution.

IX. APPLICATION

There are multiple applications for our partitioning scheme to contribute to the security of large-scale static homogeneous WSNs. Our partitioning schemes facilitate an equitable

TABLE 6. Results for average node degree deg = 6, partition sizes $n \in \{3, 4, 5\}$ and node numbers $|V| = \{20, 40, \ldots, 300\}$ for optimal and maximal n-soft domatic partitions. Showing the mean of the number of missing coverages $\overline{e_{\text{miss\_cov}}}$ and number of incompletely covered nodes $\overline{e_{\text{inc\_nodes}}}$.

n	$\lvert V\rvert$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$	$\overline{e_{\text{miss\_cov}}}$	$\overline{e_{\text{inc\_nodes}}}$
		opt n-soft domatic partition		max n-soft domatic partition	
3	20	0.10	0.10	0.31	0.11
	40	0.15	0.15	0.45	0.15
	60	0.30	0.30	0.90	0.30
	80	0.30	0.30	0.90	0.30
	100	0.20	0.20	0.60	0.20
	120	0.35	0.35	1.05	0.35
	140	0.50	0.50	1.50	0.50
	160	0.45	0.45	1.35	0.45
	180	0.40	0.40	1.20	0.40
	200	0.80	0.80	2.40	0.80
	220	0.85	0.85	2.55	0.85
	240	0.80	0.80	2.40	0.80
	260	0.90	0.90	2.70	0.90
	280	0.90	0.90	2.70	0.90
	300	0.80	0.80	2.40	0.80
4	20	0.75	0.65	2.60	0.65
	40	1.55	1.40	5.60	1.40
	60	2.15	1.85	7.40	1.85
	80	2.35	2.05	8.20	2.05
	100	2.70	2.50	10.00	2.50
	120	3.50	3.15	12.60	3.15
	140	3.65	3.15	12.60	3.15
	160	3.70	3.25	13.00	3.25
	180	4.10	3.70	14.80	3.70
	200	4.55	3.75	15.00	3.75
	220	5.45	4.60	18.40	4.60
	240	5.45	4.65	18.60	4.65
	260	6.15	5.25	21.00	5.25
	280	6.20	5.30	21.20	5.30
	300	6.85	6.05	24.20	6.05
5	20	2.75	2.00	10.00	2.00
	40	6.45	4.90	24.50	4.90
	60	8.65	6.50	32.50	6.50
	80	9.80	7.50	37.50	7.50
	100	11.50	8.80	44.00	8.80
	120	14.30	10.80	54.00	10.80
	140	15.20	11.55	57.75	11.55
	160	14.90	11.20	56.00	11.20
	180	18.25	14.15	70.75	14.15
	200	18.70	14.20	71.00	14.20
	220	21.45	16.00	80.00	16.00
	240	23.90	18.45	92.25	18.45
	260	23.25	17.10	85.50	17.10
	280	26.10	19.95	99.75	19.95
	300	28.40	21.55	107.75	21.55

distribution of security means, aiming to ensure the availability of n distinct security means in close proximity to each node. The selection of a security configuration is highly contingent upon the application area (topology, environmental conditions, accessibility for potential attackers), the specific security requirements, and other factors such as sensor node hardware and network lifespan. When designing and practically implementing a security framework which combines an ensemble of security features it is essential to assess possible emerging vulnerabilities. Here, we are drawing a rough picture on how to utilise our proposed partitioning schemes for a straightforward ensemble security framework. Hence, we will not examine all the implementation details and skip a comprehensive security analysis to determine arising vulnerabilities. We contemplate a large-scale static WSN deployed in a forest for the purpose of environmental monitoring, comprising several hundred nodes. The data's confidentiality within this network is not paramount due to the low sensitivity of individual measurement data. However, the integrity and authenticity of the data is crucial for detecting dangers to the ecosystem and potentially extreme events (e.g., wildfires). Therefore, we create an area of application and elucidate the choice of the two security means to be distributed in the WSN contributing to those requirements. We outline a combination of intrusion detection with an agent-based rerouting concept [67] and an information hiding scheme (invisible watermarking/steganography) [68], [69], [70], [71], [72]. The *Antilizer*, as proposed in [67] is a network-level IDS and automated trust-based response system. It uses an agent-based notification (ANT) scheme to detect malicious behaviour. Therefore, each node builds a trust-model of its neighbours. The trust-model is used to make routing decisions at each node. The ANTs are sent to the BS and notify intermediate nodes along the way that routing changes are the result of malicious behaviour. To detect those changes in the first place, nodes overhear their one-hop neighbours. Each node self-collects information about transmissions, receptions and further communication events ignoring the untrusted message content. Additionally, the trust-model can be adjusted by responses of the BS. The corresponding routing decisions are then made using the trust model. The *Antilizer* contributes to the node integrity and authenticity by detecting malicious behaviour and notifying the BS. As reaction, the BS utilises a filtering mechanism to determine the validity of notifications. We combine the *Antilizer* scheme with an invisible watermarking scheme. The combination implicates that not all nodes maintain their own trust-model and therefore, are required to rely on trust-models of neighbouring nodes.

