Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Survey on Secure Group Communication Schemes with Focus on IoT Communication

THOMAS PRANTL[1], TIMO ZECK[1], ANDRE BAUER[1], PETER TEN[1], DOMINIK PRANTL[1], ALA EDDINE BEN YAHYA[1], LUKAS IFFLAENDER[1], ALEXANDRA DMITRIENKO[1], CHRISTIAN KRUPITZER[2], and SAMUEL KOUNEV[1]

[1]Julius-Maximilians-Universität, Würzburg, Germany (e-mail: {firstname.lastname@uni-wuerzburg.de})
[2]University of Hohenheim & Computational Science Lab, Hohenheim, Germany (e-mail: christian.krupitzer@uni-hohenheim.de)

Corresponding author: Thomas Prantl (e-mail: thomas.prantl@uni-wuerzburg.de).

This research has been funded by the Federal Ministry of Education and Research of Germany in the framework KMU-innovativ - Verbundprojekt: Secure Internet of Things Management Platform - SIMPL (project number 16KIS0852) [1]. This publication was also supported by the Open Access Publication Fund of the University of Würzburg.

ABSTRACT A key feature for Internet of Things (IoT) is to control what content is available to each user. To handle this access management, encryption schemes can be used. Due to the diverse usage of encryption schemes, there are various realizations of 1-to-1, 1-to-n, and n-to-n schemes in the literature. This multitude of encryption methods with a wide variety of properties presents developers with the challenge of selecting the optimal method for a particular use case, which is further complicated by the fact that there is no overview of existing encryption schemes. To fill this gap, we envision a cryptography encyclopedia providing such an overview of existing encryption schemes. In this survey paper, we take a first step towards such an encyclopedia by creating a sub-encyclopedia for secure group communication (SGC) schemes, which belong to the n-to-n category. We extensively surveyed the state-of-the-art and classified 47 different schemes. More precisely, we provide (i) a comprehensive overview of the relevant security features, (ii) a set of relevant performance metrics, (iii) a classification for secure group communication schemes, and (iv) workflow descriptions of the 47 schemes. Moreover, we perform a detailed performance and security evaluation of the 47 secure group communication schemes. Based on this evaluation, we create a guideline for the selection of secure group communication schemes.

INDEX TERMS Secure Group Communication (SGC), SGC Classification, Security Features, Performance Metrics, Communication Costs, Computation Costs, Guidelines

I. INTRODUCTION

The recent emergence of rapid network technologies has led to a significant increase in the Internet speed and connectivity [2]. As electronic communications and information services become more sophisticated, applications involving multiple entities grow in importance [3]. Various applications have emerged in which multiple users are simultaneously connected, such as video conferences, Pay-Tv, group chats on social networks, or online games. Also, devices communicating in smart environments [2], [4] belong in this category. Such applications require efficient distribution of messages between the many involved participants with different requirements concerning the security level.

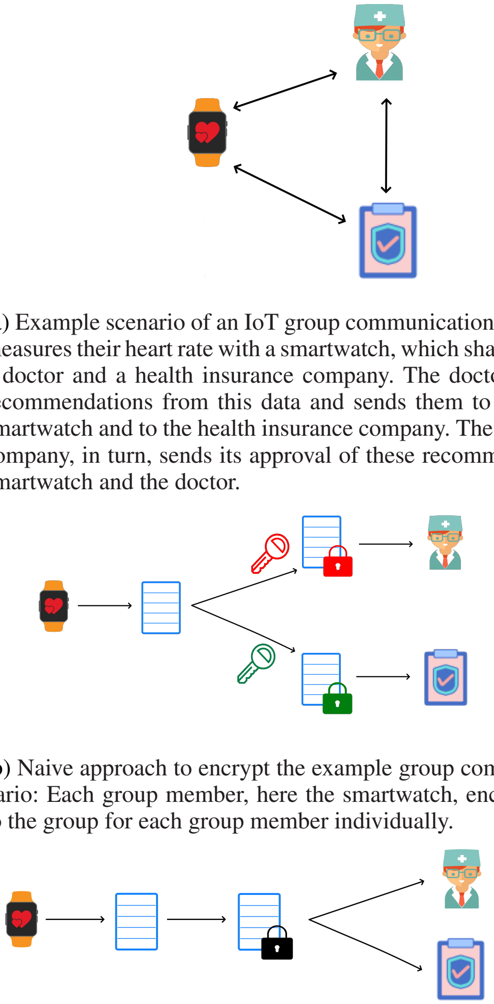
The term Internet of Things (IoT) refers to the interconnection of objects (things) that communicate via networks using various identifying and communication technologies [5]. The steadily growing numbers of online services and IoT devices gather and share vast amounts of data [6]. In addition, more and more IoT applications involving group communication permeate various important areas of our everyday lives. These areas include smart factory, remote healthcare, smart home, smart mobility, traffic management, and more [5]. The new 5G technology immensely accelerates data transfer and allows further scaling of the connectivity process [6]. The introduction of 5G will result in faster broadband speeds and more reliable mobile networks and accelerate progress in smart cities, smart vehicles, and smart manufacturing [7]. These developments open up new possibilities for numerous applications involving multiple communicating parties.

However, this promising digital transformation will not reach its potential unless consumers can trust in the privacy and security of their data [6]. IoT applications gather and share vast amounts of data, including sensitive data, for example, monitored healthcare information [5]. Therefore, it is important that users are always in control of their data and can control access to it. Unfortunately, in the past companies developing IoT devices often fail to address this need for security and privacy [6]. IoT devices had been often deployed without even bearing security in mind [8]. This led to the largest Distributed Denial of Service (DDoS) attack in 2016, executed by thousands of compromised IoT devices transformed into a botnet to knock down various Internet services, such as Netflix or Spotify [8].

To prevent such attacks in the future and to better protect users' data, it is essential to ensure more security by encrypting IoT communication. In addition, legal regulations such as the General Data Protection Regulation [9] now indirectly prescribe the use of encryption. This is not an easy task for developers, as group communication (n-to-n communication) is more difficult to encrypt than 1-to-1 communication. In n-to-n communication, messages must be encrypted for a group of recipients. An example n-to-n communication scenario in the IoT environment consisting of a smartwatch, a doctor, and a health insurance company is illustrated in Figure 1a. A naive approach to encrypt such n-to-n communication would be for the sender of a message to encrypt the messages for each recipient individually, as shown in Figure 1b. However, this approach would be very inefficient because the encryption overhead would grow linearly with the group size, which would not be feasible on resource-constrained IoT devices. A more efficient way to handle n-to-n encryption would be to encrypt a message only once so that each group member can decrypt the message, as shown in Figure 1c. In literature, there are numerous so-called Secure Group Communication (SGC) schemes [5], [10] providing precisely this functionality.

However, the large number of different SGC schemes complicates the selection of a specific scheme for a given use case. One of the reasons is fundamental differences in the process architecture and workflow. For example, some schemes require the presence of a trusted third party (e.g., [12] and [13]) while others do not (e.g., [14] and [15]). In addition, the schemes offer a wide variety of security features and differ in terms of performance. Thus, developers need an appropriate overview of the schemes' properties and workflows as well as guidelines supporting the selection of a specific scheme for a given use case.

However, there is only one survey by Cheikhrouhou [10] that provides both an overview and a guideline for selecting a particular scheme. Specifically, Cheikhrouhou's overview surveys the analysis of 22 schemes based on ten different aspects. Based on this analysis, Cheikhrouhou recommends one of the 22 schemes using three decision criteria. In this survey paper, we extend the survey of Cheikhrouhou by analyzing additional 25 schemes considering 12 different aspects. We use this expanded knowledge base to recommend a concrete scheme that considers one additional decision criterion besides Cheikhrouhou's decision criteria. Moreover, when recommending a scheme, we consider not only its performance but also its security features.



(a) Example scenario of an IoT group communication scenario: A user measures their heart rate with a smartwatch, which shares this data with a doctor and a health insurance company. The doctor derives health recommendations from this data and sends them to the user via the smartwatch and to the health insurance company. The health insurance company, in turn, sends its approval of these recommendations to the smartwatch and the doctor.

(b) Naive approach to encrypt the example group communication scenario: Each group member, here the smartwatch, encrypts a message to the group for each group member individually.

(c) Instead of the naive approach of encrypting the message individually for each group member, it is more efficient to use a Secure Group Communication Scheme, which allows a message to be encrypted only once so that any group member can decrypt it.

FIGURE 1: Illustration of a group communication scenario in (a) and its naive or efficient encryption in (b) or (c), respectively. (Used image sources [11])

More specifically, the contributions of our survey paper are:

1) Thoroughful literare analysis of the state-of-the-art for SGC schemes.
2) Analysis and comparison of 47 SGC schemes based on twelve performance metrics and security features.
3) Definition of selection guidelines for SGC schemes.

The remainder of this paper is structured as follows. In

Section II, we provide background information required for understanding SGC schemes. In Section III, we give an overview of related work and a delimitation to this work. Next, in Section IV, we present our survey methodology. In addition to the procedure of our literature review, we also show the security features and performance metrics we determined and the classification approach for SGC schemes. We also present the decision criteria used to create guidelines for the selection of methods. In Sections V, VI, and VII, we present the schemes we found using our survey methodology, grouped by category, and identify their security features and performance. We present the derived method selection guidelines in Section VIII. Finally, we conclude the paper in Section IX.

II. BACKGROUND

In this section, we provide the required background information on group communication schemes, security, and encryption in general. We first define SGC schemes and then describe the used cryptographic techniques.

A. SHARED SECRET—SECURE GROUP COMMUNICATION SCHEMES

According to the definition by Cheikhrouhou et al. [10] and Sakarindr et al. [16], an SGC scheme comprises two components: the *group key management* (GKM) and the *group membership management* (GMM).

Group Key Management (GKM): This component represents the fundamental security service in SGC schemes. Its purpose is to provide a secret *group key*, also known as the Traffic Encryption Key (TEK) [17], which is shared among the group members [10]. The shared group key allows to encrypt and sign group messages, authenticate members and messages, and authorize access to traffic and group resources [16]. Consequently, the strength of an SGC scheme depends mainly on the cryptographic strength of this group key and the key management protocol [16]. The GKM protocol defines how to generate, distribute, and update the group key. This group key update—also called the *rekeying process*—occurs either when the membership changes or periodically at fixed intervals [10].

Group Membership Management (GMM): This component securely specifies a group's membership operations. These operations run during the group creation process as well as the join and leave process. The GMM only defines the inclusion or exclusion of members, for example, in a list maintained by the group controller. The GMM takes over everything else that deals with corresponding key updates or the key distribution in the joining or leaving process. Moreover, when a new member joins the group, it should authenticate with the GMM before gaining access to the group. This restricted access to authorized members may mitigate potential identity-based attacks, such as impersonation, but also Denial of Service attacks. Additionally, compromised members leave the group. Executing a membership operation requires renewing the group key and potentially other keys using the GKM component.

B. CRYPTOGRAPHIC TECHNIQUES

This section describes different types of cryptographic techniques, such as symmetric or asymmetric cryptography. These techniques are prevalent in the compared SGC schemes and impact their performance and also their level of security.

1) Symmetric Cryptography

In symmetric cryptography, algorithms use the same cryptographic key for both encryption of the message and decryption of the corresponding ciphertext. They are faster than asymmetric algorithms and allow the encryption of large datasets. However, they require sophisticated mechanisms to securely distribute the secret keys to the communicating parties [18]. The only symmetric scheme used in SGC is the XOR Cipher.

2) Public-key Cryptography

Asymmetric cryptography, also known as public and private key cryptography, uses two keys: a public key for encrypting messages and a private key for decrypting ciphertexts. The public key of a communicating party is publicly available, and everybody can use it to encrypt a message. The idea of asymmetric encryption is that only the owner of the corresponding private key, which is unknown to anybody else, can decrypt the message [19]. Asymmetric algorithms are much slower than symmetric ones. In practice, this performance disadvantage of asymmetric encryption schemes is mitigated by using asymmetric encryption only to exchange a key, which is then used with higher-performance symmetric encryption. In the following, we present asymmetric schemes used for SGC.

