International Journal of Information Security
https://doi.org/10.1007/s10207-023-00692-0

REGULAR CONTRIBUTION

Recommendation of secure group communication schemes using multi-objective optimization

Thomas Prantl[1] · André Bauer[3] · Lukas Iffländer[1] · Christian Krupitzer[2] · Samuel Kounev[1]

Abstract
The proliferation of IoT devices has made them an attractive target for hackers to launch attacks on systems, as was the case with Netflix or Spotify in 2016. As the number of installed IoT devices is expected to increase worldwide, so does the potential threat and the importance of securing these devices and their communications. One approach to mitigate potential threats is the usage of the so-called Secure Group Communications (SGC) schemes to secure the communication of the devices. However, it is difficult to determine the most appropriate SGC scheme for a given use case because many different approaches are proposed in the literature. To facilitate the selection of an SGC scheme, this work examines 34 schemes in terms of their computational and communication costs and their security characteristics, leading to 24 performance and security features. Based on this information, we modeled the selection process for centralized, distributed, and decentralized schemes as a multi-objective problem and used decision trees to prioritize objectives.

Keywords Secure group communication scheme · Recommendation · Multi-objective optimization · Pareto front · Guidelines

1 Introduction

In 2016, various Internet services such as Netflix or Spotify were brought down by the largest *distributed denial of service* (DDoS) attack. This DDos attack was carried out by thousands of *Internet-of-Things* (IoT) devices [1] and was made possible in the first place because companies that develop such devices fail to secure their products [2, 3]. And on top of that, IoT devices are often installed without considering security [1].

According to Cisco [4], the number of installed IoT devices will grow to 14 billion worldwide, enabling an even greater potential for threats than the DDoS attack in 2016. Indeed, this growth also offers benefits in different areas, such as smart home, smart factory, remote healthcare, or traffic management. To offset the potential threats with the benefits, it is of utmost importance to secure these devices and especially their communications.

In contrast to the standard 1-to-1 communication encryption, the n-to-n communication that IoT devices use is more difficult to encrypt. The reason for this is that messages need to be encrypted for a group of recipients. To handle the n-to-n encryption in an efficient manner, so-called *Secure Group Communication* (SGC) schemes [5] can be applied. The idea of these schemes is to perform encryption of messages only once for the whole group instead of encrypting them individually for each group member [6].

The choice of which SGC scheme should be applied for a specific use case is crucial as there exist a large number of proposed schemes in the literature. They differ in their architecture, workflow, security features, and performance. For instance, some schemes require the presence of a trusted third party [7] while others do not ([8, 9]). In order to facilitate the selection of a suitable scheme, we have elaborated an overview as well as guidelines in our previous work [10]. These guidelines and overviews were also intended to make it easier for developers to comply with legal requirements

✉ Thomas Prantl
Thomas.Prantl@uni-wuerzburg.de

André Bauer
andrebauer@uchicago.edu

Christian Krupitzer
christian.krupitzer@uni-hohenheim.de

Samuel Kounev
Samuel.Kounev@uni-wuerzburg.de

1 Julius-Maximilians-Universität Würzburg, Würzburg, Germany

2 Universität Hohenheim, Stuttgart, Germany

3 Universität Chicago, Chicago, USA

Published online: 25 May 2023

such as the General Data Protection Regulation in Europe or the Internet of Things Cybersecurity Improvement Act in the USA, which mandate the use of appropriate encryption techniques to protect privacy [3, 11].

Prior to our initial survey, there existed only the survey [5] from Cheikhrouhou, which provided corresponding overviews and guidelines. Cheikhrouhou analyzed 22 schemes based on 10 criteria. These 10 criteria included both performance metrics and security features, but the guidelines were based only on the performance metrics. This work by Cheikhrouhou has already been extended by our initial survey [10], which analyzed additional schemes based on 12 criteria and provided guidelines that consider both performance metrics and security features.

In this article, we heavily extend our previous work [10] by providing a more detailed overview, extracting more features, and refining our guidelines. More precisely, our contributions are (i) we now break down the computational cost of an SGC scheme regarding the group operations addition and revocation of group members as well as the group creation; (ii) We also break down the total communication costs of an SGC scheme in terms of the group operations of adding and revoking group members and group creation and consider also the number of send messages as well the message size; (iii) The guideline considers now also the life cycle and the applied cryptography technique. Due to these enhancements, our original considered 12 performance and security features have doubled to 24, allowing for a more refined selection guide. We also altered our previous guideline by modeling the selection process as a multi-objective problem considering both performance and security features. To find now the best-suited SGC scheme, we first had to determine the Pareto fronts. Those Pareto fronts symbolize the best possible solutions found in the optimization procedure. Due to the multiple objectives, several solutions might be the result of the optimization, all having the same utility but none of the objectives can be improved without degrading some of the other objectives. Nevertheless, in order to select from the schemes that form the Pareto front, we prioritize the objectives differently using decision trees. To make a decision in our previous work, we also used decision trees, but they were much simpler and did not require the determination of the Pareto fronts because (1) they considered only half of the decision factors and (2) they were either only performance or security oriented.

To underpin the applicability of our contributions, we would like to show how our decision trees can support the selection of a concrete scheme. For this purpose, we consider an Industry 4.0 scenario, which is a typical IoT use case [12–14]. In this example scenario, we assume that the owner is most concerned with control over the system. Therefore, the creation and updating of groups should be controlled by a single central entity, which alone has the authority to issue

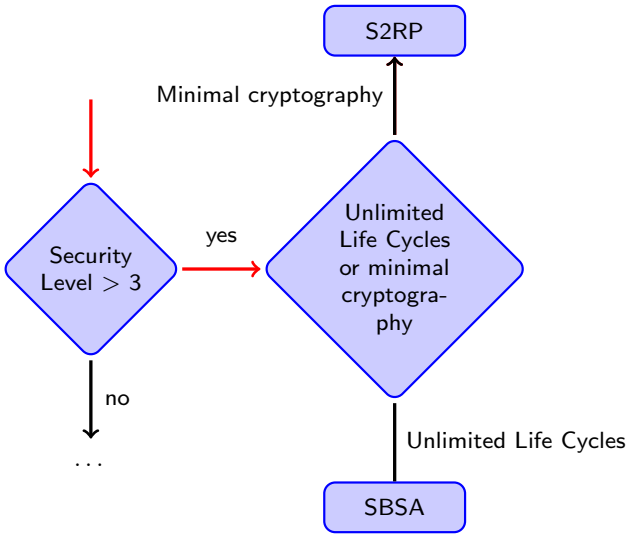

Fig. 1 Shortened decision tree for the selection of a centralized scheme for the example scenario of an Industry 4.0 use case. For this scenario, it is assumed that (1) a centralized scheme should be used, (2) security features are more important than performance, and (3) the highest possible security level should be maintained

instructions for group operations. Therefore, as we will see later, only centralized systems are suitable for the owner. Moreover, security is more important to the owner than performance, and the owner therefore wants to have the highest possible level of security. If we now consider the corresponding decision tree for centralized systems, we would first select the highest security level, which would be level 4 in the case of centralized systems. The corresponding decision tree, shortened accordingly, is shown in Fig. 1. In this decision tree, the facility owner would answer yes to the question of whether the security level should be greater than 3. Thus, the choice of possible schemes is already limited to the two schemes S2RP and SBSA. To make a choice between these two schemes, the owner must decide which is more important to him: that unlimited group operations are possible or that the number of cryptographic techniques used is minimal. In the former case, the choice would be SBSA; in the latter, S2RP.

The remainder of this paper is structured as follows. In Sect. 2, we review the foundations that are essential for understanding our work. This includes the definition of Secure Group Communication (SGC) schemes, the classification of SGC schemes, the features of SGC schemes, cryptographic techniques, and the problem of multi-objective optimization.

Then, in Sect. 3, we show how we extended and adapted the information about SGC schemes from our previous survey [10] for multi-objective optimization analysis and how a Pareto front can be determined for SGC schemes.

Based on this, we determine guidelines for selecting centralized, distributed, and decentralized schemes in Sects. 4,

5, and 6, respectively. In doing so, we establish guidelines for each category for the following cases, respectively: The selection of a scheme depends (1) only on its performance, (2) only on its features, and (3) on its performance and features. We then present cross-category guidelines in Sect. 7, again distinguishing whether only performance, only features, or performance and features should be considered for selection. A discussion of the designed decision trees is provided in Sect. 8. In Sect. 9, we provide an overview as well as a differentiation from related work and conclude the paper with a summary in Sect. 10.

2 Background

In this section, we briefly introduce the foundations that are important for understanding the following content. Specifically, it highlights the definition and classification of secure group communication (SGC) schemes, what features SGC schemes can have, and what cryptographic techniques they are built on. Finally, we introduce the concept of multi-objective optimization along with the Pareto principle.

2.1 Secure group communication schemes definition

According to the definition by Sakarindr et al. [15] and Cheikhrouhou et al. [5], an SGC scheme consists of the following two components: the *group membership management* (GMM) and the *group key management* (GKM).

Group membership management (GMM) includes the required operations for maintaining the group, i.e., group creation, the addition of members to a group, or removal of members from a group. The GMM component has to securely specify such operations. However, the GMM component only defines the exclusion or addition of members, e.g., in the form of a list maintained by the group controller. All other operations, which take care of updating and distribution of the keys, are handled by the GMM component.

Group key management (GKM) provides a secret group key that can be used by the group members. For this purpose, a corresponding protocol must be offered by the GKM component, which defines the generation, distribution, and updates of the group key.

2.2 Secure group communication schemes classification

SGC schemes can be divided into the following three classes: (i) centralized, (ii) distributed/contributory, and (iii) decentralized/hybrid. In the following, we introduce these categories in more detail, which are also illustrated in Fig. 2.

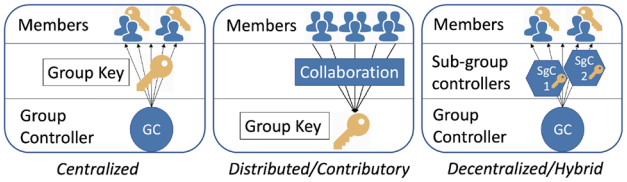

Fig. 2 Illustration of the three classes of SGC schemes from our survey [16]

Centralized SGC schemes have a trusted third party, which is also called a central authority (CI). The CI handles the creation and updating of the group, calculates the parameters needed for it, and sends them to the group members. Thus, for the group creation only communication between the CI and the respective members is necessary and no communication of the group members among themselves. The complete counter design to centralized SGC schemes are the distributed/contributory SGC schemes. This class of SGC schemes does not have a central authority, so members must communicate among themselves to agree on a group key. A mixture of centralized and distributed/contributory SGC schemes are the decentralized/hybrid SGC schemes. In this category, the group is typically divided into subgroups, with each subgroup managed by a group member called the subgroup controller. In this case, the CI communicates only with the subgroup controllers, which in turn communicate with the group members of their subgroup. Thus, there is both a CI and communication among the group members.

2.3 Secure group communication schemes security features

SGC schemes can provide eight security features, which we briefly present in the following. The first two features that SGC schemes can have are *Backward Secrecy* and *Forward Secrecy*. *Backward Secrecy* describes that members who are newly added to the group cannot decrypt group messages that were encrypted before they joined the group. *Forward Secrecy*, on the other hand, ensures that members who have been removed from the group cannot decrypt group messages that were encrypted after their removal. SGC schemes that provide the *Instant Rekey* feature update their group key immediately after any change in the group composition and not later when, for example, a certain period of time has elapsed. If an SGC scheme provides the *Message Integrity* or *Message Confidentiality* feature, messages are protected from malicious modification by third parties or third parties cannot learn any meaningful information from the messages. In addition to tamper resistance and confidentiality, messages can also have the property that the sender can be determined from them. This property is called *Member Authentication*. Another property that SGC schemes can have is *Compromise Robustness*. If an SGC scheme satisfies this feature, then in

the case of one or more compromised group members, the affected group members can be removed in such a way that subsequently group communication is secure again. The last feature that SGC schemes can have is *Group Independence*. This feature deals with the problem that a group member can be part of several groups and that a compressed group can have effects on other groups. If this is not the case, then an SGC scheme provides the feature *Group Independence*.

2.4 Cryptographic techniques

The cryptographic techniques on which SGC schemes can be based can be grouped thematically into three blocks: symmetric cryptography, asymmetric cryptography, and pseudo-random numbers/functions.

Symmetric cryptography schemes use the same key to encrypt messages and decrypt messages. Symmetric schemes are faster than asymmetric schemes, but they also rely on sophisticated mechanisms to securely create and distribute the required keys. In the context of SGC schemes, for example the XOR cipher [17] is used as a symmetric method [18].

In contrast to symmetric methods, asymmetric methods use different keys for encryption and decryption. The key for encryption is called public key and does not have to be kept secret. In contrast, the key for decryption must be kept secret, which is also referred to as the private key. In terms of performance, asymmetric methods are slower than symmetric methods. In the context of SGC schemes, one-way functions [19], Diffie–Hellman key exchange [20], and elliptic curve cryptography [21] are used as asymmetric methods.

Since SGC schemes are typically strictly deterministic algorithms, the input parameters must be appropriately randomized, e.g., for key generation. Therefore, corresponding random numbers have to be generated. For this purpose, so-called *pseudorandom generators* (PRGs) [22] or *pseudo-random functions* (PRFs) [23] are used. In contrast to PRGs, PRFs can accept any input data in addition to the internal state. For both types of random number generation, it is important that the output is indistinguishable from random sequences and that the next numbers cannot be calculated by an attacker from random numbers that have already been generated.

2.5 Multi-objective optimization problem

The following definitions of a multi-objective optimization problem are all based on the publication [24]. Multi-objective optimization problems aim to determine the optimum of the following function f:

$$f : X \rightarrow F$$

Here, F represents an m-dimensional objective space (m ≥ 2) and X represents an n-dimensional decision space. Thus, the function f can also be written as $f : (x_1, ..., x_n) \rightarrow (f_1(x_1, ..., x_n), ..., f_m(x_1, ..., x_n))$. We denote each vector $f(x) \in F$ as an objective vector and each vector x $\in$ X as a decision vector. In addition, we assume that for all i $\in$ {1,..., m} the respective $f_i : X \rightarrow R$ is to be minimized and that $F \subseteq R^m$. We call an objective vector f' pareto-optimal if no objective vector $f \in F$ exists such that $f < f'$ holds. For this definition of pareto-optimality, it is assumed that optimization means minimization. This suits the context of this work, as we target the minimization of various costs related to the communication. The set which consists of all pareto-optimal objective vectors forms the so-called Pareto front. However, to determine the Pareto front, it is still necessary to define when one objective vector is smaller than another. For this purpose, the following assumption is made in [24]: An objective vector $f^A = (f_1^A, ..., f_m^A)$ is smaller than an objective vector $f^B = (f_1^B, ..., f_m^B)$, i.e., $f^A < f^B$ if for $\forall i \in \{1, ..., m\}$ it holds that $f_i^A \leq f_i^B$ and additionally $f^A \neq f^B$.

3 Survey result extension and preparation for multi object optimization analysis

This section presents how we extend the results of our survey on secure group communication schemes [10] with the multi-objective analysis. This includes the summary of our initial survey [10], the unification and extension of the obtained performance results, and the adaptation of the definition of the Pareto front to the problem of proposing an SGC scheme.

3.1 Initial data collection and analysis

Before we show the adaptation of our initial survey to a multi-objective optimization problem, we briefly introduce the approach of our initial survey, which is illustrated in Fig. 3. As an inital data source for our former survey, we used existing surveys ([5, 15, 25–28]) about SGC schemes. We applied Forward Snowballing to this dataset to find more SGC schemes. We repeated this procedure until we could not identify any new schemes through it. We structured the 47 collected schemes into three categories: centralized, distributed/contributory, and decentralized/hybrid. The result of this classification can be seen in Table 1. Based on this classification, we determined the features and performance of the collected schemes. With regard to features, the eight security features introduced in Sect. 2 (namely, *backward secrecy*, *forward secrecy*, *instant rekey*, *message integrity*, *message confidentiality*, *member authentication*, *compromise robustness*, and *group independence*) were taken into account. Additionally, the cryptography used and the key update frequency were determined for each scheme. The performance was determined on the basis of the following aspects: Mem-

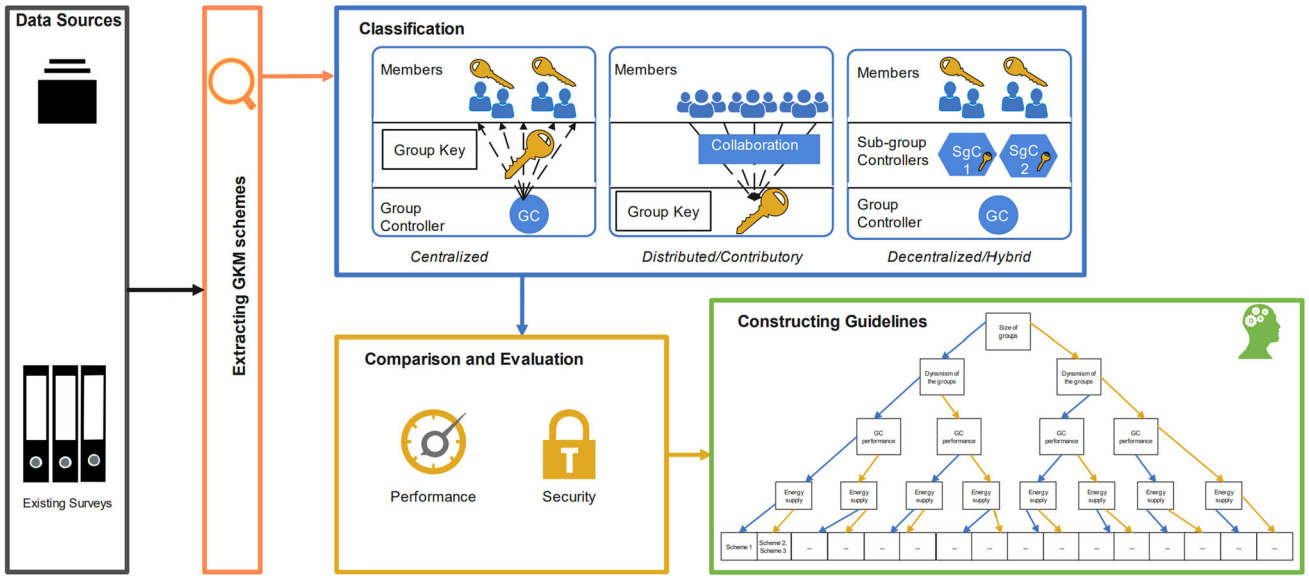

Fig. 3 Visualization of the major steps of the approach of our initial survey [10]

ory requirements for a group member, memory requirements for the CI, total computation costs of a group member, total computation costs of the CI, the total communication costs, and the number of rounds required for group creation. Based on this information, we built two decision trees to assist in the selection of a scheme. One decision tree focuses only on performance aspects and the other one on the security features.

3.2 Secure group communication schemes survey adjustment

In order to prepare and extend the data collected in our survey for multi-objective analysis, we set three goals regarding performance: (1) the original performance metric computational costs should be broken down more precisely in terms of group operation, (2) the original performance metric communication costs should be broken down more precisely in terms of the number of messages sent and the size of the messages per group operation, and (3) the performance values of the individual schemes should be directly comparable. For the implementation of the first two points, we have repeated our literature analysis from [10] and extended the performance metrics as shown in Table 2 and collected the required performance values. However, we were only able to collect this information for 34 out of the original 47 schemes from our Survey [10]. From the original 47 schemes, we had to exclude those (1) that were not described in sufficient detail for us to determine the new required performance values ourselves and (2) for which no other sources existed that contained the information on the new required performance values. An overview of the remaining 34 schemes can be seen in Table 3. (Note: We have taken the classification of the schemes from our initial Survey [10], in which the classification was explained in more detail.)

To derive enough information about the performance of the different SGC schemes, we first analyzed the parameters from Table 4, which were originally used to specify the performance of the different SGC schemes. In this table, it is noticeable that these parameters (1) are either constants or (2) can be estimated upwards with the parameter n, which stands for the number of group members. Since in the literature the performance of the SGC schemes is additionally indicated only by means of the Landau notation, in which constants simply disappear, we were able to express the performance of the SGC schemes in the Landau notation only as a function of the group size. This makes it easier to compare the performance of the different SGC schemes, since they now only depend on the parameter n.