To justify our choice of an information hiding scheme, we need to delve into the necessary background and provide some context. We consider a lightweight information hiding scheme with focus on data integrity and the protection of the source of origin of transmitted data (authenticity). Suitable concepts are fragile invisible watermarking or steganographic information hiding schemes [68]. We deem a fragile scheme to be satisfactory, assuming that the absence of a watermark serves as a sufficient indication of tampered data.

We propose to utilise pseudo-image watermarking to allow aggregation of data [68], [73]. Already watermarked data, presented as pseudo-images, that traverse through an additional watermarking/aggregation node are exclusively forwarded to safeguard the fragile watermark. Alternatively, an aggregation tolerant watermark can be considered [74]. To reach the critical amount of data necessary to create a pseudo-image, either forwarded data of other nodes or, in the case of insufficient forwarded data, the node's own data are collected and aggregated. The temporal accumulation of own data potentially necessitates the usage of time codes or an order as meta information for the evaluation at the BS. The watermarking scheme we propose does not yet exist in the literature, but its components and their integration to some extent do [73]. Modifications to certain levels are imperative to ensure their

applicability in WSNs, where only a subset of nodes employ them.

Using our partitioning scheme, we distribute the proposed security means in a WSN. We compute the n-soft domatic partition of nodes for partition sizes n of 2 and 3. The sets in a partition can differ in size affecting the ratio of different security means associated with them. Further, the union of dominating sets of a partition results in a dominating set. We can use this property to control the ratio of nodes implementing selected security means. Even so, we intend to distribute two types of security means, if a partition size of 3 is achievable with a small number of missing coverages, it potentially provides a certain amount of control over the ratio between watermarking/aggregation nodes and *Antilizer* nodes. Further, we provided auxiliary tools in the preceding sections to adjust the set sizes in n-soft domatic partitions. Under certain assumptions, it is helpful to set selected nodes, e.g. nodes on the outer rim of a WSN, to a specific security mean. Such adjustments can be considered in the partitioning by constraining the corresponding variables to a specific set of a partition and computing the remaining variables accordingly. In our case, the nodes on the outer rim can potentially be bound to the watermarking/aggregation scheme. We assume that outer nodes are more prone to attacks in the considered scenario. Another positive effect is the limited frequency of data transmissions, since those nodes are assumed to aggregate mostly own data. Further, the watermarking allows an early detection of attacks on the data integrity. Subsequently, the nodes utilising *Antilizer* would be placed on inner nodes of the network, enabling them to potentially overhear an increased number of nodes.

In general, nodes are gathering data and forwarding them hop-by-hop to a BS. The BS validates the data by comparing measurements with past measurement data as well as the variance of data in proximity. To have reliable data in the proximity of nodes, we watermark data at certain nodes and mark the data collected by those nodes specifically. The presence/absence of a watermark is validated by the BS. Since, we expect measurement data to be similar in local proximity (or their respective gradients are smooth to some degree), the BS validates the data by comparing them to watermarked measurements. While missing watermarks lead to a decrease in reputation of nodes, the presence can restore or maintain a level of reputation. Parallel to the watermarking scheme, the *Antilizer* nodes provide a timely threat reaction by monitoring network-level behaviour, adjusting the reputation of neighbouring nodes and influence routing decisions based on those. Those information is passed by agents (ANTs) to the BS for further evaluation. The BS can then create an overarching picture of network metric changes and measurement deviations to determine suitable adjustments. The evaluation result of the BS can be used to adjust trust-models of nodes. Routing decisions of nodes not implementing *Antilizer* rely on BS responses in combination with trust-models determined by neighbouring nodes.