- One-way Function: Informally described, a one-way function is a function where the computation in one direction is simple, while the computation in the reverse direction is much more difficult. More formally, it is a function f with domain X and range Y, where $f(x)$ is 'simple' to compute for all $x \in X$, but for 'virtually all' $y \in Y$ it is 'computationally unfeasible' to find any x so that $f(x) = y$ [20, pp. 1-2].
- Diffie-Hellman Key Exchange: The Diffie-Hellman (DH) Key Exchange or key agreement protocol enables two users who have not previously 'met' to establish a shared, symmetric key over an insecure channel. The original protocol uses integer operations in a multiplicative group and was susceptible to Man in the Middle attacks. Variations and updated protocols of DH have since been proposed that provide key authentication to mitigate such attacks [21].
- Elliptic Curve Cryptography (ECC): ECC is an accepted public key cryptosystem and an alternative to conventional cryptosystems such as RSA and ElGamal.

It provides the highest strength-per-bit of any other cryptosystem known today. Significantly smaller key sizes compared to other public key cryptosystems allow obtaining equivalent security. This is an ideal feature, especially for applications such as smart cards or wireless sensor networks where memory and computing power are limited. Elliptic curves apply to key agreement, digital signatures, pseudorandom generators, and other tasks [22].

3) Pseudorandom Generator (PRG)

Pseudorandom generators create seemingly random sequences of numbers in deterministic devices such as computers. Since all algorithms are strictly deterministic and therefore would be easily reversible, we require randomly chosen sequences. Therefore, PRGs must be unpredictable, and there must not exist any efficient algorithm that can predict the next 'random' bit with a probability non-negligibly higher than 0.5 after receiving the previous output bits from the PRG [23].

4) Pseudorandom Function (PRF)

A pseudorandom function is a deterministic and efficient function that returns seemingly random output, indistinguishable from truly random sequences. Such functions rely on PRGs. In contrast to PRGs, they can accept any input data in addition to the internal state. The input may be arbitrary, but the output must always 'look' completely random. A PRF with an output indistinguishable from random sequences is a secure one [24].

III. RELATED WORK

As defined in Section II-A, Secure Group Communication scheme (SGC) schemes consist of two main components: group key management and group membership management. Several surveys cover the former, which is the core component of SGC schemes [2], [16], [17], [25]–[31]. As for the latter, it has not received the same amount of attention. Many papers only define a group key management (GKM) component when presenting a new scheme.

Existing surveys mention various relevant factors when comparing GKM or SGC schemes regarding security and efficiency. However, in most cases, these factors appear sporadically without a systematic comparison of every considered scheme regarding each factor in detail. Li et al. [31] name and partly investigate the factors' computation efficiency, transmission efficiency, and storage efficiency of their surveyed GKM schemes. Furthermore, for some studied schemes, they also examine whether they meet the security requirements of forward and backward secrecy or collusion resistance. Jiang et al. [29] and others [10], [16], [17], [25], [27] mention similar metrics for comparison, namely computation costs, storage requirements, communication cost of the rekeying process, and the frequency of key updates. However, similar to Li et al. [31], most of these surveys do not present a systematic and comprehensive evaluation of GKM schemes regarding the different relevant factors as provided in our survey. Mapoka et al. [17] also name key independence as another security requirement. In addition to forward and backward secrecy, Xiao et al. [27] and others [10], [16] mention more security features, namely member authentication as well as message confidentiality and integrity. In surveys focused on group key management in Wireless Sensor Networks (WSNs) [10], [16], the authors also consider the network model of WSNs required by the schemes. Moreover, they name further security requirements, such as node compromise robustness and group independence.

Table 1 gives an overview of the factors existing surveys use or mention when evaluating GKM or SGC schemes compared to our work. In contrast to existing work, our survey examines these factors in detail for each scheme, presenting a systematic and extensive comparison. We also consider more than twice as many SGC schemes as the existing surveys.

Many surveys focus on one category of SGC schemes, typically either the centralized [28] or contributory [32], [33] category. Only a few papers have surveyed more than one category. For example, [26] discusses centralized and distributed dynamic key management schemes. Additionally, existing surveys either discuss general schemes [27], [34] or focus on a specific type of GKM [26], [28], [30]. Jiang et al. [29] only survey centralized and decentralized schemes, while Mapoka et al. [17] categorize the protocols into network independent and network dependent. Li et al. [31] divide key management (according to differences in the topology) into five classes: centralized, broadcast, hierarchical, subgroups, and distributed architectures. Xiao et al. [27] classify schemes focusing on the following seven techniques of key management: single network-wide key, pairwise key establishment, trusted base station, public key schemes, key predistribution, dynamic key management, and hierarchical key management. As for Klaoudatou et al. [30], they focus only on cluster-based approaches. Rafaeli et al. [25], and Renugadevi et al. [2] survey the categories centralized, distributed, and decentralized; however, these surveys do not provide a detailed comparison of the efficiency and security of a large number (47) of SGC schemes, as we do.

We use the metrics storage costs, communication costs, computation costs, key update frequency, and types of used cryptography to compare the efficiency of the schemes. Additionally, we evaluate the security of the 47 schemes regarding forward and backward secrecy, instant rekeying, message integrity, message confidentiality, member authentication, compromise robustness, and group independence. Of the existing surveys, only [10] directly addresses the important question of scheme suitability for a given application scenario. Moreover, [25] is deprecated and missing important new schemes since it was published in 2003. Only [10] provides detailed assistance for developers in choosing an appropriate scheme from the large number of existing schemes. We also provide such decision support in the form of guidelines supported by a decision tree. Compared to [10], our decision tree (1) considers not only 22 but 47 SGC

TABLE 1: Delimitation of our survey form related work.

	[10]	[16]	[25]	[27]	[17]	[29]	[31]	This Paper
Systematic comparison	✗	✗	✗	✗	✗	✗	✗	✓
Computation cost	✓	✓	✓	✓	✓	✓	✓	✓
Storage requirements	✓	✓	✓	✓	✓	✓	✓	✓
Communication cost	✓	✓	✓	✓	✓	✓	✓	✓
Forward, backward secrecy	✓	✓	✓	✓	✓	✓	✓	✓
Key update frequency	✓	✗	✓	✗	✗	✗	✗	✓
Instant rekey	✗	✗	✗	✗	✗	✗	✗	✓
Member authentication	✓	✓	✗	✓	✗	✗	✗	✓
Message confidentiality	✓	✓	✗	✗	✗	✗	✗	✓
Message integrity	✓	✓	✗	✗	✗	✗	✗	✓
Compromise robustness	✓	✓	✗	✗	✗	✗	✓	✓
Group independence	✗	✓	✗	✗	✗	✗	✗	✓
Amount of SGC schemes	22	13	8	24	23	11	10	47
Guidelines	✓	✗	✗	✗	✗	✗	✗	✗

systems, (2) analyzes not only ten but twelve aspects, (3) considers not only three but four decision factors, and (4) considers not only performance but also security features.

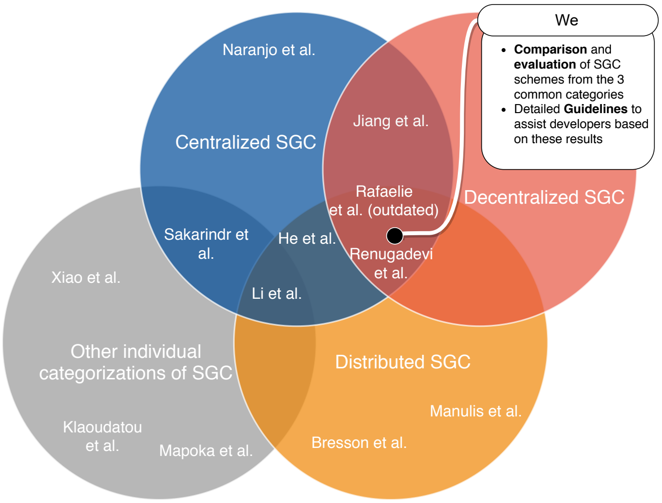

We
Comparison and evaluation of SGC schemes from the 3 common categories
Detailed Guidelines to assist developers based on these results
Naranjo et al.
Centralized SGC
Jiang et al.
Rafaelie et al. (outdated)
Decentralized SGC
Sakarindr et al.
He et al.
Renugadevi et al.
Xiao et al.
Li et al.
Other individual categorizations of SGC
Distributed SGC
Manulis et al.
Klaoudatou et al.
Mapoka et al.
Bresson et al.

FIGURE 2: Illustration of related surveys on SGC schemes and the scope of this survey.

IV. SURVEY METHODOLOGY

In this section, we first describe our literature research approach for collecting SGC schemes. Then, we present our classification, comprising the three main categories centralized, distributed/contributory, and decentralized/hybrid. Based on this classification, we classify our collected SGC scheme set. After that, in Section IV-B, we describe the performance metrics and security features we use as a basis for comparing different SGC schemes. In Section IV-C, we subsequently present our approach for deriving guidelines together with a decision tree to provide assistance for developers. These recommendations aim at supporting the scheme selection for a given application scenario. Figure 3 illustrates our general approach, visualizing the sequence of its main steps described in the following sections.

A. SECURE GROUP COMMUNICATION SCHEME CATEGORIES

First, we conducted a literature research on secure group communication and group key management schemes. As initial data sources, we used the already existing surveys on Group Key Management (GKM) and SGC [10], [16], [17], [25], [27], [29]. We applied the forward snowballing technique on these surveys to discover further schemes. This allows us to find more surveys and other papers describing or proposing new schemes. For data selection, we defined the following inclusion and exclusion criteria. If a paper proposes an SGC scheme that fully defines a GKM, we include it into our set of schemes if it is not already present. On the other hand, we exclude schemes from our set that do not fully define a GKM or fit none of our three categories. As a result, we identified and gathered a total of 47 different protocols as the main schemes proposed in literature.

While investigating existing GKM and SGC surveys, we discovered minor inconsistencies regarding the terminology used for the categories of group key management. Many researchers, such as [2], [25], categorize key management into centralized, decentralized, and distributed protocols. Sakarindr et al. [16] use the categories centralized, distributed, and contributory. Cheikhrouhou et al. [10] divide key management into centralized, contributory, and hybrid protocols. These inconsistent labels mainly refer to the same categories and are, therefore, interchangeable. Researchers using the term distributed refer to the same type of GKM protocols as the ones using the term contributory. Another name for this category is Group Key Agreement (GKA), as each member contributes to establish a common group key [2], without the presence of a trusted third party. The counterpart to the distributed/contributory approaches are the centralized approaches, not requiring communication between the group members to establish a group key. However, with centralized approaches, a trusted third party must be present. A mixture of both approaches are the hybrid approaches in [10], in which on the one hand a trusted third party is present, but on the other hand the group members also have to communicate among each other to establish a common group key. This

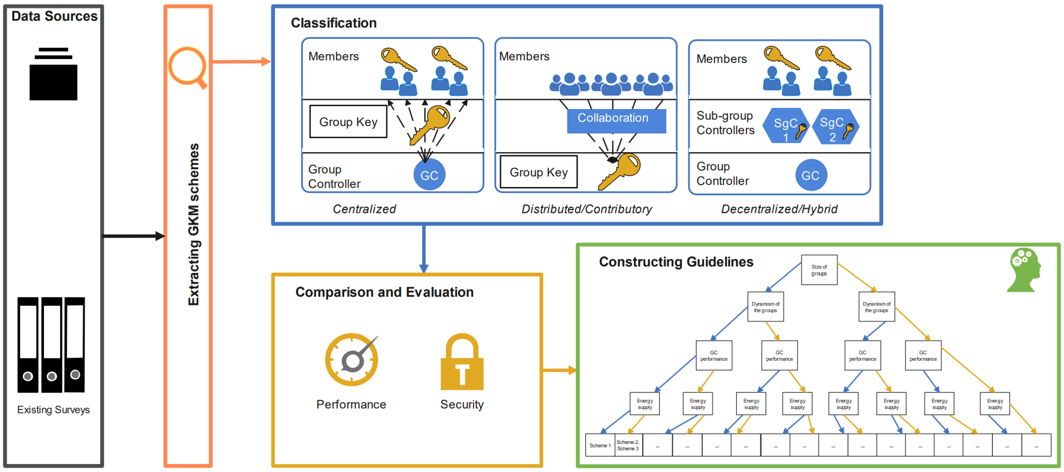


FIGURE 3: Visualization of the major steps in our approach in this survey.