Regarding features, we have made only one change compared to the original survey [10], which is to include the feature *life cycle*. This feature describes whether a scheme allows any number of further group operations after group creation. Depending on the type, this feature can be limited or unlimited. Thereby, we were able to determine this feature for all 34 schemes considered so far.

3.3 Pareto front definition adjustment

Now that we have shown the adaptations to our data basis, we next present how we adapted the problem of recommending an SGC scheme to a multi-objective optimization problem. To do this, we first define the function $f : X \rightarrow F$ to be optimized. The decision vector X in this case consists of an

Table 1 Classification of all 47 schemes collected in the survey [10] into the categories centralized, decentralized/contributory, and decentralized/hybrid

Centralized	Distributed/contributory	Decentralized/hybrid
SKDC [29]	D-LKH [8]	SMKD [30]
GKMP [31, 32]	DH-LKH [33, 34]	IGKMP [35]
LKH [36]	D-OFT [37]	Iolus [38]
LKH+ [9]	SHM [39]	MARKS[40]
OFT [41]	PCGR [42]	Kronos [43]
OFCT [44]	CRGR [45]	Slimcast [46]
S2RP [47]	BKM [48]	RiSeG [49]
LARK [50]	EGKMST [51]	LNT [52]
TKH [53]	SGRS [54]	DEP [55]
CFKM/FT [9]	BD [56]	CS [57]
ELK[58]	G-DH [59]	Hydra [60]
SGCH [61]	Octopus [62]	Alohali [6]
HSHKD [63]	CKA [64]	
LEAP [65]	DFT [9]	
EBS [66, 67]		
SeGCom [68]		
XKFS [69]		
SBSA [70]		
KMGC [71]		
CL-EKM [72]		

Table 2 More precise splitting of the original metrics computation costs and communication costs

Computation Costs	Communication Costs
...	...

⇓

Computation Costs			Message Size			Message Size		
Creation	Join	Leave	Creation	Join	Leave	Creation	Join	Leave
...	...	...	...	...	...	...	...	...

The computation costs are no longer given in general but per group operation. The metric communication costs are now split with regard to the number of messages and the size of the messages, whereby these two new metrics are also analyzed per group operation

Table 3 Overview of the 34 schemes for which the extended performance analysis could be performed

Centralized	Distributed/contributory	Decentralized/hybrid
SKDC [29]	D-LKH [8]	SMKD [30]
GKMP [31, 32]	DH-LKH [33, 34]	IGKMP [35]
LKH [36]	D-OFT [37]	Iolus [38]
LKH+ [9]	SHM [39]	MARKS[40]
OFT [41]	PCGR [42]	Kronos [43]
OFCT [44]	CRGR [45]	Slimcast [46]
S2RP [47]	CKA [64]	RiSeG [49]
SBSA [70]	G-DH [59]	LNT [52]
XKFS [69]	Octopus [62]	DEP [55]
CFKM [9]	BD [56]	CS [57]
EBS [66, 67]		Hydra [60]
SGCH [61]		Alohali [6]

Table 4 Originally used parameters from [10] to specify the performance of SGC schemes

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 life cycle
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

one-element vector whose range of values consists of the schemes in Table 3. For the definition of the objective vector F, we distinguish, following the original survey [10], the two cases that (1) only performance is considered and (2) only features are considered. Thus, for the first case, we define F_{perf} as shown in Eq. 1. In this function, the subfunctions $f_1(x)$ to $f_{23}(x)$ give the respective costs according to their names in Landau notation. For example, the subfunction $f_1(x)$ gives the member's memory cost in Landau notation for scheme x, or the subfunction $f_2(x)$ gives the CI's memory cost in Landau notation for scheme x. The range of values of the subfunctions $f_1(x)$,..., $f_{23}(x)$ thereby consists of the set $\{n/a, O(log(n)), O(n), O(n * log(n), O(n^2)\}$ which is based on the performance values determined in the literature research. Noticeable here is the symbol n/a which we have included to indicate if a scheme does not support the particular group operation.

$$F_{perf}(x) = \begin{bmatrix} f_1(x) = \text{Storage}_{Member}(x) \\ f_2(x) = \text{Storage}_{CI}(x) \\ f_3(x) = \text{ComputationCosts}_{Creation,Member}(x) \\ f_4(x) = \text{ComputationCosts}_{Join,Member}(x) \\ f_5(x) = \text{ComputationCosts}_{Leave,Member}(x) \\ f_6(x) = \text{ComputationCosts}_{Creation,CI(x)} \\ f_7(x) = \text{ComputationCosts}_{Join,CI}(x) \\ f_8(x) = \text{ComputationCosts}_{Leave,CI}(x) \\ f_9(x) = \text{MessageSize}_{Creation,Member}(x) \\ f_{10}(x) = \text{MessageSize}_{Join,Member}(x) \\ f_{11}(x) = \text{MessageSize}_{Leave,Member}(x) \\ f_{12}(x) = \text{MessageSize}_{Creation,CI}(x) \\ f_{13}(x) = \text{MessageSize}_{Join,CI}(x) \\ f_{14}(x) = \text{MessageSize}_{Leave,CI}(x) \\ f_{15}(x) = \text{MessageAmount}_{Creation,Member}(x) \\ f_{16}(x) = \text{MessageAmount}_{Join,Member}(x) \\ f_{17}(x) = \text{MessageAmount}_{Leave,Member}(x) \\ f_{18}(x) = \text{MessageAmount}_{Creation,CI}(x) \\ f_{19}(x) = \text{MessageAmount}_{Join,CI}(x) \\ f_{20}(x) = \text{MessageAmount}_{Leave,CI}(x) \\ f_{21}(x) = \text{Rounds}_{Creation}(x) \\ f_{22}(x) = \text{Rounds}_{Join}(x) \\ f_{23}(x) = \text{Rounds}_{Leave}(x) \end{bmatrix} \quad (1)$$

$$F_{feat}(x) = \begin{bmatrix} f_{24}(x) = \text{BackwardSecrecy}(x) \\ f_{25}(x) = \text{ForwardSecrecy}(x) \\ f_{26}(x) = \text{InstantRekey}(x) \\ f_{27}(x) = \text{MessageIntegrity}(x) \\ f_{28}(x) = \text{MessageConfidentiality}(x) \\ f_{29}(x) = \text{MemberAuthentication}(x) \\ f_{30}(x) = \text{CompromiseRobustness}(x) \\ f_{31}(x) = \text{GroupIndependence}(x) \\ f_{32}(x) = \text{UsedCryptography}(x) \\ f_{33}(x) = \text{LifeCycle}(x) \end{bmatrix} \quad (2)$$

After, we have treated the case that only considers the performance, we now define the target vector F_{feat} for the case that only the features are considered in Eq. 2. The subfunctions $f_{24}(x)$ to $f_{31}(x)$ of this target vector indicate whether the scheme x provides the respective feature. For example, the subfunction f_{24} indicates whether the scheme x has the feature *backward secrecy*, or the subfunction f_{25} reflects whether the scheme x provides the feature *forward secrecy*. The range of values of the subfunctions $f_{24}(x)$ to $f_{31}(x)$ consists of the set $\{0, 1\}$, where 0 is returned if the scheme

provides the respective feature and 1 otherwise. This behavior may be counterintuitive at first glance, since one would actually expect 1 to be returned if a feature is provided. The reason why we chose the other assignment is that the Pareto front in Sect. 2 was defined as a minimization problem and thus 0 is evaluated better than 1. The subfunction f_{32} returns a set consisting of the cryptographic techniques on which the scheme x is based. Here, building on the results of our survey $f_{29} \subseteq$ {PK, DH, Symmetric, RSA, Hash, XOR, PRG, Combinatorial}. The last subfunction f_{33} returns either the value $limited$ or $unlimited$ for a scheme x. The value $unlimited$ is returned if the scheme x allows any number of group operations after group creation. If the number of group operations is limited the value $limited$ is returned.

After defining the objective vectors, we still need to determine when an objective vector is smaller than another objective vector so that the objective function can be minimized accordingly and the Pareto front can be determined. To do this, we use the definition in Sect. 2.5, which states that an objective vector is smaller than another objective vector if each component of the objective vector is smaller than the corresponding component of the other objective vector. Thus, we must now define for the subfunctions $f_1(x)$ to $f_{33}(x)$ corresponding order relations on their ranges of values. To do this, we first consider the subfunctions f_1 to f_{23} which return an element of the set $\{n/a, O(log(n)), O(n), O(n * log(n), O(n^2)\}$. To define an order relation for these subfunctions, we define the following order relation for the possible Landau values: $O(log(n)) < O(n) < O(n * log(n) < O(n^2)$. In this order relation we have deliberately not included the symbol n/a. The reason for this is that, in our opinion, there are also use cases where it is important that members cannot be removed from and/or added to a group. For example, in a smart health scenario where patients with critical health conditions are to be monitored and it should therefore not be technically possible to exclude, e.g., the attending physician from the group. To accommodate use cases where it is important that not all group operations are supported, we have not included the n/a symbol. This has the consequence that in the Pareto front every occurring combination of supported group operations is represented and therefore will be considered when selecting a scheme.

Next, we consider the functions f_{24} to f_{31}, which return 0 or 1. We use the corresponding order relation of the integers as the order relation for these subfunctions. This leaves the subfunctions f_{32} and f_{33}. Since the range of values of the subfunction f_{32} consists of sets, we define the following order relation for sets: $f_{32}(x)$ is smaller than $f_{32}(x')$ if (1) $f_{32}(x) \subset f_{32}(x')$ and (2) there exists no $x'' \neq x$ such that $f_{32}(x'') \subset f_{32}(x)$. Lastly, the subfunction f_{33} returns either the value $limited$ or $unlimited$, for which we define the following order relation: unlimited < limited.

Now that we have defined the objective vectors F_{perf} and F_{feat} and their corresponding order relations, we would like to illustrate the meaning of these two vectors once again The vector F_{perf} gives us an overview of the performance of the respective scheme, since its first component indicates the memory requirement of the members in Landau notation, the second component indicates the memory requirement of the CI in Landau notation and so on. The meaning of the other components can be found in Eq. 1. The vector F_{feat}, on the other hand, gives an overview of the features of the respective scheme. For example, a scheme provides the feature backward secrecy if the first component of F_{feat} is 0. However, if this component is 1, the scheme does not have the feature backward secrecy. The second component indicates that the scheme provides the forward secrecy feature if it contains the value 0. However, if the value is 1, the scheme does not have the forward secrecy feature. The meaning of the remaining components of the vector F_{feat} can be found in Eq. 2.

4 Selection guidelines for centralized SGC scheme

For the creation of guidelines, we first consider the case where the use case dictates that a centralized scheme should be used. In practice, this could be the case in a smart city scenario, for example, where the city hall always wants to have direct control over its IoT devices. The city hall can achieve this by acting as a CI and communicating directly with the IoT devices and not indirectly through, for example, subgroup controllers.

For creating guidelines for centralized schemes, we first review the features and performance of the collected centralized schemes from Table 3. Building on these reviews, we determine the corresponding Pareto fronts by weighting each objective equally. By forming these fronts, we are able to narrow down the selection to those schemes for which no clearly better scheme exists. In order to finally make a selection from the narrowed down choice, we then designed corresponding decision trees, which take into account which objectives are most important for the respective use case. In total, we present three decision trees, once for the case when only performance is considered, once for the case when only features are considered, and once for the case when both performance and features are considered.

4.1 Performance-driven analysis of centralized SGC Schemes

We begin the creation of our guidelines with the case where the user is only interested in the performance of the centralized schemes and not in their features. This could be the case, for example, if an existing system, that already provides

the required features, is to be extended to include the aspect of group encryption. For example, it may be that the existing system already provides a secure connection between the individual group members and the central instance, so that the group encryption scheme does not need to provide these properties. (Note that we are referring here to the communication to create the group key and not to the subsequent communication between the group members).

For the creation of the purely performance-oriented guideline, we first consider Tables 5 and 6, which provide an overview of the performance of the centralized schemes. The performance is given in Landau notation, where the parameter n stands for the number of group members In doing so, we have taken the values of the Tables 5 and 6 from our initial survey [10]. Based on these tables, it can be seen that the SKDC, XFKS and SBSA schemes are identical in terms of their performance. The same statement can also be made for the schemes (i) LKH+ and CFKM or (ii) OFT and OFCT. The schemes SKDC, XFKS and SBSA have constant message sizes and the computational cost as well as memory requirements of the members is constant. For this, linear costs are imposed on the CI with respect to memory requirements and computational costs in each case, and a linear amount of messages are required in each case.

Also linear many messages are used by the scheme LKH for the group creation and also the storage requirements to the CI are linear. On the other hand, the computational costs of the CI for adding or removing members are logarithmic. In addition, only logarithmically many messages need to be sent per group operation. These advantages for the CI go, however, at expense of the group members, which have logarithmic costs. Almost identical in terms of performance to LKH is the scheme EBS. The only difference is that removing and adding group members only requires constant-sized messages.

Also almost identical to LKH are the schemes CFKM and LKH+. In terms of performance, these two schemes differ only in that adding a group member requires only a constant number of messages. Even better in terms of performance are the schemes S2RP, OFT and OFCT compared to CFKM and LKH+. Compared to LKH, S2RP not only requires a constant number of messages for adding group members, but also only constant computations. Schemes OFT and OFCT outperform LKH in terms of performance for the number of messages for removing and adding group members, which is only constant in each case.

The scheme SGCSH requires only constant costs for the group members and requires only messages of constant size. For this, however, the costs of the CI are in each case always linear, apart from the computation costs for the removing and adding of group members, which are always only constant. Also only a constant number of messages is required for adding group members, in contrast to the creation of a group

Table 5 Performance of centralized secure group communication schemes (Part 1/2)

		SKDC	GKMP	LKH	LKH+	S2RP	OFT	OFCT	CFKM	SGCSH
Storage		CI: O(n)	CI: O(1)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)
		M: O(1)	M: O(1)	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(1)
Computation costs	Creation	CI: O(n)	CI: O(1)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)
		M: O(1)	M: O(1)	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(1)
	Addition	CI: O(n)	CI: O(1)	CI: O(log(n))	CI: O(log(n))	CI: O(1)	CI: O(log(n))	CI: O(log(n))	CI: O(log(n))	CI: O(1)
		M: O(1)	M: O(1)	M: O(log(n))	M: O(log(n))	M: O(1)	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(1)
	Revocation	CI: O(n)	n/a	CI: O(log(n))	CI: O(log(n))	CI: O(log(n))	CI: O(log(n))	CI: O(log(n))	CI: O(log(n))	CI: O(1)
		M: O(1)	n/a	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(log(n))	M: O(1)
Message size	Creation	O(1)	O(1)	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(1)
	Addition	O(1)	O(1)	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(1)
	Revocation	O(1)	O(1)	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(1)
Message amount	Creation	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
	Addition	O(n)	O(1)	O(log(n))	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)
	Revocation	O(n)	O(1)	O(log(n))	O(log(n))	O(log(n))	O(1)	O(1)	O(log(n))	O(n)

Table 6 Performance of centralized secure group communication schemes (Part 2/2)

Storage		EBS CI: O(n) M: O(log(n))	XKFS CI: O(n) M: O(1)	SBSA CI: O(n) M: O(1)
Computation costs	Creation	CI: O(n)	CI: O(n)	CI: O(n)
		M: O(log(n))	M: O(1)	M: O(1)
	Addition	CI: O(log(n))	CI: O(n)	CI: O(n)
		M: O(log(n))	M: O(1)	M: O(1)
	Revocation	CI: O(log(n))	CI: O(n)	CI: O(n)
		M: O(log(n))	M: O(1)	M: O(1)
Message size	Creation	O(log(n))	O(1)	O(1)
	Addition	O(1)	O(1)	O(1)
	Revocation	O(1)	O(1)	O(1)
Message amount	Creation	O(n)	O(n)	O(n)
	Addition	O(log(n))	O(n)	O(n)
	Revocation	O(log(n))	O(n)	O(n)

and the removal of members which requires linearly many messages.

GKMP is the only scheme that has constant costs everywhere. However, this is bought by the fact that members can neither be added nor removed after group creation.

Building on the previous performance review, we now determine the Pareto front with respect to performance for the centralized schemes by first ranking the considered centralized schemes with respect to F_{perf}. To obtain these rankings, we first determine the corresponding vector F_{perf} for each of the centralized schemes from Table 1 and compared them in pairs. We were able to determine the following rankings:

- $F_{perf}(XFKS) = F_{perf}(SBSA) = F_{perf}(SKDC) > F_{perf}(SGCSH)$
- $F_{perf}(LKH) > F_{perf}(EBS)$
- $F_{perf}(LKH) > F_{perf}(LKH+) = F_{perf}(CFKM)$
- $F_{perf}(LKH+) = F_{perf}(CFKM) > F_{perf}(OFCT) = F_{perf}(OFT)$
- $F_{perf}(LKH+) = F_{perf}(CFKM) > F_{perf}(S2RP)$

Based on the above rankings, we can directly identify the schemes that form the Pareto front with respect to F_{perf}. Specifically, these are the schemes SGCSH, EBS, OFT, OFCT, S2RP, and GKMP. Thus, the selection of centralized schemes for our purely performance-based decision tree can be directly restricted to these schemes. Of the remaining schemes, the scheme GKMP stands out as the only scheme that does not support the removal of members, but otherwise has only constant costs. Therefore, we can conclude that the scheme GKMP should be directly recommended by our decision tree if the use case does not require the removal of members. Thus, the first question our decision tree asks in Fig. 4 is whether the use case requires the removal of members.

If it does not, then the scheme GKMP can be recommended directly. Otherwise, the use case must be further narrowed down using additional questions, since none of the remaining schemes is clearly best. Of the remaining schemes, SGCSH stands out as the best in terms of performance everywhere except for the number of messages needed to remove a group member. Therefore, our decision tree next considers what the message amount requirements are for removing a group member .[1]

If only a constant or linear number of messages may be sent, the schemes OFT and OFCT, which are identical with respect to F_{perf}, are clearly the best choice, respectively, the scheme SGCSH. If the use case allows the number of messages for removal to be at most logarithmic, then a choice must be made from the OFT, OFCT, EBS, and S2RP schemes. The selection can be restricted directly to the two schemes EBS and S2RP, since these perform better than the schemes OFT and OFCT with respect to F_{perf} in the so restricted use case. To make a choice between these two schemes we need to further restrict the use case. To do this, we consider what requirements the use case places on message sizes for adding and removing members. If these are allowed to be at most constant, then EBS is clearly the best choice and S2RP otherwise.

[1] Note: To make the decision trees clearer, we use the abbreviation "≤ 1" for "At most constant," the abbreviation "$\leq log(n)$" for "At most logarithmic" and the abbreviation "$\leq n$" for "At most linear." Logarithmic or linear always means that the costs increase logarithmically or linearly with the group size.

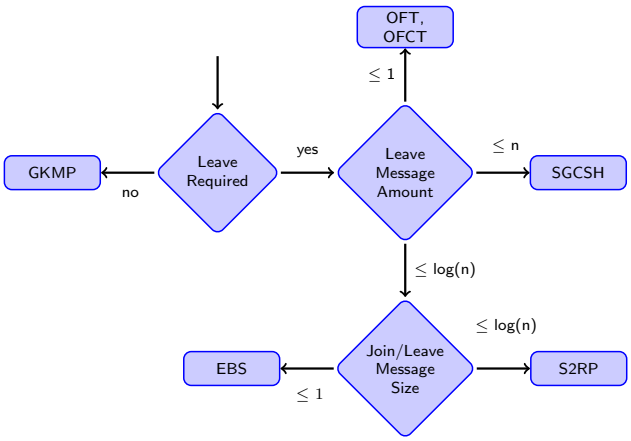

Fig. 4 Decision tree for the selection of centralized SGC schemes if only performance is considered

4.2 Feature-driven analysis of centralized SGC schemes

After creating a guideline for the selection of centralized schemes, which only takes into account the performance of the schemes, we now create in this subsection a guideline for centralized schemes that only takes into account the features. This guideline is intended for use cases where security is a top priority and no additional system can compensate for missing features.

Tables 7 and 8 list the features of the centralized SGC schemes. Thereby, Table 7 specifically focuses at the features key update frequency, types of cryptography, and life cycle. A closer look at this table reveals that (1) only the SGCSH scheme updates its keys periodically, while all other schemes do so after each change in group composition, (2) except for the two schemes S2RP and SGCSH, all other schemes have unlimited *life cycle*, and (3) only the schemes of the sets {LKH, LKH+, S2RP} and {OFT, XFKS} are each based on the same cryptographic functions.