With limited resources, it is impossible to fend off all possible attacks and to achieve perfect security. However, the combination of security means can strengthen various security features and provide resilience against a certain type of attackers. In general, we know from nature that versatility is a key to a strong and efficient, but imperfect security [75].

X. CONCLUSION AND FUTURE PROSPECTS

In this paper, we determined a distribution of security means based on the concept of a neighbourhood watch introduced by Langendörfer [4]. The concept aims to maximise the spectrum of security threats a large-scale static homogeneous WSN can detect or avert while minimising the load that will be put on individual nodes. To develop a complex security framework of this kind, there are several steps that have to be taken. Here, we introduced a graph partitioning scheme for the node distribution. While sleep scheduling themes allow partitioning schemes that determine non-disjoint minimal dominating sets, we were looking for a partition that creates disjoint partitions that approximate the definition of dominating sets. Therefore, we defined two terms, the number of missing coverages and the number of incompletely covered nodes. To determine the partitions based on those terms, we introduced two $0-1$ LPs for the maximal n-soft domatic partition and for the optimal n-soft domatic partition. Furthermore, we proposed several variations of those LPs allowing advanced distributions of security means that fit to the needs of differently equipped WSNs and to different levels of security threats. To validate the computability of the proposed NP hard $0-1$ LPs, a test setup has been designed. On its basis, we have verified the computability on graphs as representations of large-scale static homogeneous WSNs. This also implied the need for a suitable graph generator that enables to create realistic WSN models. The introduced graph generator allows to control the properties of resulting graphs via its input parameters, allowing an improved comparability of test results. Further, our graph generator aims at the creation of connected graphs as far as possible by purposive construction from the beginning. This feature avoids expensive trial-and-error strategies by iterating over a large number of insufficient graphs. Along with algorithmic design, we had to cope with the requirement that the constructive generation of connected graphs does not interfere with the desired uniform node distribution. As a result, we developed a new graph generator for λ-precision UDGs introduced in this publication. Its Python source code is available from the first author upon request. Additionally, further major properties we are able to control to some extent are the average node degree, the local clustering coefficient and the general coverage of the generation plane. Beyond, we provide several methods to further adapt the resulting graphs while maintaining their characteristics as representations of WSNs.

To evaluate the introduced LPs, we introduced a generator for λ-precision UDGs. The generator enabled us to evaluate

which parameters affect the computation time at most by providing appropriate graphs. Our results have shown that the computation time is affected the most by the average node degree and the desired partition size. In the range of 20 to 300 nodes within a graph, the node degree has almost only a linear effect on the computation time. The application in the last section intends to visualise the generic applicability of our partitioning scheme for the timely efficient design of complex and cooperative security configurations.

We have presented a number of variations towards the $0-1$ LPs allowing distributions of a fixed number of security means per node and even distributions based on the performance cost of each security mean. The latter one allows to distribute varying numbers of security means per node based on their individual resource requirements. Our future work will address a number of applications with the goal to design a DSE framework for requirement-based cooperative/collaborative security configurations for WSNs. The process can conclude by tailoring the chosen security means to align with the requirements of a specific WSN, limiting the size of the design space as outlined in [4], utilising the proposed CADRT. Our graph generator is suited for adaptation to determination of automatic node distributions for given topologies. Those adaptations include the consideration of obstacles and elevation profiles as well as node capabilities. The current version of the λ-precision UDG generator is effortless adjustable to arbitrary formed areas, automatically identifying appropriate uniform node distributions to achieve desired coverages. Only if desired coverages for a set of given sensor nodes and its capabilities are technically not achievable, manual intervention is necessary, e.g. adjusting the size of the given distribution area.

REFERENCES

[1] D. Ki-Aries, H. Dogan, S. Faily, P. Whittington, and C. Williams, "From requirements to operation: Components for risk assessment in a pervasive system of systems," in *Proc. IEEE 25th Int. Requirement Eng. Conf. Workshops (REW)*, Sep. 2017, pp. 83–89.

[2] R. Jiang, J. Luo, and X. Wang, "An attack tree based risk assessment for location privacy in wireless sensor networks," in *Proc. 8th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2012, pp. 1–4.