TABLE 2: Our classification of all collected 47 schemes into the categories centralized, distributed/contributory, and decentralized/hybrid.

Centralized	Distributed/ Contributory	Decentralized/ Hybrid
SKDC [31]	D-LKH [14]	SMKD [35]
GKMP [36], [37]	DH-LKH [38], [39]	IGKMP [40]
LKH [41]	D-OFT [42]	Iolus [43]
LKH+ [15]	SHM [44]	MARKS [45]
OFT [46]	PCGR [47]	Kronos [48]
OFCT [49]	CRGR [50]	Slimcast [51]
S2RP [52]	BKM [53]	RiSeG [54]
LARK [55]	EGKMST [56]	LNT [57]
TKH [58]	SGRS [4]	DEP [59]
CFKM/FT [15]	BD [60]	CS [61]
ELK [62]	G-DH [63]	Hydra [64]
SGCH [65]	Octopus [66]	Alohali [5]
HSHKD [67]	CKA [68]	
LEAP [69]	DFT [15]	
EBS [70], [71]		
SeGCom [72]		
XKFS [73]		
SBSA [74]		
KMGC [75]		
CL-EKM [76]		

is usually done by dividing a group into subgroups managed by group members. Literature refers to this approach also as decentralized, even though a third party is present. This leads us to our SGC scheme classification into three categories: centralized, distributed/contributory, and decentralized/hybrid. In Figure 3, the classification step illustrates the differences between these categories in terms of group key management. We classify all 47 SGC schemes into one of these three categories (see Table 2). In the following, we describe these categories in more detail:

1) Centralized SGC schemes:

In centralized schemes, a central trusted entity, called the Group Controller (GC), manages the group. This includes managing the joining and leaving of members and the renewal of the group key. The GC is the only entity that controls both the GKM and the GMM component of an SGC scheme. Therefore, the GC handles the majority of the workload [10]. This centralized approach seeks to minimize computational costs and storage requirements on the side of the group members [25]. Advantages of centralized schemes are the high security of key selection and generation, as well as the efficiency of the symmetric key encryption [10]. However, the GC represents a single point of failure and also a possible performance bottleneck. If the GC of a centralized system fails, the system can no longer function. Since the GC is the only entity managing the whole group, it is the main target of attacks on centralized systems [10]. The majority of the SGC schemes we compared belong to this category.

2) Distributed/Contributory SGC schemes:

In distributed SGC schemes, the group members collaborate to manage the group without a central authority [25]. Thus, distributed schemes have the advantage of fault tolerance [10], since no single entity is responsible for distributing and generating the keys [25]. However, this comes at the expense of higher computational costs on the side of the group members and other drawbacks, such as increased energy consumption for the devices [10].

3) Decentralized/Hybrid SGC schemes:

In decentralized architectures, a central unit carries out some tasks, while others require cooperation. These decentralized protocols aim at achieving both efficiency and fault tolerance [10]. A very common approach in decentralized schemes is to divide the group management among SGC. The goal of using SGC schemes is to minimize the problem of concentrating all the workloads on a single entity [25]. Another approach is to assign the group key generation to a group controller in a centralized manner, while having the group key distribution done contributory by all group members [10]. We classify such hybrid schemes as decentralized, since they generally have similar characteristics as traditional decentralized protocols. Therefore, we named this category decentralized/hybrid.

B. EVALUATION CRITERIA FOR SGC SCHEMES

We have to compare these schemes in terms of security features, and performance metrics to derive SGC scheme selection guidelines for specific use cases. We identify relevant security features and performance metrics that we extracted from multiple surveys [10], [16], [17], [25], [27], [29]. In the following, we first present the relevant security features [10], [16]:

Message integrity and confidentiality: Preventing message manipulation, for example, by hashing and signing the message using strong encryption keys [16]. Message confidentiality means that only authorized members can learn meaningful information from the message. In addition, data circulating within a group must stay confidential and accessible only to group members. This is done by encrypting the message with a key shared among the members [10].

Backward and forward secrecy: Backward/forward secrecy ensures that recipients can only decrypt messages exchanged while they are a group member [25]. Each time the membership changes, the group key changes to ensure backward/forward secrecy. This way, new/revoked members cannot decrypt messages sent before/after they joined/left the group [25].

Member authentication: The identity of a potential new member requires verification before giving it access to the group communication. This authentication mitigates identity-based attacks. It can be achieved by using a group key, a pairwise key, or a certificate [16].

Robustness against compromised members: An attacker can compromise one or more group members. In this case, the SGC scheme should reject these members from the group to stop data from being further revealed. This can be done, for example, by updating the group key upon detection and not revealing this new key to the compromised members [10].

Group independence: Since members of one group may also belong to other groups, security parameters should be independent. Consequently, a compromised group does not affect other groups with overlapping sets of members [10].

In addition to security requirements, SGC schemes should be as efficient as possible. Many devices in modern group communication applications have limited resources, such as low memory, computing capacity, and battery life [26]. Thus, the following metrics can be used to determine the performance of an SGC scheme:

Storage requirements: Many devices, especially in wireless group communication (e.g., sensor nodes), only have limited memory capacity. Hence, the number of keys that must be stored on devices to protect the group communication must be as low as possible [10].

Computation and communication costs: Usually, devices in wireless communication, especially in modern IoT applications, only have low-power CPUs. Thus, SGC schemes must limit their computation costs. Additionally, the number of messages exchanged by a component of an SGC scheme should be minimal. This low communication cost avoids battery/energy drain and possible failures of group member devices [10].

Cryptographic techniques: The used cryptographic techniques in SGC schemes impact their performance.

Key update frequency: Group key renewal as part of the rekeying process either occurs periodically or at membership change [10]. This time- or member-driven frequency significantly impacts the performance of an SGC scheme. Every key renewal implies generating and distributing a new key, causing expensive communication and computation overhead [29]. Thus, the number of key updates should be as low as possible.

We use the *big O notation* to describe the storage, communication, and computation costs asymptotically, enabling comparability between the schemes. For schemes where the costs are not already explicitly stated in this form, we map costs into this standardized form. The storage cost refers to the number of keys stored at the GC and at group members. The number of messages that must be sent for rekeying, or at a join or leave event, comprises the communication cost of an SGC scheme.

C. DECISION FACTORS SGC SCHEME SELECTION

We identify the main scenarios and constraints related to group communication applications that serve as decision factors to develop recommendations and construct a decision tree to select SGC schemes for different application domains.

One decision factor that nearly all papers on key management or SGC mention is the *group size* [2], [10], [16], [17], [25]–[31]. The group size has an immense impact on the performance of an SGC scheme since the number of members can be highly diverse [77]. The number of members can range from less than 10 (e.g., devices communicating in a smart home) to far more than 1000 (e.g., sensor nodes deployed in military applications) [10]. We consider a group to be small if it comprises less than 100 members. Otherwise, the group is as large [77].

Another decision factor frequently appearing in the literature is the group's *dynamism* [2], [10], [29], [31], which refers to two group characteristics. First, group membership can change in a highly dynamic way or remain rather static. This can possibly require an SGC scheme to handle very frequent key updates [31]. Second, the dynamism of a group can also refer to its topology. For example, members are often static in indoor applications or environmental monitoring. In contrast, there are applications where members, such as devices or sensors, might move or be destroyed. This can lead to dynamic changes in the group's topology [10]. In addition, membership changes mentioned before may also make the group topology more or less dynamic, depending on their frequency [2].

Furthermore, we identified the *group controller performance* to be another decision factor [25], [29], [77]. Some schemes, especially centralized ones, require a powerful GC [29]. Group communication applications may include a strong GC with abundant resources, such as a PC at home in a smart home environment [5]. However, other applications may only support a resource-constrained GC. This is common in many WSNs (e.g., in healthcare applications, where communicating sensors may be attached to a patient's body [10]). Therefore, the choice of a scheme must also consider whether its application can provide a powerful group controller or not.

Another decision factor that we identified is *energy supply* [2], [77]. In some applications, devices might be able to get power directly from an unlimited energy source, such as in industrial process control [77]. Scenarios with similar conditions may also offer the possibility of regularly restoring batteries or other energy sources [77]. However, many applications (e.g., in environmental monitoring) have limited non-replenishable energy resources [10], [77]. Hence, we also have to consider an SGC's energy consumption.

Besides these decision factors, each application might have different *security* requirements [77]. Especially IoT applications can have various levels of required security [5]. For example, security is a crucial aspect of WSNs in military applications [10]. Thus, the required security features, described in Section II, are also decision factors.

Based on these decision factors, we construct decision trees to support developers in choosing SGC schemes. One practical approach is to introduce binary categories. For example, groups can be classified into small, with less than 100 members, or large, with more than 1000 members. Accordingly, SGC schemes differ based on whether they support large groups or not [77]. Similarly, a binary distinction can be introduced for the remaining decision factors. We represent this in our decision tree by introducing a decision node that splits the tree into two subtrees for each main decision factor. However, we first need corresponding information about the performance and security features of the different schemes to build our decision tree and thus guidelines for the choice of SGC schemes.

V. CENTRALIZED GROUP MANAGEMENT APPROACHES

In this section, we discuss the performance and security of centralized SGC schemes. A more detailed description of the functionality of the considered schemes can be found in the supplemental material. We have divided the comparison into three tables. Table 3 explains the notation used in these tables. Table 4 and Table 5 summarize the performance of the centralized schemes. The terms low, medium, and high describe how a scheme performs in that specific category compared to the other approaches. Table 6 summarizes the security features of centralized schemes. Tables 4 and 5 illustrate that different schemes achieve varying results in terms of performance and apply different techniques. Some schemes are significantly more efficient than others; however, they may pose unbearable risks to the security of the group to achieve such results.

TABLE 3: Notation used in Tables 4, 5, 7, and 10.

Notation	Description	Notation	Description
n	number of group members	E	encryption operation
b	number of bits in the member ID	D	decryption operation
s	number of sessions in a group lifecycle	K	size of a key (in bits)
a	average number of neighbors	p,p_1, p_2	polynomial degree
r	set of random numbers	h	hash values
c	number of subsets in EBS	pk/sk	public key/secret key
t	threshold	g	large number
i	index of member	m	number of members in subgroup
x	length of key range a member requires	t,q	Blundo parameters
th	threshold number		

GKMP provides remarkable performance in terms of storage, communication, and computation costs, but this is achieved at the expense of compromising the forward secrecy. ELK, SGCSH, SGR, and HSHKD use a timed rekey to decouple the refreshing of the group key from any membership changes. The downside of this periodic rekey is that it could either violate forward and backward secrecy for a short time until the next update happens, or impose small delays in the joining or leaving process, such as in ELK. Among the centralized approaches, the ones using a key tree hierarchy seem to be preferable. The first scheme of that kind is LKH together with its improved extensions and variants, such as LKH+, OFT, OFCT, S2RP, LARK, and TKH. These schemes reduce the communication cost to $O(log(n))$, but this could still be high for devices with limited resources, such as sensor nodes in WSNs.

The schemes SGCSH, SGR, and HSHKD try to solve the problem of not receiving a key update due to unreliable channels or other similar problems. In these so-called self-healing protocols, members can recover lost keys based on previously received ones. This avoids repetitive retransmission of key update messages. These advantages come at the cost of imposing a limited group lifetime, after which a new group has to be reestablished. SKDC, XKFS, SBSA, KMGC, and CL-EKM rekey the group by sending an encrypted message for each member, limiting the scalability of these schemes. SeGCom and LEAP require synchronization between members since they use $\mu Tesla$. According to [78], [79], this can be hard to achieve in highly distributed applications, such as WSNs. Most schemes do not provide the GMM component, but only describe the GKM component. Therefore, member authentication has to be handled separately. SGR, SegCom,

TABLE 4: Comparison of centralized SGC schemes in terms of performance (Part 1).