The remaining features are listed for the centralized SGC schemes in Table 8, which we refer to below. Based on Table 8, it can be seen that the SKDC, LKH, LKH+, OFT, OFCT, EBS, and XKFS schemes each provide the features *forward secrecy*, *backward secrecy*, *instant rekey*, *message confidentiality*, and *group independence*. In comparison, the CFKM and GKMP schemes each have one and two fewer features, respectively, while the SBSA scheme provides one feature more, respectively. For example, the CFKM scheme lacks the *message confidentiality* feature and the GKMP scheme lacks the *backward secrecy* and *message confidentiality* features. The scheme SBDA offers the additional feature of *compromise robustness*. In Table 8, the scheme S2RP offers the most features, as it has *forward secrecy*, *backward secrecy*, *instant rekey*, *message integrity*, *message confidentiality*, and *group independence*. The least features are provided by the SGCSH scheme, which has only the *message integrity*, *message confidentiality* and *group independence* features.

In order to construct a feature-oriented decision tree, we first determine the corresponding Pareto front for the target vector F_{Feat}. To do this, we first collected all possible rankings by determining the corresponding vectors F_{Feat} for each scheme and then comparing them in pairs. This allowed us to determine the following rankings:

- $F_{Feat}(\text{SBSA}) < F_{Feat}(\text{EBS}) = F_{Feat}(\text{SKDC})$ $= F_{Feat}(\text{OFT}) = F_{Feat}(\text{XFKS}) = F_{Feat}(\text{OFCT})$
- $F_{Feat}(\text{EBS}) = F_{Feat}(\text{SKDC}) = F_{Feat}(\text{OFT})$ $= F_{Feat}(\text{XFKS}) = F_{Feat}(\text{OFCT}) < F_{Feat}(\text{GKMP})$
- $F_{Feat}(\text{LKH+}) = F_{Feat}(\text{LKH}) < F_{Feat}(\text{CFKM})$
- $F_{Feat}(\text{LKH}) = F_{Feat}(\text{LKH+}) < F_{Feat}(\text{EBS})$ $= F_{Feat}(\text{SKDC}) = F_{Feat}(\text{OFT})$ $= F_{Feat}(\text{XFKS}) = F_{Feat}(\text{OFCT})$
- $F_{Feat}(\text{S2RP}) < F_{Feat}(\text{SGCSH})$

Building on these rankings, we can directly determine the Pareto front for the target vector F_{Perf}, which consists of the following vectors: $F_{Feat}(\text{SBSA})$, $F_{Feat}(\text{LKH})$, $F_{Feat}(\text{LKH+})$, and $F_{Feat}(\text{S2RP})$. To finally propose a scheme from this already limited selection, we designed a corresponding decision tree in Fig. 5. In creating this decision tree, we took care to keep it as small as possible without limiting its expressiveness. The decision tree in Fig. 5 first restricts the use case as to whether the cryptography used must be minimal. If so, the decision tree next considers the life cycle requirements of the use case. If they must be unlimited, then both LKH and LKH+, which both offer unlimited life cycles and are identical in terms of F_{Feat}, are the best choices. If, on the other hand, the *life cycle* is to be limited, then the scheme SBSA is clearly the best choice. If the cryptography used does not need to be minimal after all, the decision tree also next asks whether the *life cycle* should be unbounded or not. If yes, then the scheme S2RP can be proposed directly. If no, then the use case must be narrowed down further for the final decision. This is done by asking whether the use case requires *message integrity* or not. If this is the case, S2RP is the best choice and otherwise SBSA.

4.3 Performance and feature-driven analysis of centralized SGC schemes

After creating a pure performance and a pure feature-oriented decision tree for the selection of a centralized SGC scheme, we now want to design a decision tree for centralized SGC schemes which is both performance- and feature-oriented. For this purpose, analogous to the two decision trees already created, we could first determine the corresponding Pareto front in terms of performance and features of the schemes.

Table 7 Key update frequency, used cryptography and life cycles of centralized Secure Group Communication Schemes

Scheme	Key update frequency	Types of cryptography	Life cycles
SKDC	Membership Change	PK, DH	Unlimited
GKMP	Membership Change	Symmetric, DH RSA	Unlimited
LKH	Membership Change	Symmetric	Unlimited
LKH+	Membership Change	Symmetric	Unlimited
S2RP	Membership Change	Symmetric	limited
OFT	Membership Change	Symmetric, Hash XOR	Unlimited
OFCT	Membership Change	Hash, PRG Symmetric	Unlimited
CFKM	Membership Change	DH	Unlimited
SGCSH	Time Period	XOR, Hash	limited
EBS	Membership Change	Combinatory Symmetric	Unlimited
XFKS	Membership Change	Hash, XOR Symmetric	Unlimited
SBSA	Membership Change	PRG Symmetric	Unlimited

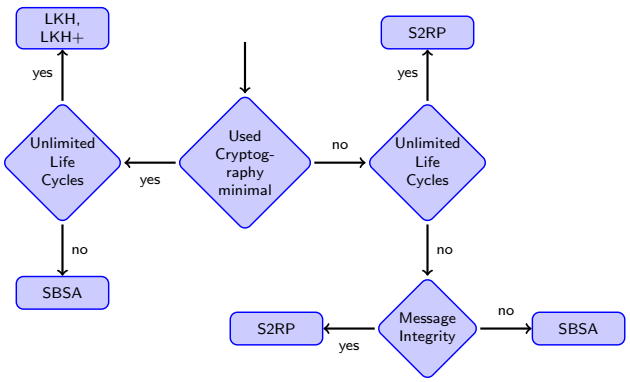

Fig. 5 Decision tree for the selection of centralized SGC schemes if only features are considered

However, this approach would imply that features and performances are considered equally important when selecting a scheme. In our opinion, this is not the case, as we believe that in order to select an SGC scheme, one first determines what level of security (e.g., if the scheme provides message integrity) or features (e.g., if the scheme provides unlimited life cycles) are needed and then makes the selection from the eligible schemes based on performance.

To realize this approach, we would first like to discuss an observation that we noticed while building the purely feature-oriented decision tree. We observed that when considering only its partial vector $F_{Feat}^{f_{24}-f_{31}}$ instead of the goal vector F_{perf}, the following rankings for centralized SGC schemes emerge[2]. Here we abbreviate the vector $F_{Feat}^{f_{24}-f_{31}}$ with F_p for the sake of simplicity.

- $F_p(\text{SBSA}) < F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS})$
- $F_p(\text{S2RP}) < F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS})$
- $F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS}) < F_p(\text{CFKM})$
- $F_p(\text{CFKM}) < F_p(\text{GKMP})$
- $F_p(\text{S2RP}) < F_p(\text{SGCSH})$

[2] The partial vector $F_{Feat}^{f_{24}-f_{31}}$ is of interest to us, since it contains the classical security features such as message integrity or message confidentiality. The only two features missing in this partial vector describe whether the trust in the cryptography is minimal and how many times a scheme allows the addition or removal of members. Therefore, the partial vector $F_{Feat}^{f_{24}-f_{31}}$ of all features represents the classical security features.

Table 8 Overview of the security features of SGC schemes

	Scheme	Backward/Forward Secrecy	Instant Rekey	Message Integrity	Message Confidentiality	Member Authentication	Compromise Robustness	Group Independence
Centralized	SKDC	y/y	y	n	y	n	n	y
	GKMP	y/n	y	n	n	n	n	y
	LKH	y/y	y	n	y	n	n	y
	LKH+	y/y	y	n	y	n	n	y
	OFT	y/y	y	n	y	n	n	y
	OFCT	y/y	y	n	y	n	n	y
	S2RP	y/y	y	y	y	n	n	y
	CFKM	y/y	y	n	n	n	n	y
	SGCSH	n/n	n	y	y	n	n	y
	EBS	y/y	y	n	y	n	n	y
	XKFS	y/y	y	n	y	n	n	y
	SBSA	y/y	y	n	y	n	y	y
Distributed	D-LKH	y/y	y	n	y	n	n	y
	DH-LKH	y/y	y	n	y	n	n	y
	D-OFT	y/y	y	n	y	n	n	y
	SHM	y/y	y	n	y	n	n	y
	PCGR	n/n	n	y	y	n	n	y
	CRGR	y/y	y	y	y	n	y	y
	BD	y/y	y	n	y	n	n	y
	G-DH	y/y	y	n	y	n	n	y
	Octopus	y/y	y	n	y	n	n	y
	CKA	y/y	y	n	y	n	n	y
Decentralized	SMKD	y/n	n	n	n	y	n	y
	IGKMP	y/y	y	n	y	y	n	y
	Iolus	y/y	y	n	y	n	n	y
	MARKS	n/n	n	n	n	n	n	y
	Kronos	y/n	n	n	y	y	n	y
	SLIMCAST	y/y	y	y	y	y	n	y
	RiSeG	y/y	y	y	y	y	n	y
	LNT	y/y	y	y	y	y	n	y
	DEP	y/n	n	n	y	y	n	y
	CS	y/y	y	n	y	n	n	y
	Hydra	y/y	y	n	y	y	n	y
	Alohali	y/y	y	n	y	y	y	y

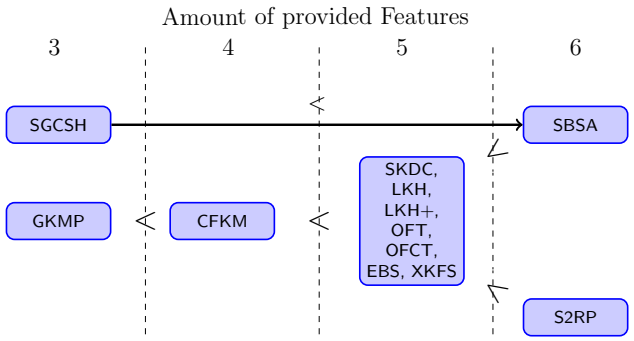

Fig. 6 Arrangement of the centralized SGC schemes based on the number and subset relationships of the features provided. The "$<$" operator means that the feature set of A is a true subset of the feature set of B if "Schema A $<$ Schema B." For schemas which are in the same box, their feature sets are identical

To make the above rankings easier to understand, we have visualized them in Figure 6. In this figure, the operator "$<$" means that if a "Scheme A $<$ Scheme B," then the feature set of Scheme A is a real subset of the feature set of Scheme B. If schemes are in the same box, this means that their feature sets are identical.

Based on the observation that we can arrange the scheme in an order that arranges the features of one scheme as subset of another, we believe it makes sense to define feature levels for centralized SGC schemes, which we refer to as security levels in the following. It can be concluded from the above diagram that the security level of the GKMP scheme should be lower than that of the CFKM scheme and that the security level of CFKM should in turn be lower than the security level of the SKDC, LKH, LKH+, OFT, OFCT, EBS and XKFS schemes. It is also directly evident that the safety levels of the SBSA and S2RP schemes are greater than the level of the SKDC, LKH, LKH+, OFT, OFCT, EBS, and XKFS schemes, but it is not directly evident how the safety levels of the SBSA and S2RP schemes should relate to each other. Due to the fact that both schemes provide the same number of features, we decided to give them the same security level. Thus, only the security level of the scheme SGCSH has to be ranked. Unfortunately, the only conclusion that can be drawn from the figure is that the security level of scheme SGCSH must be lower than that of scheme SBSA. In order to classify the security level of the SGCSH scheme in the ranking, we proceed in the same way as for the SBSA and S2RP schemes and use the number of features provided. Thus, the scheme SGCSH is on the same level as the scheme GKMP and we can formally define the security levels based on the feature sets of the schemes as follows:

- $S_{1.1}$ = {*backward secrecy*, *instant rekey*, *group independence*}
- $S_{1.2}$ = {*message integrity*, *message confidentiality*, *group independence*}.
- $S_2 = S_{1.1} \cup$ {*forward secrecy*}
- $S_3 = S_2 \cup$ {*message confidentiality*}
- $S_{4.1} = S_3 \cup$ {*compromise robustness*},
- $S_{4.2} = S_3 \cup$ {*message integrity* }

Using these levels, we now build our decision tree (see Fig. 7) for centralized SGC schemes that take performance and features into account. To do this, we proceed as follows: First, the decision tree determines the required security level. Second, the requirements regarding life cycles are considered. Third, the used cryptography is considered. Fourth, the final choice is made from the limited selection of schemes based on performance. Thus, the first question in the decision tree (see Fig. 7) is related to whether the security level should be greater than 3. If this is the case, only the SBSA and S2RP schemes are available for selection. Since S2RP uses minimal cryptography while SBSA does not, and SBSA has unlimited life cycles unlike S2RP, our decision tree makes the choice between these two schemes based on which the use case prioritizes higher. If unlimited life cycles are prioritized higher than minimal cryptography, SBSA is recommended. However, if minimal cryptography is prioritized higher than unlimited life cycles, S2RP is recommended.

However, if the safety level only needs to be greater than 2, our decision tree again first considers whether the *life cycle* needs to be limited or not. If this is the case, the selection is restricted to the S2RP scheme. If this is not the case, we next consider whether or not the use case requires minimal cryptography. If so, the selection is restricted to the schemes LKH and LKH+. Since the performance of LKH+ is clearly better than that of LKH, the LKH+ scheme can be recommended directly for this case. If, on the other hand, the cryptography used must be non-minimal, the schemes SBSA, SKDC, LKH, LKH+, OFT, OFCT, EBS and XFBS are available for selection. If we take into account the rankings already determined with respect to F_{perf}, we can directly narrow down this choice to the sets {SBSA, XFBS, SKDC}, {OFT, OFCT}, and {EBS}, where we have grouped schemes with identical performance into one set each. Since these schemes are also the Pareto front for the case where the security level must be greater than 2, the cryptography used must not be minimal, and the life cycles should be unlimited we need to further restrict the use case for the final selection of a scheme. Therefore, our decision tree next considers the use case requirements for the number of messages that must be sent for removing and adding members. If this number has to be constant in each case, the choice narrows down to the two schemes OFT and OFCT. Since they are identical in terms of performance, both schemes are recommended in this case. If the number of messages for removing and adding may also be logarithmic, then the scheme EBS is clearly the best choice. However, if the message count for removal and addition is also allowed to be linear, the decision tree next considers the computational cost requirements of the CI for removing and adding group members. If these computational costs are allowed to be logarithmic at most, then the choice narrows down to Scheme EBS. However, if these costs can also be linear, then the SKDC, SBSA and XFBS schemes are clearly the best choice, which have identical performance.

The case where the safety level only has to be greater than 1 is largely analogous to the case where the safety level has to be greater than 2. The only difference between the two cases is that if the safety level only has to be greater than 1 and the *life cycle* is to be unlimited, the CFKM scheme is now also available for selection. So we can adapt the subtree accordingly from this point on. Thus, we must consider next the cases where the used cryptography must be minimum and/or must not be minimum. If the cryptography used must be minimal, we now have the two schemes LKH+ and CFKM to choose from, both of which we recommend because they are identical in terms of performance. If, on the other hand, the cryptography used must not be minimal, SBSA, XFBS, SKDC, OFT, OFCT, EBS and CFKM are available for selection. Since $F_{perf}(CFKM) > F_{perf}(OFT) = F_{perf}(OFCT)$ holds, this choice can be directly restricted to the SBSA, XFBS, SKDC, OFT, OFCT, and EBS schemes. This corresponds to the selection for the case where the security level must be greater than 2, the *life cycle* should be limited, and the cryptography used must be non-minimal. Therefore, we refer to the corresponding subtree starting at this point.

Finally, the decision tree considers the case where the security level does not play a role in the selection of a scheme. In this case, the decision tree first checks again whether the *life cycle* should be limited or not. If yes, the selection is limited to the two schemes S2RP and SGCSH, which both use minimal cryptography. Therefore, the choice between these two schemes must be based on their performance. Scheme SGCSH is superior to S2RP in terms of performance almost everywhere except for the number of messages required to remove a member. Scheme SGCSH requires a linear number of messages to do this, while Scheme S2RP requires only a logarithmic number of messages. Therefore, we can recommend the scheme SGCSH if the use case allows the number of messages for the removal to be linear and recommend the scheme S2RP otherwise. However, if the *life cycle* are not to be limited, the decision tree next asks whether the cryptography used must be minimal or not. If this is the case, then only the schemes LKH, LKH+ and CFKM are available for selection. This selection can be directly restricted to the two schemes LKH+ and CFKM, since $F_{perf}(LKH) > F_{perf}(LKH+) = F_{perf}(CFKM)$

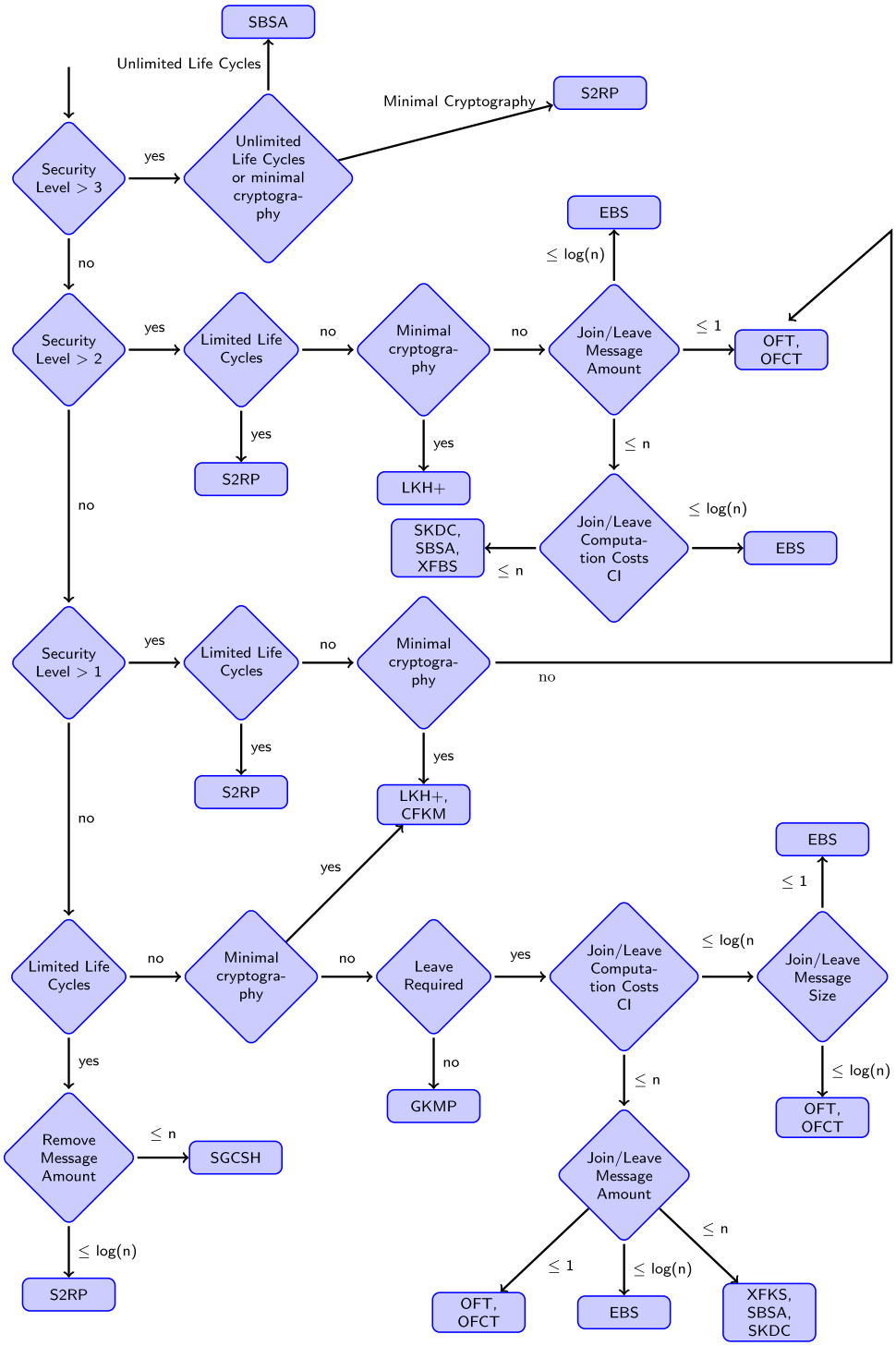

Fig. 7 Decision tree for the selection of centralized SGC schemes if both the performance and features are considered

holds. Since LKH+ and CFKM are identical in terms of performance, both schemes are recommended in this case. If the *life cycle* must be limited, but the cryptography used must also not be minimal, the selection consists of all schemes except S2RP and SGCSH. To make a selection from the remaining schemes, we have determined the Pareto front for this case, which consists of the schemes {GKMP}, {XFKS, SKFC, SBSA}, {EBS}, and {OFCT, OFT}, again grouping schemes with identical performance into a set. Of these schemes, the scheme GKMP stands out as it does not support removal of members, but otherwise has only constant cost. Accordingly, our decision tree next considers whether or not the use case requires member removal. If no, it directly recommends the scheme GKMP. If no, then a selection must be

made from the remaining schemes {XFKS, SKFC, SBSA}, {EBS}, and {OFCT, OFT}. To do this, the decision tree next further constrains the use case in terms of the computational costs of CI for removal and addition. If these costs may only be logarithmic, only the schemes {EBS} and {OFT, OFCT} are left to choose from. The choice between these two sets is based on the question whether the message sizes for adding and removing may be logarithmic or must be constant. If they must be constant, the scheme EBS is clearly the best choice and otherwise the schemes OFT and OFCT are recommended, which are identical with respect to F_{perf}. In the case that the computational cost of the CI for adding and removing may also be linear, the sets {EBS}, {OFT, OFCT} and {SKDC, SBSA, XFKS} are available for selection. The choice between these sets is made on the basis of whether the use case for removing and adding allows at most constant many, logarithmic many, or linear many number. In the first case, the schemes OFT and OFCT are clearly the best choices. In the second case, the scheme EBS is recommended. In the last case, the schemes KDC, SBSA and XFKS are clearly the best choice, which are identical with respect to F_{perf}.