[3] S. Garg, G. S. Aujla, N. Kumar, and S. Batra, "Tree-based attack–defense model for risk assessment in multi-UAV networks," *IEEE Consum. Electron. Mag.*, vol. 8, no. 6, pp. 35–41, Nov. 2019.

[4] P. Langendörfer, "Security engineering for cyber physical systems," in *Proc. 22nd EUROMICRO Conf. Digit. Syst. Design Softw. Eng. Adv. Appl.-Session Work Prog.*, 2019.

[5] B. Bhushan, K. Kaushik, and G. Sahoo, "Concealed data aggregation with dynamic intrusion detection system to remove vulnerabilities in wireless sensor networks," in *Proc. CS IT Conf.*, vol. 6, 2016, pp. 81–96.

[6] M. Valero, S. S. Jung, A. S. Uluagac, Y. Li, and R. Beyah, *Di-Sec: A Distributed Security Framework for Heterogeneous Wireless Sensor Networks*. Piscataway, NJ, USA: IEEE Press, 2012.

[7] H. Saxena, C. Ai, M. Valero, Y. Li, and R. Beyah, "DSF—A distributed security framework for heterogeneous wireless sensor networks," in *Proc. Mil. Commun. Conf.*, Oct. 2010, pp. 1836–1843.

[8] R. Sharma and V. A. Athavale, "Survey of intrusion detection techniques and architectures in wireless sensor networks," *Int. J. Adv. Netw. Appl.*, vol. 10, no. 4, pp. 3925–3937, 2019.

[9] W. Li, W. Meng, and L. F. Kwok, "Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 280–305, 1st Quart., 2022.

[10] Y. Du, J. Xia, J. Ma, and W. Zhang, "An optimal decision method for intrusion detection system in wireless sensor networks with enhanced cooperation mechanism," *IEEE Access*, vol. 9, pp. 69498–69512, 2021.

[11] S. Peter, K. Piotrowski, and P. Langendorfer, "In-network-aggregation as case study for a support tool reducing the complexity of designing secure wireless sensor networks," in *Proc. 33rd IEEE Conf. Local Comput. Netw. (LCN)*, May 2008, pp. 778–785.

[12] V. Cionca, T. Newe, and V. T. Dădârlat, "A tool for the security configuration of sensor networks," *J. Phys., Conf. Ser.*, vol. 178, Jan. 2009, Art. no. 012043.

[13] S. Peter, "Tool-supported development of secure wireless sensor networks," Ph.D. thesis, BTU Cottbus-Senftenberg, 2011.

[14] S. Peter and P. Langendörfer, "Tool-supported methodology for component-based design of wireless sensor network applications," in *Proc. IEEE 36th Annu. Comput. Softw. Appl. Conf. Workshops*, Jul. 2012, pp. 526–531.

[15] V. Cionca, T. Newe, and V. T. Dădârlat, "Configuration tool for a wireless sensor network integrated security framework," *J. Netw. Syst. Manage.*, vol. 20, pp. 417–452, Jan. 2012.

[16] S. Lange, J. Lösche, and K. Piotrowski, "Tool-supported requirements-based topology design for wireless sensor networks," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2014, pp. 1043–1047.

[17] S. Peter and T. Givargis, "Component-based synthesis of embedded systems using satisfiability modulo theories," *ACM Trans. Design Autom. Electron. Syst.*, vol. 20, no. 4, pp. 1–27, 2015.

[18] D. Kirov, P. Nuzzo, R. Passerone, and A. Sangiovanni-Vincentelli, "ArchEx: An extensible framework for the exploration of cyber-physical system architectures," in *Proc. 54th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2017, pp. 1–6.

[19] D. Kirov, P. Nuzzo, R. Passerone, and A. Sangiovanni-Vincentelli, "Optimized selection of wireless network topologies and components via efficient pruning of feasible paths," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[20] P. Nuzzo, N. Bajaj, M. Masin, D. Kirov, R. Passerone, and A. L. Sangiovanni-Vincentelli, "Optimized selection of reliable and cost-effective safety-critical system architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2109–2123, Oct. 2020.