Scheme	Storage Costs		Communication Costs	Computation Costs
	GC	Member		
SKDC	High: O(n)	Low: 1K	High: O(n)	GC: High: O(n) Member: 1D
GKMP	Low: 2K	Low: 2K	Low: 2K	Low: 2E / 2D
LKH	High: O(n)	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)
LKH+	High: O(n)	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)
OFT	High: O(n)	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)
OFCT	High: O(n)	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)
S2RP	High: O(n)	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)
LARK	High: O(n)	Medium: O(log n)	O(n) - O(log n)	Medium: O(log n)
TKH	Low: 4K	Low: 4K	Medium: O(log n)	Medium: O(log n)
CFKM	Medium: O(b)	Medium: O(b)	Medium: O(b)	Medium: O(b)
ELK	High: O(n)	Medium: O(log n)	Medium: O(log n), 0 multicast on join	GC: High: O(n) Member: O(log n)
SGCSH	High: O(n+s)	Medium: 3K+r	Join: High O(s), Rekey: Low O(1)	Low: XOR and hash for rekeying
SGR	High: O(n+s)	Medium: 2K+h	Join: High O(m), Rekey: Low O(1)	Medium: Hash and polynomial computations
HSHKD	High: O(s*p)	Low: O(1)	Very High: O(p)	High O(p)
LEAP	Low: O (a+s)	O(a)	Low: O(a)	Low: O(a)
EBS	High: O(n)	O(c)	Medium: O(log n)	Medium: O(log n)
SeGCom	Low: (t+1)log q	for all	High: O(n) leave	High O(n)
XKFS	High: O(n)	Low: O(1)	High: O(n)	High O(n)
SBSA	High: O(n)	Low: O(1)	High: O(n)	High O(n)
KMGC	Low: pk/sk		High: O(n)	High O(n)
CL-EKM	Low: pk/sk		High: O(n)	High O(n)

KMGC, and CL-EKM provide member authentication.

VI. DISTRIBUTED/CONTRIBUTORY GROUP MANAGEMENT APPROACHES

In this section, we compare distributed/contributory SGC schemes in terms of security and performance. A more detailed description of the functionality of the considered schemes can be found in the supplemental material. Tables 7, 8, and 9 summarize the performance and security of the distributed/contributory schemes. Table 3 explains the used notation. In the case of distributed/contributory approaches, we only have one column for storage costs since most schemes from this category do not have a GC. Moreover, we introduce a new column called rounds. One round includes all the messages that can be communicated simultaneously. It is an important property in distributed approaches, whether the number of protocol rounds is independent of the number of group members [3]. Table 9 compares the schemes according to their security features.

In distributed/contributory schemes that do not depend on a group leader, such as DH-LKH, D-OFT, BD, and G-DH, all members are equal. The failure of a member will not affect the whole group. Contrary, schemes that require a leader, such as D-LKH, Octopus, CKA, or DFT, suffer heavily from leader failure. Schemes that include a fixed number of rounds to set up a common group key (e.g., CKA) are independent of the number of members in terms of interactions among them. They can reduce the setup time by offering the possibility to do a set of computations in parallel. Nevertheless, they rely on a leader to accumulate the contributions from all other members and broadcast them. This dependence on a powerful leader is a downside we discussed before.

The contributory approaches generally impose high computation costs, especially on the group leader, since they are often based on asymmetric or polynomial computations. For this reason, these schemes, such as SHM, CRGR, or BKM, are less suitable for applications that include devices with limited resources. GKMST and PCGR are appropriate for applications with constrained resources due to their performance. However, they introduce other downsides, such as PCGR not supporting joining and leaving of members after setup or EGKMST being susceptible to member compromise attacks.

VII. DECENTRALIZED/HYBRID GROUP MANAGEMENT APPROACHES

In this section, we discuss the performance and security of decentralized/hybrid SGC schemes. A more detailed description of the functionality of the considered schemes can be found in the supplemental material. Table 10 and Table 11 summarize the performance of the distributed schemes. Table 3 explains the notation. Table 12 compares the schemes according to their security features.

SMKD achieves good performance results, but just like centralized GKMP, it uses a KEK known to all members to encrypt the next group key, compromising forward secrecy. MARKS and Kronos do not provide key independence. Kronos generates the new keys based on previous ones. If an attacker compromises any of the old keys, the attacker will have access to all future keys. This is also true for MARKS with a compromised seed. MARKS, Kronos, and DEP use a timed rekey that leads to delays of the group key after a member has joined or left the group. That member may then have unauthorized access to group communication until the

TABLE 5: Comparison of centralized SGC schemes (Part 2).

Scheme	Key Update Frequency	Types of Cryptography	Discussion
SKDC	Membership change	Pk, DH	Simplest approach; requires extremely powerful GC; only for small groups
GKMP	Membership change	Symmetric, DH, RSA	To provide forward secrecy, new group has to be created on leave event (O(n) communication and computation costs).
LKH	Membership change	Symmetric	Requires strong GC
LKH+	Membership change	Symmetric	Halves rekey message size for joining
OFT	Membership change	Hash, XOR, symmetric	Improved key hierarchy halving size of rekey messages from LKH on join and leave events
OFCT	Membership change	Hash, PRG, symmetric	Same communication and storage requirements as OFT; uses PRG only applied in leave process rather than one-way function
S2RP	Membership change	Symmetric	Authentication of rekeying messages; LKH improvement
LARK	Membership change	Symmetric	Extension to S2RP to support more grouping topologies; communication depends on topology
TKH	Membership change	Symmetric	Tree mapped to the physical topology to reduce rekey messages; improvement to LKH
CFKM	Membership change	DH	Performance depends on b instead of n; vulnerable to collusion
ELK	Time period	Symmetric, PRF	No multicast message for join since timed rekey used; small delay on join; hints to recover lost rekey messages
SGCSH	Time period	XOR, hash	Cons: Limited life cycle; Pros: key update messages not encrypted, instead one-way hash fct. and XOR used, recover of lost group key possible
SGR	Time period	Tesla, Poly	Pro: Supports self-healing
HSHKD	Time period	Polynomial	GC configuration before deploying; Self-healing
LEAP	Membership change	μTesla, symmetric	Synchronization required
EBS	Membership change	Combinatory + symmetric	Members are anonymous
SeGCom	Membership change	μTesla, symm.	Synchronization required between members
XKFS	Membership change	Hash, XOR, symmetric	Cons: requires powerful GC, high communication overhead; Pros: member operations are kept simple
SBSA	Membership change	PRG, symmetric	Cons: requires powerful GC, high comm. overhead; Pros: member operations are kept simple
KMGC	Membership change	Asymmetric	Cons: requires powerful GC, high comm. and comp. costs; Pros: Low storage
CL-EKM	Membership change	Asymmetric	Cons: requires powerful GC, high comm. and comp. costs; Pros: Low storage

TABLE 6: Comparison of the centralized SGC schemes in terms of security features.

Scheme	Backward/Forward Secrecy	Instant Rekey	Message Integrity	Massage Confidentiality	Member Authentication	Compromise Robustness	Group Independence
SKDC	✓/✓	✓	✗	✓	✗	✗	✓
GKMP	✓/✗	✓	✗	✗	✗	✗	✓
LKH	✓/✓	✓	✗	✓	✗	✗	✓
LKH+	✓/✓	✓	✗	✓	✗	✗	✓
OFT	✓/✓	✓	✗	✓	✗	✗	✓
OFCT	✓/✓	✓	✗	✓	✗	✗	✓
S2RP	✓/✓	✓	✓	✓	✗	✗	✓
LARK	✓/✓	✓	✓	✓	✗	✗	✓
TKH	✓/✓	✓	✗	✓	✗	✗	✓
CFKM	✓/✓	✓	✗	✗	✗	✗	✓
ELK	✓/✓	✗	✗	✓	✗	✗	✓
SGCSH	✗/✗	✗	✓	✓	✗	✗	✓
SGR	✗/✗	✗	✓	✓	✓	✓	✓
HSHKD	✗/✗	✗	✓	✓	✗	✓	✓
LEAP	✓/✓	✗	✗	✓	✗	✗	✓
EBS	✓/✓	✓	✗	✓	✗	✗	✓
SeGCom	✓/✓	✗	✓	✓	✓	✗	✓
XKFS	✓/✓	✓	✗	✓	✗	✗	✓
SBSA	✓/✓	✓	✗	✓	✗	✓	✓
KMGC	✓/✓	✓	✓	✓	✓	✓	✓
CL-EKM	✓/✓	✓	✓	✓	✓	✓	✓

TABLE 7: Comparison of distributed SGC schemes (Part 1).

Scheme	Storage Costs	Communication Costs	Computation Costs	Key Update Frequency	Rounds
D-LKH	Medium: O(log n)	High: O(n)	Medium: O(log n)	Membership change	O(log n)
DH-LKH	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)	Membership change	O(log n)
D-OFT	Medium: O(log n)	Medium: O(log n)	Medium: O(log n)	Membership change	O(log n)
SHM	Low: O(1)	High: O(n)	High: O(n)	Membership change	2
PCGR	Medium: O(p)	High: O($p_1 * p2$)	Medium: O($p_1 \ * \ p^2 \ + p_2^3$)	Time period	O(a)
CRGR	Low: O(1)	High: O(n)	High: O(n^2)	Membership change	3
BKM	High: O(n)	High: O(n)	High: O(n^2)	Membership change	3
EGKMST	Low: O(th)	High: O(n)	Low: polynomial evaluation	Membership change	3
SGRS	High: O(n)	High: O(n)	Low: each Member: O(1)	Membership change	3
BD	High: O(n)	Low: O(1)	High: O(n)	Membership change	O(n)
G-DH	Low: O(1)	High: O(n)	High: O(i)	Membership change	3
Octopus	Low: O(1)	High: O(n)	High: O(n)	Membership change	O(n)
CKA	Low: O(1)	High: O(n)	High: O(n)	Membership change	3
DFT	Low	High: O(n)	High: O(i)	Membership change	O(n)

TABLE 8: Comparison of distributed SGC schemes (Part 2).

Schemes	Types of Cryptography	Discussion
D-LKH	Symmetric	No GC; LKH is generated among the leaders of the members
DH-LKH	DH, symmetric	Members collaboratively generate keys in the hierarchy by using Diffie-Hellman algorithm
D-OFT	Hash, XOR, symmetric	Uses OFT without GC; members generate keys and distribute blinded key
SHM	DH	High computation and communication costs; low storage costs; highly secure
PCGR	Polynomial	Cons: Synchronization of members required, no join or leave; Pros: member compromise resistant
CRGR	Polynomial	High computation cost for powerful member (leader) and high communication costs
BKM	Polynomial	High computation and storage costs for leader; high communication costs
EGKMST	Polynomial	Cons: susceptible to member compromise attacks; Pros: low computation and storage costs
SGRS	One-way fct, XOR	Members arranged in a logical circular linked list
BD	DH	Only needs 3 rounds to execute
G-DH	DH	Key agreement for a whole group; setup message and number of exponential operations increases as the sequence reaches last member
Octopus	DH	Group split into 4 subgroups that agree on a intermediate DH key
CKA	One-way fct, public key	Group key is generated by combining the contributions of all members
DFT	DH	Extension to CFKM without a GC; synchronization delays when multiple members change same key at the same time

TABLE 9: Comparison of the distributed SGC schemes in terms of security features.

Scheme	Backward/Forward Secrecy	Instant Rekey	Message Integrity	Massage Confidentiality	Member Authentication	Compromise Robustness	Group Independence
D-LKH	✓/✓	✓	✗	✓	✗	✗	✓
DH-LKH	✓/✓	✓	✗	✓	✗	✗	✓
D-OFT	✓/✓	✓	✗	✓	✗	✗	✓
SHM	✓/✓	✓	✗	✓	✗	✗	✓
PCGR	✗/✗	✗	✓	✓	✗	✗	✓
CRGR	✓/✓	✓	✓	✓	✗	✓	✓
BKM	✓/✓	✓	✗	✗	✗	✗	✓
EGKMST	✓/✓	✓	✓	✓	✓	✗	✗
SGRS	✓/✓	✓	✗	✓	✗	✗	✓
BD	✓/✓	✓	✗	✓	✗	✗	✓
G-DH	✓/✓	✓	✗	✓	✗	✗	✓
Octopus	✓/✓	✓	✗	✓	✗	✗	✓
CKA	✓/✓	✓	✗	✓	✗	✗	✓
DFT	✓/✓	✓	✗	✗	✗	✗	y

TABLE 10: Comparison of decentralized SGC schemes. Part 1.