5 Selection guidelines for distributed/contributory SGC Schemes

After creating the guidelines for centralized schemes, we next consider the case where the use case dictates that no CI should be used and thus only distributed/contributory schemes can be considered. This may be the case when there is no trusted party that can act as a CI or when reliable communication with a CI is not possible. A concrete use case for this scenario could be self-driving cars that want to form a platoon but are traveling in a rural area with poor internet coverage.

When creating the decision trees for distributed/contributory SGC schemes, we proceed analogously to the centralized schemes. Again we first review the performance and features of the distributed/contributory SGC systems and determine the corresponding Pareto fronts to constrain the scheme choices. To propose a scheme from the constrained selection, we again create corresponding decision trees that consider the weight of objectives for the use case. When creating the decision trees, we again consider the three cases that only performance is considered, that only features are considered, and that both performance and features are considered

5.1 Performance-driven analysis of distributed/contributory SGC Schemes

Tables 9, 10 and 11 provide an overview on the performance of the distributed/contributory schemes using the Landau notation, where n again stands for the number of group members. These tables are based on the corresponding tables from our initial survey [10]. In contrast to the centralized schemes, for the distributed/contributory schemes we also consider the number of rounds needed for group creation or adding/removing members. Here, we understand the number of rounds as the number of times a group member must receive new information in order to continue the calculation of the corresponding group key. We did not consider this metric for centralized schemes, which was due to the fact that centralized schemes only require a constant number of rounds each for all group operations. Analogous to the procedure for the centralized schemes for constructing a purely performance-based decision tree, we also first determine all possible rankings with respect to F_{perf} for the distributed/contributory schemes in the following:

- $F_{Perf}(\text{CKA}) < F_{Perf}(\text{BD})$
- $F_{Perf}(\text{BD}) < F_{Perf}(\text{Octopus})$
- $F_{Perf}(\text{BD}) < F_{Perf}(\text{PCGR})$
- $F_{Perf}(\text{SHM}) < F_{Perf}(\text{CRGR})$

Building on the above rankings, we can make the statement that no two schemes are identical with respect to F_{perf} and that the Pareto front consists of all schemes except BD, Octopus, PCGR, and CRGR, since there is a clearly better choice for each of these schemes. In addition, by including the three Tables 9, 10 and 11 we can make the following further statements: (1) only the schemes CKA, BD, Octopus and PCGR the formation allow exclusively the creation of a group, but not the removal or addition of members and (2) of these four schemes CKA performs best in terms of performance. Thus, our decision tree for purely performance-oriented selection of SGC schemes in Fig. 8 first considers whether the use case requires only group creation, or also group member removal and addition. If only the group creation is needed, the scheme CKA can be recommended directly. However, if the use case also requires the removal and addition of members, then a selection must be made from the remaining schemes of the Pareto front. For this selection, our decision tree must further constrain the use case, again trying to keep the decision tree as small as possible.

Next, our decision tree further restricts the use case in terms of the requirements for the number of rounds needed to create a group. If the group creation may only require a constant number of rounds, only the two schemes SHM and DH-LKH remain to be selected. To make a decision between these two schemes, the decision tree additionally considers whether the computational cost of removing and adding group members may be at most logarithmic or linear. If the number of rounds is allowed to be at most logarithmic, DH-LKH is clearly the best choice and SHM otherwise. In case that group creation may also require logarithmically many rounds, the choice expands to the schemes SHM, D-OFT, and DH-LKH. Since the scheme D-OFT is the only one of

Table 9 Performance of distributed/contributory Secure Group Communication Schemes (Part 1/3)

		CRGR	SHM	D-OFT	DH-LKH	D-LKH	Octopus	G-DH	BD
Storage		O(n)	O(1)	O(log(n))	O(n)	O(log(n))	O(1)	O(n)	O(1)
Computation costs	Creation	$O(n^2)$	O(n)	O(log(n))	O(n*log(n))	O(n*log(n))	O(n)	O(n)	O(n)
	Addition	$O(n^2)$	O(n)	O(log(n))	O(log(n))	O(log(n))	n/a	O(n)	n/a
	Revocation	$O(n^2)$	O(n)	O(log(n))	O(log(n))	O(log(n))	n/a	O(n)	n/a
Message size	Creation	O(n)	O(n)	O(1)	O(n)	O(n*log(n))	O(n)	O(n)	O(n)
	Addition	O(n)	O(n)	O(1)	O(n)	O(1)	n/a	O(n)	n/a
	Revocation	O(n)	O(n)	O(1)	O(n)	O(log(n))	n/a	O(n)	n/a
Message amount	Creation	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
	Addition	O(n)	O(n)	O(log(n))	O(1)	O(log(n))	n/a	O(1)	n/a
	Revocation	O(n)	O(n)	O(log(n))	O(1)	O(log(n))	n/a	O(1)	n/a

these schemes that requires only constant-sized messages for each group operation, our decision tree next considers what the use case requires in terms of this number. If the messages can only be of constant size, the scheme D-oFT can be recommended immediately. If, on the other hand, the messages may also be linear, our decision tree next asks what requirements the use case has with respect to the number of rounds for removing and adding members. If this number is allowed to be at most constant, the choice is restricted to the two schemes DH-LKH and SHM. The choice between these two schemes is made on the basis of the requirements for the calculation cost of removing and adding members. If this cost can be at most logarithmic, the DH-LKH scheme is recommended, but if it can be linear, the SHM scheme is clearly the best choice. If the number of rounds for removing and adding members is also allowed to be logarithmic, the decision tree next restricts the use case further in terms of storage requirements. If the storage requirements may be at most constant or logarithmic, then the SHM or D-OFT scheme is clearly the best choice. In the case that the storage costs are also linear, a final restriction of the use case is made on the basis of the number of messages for the removal and addition of group members. If this number must be constant at most, then D-OFT is recommended and otherwise DH-LKH.

If the creation of a group is also allowed to require a linear number of rounds, our decision tree next further restricts the use case in terms of the requirements for the number of rounds for removing and adding members. In the case where this number of rounds may be at most constant, the schemes DH-LKH, D-LKH and G-DH are available for selection. From this selection, we can directly recommend the scheme D-LKH, if in addition it is valid that the SIZE of messages for adding group members may be constant at maximum. If, however, the size of the messages for the addition may also be linear, we still recommend the scheme D-LKH, if in addition it applies that the size of the messages for the removal of group members may be at most logarithmic. If the requirements of the use case on the size of messages for removing and adding group members are less strict and these messages may also be linearly large, our decision tree makes the next restriction on the basis of the computational costs for removing and adding members. If these costs are allowed to be the highest logarithmic, the choice narrows down to schemes D-LKH and DH-LKH. The final choice between these schemes is made by our decision tree based on the storage cost requirements of the use case. If the storage cost can be at most logarithmic, the scheme D-LKH is clearly the best choice, and the scheme DH-LKH if the storage cost can also be linear. However, if the calculation costs for removing and adding members may also be linear, the schemes G-DH, SHM, D-LKH and DH-LKH are available. To select from these schemes, our decision tree restricts the use case based on the requirements for the number of messages for removing and adding group members. If this number is allowed to be linear, the scheme SHM is clearly the best choice. In case this number may be constant at most, the G-DH scheme is recommended. If, however, the number of messages for entering and adding may be at most logarithmic, then the application case must be further restricted again. This time this is done on the basis of the storage costs. If these may be at most logarithmic, then D-LKH is the best choice and if the storage costs may also be linear the scheme G-DH.

In the case that a maximum logarithmic number of rounds may be required for adding and removing members, the schemes SHM, D-OFT, D-LKH, DH-LKH and G-DH are available for selection. In order to propose a scheme from this selection, the decision tree next restricts the application case with respect to the required number of messages for entering and adding members. If this number can be at most linear, the selection is reduced to the schemes G-DH and DH-LKH. The choice between these two schemes is made by the decision tree on the basis of the question, which requirements the use case makes on the computational effort for removing and adding. If this effort may be at most logarithmic, the

Table 10 Performance of distributed/contributory Secure Group Communication Schemes (Part 2/3)

		PCGR	CKA
Storage		O(n)	O(1)
Computation costs	Creation	O(n)	O(n)
	Addition	n/a	n/a
	Revocation	n/a	n/a
Message size	Creation	O(n)	O(1)
	Addition	n/a	n/a
	Revocation	n/a	n/a
Message amount	Creation	O(n)	O(1)
	Addition	n/a	n/a
	Revocation	n/a	n/a

Table 11 Performance of distributed/contributory secure group communication schemes (Part 3/3)

Scheme	Rounds		
	Creation	Join	Leave
CRGR	O(1)	O(1)	O(1)
SHM	O(1)	O(1)	O(1)
D-OFT	log(n)	log(n)	log(n)
D-LKH	O(1)	O(1)	O(1)
DH-LKH	O(n)	O(1)	O(1)
Octopus	O(n)	n/a	n/a
G-DH	O(n)	O(1)	O(1)
BD	O(1)	n/a	n/a
PCGR	O(1)	n/a	n/a
CKA	O(1)	n/a	n/a

scheme DH-LKH is the best choice. The scheme G-DH is the best choice, if this effort may also be linear. If the number of messages for removing and adding group members is allowed to be at most logarithmic, the scheme D-OFT is clearly the best choice. If, however, the number of messages for the removal and addition of group members may also be linear, then there is still the choice between the two schemes SHM and D-OFT. The decision tree makes this choice based on the memory requirements of the application. If this may be at most constant, SHM is the best choice and D-OFT if the memory requirement may also be logarithmic.

5.2 Feature-driven analysis of distributed/contributory SGC Schemes

The features for distributed/contributory SGC schemes are listed in Tables 12 and 8 and correspond to the features that were also analyzed for the centralized SGC Schemes. These tables are from our first survey [10]. Tables 12 and 8 allow us to directly conclude the following: (1) only the scheme PCGT changes its group keys periodically, while all other schemes do so after any change in group composition, (2) the *life cycle* of all schemes are unlimited, (3) only of the schemes SHM, Octopus, G-DH, and BD, the cryptographic techniques on which these schemes are based are identical, and (4) the schemes D-LKH, DH-LKH, D-OFT, SHM, BD, G-DH, Octopus, and CKA have the same features as shown in Table 8 have the same features. Analogous to the creation of the purely feature-oriented decision tree for centralized SGC schemes, we first determine the possible rankings with respect to F_{Feat} for the creation of the purely feature-oriented decision tree for distributed/contributory SGC schemes:

- $F_{Feat}(\text{CKA}) = F_{Feat}(\text{Octopus}) = F_{Feat}(\text{BD}) = F_{Feat}(\text{SHM}) = F_{Feat}(\text{D-OFT}) = F_{Feat}(\text{D-LKH}) = F_{Feat}(\text{G-DH}) < F_{Feat}(\text{DH-LKH})$
- $F_{Feat}(\text{CRGR}) < F_{Feat}(\text{DH-LKH})$

Based on the above rankings, we can now determine the Pareto front for the target vector F_{Feat}, which consists of the schemes {PCGR}, {CRGR} and {D-LKH, D-OFT, SHM, BD, Octopus, CKA, G-DH}, where we have grouped schemes that are identical with respect to F_{Feat} into sets. To make a selection from the Pareto front, we have created a corresponding decision tree in Fig. 9. This decision tree first constrains the use case as to whether massage requires confidentiality or not. If this is the case, the PCGR scheme is recommended if the cryptography used must also be minimal, and the CRGR scheme is recommended otherwise. However, if the use case does not require *message confidentiality*, the decision tree also next considers the requirements for the cryptography used. If this must be minimal, then the schemes D-LKH, D-OFT, SHM, BD, Octopus, CKA, G-D are suggested, which are all identical with respect to F_{Feat} and otherwise the scheme CRGR.

5.3 Performance and feature-driven analysis of distributed/contributory SGC Schemes

Building on the separate analyses regarding the performance and properties of distributed/contributory schemes, we now construct a decision tree for the selection of distributed/contributory schemes in Fig. 11 that takes both performance and features into account. To do this, we first define security levels in the same way as for the centralized schemes. Therefore, we again determine the possible rankings of the partial vector $F_{Feat}^{f_{24}-f_{31}}$. Therefore, we again abbreviate the vector $F_{Feat}^{f_{24}-f_{31}}$ with F_p for the sake of simplicity:

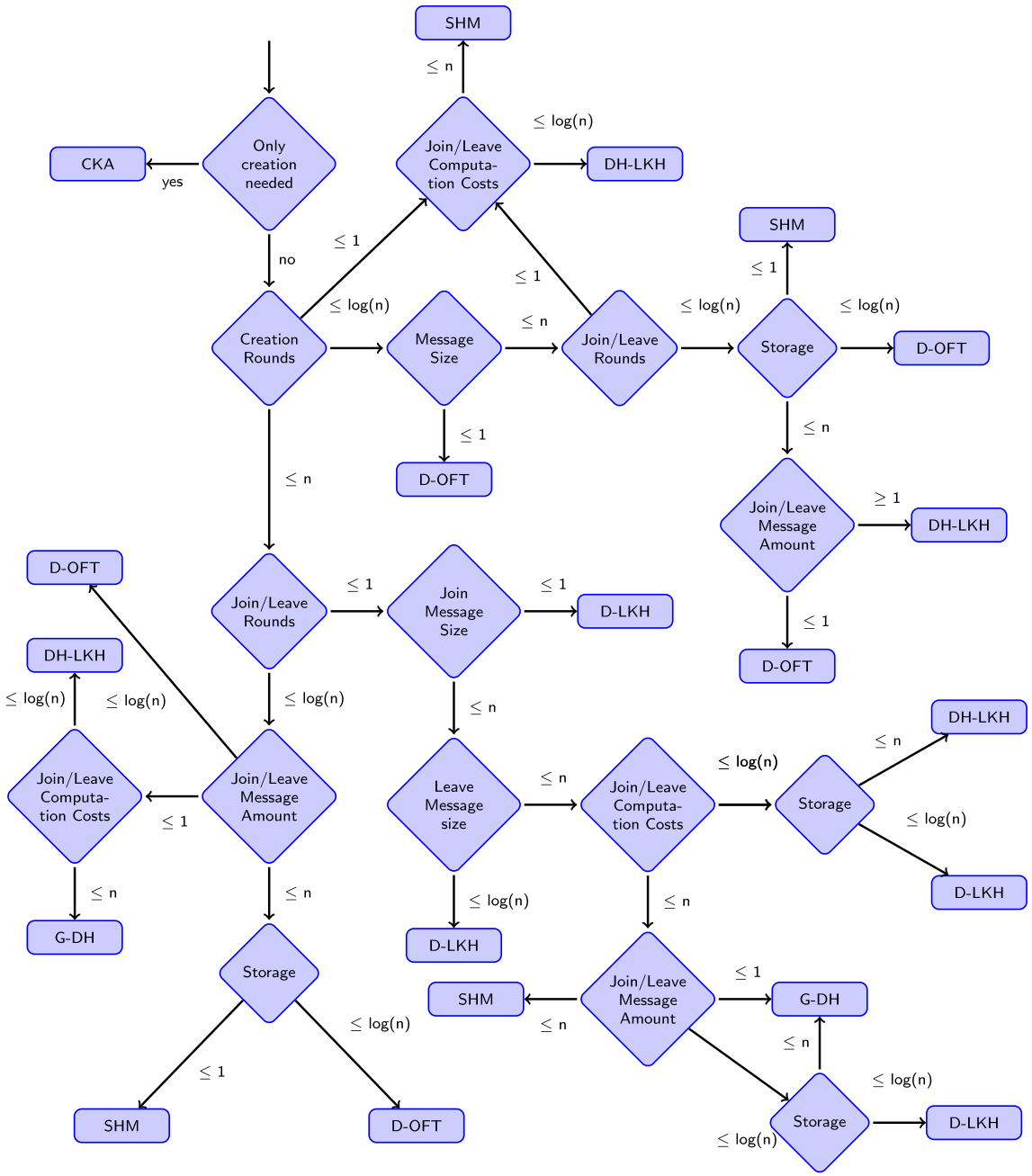

Fig. 8 Decision tree for the selection of distributed/contributory SGC schemes if only performance is considered

- F_p(CRGR) < F_p(PCGR) < F_p(D-LKH) = F_p(DH-LKH) = F_p(D-OFT) = F_p(SHM) = F_p(BD) = F_p(G-DH) = F_p(Octopus) = F_p(CKA)
- F_p(CRGR) < F_p(PCGR)

Also analogous to the procedure for the centralized schemes, we visualize the above rankings in Fig. 10. Here, the "<" operator in this figure again means that if "Scheme A < Scheme B," the feature set of A is a true subset of the feature set of B. For schemes that are in the same box, their feature sets are again identical.

Based on the visualization above, it is immediately apparent that the feature combination of scheme CRGR should represent the highest security level. This is because the feature combinations of the other schemes each provide only a true subset of the feature combination of CRGR. Thus, all that remains to be determined is how the security levels of the {PCGR} and {D-LKH, DH-LKH, D-OFT, SHM, BD, Octopus, CKA, G-DH} schemes relate to each other,

Table 12 Key update frequency, used cryptography and life cycles of distributed/ contributory Secure Group Communication Schemes

Scheme	Key update frequency	Types of cryptography	Life cycles
CRGR	Membership Change	Polynomial Public key	Unlimited
SHM	Membership Change	DH	Unlimited
D-OFT	Membership Change	One-way fct XOR	Unlimited
D-LKH	Membership Change	Symmetric	Unlimited
DH-LKH	Membership Change	Symmetric DH	Unlimited
Octopus	Membership Change	DH	Unlimited
G-DH	Membership Change	DH	Unlimited
BD	Membership Change	DH	Unlimited
PCGR	Time Period	Polynomial	Unlimited
CKA	Membership Change	One-way fct public key	Unlimited

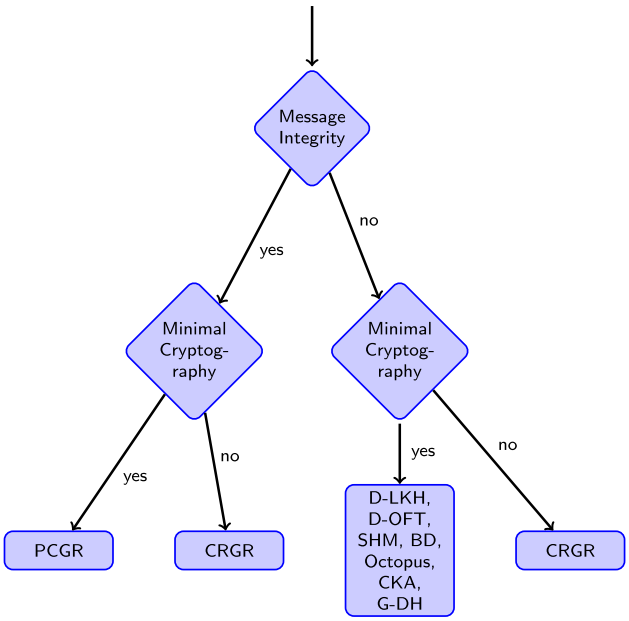

Fig. 9 Decision tree for the selection of distributed/contributory SGC schemes if only features are considered

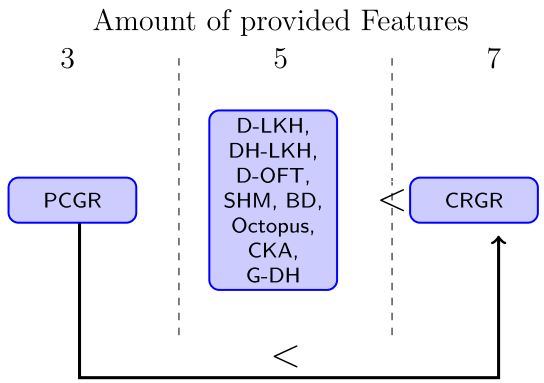

Fig. 10 Arrangement of the distributed/contributory SGC schemes based on the number and subset relationships of the features provided. The "<" operator means that the feature set of A is a true subset of the feature set of B if "Schema A < Schema B." For schemas which are in the same box, their feature sets are identical

where we have again combined schemes with the same feature combination into one set. Since no subset relationship exists between these two sets, we use the number of features they provide for the ordering of their levels, analogous to the procedure for the centralized schemes. It follows that the PCGR scheme has the lowest security level, the D-LKH, DH-LKH, D-OFT, SHM, BD, Octopus, CKA and G-DH schemes have the second lowest security level, and the CRGR scheme has the highest security level. Thus, we can define the following security levels based on the just determined ranking of the schemes and their feature sets. To avoid confusion with the security levels of the centralized schemes, we use in addition the index "dis."