[21] N. Tsiskaridze, M. Strange, M. Mann, K. Sreedhar, Q. Liu, M. Horowitz, and C. Barrett, "Automating system configuration," in *Proc. Formal Methods Comput. Aided Design (FMCAD)*, Oct. 2021, pp. 102–111.

[22] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Cham, Switzerland: Springer, 1972, pp. 85–103.

[23] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using NetworkX," Dept. Los Alamos Nat. Lab. (LANL), Los Alamos, NM, USA, Tech. Rep., 2008.

[24] W. E. Hart, J.-P. Watson, and D. L. Woodruff, "Pyomo: Modeling and solving mathematical programs in Python," *Math. Program. Comput.*, vol. 3, no. 3, pp. 219–260, 2011.

[25] *Gurobi Optimizer Reference Manual*, Gurobi Optim., 2022.

[26] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[27] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 1st Quart., 2014.

[28] A. Stetsko, L. Folkman, and V. Matyaš, "Neighbor-based intrusion detection for wireless sensor networks," in *Proc. 6th Int. Conf. Wireless Mobile Commun.*, Sep. 2010, pp. 420–425.

[29] F. Liu, X. Cheng, and D. Chen, "Insider attacker detection in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, May 2007, pp. 1937–1945.

[30] I. Krontiris, T. Giannetsos, and T. Dimitriou, "LIDeA: A distributed lightweight intrusion detection architecture for sensor networks," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, 2008, pp. 1–10.

[31] M. Riecker, "Lightweight intrusion detection in wireless sensor networks," Ph.D. thesis, Technische Universität Darmstadt, 2015.

[32] R. Laskar and J. Lyle, "Fall colouring of bipartite graphs and Cartesian products of graphs," *Discrete Appl. Math.*, vol. 157, no. 2, pp. 330–338, Jan. 2009.

[33] M. R. Garey and D. S. Johnson, *Computers and Intractability*, vol. 174. San Francisco, CA, USA: Freeman, 1979.

[34] P. Floreen, P. Kaski, and J. Suomela, "A distributed approximation scheme for sleep sceduling in sensor networks," in *Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2007, pp. 152–161.

[35] K. Islam, S. G. Akl, and H. Meijer, "Maximizing the lifetime of wireless sensor networks through domatic partition," in *Proc. IEEE 34th Conf. Local Comput. Netw.*, Oct. 2009, pp. 436–442.

[36] R. Misra and C. Mandal, "Efficient clusterhead rotation via domatic partition in self-organizing sensor networks," *Wireless Commun. Mobile Comput.*, vol. 9, no. 8, pp. 1040–1058, 2009.

[37] B. Mumey, K. Spendlove, and B. Zhu, "Extending the lifetime of a WSN by partial covers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 1779–1783.

[38] S. V. Pemmaraju and I. A. Pirwani, "Energy conservation via domatic partitions," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, May 2006, pp. 143–154.

[39] J. Yu, Q. Zhang, D. Yu, C. Chen, and G. Wang, "Domatic partition in homogeneous wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 37, pp. 186–193, Feb. 2014.

[40] D. Rall, "A fractional version of domatic number," *Congressus Numerantium*, vol. 74, pp. 100–106, Sep. 1990.

[41] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan, "Approximating the domatic number," *SIAM J. Comput.*, vol. 32, no. 1, pp. 172–195, 2002.

[42] A. Czygrinow, M. Hańć kowiak, E. Szymałska, W. Wawrzyniak, and M. Witkowski, "Improved distributed local approximation algorithm for minimum 2-dominating set in planar graphs," *Theor. Comput. Sci.*, vol. 662, pp. 1–8, Feb. 2017.

[43] H. Liang, "The algorithmic complexity of k-domatic partition of graphs," in *Proc. IFIP Int. Conf. Theor. Comput. Sci.* Cham, Switzerland: Springer, 2012, pp. 240–249.

[44] A. S. Rao and C. P. Rangan, "Linear algorithm for domatic number problem on interval graphs," *Inf. Process. Lett.*, vol. 33, no. 1, pp. 29–33, 1989.

[45] D. Mahjoub, A. Leskovskaya, and D. W. Matula, "Approximating the independent domatic partition problem in random geometric graphs-an experimental study," in *Proc. CCCG*, 2010, pp. 195–198.