Scheme	**Storage Costs**		**Communication Costs**	**Computation Costs**
	GC	**Member**		
SMKD	Low: O(1) 2K	Low: O(1) 2K	Low: O(1)	Low: O(1) 1E/D
IGKMP	Low: O(1)	Low: O(1)	Join: O(1) for each node Leave: O(m)	Low: O(1) E/D for each
Iolus	Low: O(1) 2/4K	Low: O(1) 3K	Medium: O(m)	Low: O(1), translations between SGCs required
MARKS	Medium: O(log x)	Medium: O(log x)	Medium: O(log x)	Medium: O(log x) mean: 2 Hashes
Kronos	Low: O(1)	Low: O(1)	Low: O(1) multicast	Low: O(1)
Slimcast	High: O(n)	High: O(n)	Low: O(m)	Low: O(1) 1E/D
RiSeG	Low: (t+1) log q	for all nodes	Low: O(1)	Low: O(1) 1E/D
LNT	Low: (t+1) log q	for all nodes	Low: O(1)	Low: O(1) 1E/D
DEP	GC: O(m) SGC: O(1) 5K	Low: O(1) 4K	Medium: O(m)	Medium: O(m)
CS	Low: O(1)	Low: O(1)	Medium: O(m)	Low: O(1)
Hydra	Low: O(1) 3K	Low: O(1) 2K	Medium: O(m)	Low: O(1)
Alohali	Low: O(1) 1K	Low: O(1) 1K	High: O(n)	Medium: O(m)

TABLE 11: Comparison of decentralized SGC schemes. Part 2

Scheme	**Key Update Frequency**	**Cryptography Types**	**Discussion**
SMKD	Membership change	Symmetric	Uses trees built by CBT protocol. Requires modifications to IGMP.
IGKMP	Membership change	Symmetric	The GC (DKD) distributes the group Management among SGCs (AKDs).
Iolus	Time period	Symmetric	Cons: Only 1-to-n communication. Pros: membership changes are treated locally
MARKS	Time period	Hash	Cons: Cannot be used if membership changes would require group key update. Pros members can subscribe to different time intervals
Kronos	Time period	Symmetric	Cons: generates new key based on old Pros: fault-tolerant
Slimcast	Membership change	Symmetric	High computation costs for multicast: O(n) encryptions and decryptions in hop-by-hop
RiSeG	Membership change	Symmetric, Elliptic Curve	Cons: Delay in rekeying O(n) Pros: Low costs, considers constrained GC
LNT	Membership change	Symmetric, Elliptic Curve	Cons: Delay in rekeying O(log n) Pros: Low costs
DEP	Time period	Symmetric	Solves untrusted SGC problem
CS	Membership change	Reversible cipher sequence	Cons: 1-to-n communication
Hydra	Membership change	PK	Cons: Requires synchronization between SGC Pros: fault-tolerant
Alohali	Membership change	One-way fct	Based on Shamir‘s Secret Sharing scheme. Very low storage requirements

TABLE 12: Comparison of the distributed SGC schemes in terms of security features.

Scheme	**Backward/Forward Secrecy**	**Instant Rekey**	**Message Integrity**	**Massage Confidentiality**	**Member Authentication**	**Compromise Robustness**	**Group Independence**
SMKD	✓/✗	✗	✗	✗	✓	✗	✓
IGKMP	✓/✓	✓	✗	✓	✓	✗	✓
Iolus	✓/✓	✓	✗	✓	✗	✗	✓
MARKS	✗/✗	✗	✗	✗	✗	✗	✓
Kronos	✓/✗	✗	✗	✓	✓	✗	✓
SLIMCAST	✓/✓	✓	✓	✓	✓	✗	✓
RiSeG	✓/✓	✓	✓	✓	✓	✗	✓
LnT	✓/✓	✓	✓	✓	✓	✗	✓
DEP	✓/✗	✗	✗	✓	✓	✗	✓
CS	✓/✓	✓	✗	✓	✗	✗	✓
Hydra	✓/✓	✓	✗	✓	✓	✗	✓
Alohali	✓/✓	✓	✗	✓	✓	✓	✓

next rekey happens.

Some schemes limit the key update to the subgroup where the change has occurred to solve the problem of making all members change their keys on a membership change. This is achieved by each subgroup using its own key for communication within the group. However, this has the consequence that when messages are exchanged between groups, the messages for the other group must be encrypted with the other group's key. Therefore, in this solution of different keys for each subgroup, direct interference with the data path is required. An example scheme for this is Iolus. Moreover, some decentralized schemes only propose a framework or an architecture for large-scale group key management without approaching the question of how to distribute keys to members within subgroups efficiently.

Schemes, such as SMKD, IGKMD, DEP, and CS, depend on the GC in the process of group key generation or for controlling access additionally to using SGCs. The failure of this central GC affects the whole group. Schemes are also less scalable when contacting the GC each time to verify whether a membership is valid. DEP solves the problem of establishing trust between third-party entities (the SGCs) by dual encryption.

Decentralized/hybrid schemes are generally scalable by design and more suitable for applications with resource-constrained devices. They require low costs regarding storage, communication, and computation. RiSeG and its improvement LNT are specially designed for WSNs.

VIII. GUIDELINES FOR CHOOSING SGC SCHEMES

In this section, we present guidelines summarizing the results from the comparison of SGC schemes in Sections V–VII. These guidelines are intended to assist developers in choosing an appropriate and efficient scheme to integrate security into their group communication application. The guidelines comprise performance and security points of view.

A. PURE PERFORMANCE-DRIVEN SGC SCHEME SELECTION

For our performance-driven scheme selection, we only cover the schemes performing best based on our evaluation results in the previous sections. We first look at the best schemes in each category and then present cross-category guidelines, illustrated in Figure 4.

1) Guidelines for Centralized SGC Schemes

SGC schemes from the central category, described and compared in Section V, require reliable communication between the GC and the members. If a member misses a key update due to a temporary connection issue, the member can no longer participate in the group communication. Additionally, frequent membership changes create much stress on the GC, the single entity managing the whole group. Therefore, these schemes are generally more suitable for static applications.

SKDC is a simple approach that scales poorly and requires a powerful GC as well as an unlimited energy supply. However, it is an option when simplicity is key there are no resource constraints. GKMP compromises forward secrecy, and its approach of creating an entirely new group on each leave event does not scale. However, if the application does not intend for members to leave the group, it is suitable for larger groups with limited energy supply, both for weak and powerful GCs.

LKH and its extensions are suitable for larger groups. The improved schemes are obviously preferable. LKH+ halves the size of rekey messages on join events, while OFT and OFCT additionally halve the key update message for leave events. S2RP introduces the authentication of rekey messages and supports a variety of different grouping topologies when using its extension LARK. In addition to reducing rekey messages, TKH also supports resource-constrained GCs.

LEAP is another very efficient centralized SGC scheme. It scales well and works power and storage limited GC. LEAP requires synchronization between members since it uses $\mu Tesla$. Furthermore, it is not suitable for highly dynamic applications, as all members must establish a communication link with each of their neighbors. SeGCom requires low storage costs for the GC and members, but imposes high communication and computation costs. Moreover, it also requires synchronization since it uses $\mu Tesla$. SGCSH, SGR, and HSHKD are so-called self-healing schemes that enable a member to recover a lost key. They are more appropriate for dynamic groups. However, they rekey periodically, which could violate backward and forward secrecy until the next key update. Membership changes should be as few as possible to reduce this security issue. Therefore, self-healing schemes should not be used in large groups, since these typically include frequent membership changes.

2) Guidelines for Distributed SGC Schemes

Distributed SGC schemes, described and compared in Section VI, impose very high communication costs. In large groups, these schemes lead to high bandwidth usage and energy consumption. Hence, these approaches should only be used in applications with few members or unlimited energy supply. Additionally, in distributed schemes, the group members generate the group key collaboratively, requiring a connection between the members. Thus, distributed approaches are more appropriate for static groups with reliable communication channels.

DH-LKH, D-OFT, BD, and G-DH require no group leader in the group key generation process. These schemes rather distribute the computation costs among all members. In contrast to the other three approaches, G-DH does not distribute the costs equally since the number of exponential operations increases with each member in the sequence of the key generation. D-LKH, DH-LKH, and D-OFT exhibit the lowest communication costs of all distributed schemes. The latter two also have a decent overall performance. DH-LKH and D-OFT require no GC and are suitable for small groups and devices with limited energy supply.

SGRS imposes very low computational costs for the members, especially since they only execute hash and XOR operations. Thus, this scheme is very suitable for small devices with a low-power CPU applications. SHM, CRGR, and BKM, as well as Octopus and CKA, require large computations by a member who takes the role of a group leader or a separate GC. Thus, they are not appropriate for applications with a weak GC or none. EGKMST is a scalable and efficient scheme regarding storage and computation, making it suitable for large static groups without a powerful GC. PCGR integrates security against member compromise attacks, but requires synchronization between members. Moreover, PCGR does not support joining and leaving after the group setup due to the pre-distribution of keys.

3) Guidelines for Decentralized SGC Schemes

Decentralized SGC schemes generally provide good scalability by dividing the group into subgroups managed by SGCs. Thus, they are more suitable for applications with large groups. Additionally, by distributing the workload, more entities can fail before affecting the group. This increases the applicability of decentralized schemes for more dynamic groups.

SMKD is a scheme that achieves excellent results in terms of performance. Thus, it is very suitable for weak GCs and devices with a limited energy supply. However, just like centralized GKMP, it compromises forward secrecy, which makes it inappropriate for applications with high security requirements. The approaches Iolus and CS are only suitable for special applications that use one-to-many instead of many-to-many communication. Iolus offers the advantage of treating membership changes locally within the subgroup. This makes it especially suitable for weak GCs and dynamic groups, but its communication costs could drain the limited energy of devices.

Since IGKMP, Kronos, RiSeG, and LNT impose low storage, communication, and computation costs, they are more appropriate for applications with constrained resources and GC. Of course, they are also suitable for applications with powerful GCs or infinite energy. Additionally, SMKD, IGKMP, and Kronos can handle frequent membership changes, making them useful in dynamic groups. Very frequent leave events in larger groups could still drain the energy of devices when using IGKMP. Hence, it is unsuited for large dynamic groups with a limited energy supply. Kronos should not be used in applications that have high security requirements since it generates the new group key based on the previous one. RiSeG includes a big delay in rekeying that scales linearly with the number of members. Consequently, RiSeG is not suitable for large groups. LNT reduces this delay to O(log(n)). Since these schemes have such a delay every time the membership changes, they are not suitable for dynamic groups with frequent membership changes. The scheme Alohali is another decentralized scheme with very low storage costs. It is appropriate for large groups as well, but its communication costs would quickly drain the energy of devices with a limited energy supply.