- $S_{dis,1}$ = {*message integrity*, *message confidentiality*, *group independence*}
- $S_{dis,2}$ = {*backward secrecy*, *forward secrecy*, *instant rekey*, *message confidentiality*, *group independence*}
- $S_{dis,3}$ = {*backward secrecy*, *forward secrecy*, *instant rekey*, *message integrity*, *message confidentiality*, *compromise robustness*, *group independence*}

In the case that the security level must be greater than 2, our decision tree directly proposes the scheme CRGR, since it is the only scheme in question. If, on the other hand, the security level only needs to be greater than 1, we next consider the requirements of the use case for the cryptography used. If this must be not minimal, then the selection problem is the same as selecting a distributed SGC scheme when only performance is considered. Therefore, at this point we refer to the corresponding subtree in Fig. 8. In the case that the cryptography used must be minimal, the selection problem is again the same as for the selection of a distributed SGC scheme with focus only on performance, except that now the DH-LKH scheme is not available for selection. Therefore, the following subtree corresponds to the subtree in Fig. 8, except that it has been truncated by the options to select the scheme DH-LKH. The case where the security level does not play a role at all again completely corresponds to the problem of selecting a distributed SGC scheme, when only performance is considered.

6 Selection guidelines for decentralized/hybrid SGC schemes

Having considered the cases where the use case dictates the use of a centralized or distributed/contributory scheme, we now consider the case where the use of a decentralized/hybrid scheme is dictated. As a concrete scenario, we can imagine the smart city scenario already mentioned for the centralized schemes. However, the city hall is now willing to forego direct communication and thus control of the IoT devices in exchange for better system reliability. The fact that in distributed/contributory schemes the group members also communicate with each other for group operations by means of a subgroup controller means that the individual subgroups can continue to operate if the CI is unavailable.

For the decentralized/hybrid schemes, we again proceed analogously to the centralized and distributed/contributory schemes. We first present the performance and features of distributed/contributory schemes, determine the corresponding Pareto front to narrow down the selection and design decision trees that take into account only the performance, only the features or both.

6.1 Performance-driven analysis of decentralized/hybrid SGC schemes

The performance of distributed/contributory secure group communication schemes is listed in Tables 13 and 14 using Landau notation. Here, n again stands for the number of group members. The performance is considered on the basis of the same aspects as for the centralized schemes. From the table, it can be directly seen that the Kronos, Slimcast, CS, and DEP schemes each have the same performance. In order to get a deeper overview of the performance of the individual schemes, we again determine the following all possible rankings with respect to F_{Perf}:

- $F_{Perf}(\text{LNT}) > F_{Perf}(\text{Kronos}) = F_{Perf}(\text{Slimcast}) = F_{Perf}(\text{CS}) = F_{Perf}(\text{DEP}) > F_{Perf}(\text{Hydra}) > F_{Perf}(\text{IGKMP})$
- $F_{Perf}(\text{LNT}) > F_{Perf}(\text{RiSeG})$

Using these rankings, the Pareto front for decentralized SGC schemes can again be directly specified, which is formed by the schemes IGKMP, RiSeG, Alohali, Marks, SMKD, and Iolus.

To make the final selection from these schemes, we again design a corresponding decision tree in Fig. 12. In this tree, we first address the peculiarities that the scheme Alohali only supports the creation of groups and the schemes SMKD and Marks only support the creation of groups and addition of members. Accordingly, our decision tree first asks whether only creating groups should be possible. If yes, then Alohali is the only possible choice. If, in addition to the creation of groups, the addition of members should also be possible, the schemes SMKD and Marks are available for selection. The choice between these two schemes is based on the question of whether the use case for the creation of a group allows a maximum constant or linear number of messages. In the former case, SMKD is clearly the best choice, in the latter case Marks.

If the use case requires not only the creation of a group and the addition of members but also the removal of members, only the schemes RiSeG, IGKMP and Iolus are available. Since Iolus is the only scheme that requires a constant number of messages for removing members, we next narrow down the use case based on this aspect. In doing so, we obviously recommend Iolus if only constant many messages are allowed for removal and otherwise need to further narrow the use case to make a choice. Therefore, we next consider whether the use case only allows a maximum constant computation cost for the CI when removing a member, or whether it may also be linear. If the CI computation cost is only allowed to be constant, the only scheme to choose is RiSeG, which we recommend accordingly. If, on the other hand, the costs are also

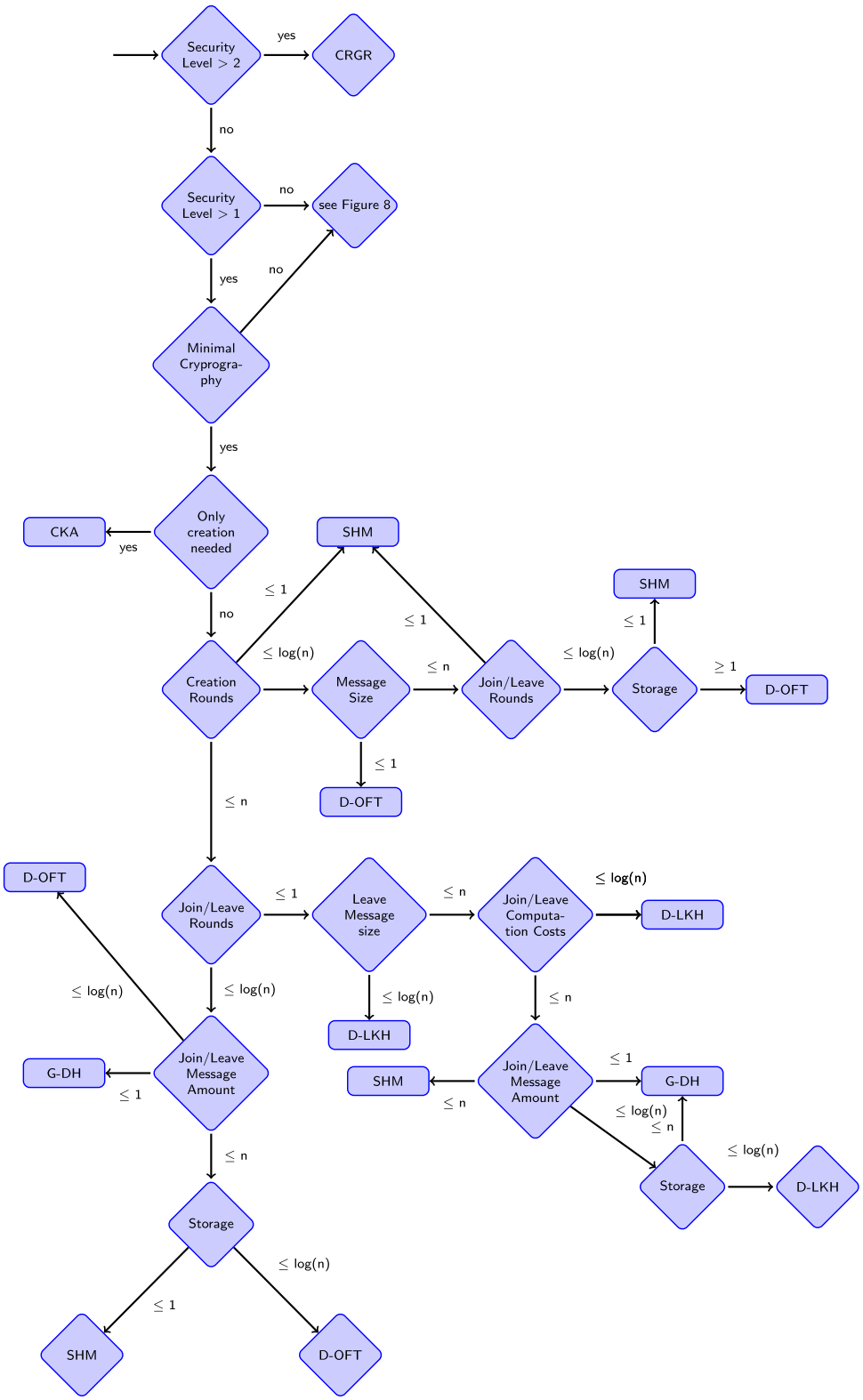

Fig. 11 Decision tree for the selection of distributed/ contributory SGC schemes if both performance and features are considered

Table 13 Performance of decentralized/hybrid secure group communication schemes (Part 1/2)

		Alohali	RiSeG	LNT	DEP	CS	Slimcast	Kronos	Marks	Iolus
Storage		CI: O(1)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(1)	CI: O(n)
		M: O(1)	M: O(n)	M: O(n)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: O(1)
Computation costs	Creation	CI: O(n)	CI: O(1)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(1)	CI: O(n)
		M: O(n)	M: O(n)	M: O(n)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: O(n)
	Addition	CI: n/a	CI: O(1)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(1)	CI: O(1)
		M: n/a	M: O(1)	M: O(n)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: O(1)
	Revocation	CI: n/a	CI: O(1)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: O(n)	CI: n/a	CI: O(n)
		M: n/a	M: O(1)	M: O(n)	M: O(1)	M: O(1)	M: O(1)	M: O(1)	M: n/a	M: O(1)
Message size	Creation	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)
	Addition	n/a	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)
	Revocation	n/a	O(1)	O(1)	O(1)	O(1)	O(1)	O(1)	n/a	O(n)
Message amount	Creation	O(n)	O(n)	$O(n^2)$	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)
	Addition	n/a	O(1)	O(n)	O(n)	O(n)	O(n)	O(n)	O(1)	O(1)
	Revocation	n/a	O(n)	O(n)	O(n)	O(n)	O(n)	O(n)	n/a	O(1)

Table 14 Performance of decentralized/hybrid secure group communication schemes (Part 2/2)

		SMKD	Hydra	IGKMP
Storage		CI: O(n)	CI: O(1)	CI: O(1)
		M: O(1)	M: O(1)	M: O(1)
Computation costs	Creation	CI: O(n)	CI: O(n)	CI: O(n)
		M: O(1)	M: O(1)	M: O(1)
	Addition	CI: O(1)	CI: O(n)	CI: O(1)
		M: O(1)	M: O(1)	M: O(1)
	Revocation	CI: n/a	CI: O(n)	CI: O(n)
		M: n/a	M: O(1)	M: O(1)
Message size	Creation	O(1)	O(1)	O(1)
	Addition	O(1)	O(1)	O(1)
	Revocation	n/a	O(1)	O(1)
Message amount	Creation	O(1)	O(n)	O(n)
	Addition	O(1)	O(n)	O(1)
	Revocation	n/a	O(n)	O(n)

allowed to be linear, we further restrict the use case based on the CI's storage requirements. For the case that the CI may require a maximum of constant memory, only the scheme IGKMP is available for selection, which we can again recommend directly. If the CI may also require linear memory, we can finally conclude the selection on the basis of another question. May the calculation costs of the CI for group creation be constant at most or linear as well? In the former case, the scheme RiSeG is clearly the best choice, in the latter case the scheme IGKMP.

6.2 Feature-driven analysis of decentralized/hybrid SGC schemes

The features of the decentralized/hybrid SGC schemes are listed in Tables 15 and 8. The features considered in these tables are the same as of the centralized and distributed schemes. Using these two tables, we can make the following statements: (1) all schemes change their group key when the group compositions change, only the DEP, Kronos and Iolus schemes change their keys periodically, (2) the schemes of the sets {Iolus, SMKD, IGKMP, SLIMCAST} and {RISEG, LNT } are each based on the same cryptographic functions, while all other schemes are based on a unique combination of cryptographic functions, (3) the schemes of the sets {DEP, Kronos }, {IGKMP, HYDRA}, {Iolus, CS} and {Slimcast,

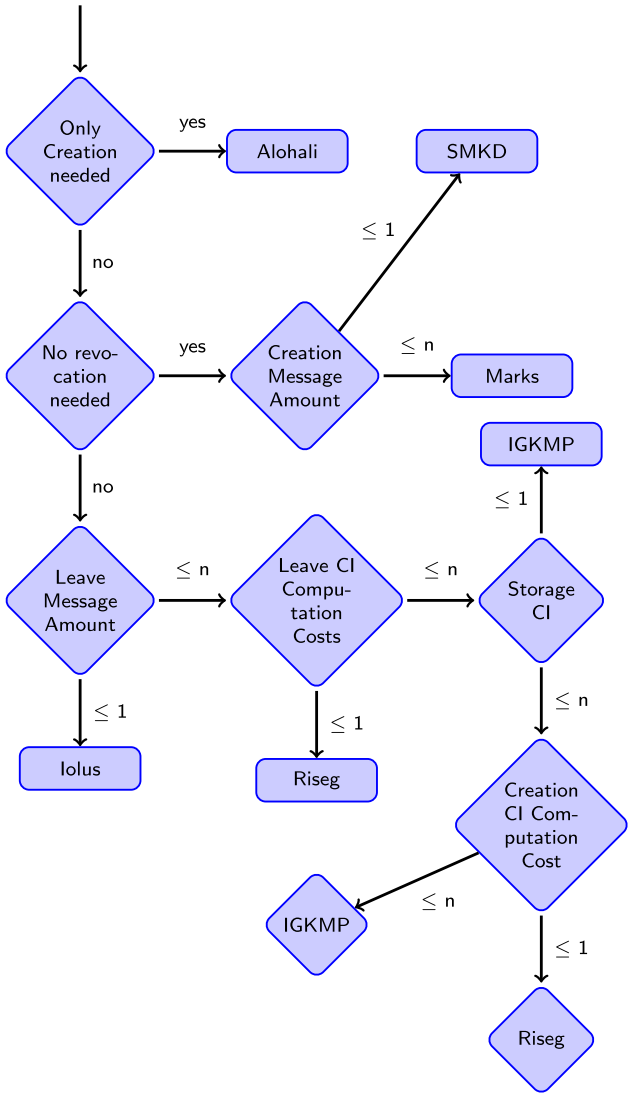

Fig. 12 Decision tree for the selection of decentralized/hybrid SGC schemes if only performance is considered

RiSeG, LNT} have the same characteristics with respect to the cryptographic functions listed in Table 8 and (4) only the *life cycle* of the scheme Marks are limited. In addition, the following rankings can be determined with respect to F_{Feat}:

- $F_{Feat}(\text{Slimcast}) < F_{Feat}(\text{Riseg}) = F_{Feat}(\text{LNT})$
- $F_{Feat}(\text{Riseg}) = F_{Feat}(\text{LNT}) < F_{Feat}(\text{Kronos}) = F_{Feat}(\text{Dep})$
- $F_{Feat}(\text{Riseg}) = F_{Feat}(\text{LNT}) < F_{Feat}(\text{Hydra}) = F_{Feat}(\text{IGKMP})$
- $F_{Feat}(\text{Alohali}) < F_{Feat}(\text{Hydra}) = F_{Feat}(\text{IGKMP})$
- $F_{Feat}(\text{Hydra}) = F_{Feat}(\text{IGKMP}) < F_{Feat}(\text{CS}) = F_{Feat}(\text{Iolus})$
- $F_{Feat}(\text{Hydra}) = F_{Feat}(\text{IGKMP}) < F_{Feat}(\text{SMKD})$
- $F_{Feat}(\text{CS}) = F_{Feat}(\text{LNT}) < F_{Feat}(\text{Marks})$
- $F_{Feat}(\text{SMKD}) < F_{Feat}(\text{Marks})$

Using the above rankings, we can directly determine the Pareto front, which is formed by the Alohali and Slimcast schemes. The difference between these two schemes is that Alohali provides *message integrity*, which is not the case with Slimcast, but Alohali does not have the compromise robustness feature that Slimcast does. Accordingly, our decision tree for the selection of a decentralized SGC scheme based on features only consists of the question whether *message integrity* or *compromise robustness* is more important for the use case. In the former case, the scheme Alohali is recommended, in the latter case Slimcast. The decision tree is also illustrated in Fig. 13.

6.3 Performance and feature-driven analysis of decentralized/hybrid SGC schemes

Now that we have created the decision trees for decentralized schemes that only take performance or features into account, we will also create a tree that takes both performance and features into account. To do this, we again proceed in the same way as for the centralized and distributed schemes and first define security levels again. To do so, we again determine the possible rankings of the partial vector $F_{Feat}^{f_{24}-f_{31}}$ and use again the abbreviation F_p for $F_{Feat}^{f_{24}-f_{31}}$:

- $F_p(\text{Marks}) > F_p(\text{SMKD})$
- $F_p(\text{Marks}) > F_p(\text{Iolus}) = F_p(\text{CS})$
- $F_p(\text{SMKD}) > F_p(\text{DEP}) = F_p(\text{Kronos})$
- $F_p(\text{Iolus}) = F_p(\text{CS}) > F_p(\text{Hydra}) = F_p(\text{IGKMP})$
- $F_p(\text{Dep}) = F_p(\text{Kronos}) > F_p(\text{Hydra}) = F_p(\text{IGKMP})$
- $F_p(\text{Hydra}) > F_p(\text{Alohali})$
- $F_p(\text{Hydra}) > F_p(\text{LNT}) = F_p(\text{Riseg}) = F_p(\text{Slimcast})$

To get a better overview of these rankings, we visualize them in Fig. 14.