[46] S. Pandit, S. V. Pemmaraju, and K. Varadarajan, "Approximation algorithms for domatic partitions of unit disk graphs," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Cham, Switzerland: Springer, 2009, pp. 312–325.

[47] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks—A comprehensive survey," *Comput. Commun.*, vol. 36, no. 2, pp. 121–134, 2013.

[48] P. Erdos and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.

[49] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[50] H. Kenniche and V. Ravelomananana, "Random geometric graphs as model of wireless sensor networks," in *Proc. 2nd Int. Conf. Comput. Autom. Eng. (ICCAE)*, vol. 4, Feb. 2010, pp. 103–107.

[51] E. N. Gilbert, "Random plane networks," *J. Soc. Ind. Appl. Math.*, vol. 9, no. 4, pp. 533–543, 1961.

[52] M. Jorgic, M. Hauspie, D. Simplot-Ryl, and I. Stojmenovic, "Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks," in *Proc. Medit. Ad Hoc Netw. Workshop*, 2004, p. 12.

[53] F. A. Onat, I. Stojmenovic, and H. Yanikomeroglu, "Generating random graphs for the simulation of wireless ad hoc, actuator, sensor, and internet networks," *Pervas. Mobile Comput.*, vol. 4, no. 5, pp. 597–615, Oct. 2008.

[54] V. K. Verma, S. Singh, and N. P. Pathak, "Towards comparative evaluation of trust and reputation models over static, dynamic and oscillating wireless sensor networks," *Wireless Netw.*, vol. 23, pp. 335–343, Aug. 2017.

[55] A. Ahmed, K. A. Bakar, M. I. Channa, K. Haseeb, and A. W. Khan, "A survey on trust based detection and isolation of malicious nodes in ad-hoc and sensor networks," *Frontiers Comput. Sci.*, vol. 9, no. 2, pp. 280–296, Apr. 2015.

[56] M. Strasser and H. Vogt, "Autonomous and distributed node recovery in wireless sensor networks," in *Proc. 4th ACM workshop Secur. Ad Hoc Sensor Netw.*, Oct. 2006, pp. 113–122.

[57] K.-S. Hung, C.-F. Law, K.-S. Lui, and Y.-K. Kwok, "On attack-resilient wireless sensor networks with novel recovery strategies," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2009, pp. 1–6.

[58] H. Hayouni, M. Hamdi, and T.-H. Kim, "A survey on encryption schemes in wireless sensor networks," in *Proc. 7th Int. Conf. Adv. Softw. Eng. Appl.*, Dec. 2014, pp. 39–43.

[59] S. El Khediri, "Wireless sensor networks: A survey, categorization, main issues, and future orientations for clustering protocols," *Computing*, vol. 104, no. 8, pp. 1775–1837, Aug. 2022.

[60] M. Pawar and J. Agarwal, "A literature survey on security issues of WSN and different types of attacks in network," *Indian J. Comput. Sci. Eng.*, vol. 8, no. 2, pp. 80–83, 2017.

[61] R. Kumar, S. Tripathi, and R. Agrawal, "An analysis and comparison of security protocols on wireless sensor networks (WSN)," in *Design Frameworks for Wireless Networks*. Cham, Switzerland: Springer, 2020, pp. 3–21.

[62] L. K. Ketshabetswe, A. M. Zungeru, M. Mangwala, J. M. Chuma, and B. Sigweni, "Communication protocols for wireless sensor networks: A survey and comparison," *Heliyon*, vol. 5, no. 5, May 2019, Art. no. e01591.

[63] G. Kovásznai, B. Erdélyi, and C. Biró, "Investigations of graph properties in terms of wireless sensor network optimization," in *Proc. IEEE Int. Conf. Future IoT Technol. (Future IoT)*, Jan. 2018, pp. 1–8.

[64] C. Basaran, K.-D. Kang, and H. S. Mehmet, "Hop-by-hop congestion control and load balancing in wireless sensor networks," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 448–455.

[65] Y. Ding, C. Wang, and L. Xiao, "An adaptive partitioning scheme for sleep scheduling and topology control in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 9, pp. 1352–1365, Sep. 2009.

[66] D. Saha and N. Das, "Distributed area coverage by connected set cover partitioning in wireless sensor networks," 2014, *arXiv:1401.8152*.