4) Cross-Category Guidelines for SGC Schemes

In our cross-category guidelines focused on performance, we aim to address the characteristics and requirements of a given group communication application as closely as possible. We use the decision factors described in Section IV (i.e., the group size, the group dynamism, the performance of the GC, and the energy supply) to construct a decision tree. Figure 4 illustrates our decision tree that recommends appropriate schemes for each application characteristic. A reader who wishes to choose a suitable SGC scheme for a given application should use the decision tree as follows. The tree works top-down, starting at the first decision node, 'size of groups'. At each decision node in the tree, the reader must follow the path that best corresponds to the specific feature of the groups in the considered application. As described in Section III, we make binary distinctions on each of these decision nodes. For example, at the first decision node, the reader has to follow the left branch if groups in the considered application scenario generally have less than 100 members and therefore are labeled small. Otherwise, the reader should follow the right branch leading to schemes that are more suitable for larger

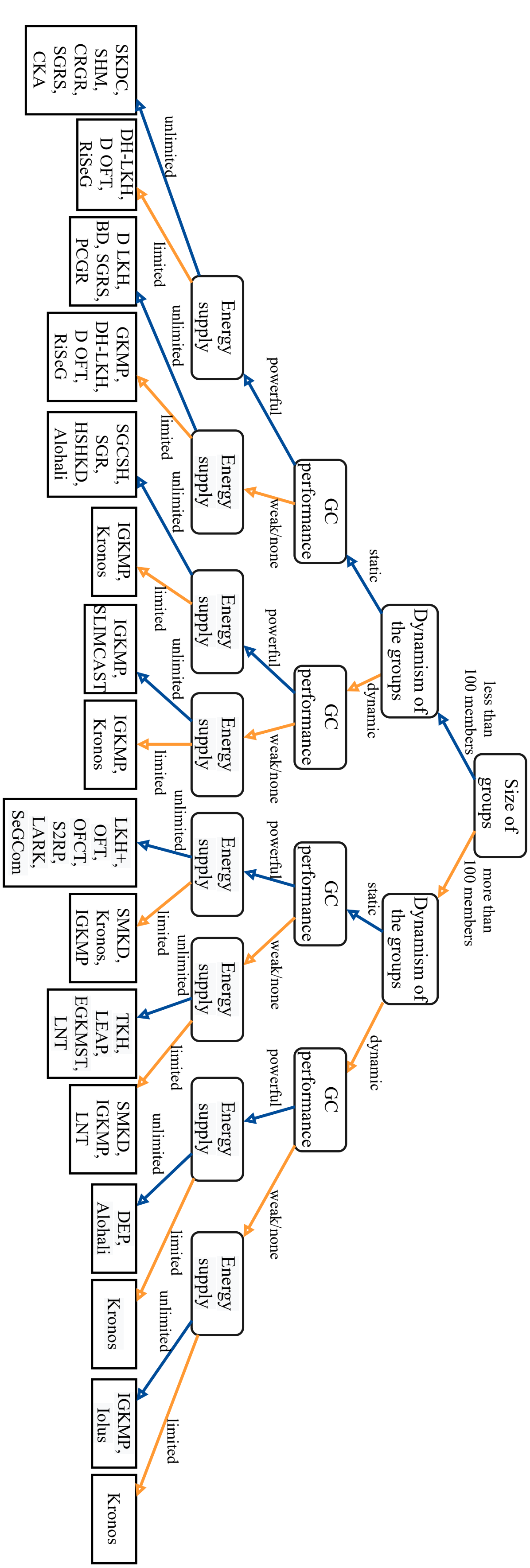


FIGURE 4: Decision tree illustrating our guidelines for the performance-driven choice of a suitable SGC scheme for an application. More demanding constraints are orange, while less demanding ones are blue.

groups. Accordingly, the groups can either be dynamic (with mobile members or highly frequent membership changes) or static (with a generally fixed topology and few membership changes). Regarding the GC performance, the application can either supply a GC with a sufficient CPU and memory, or only one with constrained resources, or none. The energy supply for devices in the group can either be unlimited or limited. Arriving at the bottom of the tree, the reader finds which schemes are most appropriate in terms of performance for the considered group communication application. Based on the decisions made on the way to each leaf, we also determined proposed schemes in each case. For example, the decision that the performance of the GC is weak/none results in the exclusion of centralized schemes, whereas no category was excluded for a powerful GC. The decision of whether the group is static or dynamic, for example, entailed whether only considering group creation costs or also the costs of adding and removing members.

In Figure 4, the more demanding application characteristic always branches to the right. Consequently, the right-most branch represents large and highly dynamic applications with a limited energy supply that can only support weak or no GC. While developing our guidelines and constructing the decision tree, we discovered that most schemes are more appropriate for rather static groups. The decision tree resembles this fact, as the recommendations on the very right side only contain few SGC schemes. Especially for very demanding applications with large and highly dynamic groups, we can only recommend a handful of approaches. In such cases, if the application additionally only supports devices with limited energy, we can just recommend a single scheme, namely Kronos. However, Kronos only achieves such excellent results by generating the new group key based on the previous one. Thus, security is at a severe risk since an attacker can compromise any of the old keys to access all future ones. This example of Kronos illustrates our second observation: SGC schemes have to fulfill two conflicting goals: maximum efficiency vs. maximum security. The problem is that enhanced security often incurs more communication and computation costs, resulting in less efficiency. Despite this, many schemes can become more efficient if we can reduce the amounts of data transmitted between group members without compromising security.

B. SECURITY-DRIVEN SGC SCHEME SELECTION GUIDELINES

Table 13 presents an overview of the security features provided by different schemes. This table shows that the methods SMKD, MARKS, Alohali, CRGR, EGKMST, GKMP, SGR, HSHKD, SeGCom, XKFS, and SBSA each offer a unique combination of security features. Since there is only one scheme for each of these combinations of security features, our guidelines for these cases are to simply select the corresponding scheme in each case.

In the following, we now consider the cases where more than one scheme enables the respective combination of security features. We start with the set of schemes consisting of S2RP and LARK. For this set, our guidelines call for the selection of S2RP. The reason is that S2RP and LARK are both centralized schemes and exhibit the same performance except for communication costs. Regarding communication costs, S2RP is in $O(log(n))$ regardless of the topology, whereas LARK may also be in $O(n)$ in rare cases depending on the topology. Thus, our guidelines recommend the selection of S2RP.

Next, we consider the combination of security features provided by the centralized schemes ELK and LEAP. Our guidelines recommend choosing LEAP because its performance costs are always in the low range, whereas ELK's costs are in the medium to high range (see Table IV).

The schemes PCGR and SGCSH provide the same unique combination of security features that no other scheme provides. Although the two schemes offer the same combination of security features, they differ in their basic functionality. SGCSH is a centralized scheme, while PCGR is a distributed scheme. Accordingly, our guidelines recommend that if a trusted, powerful, central authority is available, SGCSH should be selected; otherwise, PCGR should be selected.

The two decentralized schemes Kronos and DEP both provide the same unique combination of security features. Since both are decentralized and the performance costs of Kronos are always in the low range, whereas the costs of DEP can also be in the medium range (see Table X), our guidelines recommend the selection of Kronos.

Our guidelines must also include a recommendation for the combination of security features only provided by KMGC and CL-EKM. Since both are centralized schemes and do not differ in terms of performance (see Table IV), a free choice between the two schemes is possible.

Another set of schemes that offers a unique combination of security features consists of the distributed schemes BKM and DFT and the centralized scheme CFKM. Since CFKM achieves its good performance for group members only when a powerful, trustworthy, and centralized authority is in place, our guidelines recommend choosing CFKM only when such an authority exists. Otherwise, our guidelines recommend the choice of DFT. The reason for not recommending BKM without such an authority is that the performance costs of BKM are always in the high range. The costs of DFT are also almost all in the high range, but the storage costs in the low range (see Table VII).

The next set of schemes with a unique combination of security features consists of the decentralized SLIMCAST, RiSeG, and LNT schemes. Here, our guidelines provide for a free choice between RiSeG and LNT, since the performance costs of SLIMCAST are only in the high range and those of RiSeG and LNT are only in the low range or identical.

The largest set of schemes that offer the same combination of security features consists of SKDC, LKH, LKH+, OFT, OFCT, TKH, EBS, D-LKH, DH-LKH, Octopus, CKA, D-OFT, SHM, SGRS, BD, G-DH, Iolus, and CS. Figure 5 supports scheme selection. This decision tree is a shortened

TABLE 13: Cumulative overview of which combination of security features are provided by which schemes.

Scheme	**Backward/Forward Secrecy**	**Instant Rekey**	**Message Integrity**	**Massage Confidentiality**	**Member Authentication**	**Compromise Robustness**	**Group Independence**
SMKD	✓/✗	✗	✗	✗	✓	✗	✓
IGKMP, Hydra	✓/✓	✓	✗	✓	✓	✗	✓
MARKS	✗/✗	✗	✗	✗	✗	✗	✓
Alohali	✓/✓	✓	✗	✓	✓	✓	✓
CRGR	✓/✓	✓	✓	✓	✗	✓	✓
EGKMST	✓/✓	✓	✓	✓	✓	✗	✗
GKMP	✓/✗	✓	✗	✗	✗	✗	✓
SGR	✗/✗	✗	✓	✓	✓	✓	✓
HSHKD	✗/✗	✗	✓	✓	✗	✓	✓
SeGCom	✓/✓	✗	✓	✓	✓	✗	✓
XKFS	✓/✓	✓	✗	✓	✓	✗	✓
SBSA	✓/✓	✓	✗	✓	✗	✓	✓
S2RP, LARK	✓/✓	✓	✓	✓	✗	✗	✓
ELK, LEAP	✓/✓	✗	✗	✓	✗	✗	✓
PCGR, SGCSH	✗/✗	✗	✓	✓	✗	✗	✓
Kronos, DEP	✓/✗	✗	✗	✓	✓	✗	✓
KMGC, CL-EKM	✓/✓	✓	✓	✓	✓	✓	✓
BKM, DFT, CFKM	✓/✓	✓	✗	✗	✗	✗	✓
SLIMCAST, RiSeG, LnT	✓/✓	✓	✓	✓	✓	✗	✓
Iolus, CS, DLKH, DH-LKH, Octopus, CKA, SKDC, LKH, LKH+, OFT, OFCT, TKH, EBS, DOFT, SHM, SGRS, BD, G-DH	✓/✓	✓	✗	✓	✗	✗	✓

version of the decision tree in Figure 4, which considered all schemes, allowing for a more fine granular selection than when only considering procedures with the same security features.

IX. CONCLUSION

Applications with groups of communicating devices are rapidly growing in importance as electronic communication becomes more sophisticated. The emergence of the Internet-of-Things and fast network technologies such as 5G are increasing the level and speed of connectivity, leading to even more group communication applications. Researchers proposed several schemes for secure group communication (SGC). These SGC schemes securely manage cryptographic keys in groups that use many-to-many communication encrypted with a shared symmetric group key. The security and efficiency of SGC schemes vary significantly depending on the application and its group characteristics [3]. Developers who need to integrate security into group communication must choose one of the overwhelming number of SGC schemes. Additionally, they have to ensure that it is efficient and secure enough for their specific application. Our survey approached this problem by comparing and evaluating a total of 47 different SGC schemes in terms of security and efficiency. We covered the three main categories of SGC schemes: centralized, distributed, and decentralized. We used the metrics storage costs, communication costs, computation costs, key update frequency, and types of cryptography used. We analyzed if each of the 47 schemes achieves the requirements of forward and backward secrecy, instant rekeying, message integrity, message confidentiality, member authentication, compromise robustness, and group independence. Moreover, we identified the most suitable and best-performing schemes for different scenarios of applications with communicating groups. These scenarios cover differences in group size, group dynamism, the performance of the group controller, and the provided energy supply. Based on these results, we proposed guidelines to assist developers in choosing an appropriate scheme for their specific application requirements. We also illustrate our recommendations with decision trees that further facilitate the process of selecting a scheme for a given application scenario. While developing our guidelines, we observed that most schemes are more appropriate to use in rather static groups. Especially for very demanding applications with groups that are large in addition to being highly dynamic, we can only recommend a handful of SGC schemes. Our second observation is a fundamental problem: SGC schemes have to fulfill two conflicting goals: maximum efficiency and maximum security. The problem is that enhanced security often requires more complex computations, resulting in less efficiency.

The development of approaches to address this problem is an important objective for future work on group key management and secure group communication. Solutions have to provide increased efficiency and scalability without having a negative impact on security. Many existing schemes can become more efficient if we can reduce the amounts of data transmitted between group members without compromising security. Nonetheless, novel approaches are necessary to boost efficiency further while maintaining the level of security.

Our survey provides the following two two benefits for researchers in the field of group communication, as well as developers integrating security into group communication.