From Fig. 14, it can be immediately seen that Marks should represent the lowest security level. It can also be directly concluded that the schemes (1) Dep and Kronos, (2) Iolus and CS, (3) IGKMP and Hydra and (4) LNT, RIseg and Slimcast should each have the same level. It is also immediately obvious how the security levels of the schemes SMKD, Dep, Kronos, IGKMP and Hydra should relate to each other. The security level of SMKD should be lower than the level of Dep and Kronos. The level of Dep and Kronos should be lower than the level of IGKMP and Hydra. The question is how the level of the schemes Iolus and CS fits into this arrangement. On the one hand, Iolus clearly has to have a lower level than the IGKMP and Hydra schemes, but on the other hand, it has to have a higher level than Marks. Since Iolus and CS cannot otherwise be further compared with the SMKD, Dep and Kronos schemes, we finally rank these two schemes based on the number of features they provide. This

Table 15 Key update frequency, used cryptography and limitations of decentralized/hybrid secure group communication schemes

Scheme	Key update frequency	Types of cryptography	Life cycles
Alohali	Membership Change	One-way fct.	Unlimited
RiSeG	Membership Change	Symmetric Elliptic Curve	Unlimited
LNT	Membership Change	Symmetric Elliptic Curve	Unlimited
DEP	Time Period	Symmetric Public Key	Unlimited
CS	Membership Change	Reversible cipher sequence	Unlimited
Hydra	Membership Change	Public Key	Unlimited
Slimcast	Membership Change	Symmetric	Unlimited
Kronos	Time Period	Symmetric One-way fct	Unlimited
Marks	Time Period	Hash	limited
Iolus	Time Period	Symmetric	Unlimited
SMKD	Membership Change	Symmetric	Unlimited
IGKMP	Membership Change	Symmetric	Unlimited

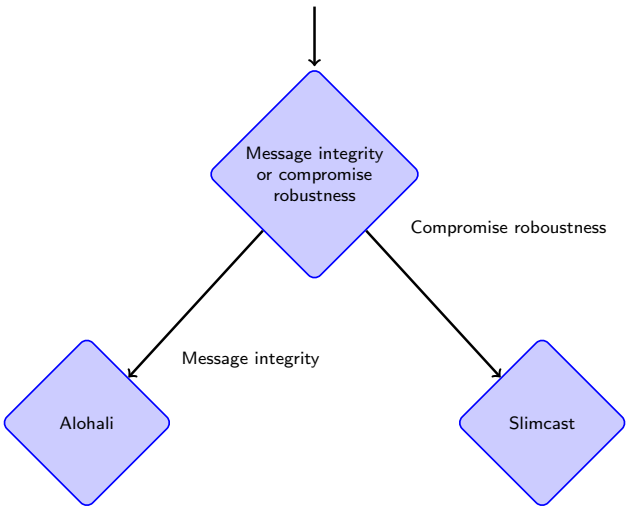

Fig. 13 Decision tree for the selection of decentralized/hybrid SGC schemes if only features are considered

results in Iolus and CS having a lower level than IGKMP and Hydra, but providing a higher level than Dep and Kronos. Thus, only the levels of the schemes Alohali or LNT, Riseg and Slimcast have to be inserted into this arrangement. From the above illustration, it can be deduced that these schemes should have a higher level than the IGKMP and Hydra schemes. All that remains is to clarify whether the Alohali scheme should have a higher, equal, or lower level than the LNT, Riseg, and Slimcast schemes. Since we cannot form any further rankings between these schemes with respect to $F_{Feat}^{f_{24}-f_{31}}$, and since these schemes all offer the same number of features, it makes sense that they should all be classified at the same level. Thus, the feature combination of Marks forms security level 1, the combination of SMKD forms security level 2, the combination of Dep and Kronos forms level 3, the combination of Iolus and CS forms level 4, the combinations of the schemes IGKMP and Hydra form level 5, and the combinations of the schemes Alohali and LNT, Riseg, and Slimcast form levels 6.1 and 6.2, respectively. Thus, we formally define the following security levels for decentralized schemes, based on their feature combinations. We use the index "de" to avoid confusion with the security levels of the centralized and distributed schemes, respectively.

- $S_{de,1}$ ={*group independence*},
- $S_{de,2} = S_{de,1} \cup$ {*member authentication*, Backward Secrecy},

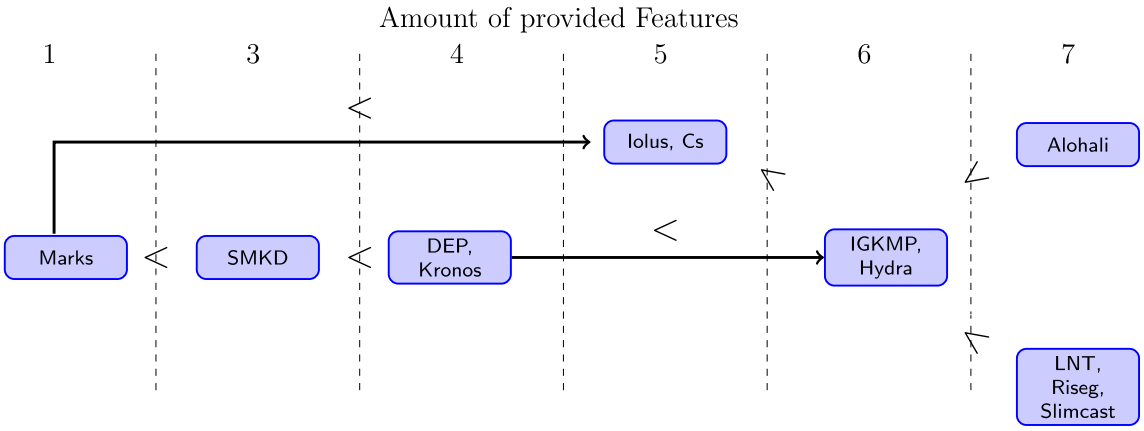

Fig. 14 Arrangement of the decentralized/hybrid SGC schemes based on the number and subset relationships of the features provided. The "$<$" operator means that the feature set of A is a true subset of the feature set of B if "Schema A $<$ Schema B." For schemas which are in the same box, their feature sets are identical

- $S_{de,3} = S_{de,2} \cup \{\textit{message confidentiality}\}$,
- $S_{de,4} = S_{de,1} \cup \{\textit{backward secrecy}, \textit{forward secrecy}, \textit{instant rekey}, \textit{message confidentiality}\}$,
- $S_{de,5} = S_{de,4} \cup \{\textit{message authentication}\}$,
- $S_{de,6.1} = S_{de,5} \cup \{\textit{message integrity}\}$ and
- $S_{de,6.2} = S_{de,5} \cup \{\textit{compromise robustness}\}$.

Our decision tree in Fig. 15, analogous to the centralized and distributed schemes, again follows the principle of first determining the required security level and then narrowing down the use case in terms of cryptography and life cycle requirements and which group operations should be supported at all. If these constraints are still not sufficient for a clear recommendation of a scheme, the use case is further constrained based on performance aspects. Thus, our decision tree first considers whether the security level should be greater than 6. For this case, only the schemes Alohali, LNT, and Riseg are available for selection, all of which have unlimited life cycles. Next, the decision tree asks whether the cryptography used should be minimal or not. If yes, the choice narrows down to the schemes Alohali and Slimcast. Since Alohali only supports the creation of a group, but Slimcast additionally allows the addition and removal of members, we make the final selection based on the criterion whether a use case requires that only a group can be created, but no members can be added or removed, or whether not. In the former case our tree recommends Alohali, in the latter case Slimcast. If, on the other hand, the use case requires that the cryptography used must not be minimal, the four schemes Alohali, LNT, RIseg and Slimcast are still available. Since Alohali is the only one of these schemes that only allows the creation of a group, but not the addition and removal of members, the decision tree asks what the use case requires in this regard. If only one group can be created, then Alohali is the only possible choice, but if members can be added or removed, then a choice must be made between the schemes LNT, Riseg and Slimcast. However, this choice can be restricted directly to the schemes Slimcast and Riseg on the basis of the determined rankings with respect to F_{perf}. Slimcast is superior to Riseg only in the aspects of memory requirements of the members and computational effort of the members during group creation, since it has only constant costs there, while Riseg has linear costs there. In all other aspects, however, Riseg is better than Slimcast. Therefore, in the next two questions, our decision tree considers whether the members' memory requirement must be constant or the members' computational cost in group creation must be constant. If either of these is true, then the scheme Slimcast is recommended. However, if these two cost aspects are also allowed to be linear, then Riseg is clearly the best choice.

If, on the other hand, the use case only requires that the security level must be greater than 5, the schemes Alohali, LNT, Riseg, Slimcast, IGKMP and Hydra are available for selection. After the question about the security level, the next question is whether the use case requires minimal cryptography or not. If so, the schemes Alohali, Riseg, LNT and IGKMP are available in principle. Based on the previously determined rankings with respect to F_{perf}, LNT can be excluded directly. Since Alohali again only allows the creation of a group, but not the addition or removal of members, as is the case with Riseg and IGKMP, the next question is whether the use case requires that only a group may be created whose composition may not change, or whether the addition and removal of members should also be supported. In the former case only the scheme Alohali remains, in the latter case the schemes Riseg and IGKMP. RISEG is worse than IGKMP in all aspects of performance, except for the cost of CI for group creation and member removal. In these

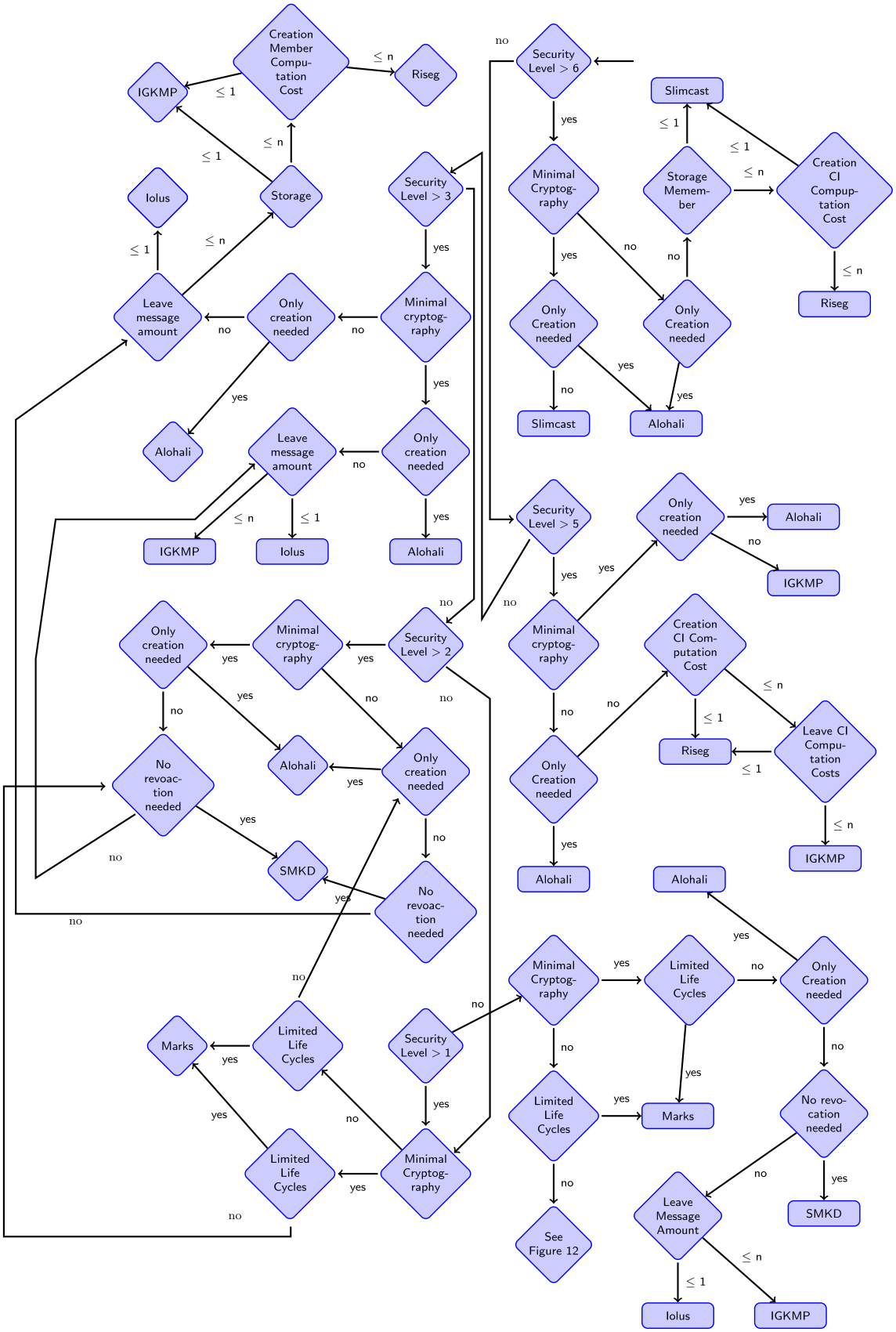

Fig. 15 Decision tree for the selection of decentralized/hybrid SGC schemes if both performance and features are considered

two aspects, Riseg has only constant costs, while IGKMP has linear costs. Therefore, in the next two questions, our decision tree considers whether the CI's computational cost for group creation and member removal, respectively, must be constant or may be linear. Riseg is recommended in the former case, and IGKMP in the latter.

If the use case only requires a security level greater than 4, the schemes Alohali, Riseg, Slimcast, LNT, Hydra, IGKMP, Iolus and CS are available for selection. In order to limit this selection, the decision tree next poses the question of the cryptographic requirements. If these are allowed to be minimal, this restricts the choice to the schemes Alohali, IGKMP, Iolus, and CS. Since CS scores worse than IGKMP with respect to F_{perf}, however, this scheme can be excluded directly. Since of the remaining schemes again only Alohali supports only the creation of a group, the question regarding the allowed group operations is posed next. If only the creation of a group should be possible, then Alohali is recommended accordingly. Otherwise a choice must be made between Iolus and IGKMP. Since Iolus is worse than IGKMP in all performance aspects, except for the number of messages needed to remove a member, we use this number of messages as the next constraint criterion. If the use case only allows a maximum constant number of messages for removal, then Iolus is clearly the best choice, otherwise IGKMP. If, on the other hand, the cryptography does not have to be minimal, then the schemes Alohali, Riseg, IGKMP, and Iolus are still basically available for selection. Since of these Alohali again only allows the creation of a group, but the other schemes also support all other group operations, we next ask whether the use case only allows the creation of a group. If yes, then Alohali is recommended, if no, then a choice must be made between the Riseg, IGKMP, and Iolus schemes. Since Iolus is the only one of these schemes that only has a constant cost with respect to the number for removing a member, the next question we ask is what are the requirements of the use case with respect to this number. If this is only allowed to be constant, then Iolus is the best choice. Otherwise, the choice narrows down to the schemes Riseg and IGKMP, since Iolus always performs worse with respect to the remaining performance aspects. Since IGKMP performs worse than Riseg in terms of performance, apart from the memory requirements of the CI and the members, as well as the authorization overhead of the members during group creation, our decision tree uses these two aspects to restrict the use case. If the use case allows only constant memory costs and only constant computation overhead of the members during group creation, IGKMP is recommended and Riseg if the costs can also be linear in each case.

Since the subtree for the case that the security level must be greater than 3 is equivalent to the case that the security level must be greater than 4, we go in our decision tree directly from "Security Level $>$ 5" to "Security Level $>$ 3." The reason why these two subtrees are equivalent is that, in contrast to "Security Level $>$ 4," "Security Level $>$ 3" only adds the schemes DEP and Kronos, which are both minimal and have unlimited *life cycle*, but both are worse in terms of F_{perf} than the scheme IGKMP, which is also available in this case.

If the security level only has to be greater than 2, only the SMKD scheme is added in comparison with "Security level $>$ 3." The SMKD scheme has minimal cryptography and unlimited *life cycle*. However, it supports the creation of a group as well as the addition of members, but not the removal of members. Therefore, the subtree for the "Security Level $>$ 2" case is almost analogous to the "Security Level $>$ 3" case. The only difference is that if the question whether the use case only requires the creation of a group is answered with yes, then the question comes whether the use case only allows the creation of a group and the addition of members and prohibits the removal of members. If this is the case, the scheme SMKD is recommended, otherwise the decision tree behaves analogously to the case "Security Level $>$ 3" if the question "Only Creation needed" was answered with no.

If the use case only requires that the security level is greater than 1, then, compared to the case "Security Level $>$ 2," the Marks scheme is added. Marks has minimal cryptography, but has limited *life cycle*. Therefore, we can again design the following subtree largely analogous to the case "Security Level $>$ 2." The difference is that after asking about cryptography, we now ask about *life cycle*. If the use case requires that these should be limited, then Marks is recommended. Otherwise, the remaining subtree behaves analogously to the case "Security Level $>$ 2" after the question regarding the cryptography has been asked.

If the security level does not matter, our decision tree next bounds the use case based on whether the cryptography used must be minimal. If not, then all schemes are available for selection and the selection problem is largely the same as the problem of selecting a decentralized SGC scheme when only performance is considered. The only difference is that the purely performance-based tree does not consider *life cycle*. Therefore, in this case, our decision tree next asks whether the *life cycle* should be limited or not. If yes, then only Marks remains, if no, then it refers to the performance-based decision tree from Fig. 12. If the cryptography used is to be minimal, then the choice of possible schemes is restricted to Marks, Alohali, Iolus, SMKD and IGKMP. Since only Marks has limited life cycles, the choice can be narrowed down by the question whether the use case requires limited life cycles or not. If yes, then Marks is the only choice, if no, then Alohali, Iolus, SMKD and IGKMP remain for selection. Of these schemes, only Iolus and IGKMP allow both the creation of a group and the addition and removal of members. Alohali, on the other hand, only allows the creation of a group, and SMKD allows the creation of a group as well as the addition of members, but not the removal of them.

Therefore, Alohali is recommended next if the use case only requires the creation of a group, or SMKD if no members need to be removed. If, on the other hand, all group operations are required, then a choice must be made between Iolus and IGKMP. Since Iolus, in terms of performance, is inferior to IGKMP in all aspects except for the number of messages required to remove members, Iolus is only recommended if removing members may only require a constant number of messages. Otherwise IGKMP is recommended.

7 Cross-category selection guidelines for SGC Schemes

Having created the decision trees when the use case already dictates the use of a concrete class of schemes, we now consider the case when such a constraint is not already present when selecting a scheme. Accordingly, we now create cross-category decision trees for the selection of a scheme.

For this purpose, we proceed analogously to the category-specific decision trees and first design a decision tree based purely on performance or feature, respectively, in order to finally design a decision tree that takes both into account.

7.1 Cross-category performance-driven analysis of SGC schemes

A cross-category recommendation of an SGC scheme, purely on the basis of performance, first raises the question of how this is even possible with the different categories. Fortunately, the three categories can be split by means of their architecture into two disjoint sets. One with CI and one without CI. Or specifically into the two sets {centralized, decentralized/hybrid} and {distributed/contributory}. The performance of the centralized and decentralized/hybrid schemes are captured using the same metrics, see Tables 5 and 6 and 13 and 14, which allows a direct comparison of the two classes. Therefore, in our opinion, it is reasonable to assume that only a comparison between centralized and decentralized/hybrid schemes makes sense and that distributed/contributory schemes cannot be directly compared with any other category. In our opinion, this is also reflected in the basic architecture and involved actors of the different categories.

Therefore, the first question our decision tree in Fig. 16 asks is whether the use case requires a CI or not. If no, then only the distributed schemes are available for selection and the further subtree corresponds to the decision tree for the selection of a distributed scheme on the basis of performance from Fig. 11. If a CI is to be present after all, then we can directly restrict the selection of eligible schemes to the schemes GKMP, S2RP, OFT, OFCT, SGCSH, EBS, SMKD, Marks, Alohali, Riseg, Iolus and IGKMP. This is because, overall, only the centralized and decentralized schemes are eligible, and of these, again, only the schemes of the corresponding Pareto fronts. This set of schemes can be directly divided into the following three sets, where the schemes of the first set only support creating a group, the schemes of the second set only support creating a group as well as adding members, and those of the third set support removing members in addition to group creation and adding members: {Alohali}, {GKMP, SMKD, Marks} and {S2RP, OFT, OFCT, SGCSH, EBS, Riseg, Iolus, IGKMP}. Since GKMP is better than SMD and Marks in terms of F_{perf}, the second set can be directly reduced to {GKMP}. For further constraining the use case, the decision tree is based on these three sets. Thus, the next step is to determine whether only the creation of a group should be allowed, but not the removal or addition of members. If yes, then Alohali can be recommended directly. If no, then we next consider whether the use case requires that the creation of a group and the addition of members should be possible, but not the removal of members. If yes, then GKMP can be recommended directly. If no, then a selection must be made from the set {S2RP, OFT, OFCT, SGCSH, EBS, Riseg, Iolus, IGKMP}. In order to select a scheme from this set, the use case must be further restricted. This is done next on the basis of the question whether the calculation effort of the CI for the group creation must be constant or not. If yes, the scheme Riseg can be recommended directly. If no, we have already restricted the use case so far that the scheme SGCSH is constant everywhere in the remaining performance aspects, except for the number of messages needed for removing members. For this, SGCSH requires a linear number. Therefore, our decision tree next asks what constraints the use case has on this number. If this is allowed to be linear, then the scheme SGCSH can be recommended directly. If not, then the two cases still need to be handled if this number is only allowed to be constant and logarithmic, respectively.

For the case that the number of messages for the removal of a member may only be constant, the selection restricts itself to the schemes OFT, OFCT and Iolus, whereby the former two are identical regarding F_{perf}. Here, Iolus is recommended if the use case requires that either the total computational cost of removing and adding members must be constant or the storage requirements for members must be constant. Otherwise, OFT and OFCT are recommended.