[67] I. Tomić, P.-Y. Chen, M. J. Breza, and J. A. McCann, "Antilizer: Run time self-healing security for wireless sensor networks," in *Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Nov. 2018, pp. 107–116.

[68] O. Evsutin, A. Melman, and A. A. A. El-Latif, "Overview of information hiding algorithms for ensuring security in IoT based cyber-physical systems," in *Security and Privacy Preserving for IoT and 5G Networks: Techniques, Challenges, and New Directions*. Cham, Switzerland: Springer, 2022, pp. 81–115.

[69] Z. Wang, "Information hiding data security based on wireless sensor networks," in *Proc. 4th Int. Conf. Electr. Electron. Eng. Comput. Sci. (ICEEECS)*, 2016, pp. 240–245.

[70] J. M. de Fuentes, J. Blasco, A. I. González-Tablas, and L. González-Manzano, "Applying information hiding in VANETs to covertly report misbehaving vehicles," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 2, Feb. 2014, Art. no. 120626.

[71] S. Sultana, M. Shehab, and E. Bertino, "Secure provenance transmission for streaming data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1890–1903, Aug. 2013.

[72] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 3, pp. 44–57, 3rd Quart., 2007.

[73] Y. Xiao and G. Gao, "Digital watermark-based independent individual certification scheme in WSNs," *IEEE Access*, vol. 7, pp. 145516–145523, 2019.

[74] A. S. Panah, R. van Schyndel, T. Sellis, and E. Bertino, "In the shadows we trust: A secure aggregation tolerant watermark for data streams," in *Proc. IEEE 16th Int. Symp. A World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2015, pp. 1–9.

[75] B. Förster, P. Langendörfer, and T. Hinze, "Novel approach to a plant inspired distributed security scheme for wireless sensor networks," in *Proc. 12th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2023, pp. 1–6.

BENJAMIN FÖRSTER was born in Frankfurt an der Oder, Germany, in 1991. He received the B.Sc. and M.Sc. degrees in computer science from Brandenburg University of Technology Cottbus-Senftenberg (B-TU), Cottbus, Brandenburg, Germany, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with IHP, Frankfurt an der Oder.

His major field of study is security in wireless embedded resource constraint systems. Prior to that, he was a Research Assistant with IHP and B-TU, and a Software Developer with Astronergy Solarmodule GmbH. His research interests include the security of wireless resource-constrained embedded systems and the application of biologically inspired computing principles.

PETER LANGENDÖRFER received the Diploma and Ph.D. degrees in computer science from B-TU, Cottbus, Brandenburg, Germany, in 1995 and 2001, respectively.

He has been with IHP, Frankfurt an der Oder, since 2000, where he is currently leading the Wireless Systems Department. From 2012 to 2020, he was leading the Chair for Security in Pervasive Systems, Brandenburg University of Technology Cottbus-Senftenberg. Since 2020, he has been the Chair of the Wireless Systems, Brandenburg University of Technology Cottbus-Senftenberg. His portfolio includes over 150 peer-reviewed technical articles that he has published, as well as 17 filed patents, with 11 already granted. His research interests include security for resource constraint devices, low-power protocols, efficient implementations of AI, and resilience.

Dr. Langendörfer is a member of "Gesellschaft für Informatik." He is an Associate Editor of IEEE Access and *Peer-to-Peer Networking and Applications*. He was the Guest Editor of many renowned journals, such as *Wireless Communications and Mobile Computing* (Wiley) and *ACM Transactions on Internet Technology*.

THOMAS HINZE was born in Halle (Saale), Germany, in 1971. He received the Diploma degree in computer science and the Ph.D. degree in engineering sciences from Dresden University of Technology, Germany, in 1997 and 2002, respectively.

In 2012, he became a Senior University Lecturer with Friedrich Schiller University Jena, Germany, along with his professorial dissertation. He is author of three text books, more than 90 publications, and holds two patents. His research interests include principles of biological and biologically inspired information processing, such as molecular computing, membrane computing, distributed computing, evolutionary computing, and systems biology.

Dr. Hinze is an Editorial Board Member of the *Journal of Membrane Computing* (Springer). He joined the Association for Computing Machinery (ACM) and engages in the International Membrane Computing Society (IMCS) and the Institute for Systems and Technologies of Information, Control and Communication (INSTICC). He was a recipient of five best paper awards.