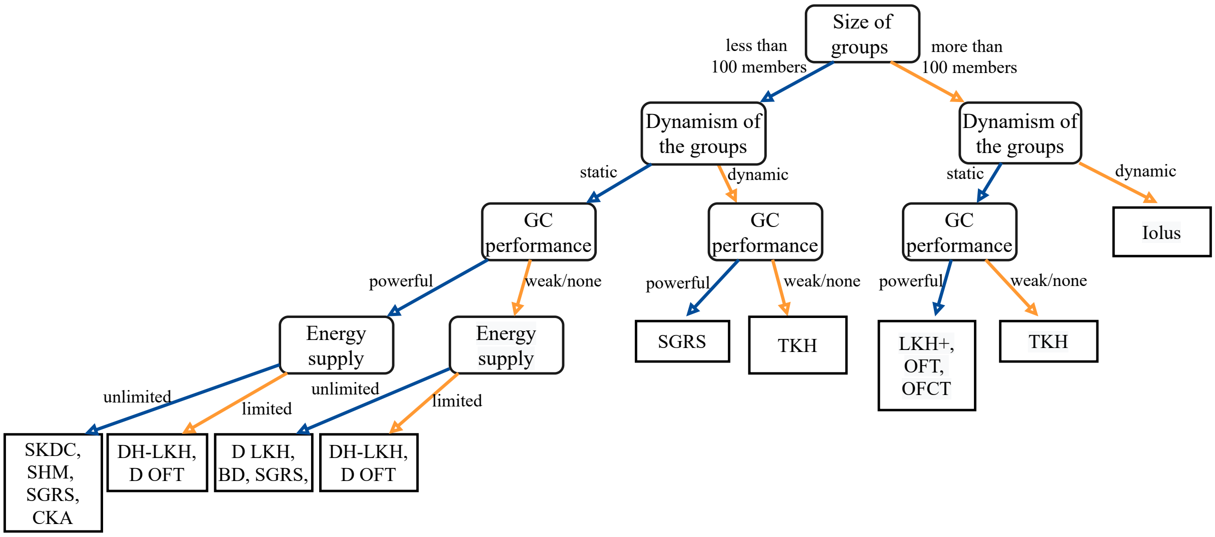


FIGURE 5: Decision tree illustrating our guidelines for the security-driven choice of a suitable SGC scheme for an application. More demanding constraints are orange, while less demanding ones are blue.

First, we give a large and detailed overview of a total of 47 different SGC schemes from all the three main GKM categories, centralized, distributed, and decentralized. Therefore, we extend our previous practical analyses (see [80]–[89]) regarding the security and performance of IoT group communication with a theoretical review of SGC schemes. Second, we provide extensive guidelines to assist developers in integrating security into their group communication applications. These guidelines allow developers to easily select a suitable SGC scheme providing efficient GKM for their specific application while satisfying all security requirements.

REFERENCES

[1] T. Prantl *et al.*, "Simpl: Secure iot management platform," in *ITSec*, ser. 1st ITG Workshop on IT Security, 2020.

[2] N. Renugadevi, G. Swaminathan, and A. S. Kumar, "Key management schemes for secure group communication in wireless networks - a survey," in *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, 2014, pp. 446–450.

[3] D. S. Colin Boyd, Anish Mathuria, *Protocols for Authentication and Key Establishment*, ser. Information Security and Cryptography. Springer-Verlag Berlin Heidelberg, 2020, vol. 2.

[4] M. Bilal and S.-G. Kang, "A secure key agreement protocol for dynamic group," *Cluster Computing*, vol. 20, no. 3, pp. 2779–2792, 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-0853-0

[5] B. A. Alohali, V. G. Vassilakis, I. D. Moscholios, and M. D. Logothetis, "A secure scheme for group communication of wireless iot devices," in *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2018, pp. 1–6.

[6] E. von Gravrock, "How 5g, ai and iot are set to accelerate digital transformation," *Forbes Los Angeles Buisiness Council*, 2019. [Online]. Available: https://www.forbes.com/sites/forbeslacouncil/2019/05/23/how-5g-ai-and-iot-are-set-to-accelerate-digital-transformation/#68ed1eef183a

[7] D. Newman, "Top 10 digital transformation trends for 2020," *Forbes Los Angeles Buisiness Council*, 2019. [Online]. Available: https://www.forbes.com/sites/danielnewman/2019/07/14/top-10-digital-transformation-trends-for-2020/#1e8e5bc276be

[8] G. Perrone, M. Vecchio, R. Pecori, and R. Giaffreda, *The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices*, 01 2017.

[9] 2018 reform of eu data protection rules. European Commission. [Online]. Available: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf

[10] O. Cheikhrouhou, "Secure group communication in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 61, pp. 115–132, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804515002349

[11] Online images sources: https://icons8.com/icon/555/key, https://icons8.com/icon/94/sperren, https://www.freepik.com/vectors/fitness-tracker, https://icons8.com/icon/hTNuSzXPYomv/insurance, https://www.freepik.com/vectors/doctor-equipment.

[12] T. Prantl, P. Ten, L. Iffländer, A. Dmitrenko, S. Kounev, and C. Krupitzer, "Evaluating the performance of a state-of-the-art group-oriented encryption scheme for dynamic groups in an iot scenario," in *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2020, pp. 1–8.

[13] K. Nishat and B. R. Purushothama, "Group-oriented encryption for dynamic groups with constant rekeying cost," *Security and Communication Networks*, vol. 9, no. 17, pp. 4120–4137, 2016.

[14] O. Rodeh, K. Birman, and D. Dolev, "Optimized group rekey for group communication systems," 03 2000.

[15] M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, "The versakey framework : versatile group key management," *IEEE Journal on selected areas in communications*, vol. 17, no. 9, pp. 1–16, 1999.

[16] P. Sakarindr and N. Ansari, "Security services in group communications over wireless infrastructure, mobile ad hoc, and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 8–20, 2007.

[17] T. T. Mapoka, "Group key management protocols for secure mobile multicast communication: A comprehensive survey," *International Journal of Computer Applications*, vol. 84, pp. 28–38, 12 2013.

[18] C. Kowalczyk. Crypto-it: Symmetric ciphers. Last accessed: 2020-09-02. [Online]. Available: http://www.crypto-it.net/eng/symmetric/index.html

[19] ——. Crypto-it: Asymmetric ciphers. Last accessed: 2020-09-02. [Online]. Available: http://www.crypto-it.net/eng/asymmetric/index.html

[20] M. J. B. Robshaw, *One-Way Function*. Boston, MA: Springer US, 2011, pp. 887–888. [Online]. Available: https://doi.org/10.1007/978-1-4419-5906-5_467

[21] M. Just, *Diffie–Hellman Key Agreement*. Boston, MA: Springer US, 2011, pp. 341–342. [Online]. Available: https://doi.org/10.1007/978-1-4419-5906-5_75

[22] S. Kalra and S. Sood, "Elliptic curve cryptography: Current status and research challenges," vol. 169, 01 2011, pp. 455–460.

[23] C. Kowalczyk. Crypto-it: Pseudorandom generator (prg). Last accessed: 2020-08-01. [Online]. Available: http://www.crypto-it.net/eng/theory/pseudorandom-generator.html

[24] ——. Crypto-it: Pseudorandom functions and permutations. Last accessed: 2020-08-01. [Online]. Available: http://www.crypto-it.net/eng/theory/prf-and-prp.html

[25] S. Rafaeli and D. Hutchison, "A survey of key management for secure group communication," *ACM Comput. Surv.*, vol. 35, pp. 309–329, 09 2003.

[26] X. He, M. Niedermeier, and H. de Meer, "Dynamic key management in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 611–622, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804512002573

[27] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11, pp. 2314–2341, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366407001752

[28] J. A. M. Naranjo and L. G. Casado, "Keeping group communications private: An up-to-date review on centralized secure multicast," in *Computational Intelligence in Security for Information Systems*, Á. Herrero and E. Corchado, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 151–159.

[29] B. Jiang and X. Hu, "A survey of group key management," in *2008 International Conference on Computer Science and Software Engineering*, vol. 3, 2008, pp. 994–1002.

[30] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, "A survey on cluster-based group key agreement protocols for wsns," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 429–442, 2011.

[31] S. Li and Y. Wu, "A survey on key management for multicast," in *2010 Second International Conference on Information Technology and Computer Science*, 2010, pp. 309–312.

[32] M. Manulis, "Contributory group key agreement protocols, revisited for mobile ad-hoc groups," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, 2005, pp. 8 pp.–818.

[33] E. Bresson and M. Manulis, "Contributory group key exchange in the presence of malicious participants," *IET Information Security*, vol. 2, no. 3, pp. 85–93, 2008.

[34] H. Annapurna and M. Siddappa, "A survey on security techniques in group communication for wireless sensor networks," *International Journal of Computer Applications*, vol. 113, pp. 6–12, 03 2015.

[35] T. Ballardie and J. Crowcroft, "Multicast-specific security threats and counter-measures," in *Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95)*, ser. SNDSS '95. USA: IEEE Computer Society, 1995, p. 2.

[36] H. Harney and C. Muckenhirn, "Rfc2094: Group key management protocol (gkmp) architecture." USA: RFC Editor, 1997.

[37] ——, "Rfc2093: Group key management protocol (gkmp) specification," USA, 1997.

[38] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *Proceedings of the 7th ACM Conference on Computer and Communications Security*, ser. CCS '00. New York, NY, USA: Association for Computing Machinery, 2000, pp. 235–244. [Online]. Available: https://doi.org/10.1145/352600.352638

[39] ——, "Tree-based group key agreement," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 60–96, Feb. 2004. [Online]. Available: https://doi.org/10.1145/984334.984337

[40] B. DeCleene, L. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, and C. Zhang, "Secure group communications for wireless networks," in *2001 MILCOM Proceedings Communications for Network-Centric Operations: Creating the Information Force (Cat. No.01CH37277)*, vol. 1, 2001, pp. 113–117 vol.1.

[41] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.

[42] L. R. Dondeti, S. Mukherjee, and A. Samal, "A distributed group key management scheme for secure many-to-many communication," 1999.

[43] S. Mittra, "Iolus: A framework for scalable secure multicasting," in *Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '97. New York, NY, USA: Association for Computing Machinery, 1997, pp. 277–288. [Online]. Available: https://doi.org/10.1145/263105.263179

[44] M. Tubaishat, J. Yin, B. Panja, and S. Madria, "A secure hierarchical model for sensor network," *SIGMOD Rec.*, vol. 33, no. 1, pp. 7–13, Mar. 2004. [Online]. Available: https://doi.org/10.1145/974121.974123

[45] B. Briscoe, "Marks: Zero side effect multicast key management using arbitrarily revealed key sequences," in *Rizzo L., Fdida S. (eds) Networked Group Communication, NGC 1999*, vol. 1736. Berlin, Heidelberg: Springer, 11 1999.

[46] A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, 2003.

[47] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: a predistribution and local collaboration-based approach," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1, 2005, pp. 503–514 vol. 1.

[48] S. Setia, S. Koussih, S. Jajodia, and E. Harder, "Kronos: A scalable group re-keying approach for secure multicast," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, ser. SP '00. USA: IEEE Computer Society, 2000, pp. 215–228.

[49] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, vol. 2, 1999, pp. 708–716 vol.2.

[50] S. Guo and A.-N. Shen, "A compromise-resilient pair-wise rekeying protocol in hierarchical wireless sensor networks," *Comput. Syst. Sci. Eng.*, vol. 25, 2010.

[51] Jyh-How Huang, J. Buckingham, and R. Han, "A level key infrastructure for secure and efficient group communication in wireless sensor network," in *Proceedings of the first International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. IEEE Computer Society, 2005, pp. 249–260.

[52] G. Dini and I. M. Savino, "S2rp: a secure and scalable rekeying protocol for wireless sensor networks," in *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 06)*, Vancouver, Canada, 2006, pp. 457–466.

[53] M. Wen, Y.-F. Zheng, W.-j. Ye, K.-F. Chen, and W.-D. Qiu, "A key management protocol with robust continuity for sensor networks," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 642–647, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548908000937

[54] O. Cheikhrouhou, A. Koubâa, G. Dini, and M. Abid, "Riseg: a ring based secure group communication protocol for resource-constrained wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 15, no. 8, pp. 783–797, 2011. [Online]. Available: https://doi.org/10.1007/s00779-011-0365-5

[55] G. Dini and I. M. Savino, "Lark: A lightweight authenticated rekeying scheme for clustered wireless sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 10, no. 4, Nov. 2011. [Online]. Available: https://doi.org/10.1145/2043662.2043665

[56] A. Diop, Y. Qi, and Q. Wang, "Efficient group key management using symmetric key and threshold cryptography for cluster based wireless sensor networks," *International Journal of Computer Network and Information Security*, vol. 6, pp. 9–18, 2014.