In the case that the message count for removing members may also be logarithmic, the schemes S2RP, OFT, OFCT, EBS, and Iolus are available for selection overall. In order to select from these schemes, the use case is next constrained by the question of what storage requirements may be imposed on the members. If the members' memory requirements may be at most constant, then Iolus is the only choice. If, on the other hand, the storage requirements of the members may also be logarithmic, then the next step is to consider the total

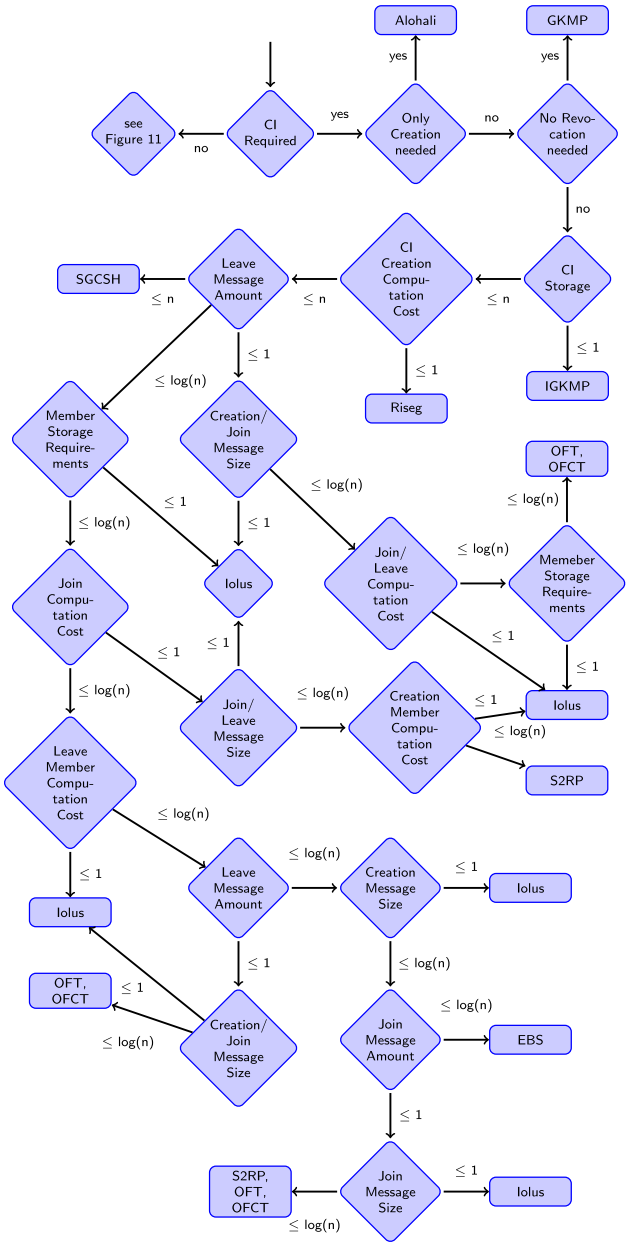

Fig. 16 Cross-category decision tree for the selection of SGC schemes if only the performance is considered

computational cost of adding members. If these costs may be only constant, then only Iolus and S2RP are to the selection. Here S2RP is recommended, if both the sizes of the messages for adding and removing members may be logarithmically large and if the computation costs of the members for removing members may be logarithmic. Otherwise, Iolus is recommended. On the other hand, if the computational cost of adding members is also allowed to be logarithmic, the use case is constrained on the basis of whether the computational cost of removing members is allowed to be at most constant or logarithmic. In the former case, only Iolus remains. In the latter case, Iolus, S2RP, OFT, OFCT, and EBS are still available for selection. Therefore, the use case must be further restricted, this time by the question of whether a maximum constant or logarithmic number of messages may be sent for the removal of members. In the former case, only the two schemes Iolus and OFT and OFCT remain. Iolus is recommended if the message size for adding and removing members may be constant at most and OFT and OFCT, which are identical with respect to F_{perf}. If, on the other hand, the message count for removing a member is also allowed to be logarithmic, then Iolus, S2RP, OFT, and OFCT and EBS are still available for selection, but the use case is now already so strongly restricted that S2RP, OFT, and OFCT are identical with respect to the remaining performance aspects. To make the final selection from these schemes, the decision tree asks three final questions. The first is whether the messages for group creation must be of constant size or may be logarithmic. In the former case, only Iolus remains. In the latter case, the second question is how many messages may be sent for adding a member. If logarithmically many messages may be sent, then EBS is clearly the best choice. If this number of messages for adding must be constant, the last question is asked, which is: may the messages for adding be at most constant or logarithmically large. In the former case, only Iolus remains; in the latter case, S2RP, OFT, and OFCT must be recommended, which are now indistinguishable due to the restrictions placed.

7.2 Cross-category feature-driven analysis of SGC schemes

For the cross-category, purely feature-based decision tree, we again divide the schemes into the two disjoint sets {centralized, decentralized/hybrid} and {distributed/collaborative}. Accordingly, the first question asked by the decision tree in Fig. 17 is whether a CI should be present or not. If it is not, then the remainder of the subtree corresponds to the decision tree for selecting distributed/contributing systems based solely on features from Fig. 9. If so, then a selection must be made from the centralized and distributed/hybrid systems. In doing so, we can restrict the selection of eligible schemes directly to the schemes of the Pareto fronts with respect to F_{Feat}. Thus, we can directly restrict the selection to the SBSA and S2RP schemes for the centralized systems and to Slimcast and Alohali for the decentralized/hybrid systems. To determine whether this selection can be further restricted, we determine whether additional rankings can be determined for these schemes with respect to F_{Feat}. In the course of this, we were able to determine the following rankings:

- $F_{Feat}(\text{Alohali}) < F_{Feat}(\text{SBSA})$
- $F_{Feat}(\text{Alohali}) < F_{Feat}(\text{LKH}) = F_{Feat}(\text{LKH+})$
- $F_{Feat}(\text{Slimcast}) < F_{Feat}(\text{LKH}) = F_{Feat}(\text{LKH+})$
- $F_{Feat}(\text{Slimcast}) < F_{Feat}(\text{S2RP})$

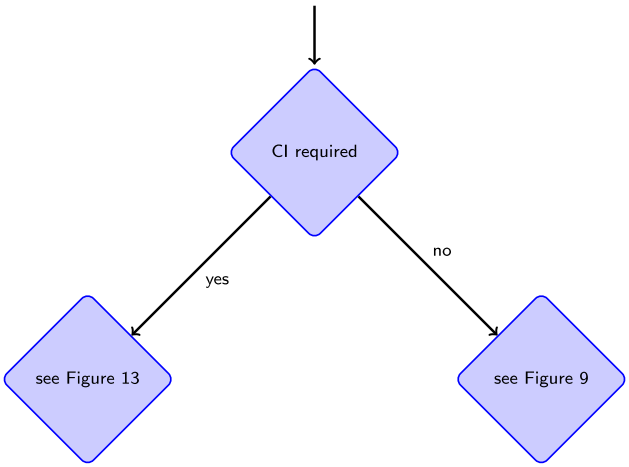

Fig. 17 Cross-category decision tree for the selection of a SGC schemes if only features are considered

Using the above rankings, the choice can be further narrowed down to the two schemes Slimcast and Alohali. Thus, the following subtree for the presence of a CI is equivalent to the decision tree for choosing a decentralized/hybrid scheme purely based on features from Fig. 13.

7.3 Cross-category performance and feature-driven analysis of SGC schemes

Now that we have created cross-category decision trees that focus only on performance or features, we will create a cross-category decision tree that takes both features and performance into account. In doing so, we proceed analogously to the creation of the category-specific decision trees which consider both features and performance. In concrete terms, this means that we first define the corresponding security levels. Again, we assume that only centralized and decentralized schemes can be meaningfully compared with each other and that our decision tree is divided into two subtrees right at the beginning. For one subtree, we assume that a centralized instance exists and therefore centralized and decentralized schemes are available for selection. For the other subtree, it is assumed that there is no central instance and therefore only distributed schemes are available for selection. Based on this initial split, we believe it is reasonable that we also consider this split when defining the security levels. Thus, the security levels for the case where no CI exists correspond to the security levels $S_{dis,1}$ - $S_{dis,3}$ of the distributed schemes. For the case where a CI is present, we need to merge the individual definitions of the security levels for centralized and distributed schemes into a common definition. To avoid confusion, we use the index "cdc," short for centralized decentralized/contributory, for the new security levels to be defined. To define these security levels, we again first determine all possible rankings of the centralized and decentralized schemes with respect to the partial vector $F_{Feat}^{f_{21}-f_{28}}$ and abbreviate $F_{Feat}^{f_{24}-f_{31}}$ again with F_p.

- $F_p(\text{Marks}) > F_p(\text{SGCSH})$
- $F_p(\text{Marks}) > F_p(\text{SMKD})$
- $F_p(\text{Marks}) > F_p(\text{GKMP})$
- $F_p(\text{DEP}) = F_p(\text{Kronos}) > F_p(\text{Iolus}) = F_p(\text{CS})$
- $F_p(\text{SGCSH}) > F_p(\text{Iolus}) = F_p(\text{CS})$
- $F_p(\text{CFKM}) > F_p(\text{Iolus}) = F_p(\text{CS})$
- $F_p(\text{CFKM}) > F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS})$
- $F_p(\text{Iolus}) = F_p(\text{CS}) > F_p(\text{SBSA})$
- $F_p(\text{Iolus}) = F_p(\text{CS}) > F_p(\text{S2RP})$
- $F_p(\text{Iolus}) = F_p(\text{CS}) > F_p(\text{IGKMP}) = F_p(\text{Hydra})$
- $F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS}) > F_p(\text{SBSA})$
- $F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS}) > F_p(\text{S2RP})$
- $F_p(\text{SKDC}) = F_p(\text{LKH}) = F_p(\text{LKH+}) = F_p(\text{OFT}) = F_p(\text{OFCT}) = F_p(\text{EBS}) = F_p(\text{XFKS}) > F_p(\text{IGKMP}) = F_p(\text{Hydra})$
- $F_p(\text{SBSA}) > F_p(\text{Alohali})$
- $F_p(\text{IGKMP}) = F_p(\text{Hydra}) > F_p(\text{Alohali})$
- $F_p(\text{S2RP}) > F_p(\text{LNT}) = F_p(\text{Riseg}) = F_p(\text{Slimcast})$
- $F_p(\text{IGKMP}) = F_p(\text{Hydra}) > F_p(\text{LNT}) = F_p(\text{Riseg}) = F_p(\text{Slimcast})$

To get a better overview of the above rankings, we visualize them in Fig. 18. In this figure, the operator "<" again means that if "Scheme A < Scheme B," the feature set of A is a true subset of the feature set of B.

Based on Fig. 18 and analogous to the definition of the category-specific security levels, we again decide that the security levels in this case can again be traced back, in simplified terms, to the number of features provided. Based on this definition, we can define the following security levels, which again correspond to the feature combinations of the individual schemes:

- $S_{cdc,1} = \{group\ independence\}$,
- $S_{cdc,2.1} = S_{cdc,1} \cup \{message\ integrity, message\ confidentiality\}$,
- $S_{cdc,2.2} = S_{cdc,1} \cup \{backward\ secrecy, member\ authentication\}$,
- $S_{cdc,2.3} = S_{cdc,1} \cup \{backward\ secrecy, instant\ rekey\}$,
- $S_{cdc,3.1} = S_{cdc,2.2} \cup \{message\ confidentiality\}$,
- $S_{cdc,3.2} = S_{cdc,2.3} \cup \{forward\ secrecy\}$,
- $S_{cdc,4} = S_{cdc,3.2} \cup \{message\ confidentiality\}$,
- $S_{cdc,5.1} = S_{cdc,2.1} \cup \{backward\ secrecy, forward\ secrecy, instant\ rekey\}$,
- $S_{cdc,5.2} = S_{cdc,4} \cup \{member\ authentication\}$,
- $S_{cdc,5.3} = S_{cdc,4} \cup \{compromise\ robustness\}$,

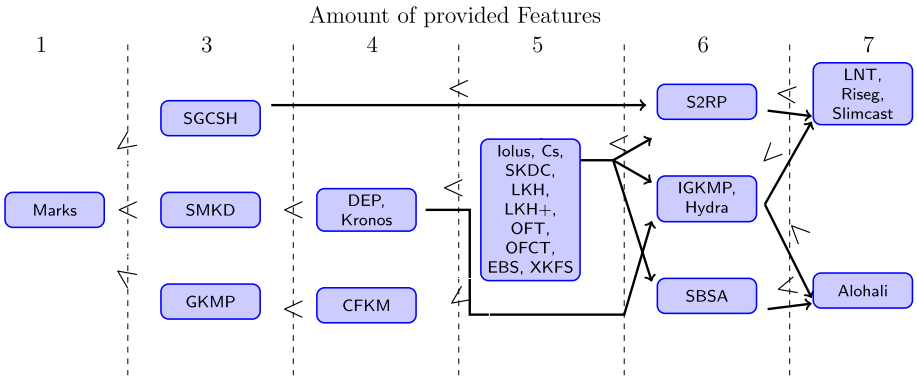

Fig. 18 Arrangement of the centralized and decentralized/hybrid SGC schemes based on the number and subset relationships of the features provided. The "<" operator means that the feature set of A is a true subset of the feature set of B if "Schema A < Schema B." For schemes which are in the same box, their feature sets are identical

- $S_{cdc,6.1} = S_{cdc,5.1} \cup \{member\ authentication\}$,
- $S_{cdc,6.2} = S_{cdc,5.2} \cup \{compromise\ robustness\}$,

Now that the security levels are defined for the case when a CI is present, we can start designing the cross-category decision tree, which considers both features and performance. For clarity, we have split this tree into several graphs, with the start of the tree shown in Fig. 19. As mentioned before, the decision tree first analyzes whether a CI should be present or not. If no, the further subtree corresponds to the decision tree for selecting a distributed SGC scheme from Fig. 11, if both performance and features are to be considered. If yes, only centralized and decentralized schemes are still considered, and the decision tree first determines what level of security S_{cgc} the use case requires. If S_{cgc} must be > 5, then only the LNT, RiSeG, Slimcast, and Alohali schemes are available for selection. This selection corresponds exactly to the schemes that form security level 6.1 and 6.2 for the decentralized SGC schemes. Therefore, the further subtree for the case that S_{cgc} must be > 5 corresponds to the subtree for the selection of a dencentralized scheme when both performance and features are to be considered and the security level S_{dis} must be greater than 6 from Fig. 15.

If the security level S_{cgc} only needs to be greater than 4, the selection expands to the LNT, RiSeG, Slimcast, Alohali, S2RP, SBSA, IGKMP, and Hydra schemes. To make a selection from these schemes, the decision tree first determines whether the cryptography used must be minimal and then whether the life cycles should be limited or unlimited. If the cryptography must be minimal, this limits the selection to the schemes Slimcast, Alohali, S2RP, IGKMP and Hydra. If the life cycles are also to be limited, only the S2RP scheme remains to be selected. If you want the *life cycle* to be unlimited, you have to choose between the schemes Slimcast, Hydra, Alohali and IGKMP. Since of these schemes Alohali only supports group creation, but the other schemes also support adding and removing members, the decision tree next asks whether the use case requires that only one group be created. If yes, Alohali is the only possible choice and if no, IGKMP is clearly the best remaining choice. In the case where the cryptography does not need to be minimal, the original schemes LNT, RiSeG, Slimcast, Alohali, S2RP, SBSA, IGKMP, and Hydra are still available for selection. Analogous to the minimal case, the use case is next restricted based on whether the *life cycle* must be limited or not. If yes, then again only S2RP remains as the only scheme. If not, the schemes LNT, RIseg, Slimcast, Alohali, SBSA, IGKMP and Hydra are available for selection. Since all of these schemes, except Alohali, support adding and removing members in addition to creating a group, the decision tree further narrows down the use case as to which group operations should be supported. If only the creation of a group should be possible, Alohali is the only choice. If additionally removing and adding should be supported, only the schemes IGKMP, Riseg and SBSA remain for selection, since the following rankings apply: $F_{perf}(\text{LNT}) > F_{perf}(\text{Slimcast}) > F_{perf}(\text{IGKMP})$. Since IGKMP is the only one of the remaining schemes that requires only constant storage cost for both CI and group members, the decision tree is next constrained to determine whether the use case requires constant storage cost. If yes, then IGKMP can be recommended directly, if no, then the computational costs of the CI for creating a group are considered next. If these are only allowed to be constant, only the scheme Riseg remains. If no, the final choice is made on the basis of the question how the calculation costs of the members for the group creation may behave. If these costs may only be linear, IGKMP is clearly the best scheme. If, on the other hand, these costs may also be linear, then Riseg is the best choice.

If the security level S_{cgc} only needs to be greater than 3, the schemes LNT, Riseg, Slimcast, Alohali, S2RP, IGKMP, Hydra, SBSA, Iolus, CS, SKDC, LKH, LKH+, OFT, OFCT, EBS, XFKS, Dep, Kronos, and CFKM are basically available. The corresponding partial decision tree is illustrated in Fig. 20. In this decision tree, we first analyze whether the use case requires the cryptography used to be minimal or not. If so, we narrow the choice down to the schemes Slimcast, Alohali, S2RP, IGKMP, Hydra, Iolus, CS, LKH, and LKH+, and next consider again the use case requirements in terms of life cycles. If the use case requires that the *life cycle* be

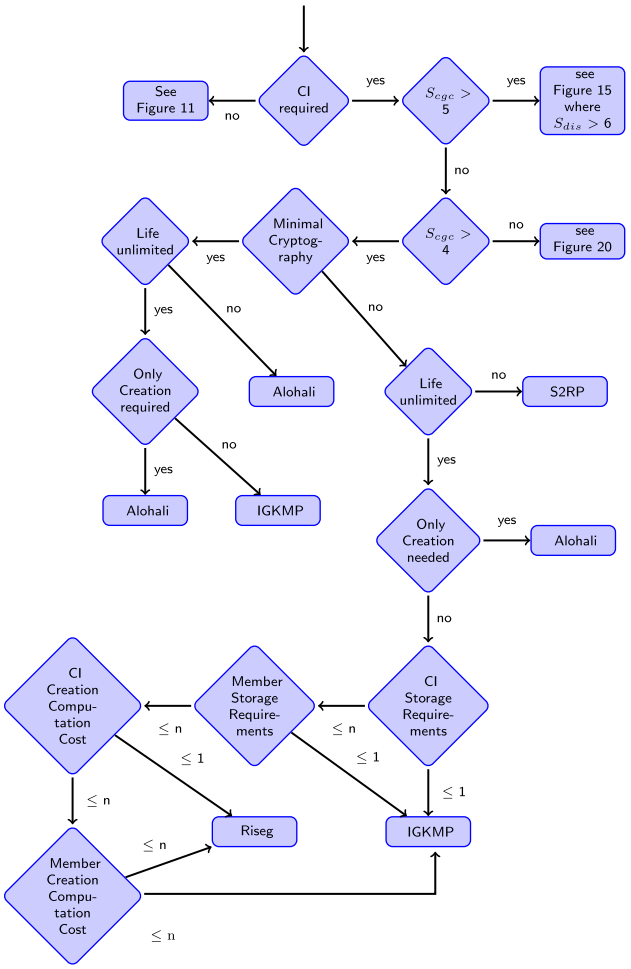

Fig. 19 Cross-category decision tree for the selection of a SGC schemes if both performance and features are considered (Part 1/5)

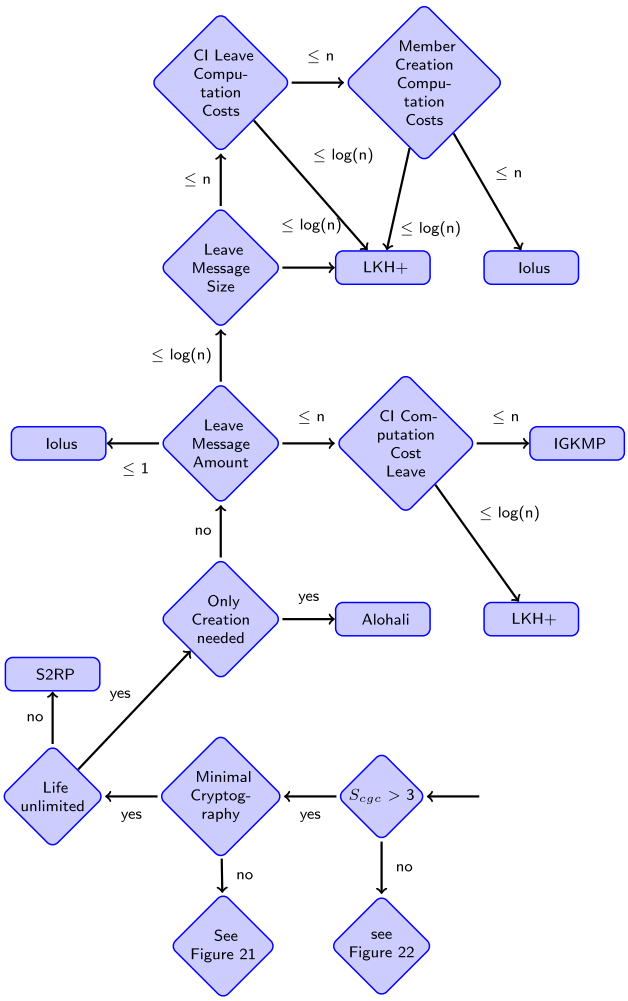

Fig. 20 Cross-category decision tree for the selection of a SGC schemes if both performance and features are considered (Part 2/5)

limited, then S2RP is the only scheme to consider. If not, the choice has been narrowed further so that the S2RP scheme is no longer available for selection. Of the schemes left to choose from, the Alohali scheme stands out because, unlike the other remaining schemes, it only allows you to create one, not add and remove members. Therefore, the decision tree next asks which group operations should be supported. If only group creation should be possible, then Alohali is the only scheme in question. If, in addition to group creation, the addition and removal of members should also be possible, a selection must be made from the schemes Slimcast, IGKMP, Hydra, Iolus, CS, LKH and LKH+. Since it additionally holds that $F_{perf}(LKH) > F_{perf}(LKH+)$, $F_{perf}(Slimcast) = F_{perf}(CS) > F_{perf}(Hydra) > F_{perf}(IGKMP)$ this selection can be further restricted to the schemes Iolus, IGKMP and LKH+. The final selection from these schemes is made based on performance, for which the next question is what the use case requires in terms of the number of messages needed for removal. If this is only allowed to be constant, then only Iolus is available for selection. If this may be also logarithmic, then LKH+ is recommended if the use case requires it that either the message sizes may be maximally logarithmically large, the computation costs of the CI for the removing may be only logarithmically large or the computation costs of the members for the group production may be only logarithmically large. If, on the other hand, all these costs just mentioned may also be linearly large, then Iolus is clearly the best choice.