[57] O. Cheikhrouhou, A. Koubâa, G. Dini, H. Alzaid, and M. Abid, "Lnt: A logical neighbor tree secure group communication scheme for wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1419–1444, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870512000704

[58] J.-H. Son, J.-S. Lee, and S.-W. Seo, "Topological key hierarchy for energy-efficient group key management in wireless sensor networks," *Wireless Personal Communications*, vol. 52, no. 2, p. 359, 2010. [Online]. Available: https://doi.org/10.1007/s11277-008-9653-4

[59] L. Dondeti, S. Mukherjee, and A. Samal, "Scalable secure one-to-many group communication using dual encryption," *Computer Communications*, vol. 23, no. 17, pp. 1681 – 1701, 2000.

[60] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *A.D. Santis (ed.) Advances in Cryptology – EUROCRYPT '94, Lecture Notes in Computer Science*, vol. 950. Springer-Verlag, 1994, pp. 275–286.

[61] R. Molva and A. Pannetrat, "Scalable multicast security in dynamic groups," in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, ser. CCS '99. New York, NY, USA: Association for Computing Machinery, 1999, pp. 101–112. [Online]. Available: https://doi.org/10.1145/319709.319723

[62] J. Tygar, A. Perrig, and D. Song, "Elk, a new protocol for efficient large-group key distribution," in *2012 IEEE Symposium on Security and Privacy*. Los Alamitos, CA, USA: IEEE

Computer Society, may 2001, p. 0247. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SECPRI.2001.924302

[63] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-hellman key distribution extended to group communication," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, ser. CCS '96. New York, NY, USA: Association for Computing Machinery, 1996, pp. 31–37. [Online]. Available: https://doi.org/10.1145/238168.238182

[64] S. Rafaeli and D. Hutchison, "Hydra: a decentralised group key management," in *Proceedings. Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Pittsburgh, PA, USA, 2002, pp. 62–67.

[65] F. Kausar, S. Hussain, J. H. Park, and A. Masood, "Secure group communication with self-healing and rekeying in wireless sensor networks," in *Proceedings of the 3rd International Conference on Mobile Ad-Hoc and Sensor Networks*, ser. MSN'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 737–748.

[66] K. Becker and U. Wille, "Communication complexity of group key distribution," in *Proceedings of the 5th ACM Conference on Computer and Communications Security*, ser. CCS '98. New York, NY, USA: Association for Computing Machinery, 1998, pp. 1–6. [Online]. Available: https://doi.org/10.1145/288090.288094

[67] Y. Yang, J. Zhou, R. H. Deng, and F. Bao, "Hierarchical self-healing key distribution for heterogeneous wireless sensor networks," in *Security and Privacy in Communication Networks*, Y. Chen, T. D. Dimitriou, and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 285–295.

[68] C. Boyd, "On key agreement and conference key agreement," in *Varadharajan V., Pieprzyk J., Mu Y. (eds) Information Security and Privacy ACISP 1997*, vol. 1270. Springer, Berlin Heidlerberg, 1997.

[69] S. Zhu, S. Setia, and S. Jajodia, "Leap+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 4, pp. 500–528, Nov. 2006. [Online]. Available: https://doi.org/10.1145/1218556.1218559

[70] M. Eltoweissy, M. H. Heydari, L. Morales, and I. H. Sudborough, "Combinatorial optimization of group key management," *Journal of Network and Systems Management*, vol. 12, no. 1, pp. 33–50, 2004. [Online]. Available: https://doi.org/10.1023/B:JONS.0000015697.38671.ec

[71] M. Eltoweissy, A. Wadaa, S. Olariu, and L. Wilson, "Group key management scheme for large-scale sensor networks," *Ad Hoc Networks*, vol. 3, no. 5, pp. 668–688, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870504000733

[72] O. Gaddour, A. Koubâa, and M. Abid, "Segcom: A secure group communication mechanism in cluster-tree wireless sensor networks," in *2009 First International Conference on Communications and Networking*, 2009, pp. 1–7.

[73] A. Ghafoor, M. Sher, M. Imran, and K. Saleem, "A lightweight key freshness scheme for wireless sensor networks," in *2015 12th International Conference on Information Technology - New Generations*, 2015, pp. 169–173.

[74] P. Szalachowski and T. H.-J. Kim, "Secure broadcast in distributed networks with strong adversaries," *Security and Communication Networks*, vol. 8, no. 18, pp. 3739–3750, 2020/06/05 2015. [Online]. Available: https://doi.org/10.1002/sec.1296

[75] Xirong Bao, Jin Liu, Lihuang She, and Shi Zhang, "A key management scheme based on grouping within cluster," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, 2014, pp. 3455–3460.

[76] S.-H. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 371–383, 02 2015.

[77] S. Raazi and S. Lee, "A survey on key management strategies for different applications of wireless sensor networks," *JCSE*, vol. 4, pp. 23–51, 03 2010.

[78] R. Du and S. Wen, "An improved scheme of mutesla authentication based trusted computing platform," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1–4.

[79] A. E. Hegazy, A. M. Darwish, and R. El-Fouly, "Reducing mutesla memory requirements," in *2007 Second International Conference on Systems and Networks Communications (ICSNC 2007)*, 2007, pp. 33–33.

[80] S. Herrnleben, R. Ailabouni, J. Grohmann, T. Prantl, C. Krupitzer, and S. Kounev, "An iot network emulator for analyzing the influence of varying network quality," in *International Conference on Simulation Tools and Techniques*. Springer, 2020, pp. 580–599.

[81] S. Herrnleben, M. Leidinger, V. Lesch, T. Prantl, J. Grohmann, C. Krupitzer, and S. Kounev, "Combench: A benchmarking framework for publish/subscribe communication protocols under network limitations," in *EAI International Conference on Performance Evaluation Methodologies and Tools*. Springer, 2021, pp. 72–92.

[82] S. Herrnleben, J. Grohmann, V. Lesch, T. Prantl, F. Metzger, T. Hoßfeld, and S. Kounev, "Investigating the predictability of qos metrics in cellular networks," in *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. IEEE, 2022, pp. 1–10.

[83] T. Prantl, L. Iffländer, S. Herrnleben, S. Engel, S. Kounev, and C. Krupitzer, "Performance impact analysis of securing mqtt using tls," in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 2021, pp. 241–248.

[84] T. Prantl, P. Ten, L. Iffländer, A. Dmitrenko, S. Kounev, and C. Krupitzer, "Evaluating the performance of a state-of-the-art group-oriented encryption scheme for dynamic groups in an iot scenario," in *2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE, 2020, pp. 1–8.

[85] T. Prantl, P. Ten, L. Iffländer, S. Herrnleben, A. Dmitrenko, S. Kounev, and C. Krupitzer, "Towards a group encryption scheme benchmark: A view on centralized schemes with focus on iot," in *Proceedings of the ACM/SPEC International Conference on Performance Engineering*, 2021, pp. 233–240.

[86] T. Prantl, D. Prantl, L. Beierlieb, L. Iffländer, A. Dmitrienko, S. Kounev, and C. Krupitzer, "Performance evaluation for a post-quantum public-key cryptosystem," in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. IEEE, 2021, pp. 1–7.

[87] T. Prantl, D. Prantl, A. Bauer, L. Iffländer, A. Dmitrienko, S. Kounev, and C. Krupitzer, "Benchmarking of pre-and post-quantum group encryption schemes with focus on iot," in *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*. IEEE, 2021, pp. 1–10.

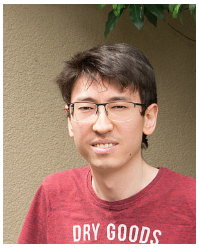
[88] T. Prantl, T. Zeck, L. Iffländer, L. Beierlieb, A. Dmitrenko, C. Krupitzer, and S. Kounev, "Towards a cryptography benchmark: A view on attribute based encryption schemes," in *2022 5th Conference on Cloud and Internet of Things (CIoT)*. IEEE, 2022, pp. 158–165.

[89] T. Prantl, S. Engel, A. Bauer, A. E. B. Yahya, S. Herrnleben, L. Iffländer, A. Dmitrienko, and S. Kounev, "An experience report on the suitability of a distributed group encryption scheme for an iot use case," in *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*. IEEE, 2022, pp. 1–7.

THOMAS PRANTL received the bachelor's and master's degrees from the University of Würzburg. He is currently a Doctoral Researcher with the Security Testing & Benchmarking Research Group at the Software Engineering Chair, University of Würzburg.

TIMO ZECK is a master's student at the Julius Maximilians University. His research interests are in the area of IoT security and deep learning. His bachelor thesis was awarded with the Würzburg Software Engineering Award 2021. He is currently writing his master's thesis on the detection of IoT botnets using deep learning.

ANDRÉ BAUER received the Ph.D. degree in computer science from the University Würzburg, Würzburg, Germany, in 2020. He is currently head of the Software Engineering for Applied Data Analytics Group Research Group at the Software Engineering Chair, University of Würzburg.



PETER TEN is a master's student at the Julius Maximilians University. His research interests are in the area of IoT and web-security and he is a member of the Secure Software Systems group at the department of Software Engineering at the Julius Maximilians University. As a member of the group, he worked on the IoT project SIMPL that was funded by the German Federal Ministry of Education and Research to research a secure framework for the communication of a heterogenous and dynamic network. Further, he was working on the web-security of the web-based eSano project, which is a research project in the area of E-Mental and E-Behavior health intervention that is developed and researched by the departments Clinical Psychology and Psychotherapy and the Institute of Databases and Information Systems at the University of Ulm, in cooperation with the Institute for Clinical Epidemiology and Biometry at the Julius Maximilians University.

DOMINIK PRANTL is a student at the Julius Maximilians University. His study interests are in the field of mathematics and computer science. He is currently working towards his bachelor's degree in computer science and natural science foundations.

ALA EDDINE BEN YAHYA is a research assistant and a member of the Secure Software Systems group at the University of Wuerzburg in Germany. He is passionate about IoT security, networks security, anonymity networks, privacy and secure communication.

LUKAS IFFLÄNDER received a bachelor's, master's, and Ph.D. degree from the University of Würzburg, Germany, where he is heading the Security Testing and Benchmarking Group of the Software Engineering Chair. He also works as a desk officer for cyber scecurity at the German Centre for Rail Transport Research (DZSF).

ALEXANDRA DMITRIENKO is an associate professor at the University of Wuerzburg in Germany, where she is heading the Secure Software Systems group. Before taking her current faculty position in 2018, she worked for about 10 years in renowned security institutions in Germany and in Switzerland: Ruhr-University Bochum (2008-2011), Fraunhofer Institute for Information Security in Darmstadt (2011-2015), and ETH Zurich (2016-2017). She holds a PhD degree in Security and Information Technology from TU Darmstadt (2015). Her PhD dissertation focused on the security and privacy of mobile systems and applications and was awarded by the European Research Consortium in Informatics and Mathematics (ERCIM STM WG 2016 Award) and recognized as outstanding by Intel – she received an Intel Doctoral Student Honor Award (2013). Today, her research interests focus on various topics on secure software engineering, systems security and privacy, and security and privacy of mobile, cyber-physical, and distributed systems.

CHRISTIAN KRUPITZER received a bachelor's, master's, and Ph.D. degree from the University of Mannheim, Germany, in 2010, 2012, and 2018, respectively. Since October 2020, he is tenure track professor and leads the Department of Food Informatics at the University of Hohenheim in Stuttgart, Germany. His research interests include applying principles of adaptive systems and machine learning for IIoT (focusing on food production), intelligent transportation, and sports. He is involved in the organization of workshops and conferences, such as IEEE PerCom and IEEE ACSOS, and a reviewer for conferences and journals, e.g., IEEE T-ITS, IEEE IoTJ, or Elsevier FGCS.

SAMUEL KOUNEV received the Ph.D. degree in computer science from TU Darmstadt, Darmstadt, Germany, in 2005.,He is currently a Professor with the Chair of Software Engineering, University of Würzburg, Würzburg, Germany. His research interests include the engineering of dependable and efficient software systems, systems benchmarking and experimental analysis, and autonomic and self-aware computing.,Dr. Kounev is a member of ACM and the German Computer Science Society.