If the number of messages required for removal may also be linear, then LKH+ is recommended if the computational cost of CI may be logarithmic at most, and IGKMP otherwise. The case where the cryptography used must be non-minimal and the corresponding decision tree is shown in Fig. 21 for a better overview. In this case, the decision tree first determines again whether the *life cycle* should be limited. If yes, then S2RP is the only choice. If no, then a choice must be made from the schemes LNT, Riseg, Slimcast, Alohali, IGKMP, Hydra, SBSA, Iolus, CS, SKDC, LKH, LKH+, OFT, OFCT, EBS, XFKS, Dep, Kronos, and CFKM. Since of these

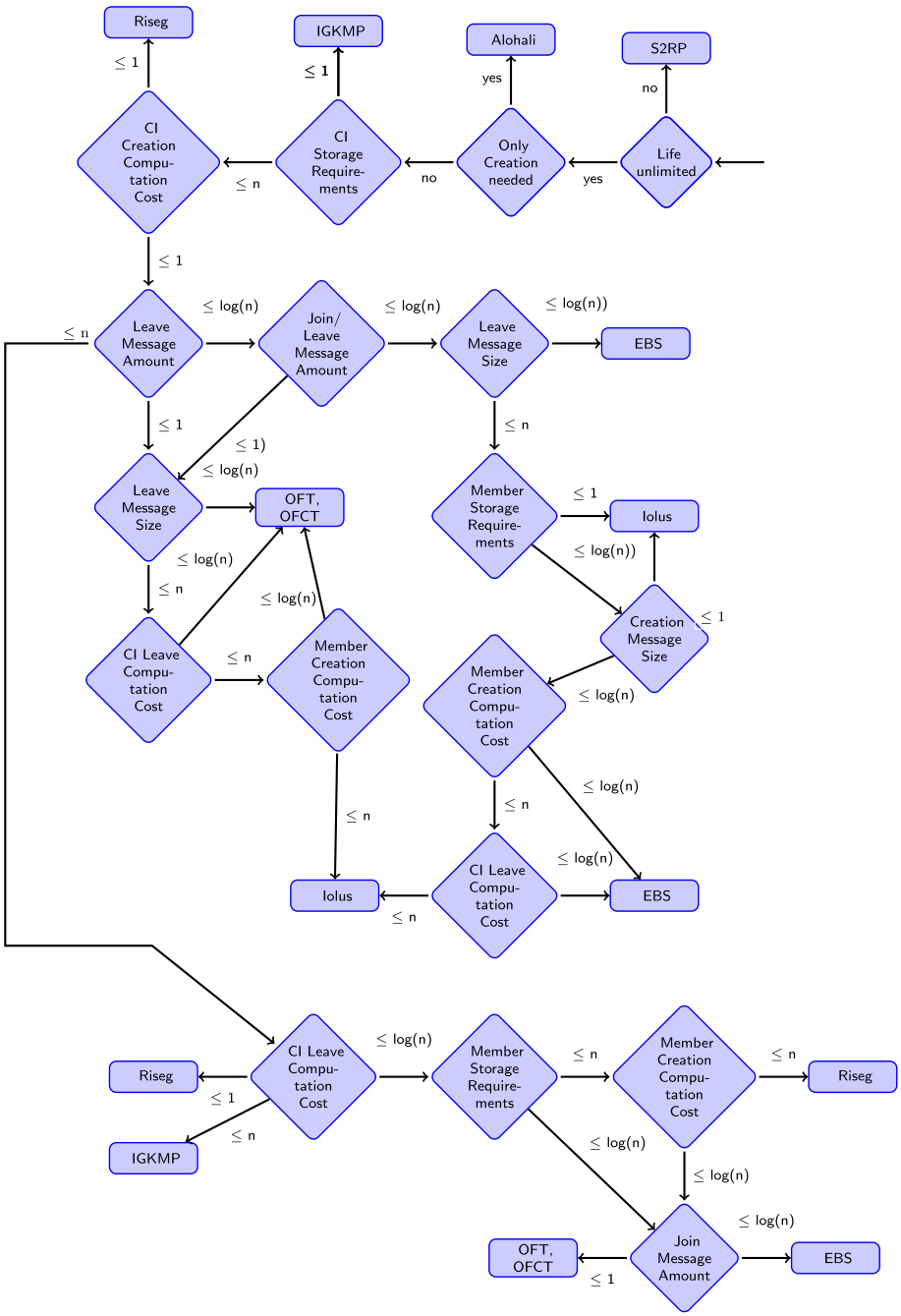

Fig. 21 Cross-category decision tree for the selection of a SGC schemes if both performance and features are considered (Part 3/5)

schemes, again all schemes except Alohali allow addition and removal of members in addition to group creation, the decision tree next further restricts the use case as to which group operations should be allowed. If only the creation of a group is to be possible, then only Alohali remains, if in addition also the removing and adding of members is to be possible, schemes LNT, Riseg, Slimcast, IGKMP, Hydra, SBSA, Iolus, CS, SKDC, LKH, LKH+, OFT, OFCT, EBS, XFKS, Dep, Kronos, and CFKM are available for selection. However, since $F_{perf}(LKH) > F_{perf}(EBS)$, $F_{perf}(LNT) > F_{perf}(Hydra) > F_{perf}(IGKMP)$ and $F_{perf}(Slimcast) > F_{perf}(Hydra)$ and $F_{perf}(CS) > F_{perf}(Hydra)$ and $F_{perf}(XKFS) > F_{perf}(IGKMP)$, the selection can be further restricted directly to the OFT, OFCT, Riseg, IGKMP, Iolus, and EBS schemes. To make a selection from these schemes, their performance properties are again used. It is noticeable that IGKMP is the only one with only constant memory requirements for CI. Therefore, the next step is to restrict the use case as to whether the memory requirements for the CI may be constant at most or linear. In the former case, IGKMP can be recommended directly. In the latter case, the use case must be further constrained. Since Riseg is the

only scheme that has only constant computational costs for the CI during group creation, this aspect is used as the next constraint criterion. If the use case requires that these costs may only be constant, then only the Riseg scheme remains. If, on the other hand, these costs may also be linear, the decision tree must further specify the use case. There are several possible ways to do this at this point, we chose to consider the number of messages for removal as the next criterion in the hope that this would keep the further decision tree as small as possible. In the case that only a constant number of messages is allowed for removal, only the schemes OFT, OFCT and Iolus are left to choose from, where OFT and OFCT are identical with respect to F_{perf}. The scheme Iolus is recommended, if in addition the size of the messages for removal may be linearly large, the computational cost of the CI for removal may be linearly large and the computational cost of the members for creation may be linearly large. If, on the other hand, these costs are allowed to be at most logarithmically large, OFT or OFCT is clearly the better choice. If the removal permits logarithmically many messages, the scheme EBS is available for selection in addition to the schemes OFT, OFCT and Iolus. The choice can be traced back to the selection between OFT, OFCT and Iolus mentioned directly before, for the case that the number of messages for adding and removing may only be constant. If, on the other hand, a logarithmic number of messages may also be sent, we consider as the next containment criterion whether the message size for removal may be at most logarithmic or linear. This is because in the former case, only the scheme EBS remains. In the second case, the selection can still be restricted to the schemes EBS and Iolus, since from this point on $F_{perf}(OFT) = F_{perf}(OFCT) > F_{perf}(EBS)$ holds. Here we recommend Iolus if the storage requirements for the members must be constant, or if the storage requirements for the members may be logarithmic and in addition the messages for creation may be constant size only, or if the storage requirements for the members may be logarithmic and the messages for creation may be logarithmically large for this and also the computational costs of the members for creation and the CI for removal may be linearly large. Otherwise, EBS is recommended. If the removal may need, however, linearly many messages, the schemes OFCT, OFT, Riseg, IGKMP, Iolus and EBS are further to the selection. By the further question about the computation costs of the CI for the removal the scheme Riseg can be recommended directly, if these may be only constant and the scheme IGKMP, if these may be linearly large. However, if they are only allowed to be logarithmically large, the use case must be further restricted. To do this, we next consider the storage requirements on the members. If these are only allowed to be logarithmic, then only OFT, OFCT and EBS are left to choose from. Here OFT and OFCT are recommended, if in addition the number for the addition may be only constant and otherwise EBS. If the storage requirement of the members may be also linear, then Riseg is recommended, if the computation costs of the members for the creation may be linear. If these costs may be only logarithmic, the further subtree corresponds to the case that the memory requirements for the members were allowed to be only logarithmic.

If the security level S_{cgc} only has to be greater than 2, the schemes Dep, Kronos and CFKM are added. The corresponding subtree is illustrated in Fig. 22. However, the two schemes Dep and Kronos can be excluded directly, since they have non-minimal cryptography as well as unlimited *life cycle* and are identical to Slimcast with respect to F_{perf}. Since Slimcast additionally has minimal cryptography and $S_{cgc} > 4$ was not recommended in the previous case, it is obvious that now Dep and Kronos are not recommended either. The scheme CFKM is identical to LKH+ in terms of F_{perf}, *life cycle*, and cryptography used. Therefore, the subtree for the case $S_{cgc} > 2$ is identical to the case $S_{cgc} > 3$ except that now that LKH+ is recommended, CFKM is also recommended.

In the case that the safety level S_{cgc} must only be greater than 1, the schemes SGCSH, SMKD and GKMP are added. Thus, the schemes SGCSH, SMKD, GKMP, Dep, Kronos, CFKM, Iolus, CS, SKDC, LKH, LKH+, OFT, OFCT, EBS, XKFS, S2RP, IGKMP, Hydra, SBSA, LNT, Riseg, Slimcast and Alohali are available. The selection from these schemes is made in the decision tree in Fig. 22. In this tree, we first determine whether the cryptography used must be minimal or not. If yes, we restrict the selection to the schemes Slimcast, Alohali, S2RP, IGKMP, Hydra, Iolus, CS, LKH, LKH+, CFKM, SGCSH and SMKD. Next, the use case is constrained as to whether the *life cycle* should be limited or not. If this is the case, only the S2RP and SGCSH schemes are available for selection. S2RP is recommended, if the removal may need only logarithmically many messages and SGCSH, if the removal may need also linearly many messages. If the *life cycle* are not to be limited, the schemes Slimcast, Alohali, IGKMP, Hydra, Iolus, CS, LKH, LKH+, CFKM and SMKD are available. Of these schemes, since the Alohali scheme supports only group creation and the SMKD scheme supports only group creation and addition, but the remaining schemes support creation and addition and removal, the decision tree uses the question of which group operations to support as the next enclosure criterion. Alohali is recommended if only the creation of a group should be supported and SMKD if the creation of a group and the addition of members should be possible. If all group operations are to be supported, the further decision tree is analogous to the case where the security level S_{cgc} must be > 2, the cryptography used must be minimal, and the *life cycle* must be unbounded, and not only group creation is to be supported. If the cryptography does not have to be minimal, the schemes SGCSH, SMKD, GKMP, Dep, Kronos, CFKM, Iolus, CS, SKDC, LKH, LKH+, OFT, OFCT, EBS, XKFS, S2RP, IGKMP, Hydra, SBSA, LNT,

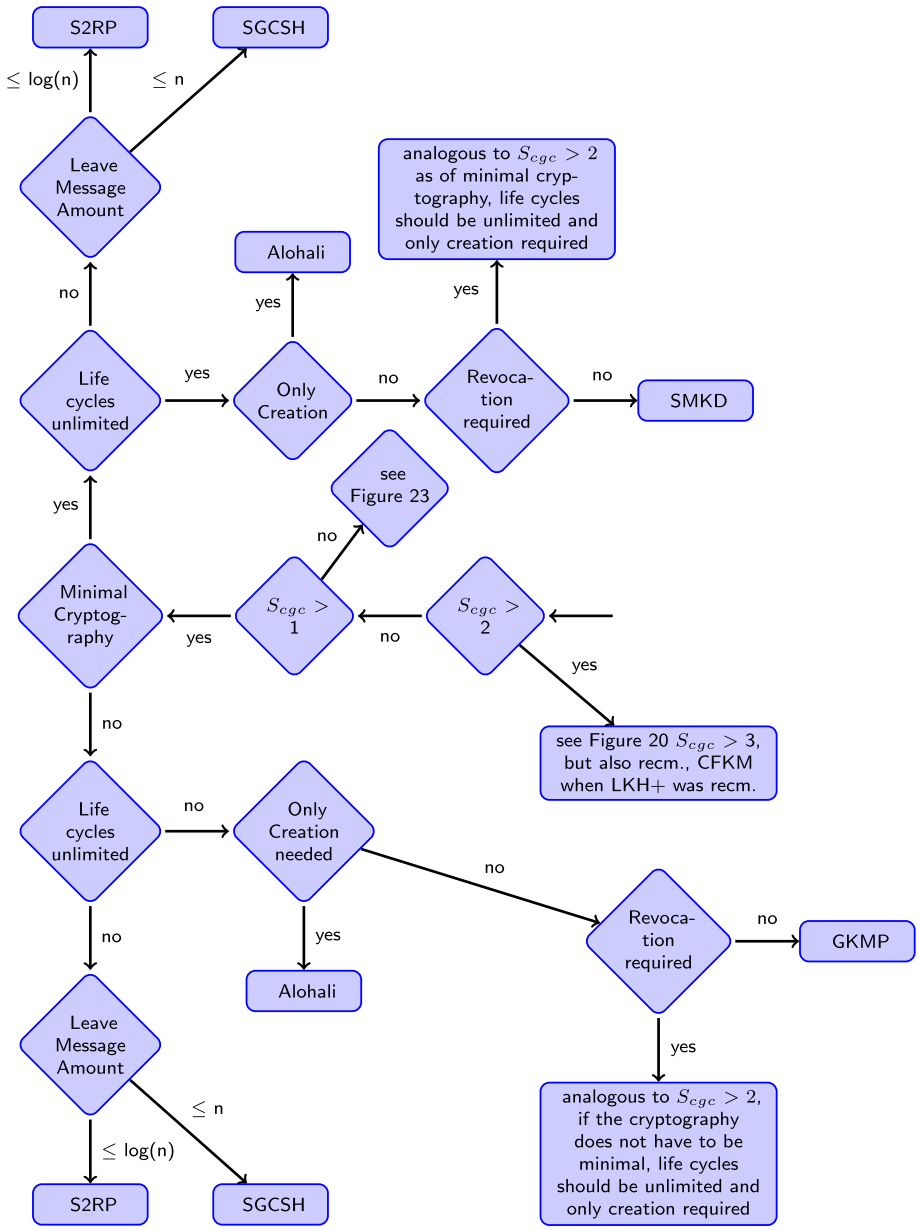

Fig. 22 Cross-category decision tree for the selection of a SGC schemes if both performance and features are considered (Part 4/5)

Riseg, Slimcast, and Alohali are still available for selection. The rest of the subtree is mostly analogous to the case where the cryptography must be minimal. The only two differences are that GKMP is now recommended instead of SMKD and that the reference $S_{cgc} > 2$ now naturally assumes that the cryptography is not minimal.

If the security level S_{cgc} is no longer important, only the Marks scheme is added compared to $S_{cgc} > 1$. The decision tree associated for this case is illustrated in Fig. 23. Since Marks only supports group creation and addition, but not removal, the case where the security level S_{cgc} does not matter is mostly analogous to the case $S_{cgc} > 1$. The only differences can occur if only in cases where the use case requires that no removal should be supported. In this case, Marks is now the best candidate for the case where the cryptography used must be minimal and the *life cycle* must be unbounded. There are no effects on the case that the cryptography does not have to be minimal, since the SMKD scheme is also available there, which is better with respect to F_{perf}.

8 Discussion of the selection guidelines

After we have created the category-specific and cross-category guidelines for the selection of an SGC scheme, we want to discuss the created decision trees. To do so, we address the quality and extensibility of the decision trees we created. In order to evaluate the quality of our decision

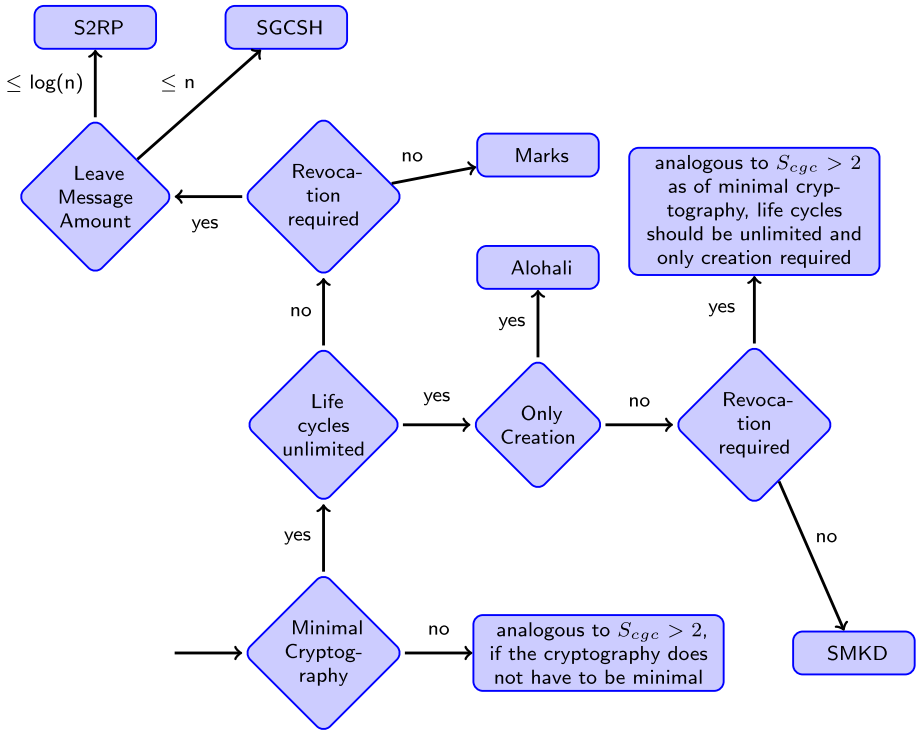

Fig. 23 Cross-category decision tree for the selection of a SGC schemes if both performance and features are considered (Part 5/5)

trees, we face the problem that there are no other works in the literature that break down the performance and features in as much detail as we did in order to create appropriate decision trees based on this information. For this reason, we can also not compare these. Thus, we decided to use the "naive" process of selecting an SGC scheme as a baseline and compare ourselves against this baseline. To show how the naive approach works, we consider a simplified example that attempts to propose a scheme based on three criteria. The associated decision tree would ask the user in turn which of the three criteria is most important to him, second most important, and least important. Thus, a total of 6 different use cases would have to be distinguished. Generalized, this means for the naive approach that it must distinguish $C!$ many use cases, where C stands for the number of selection criteria to be considered.

To evaluate the quality of our decision trees, we first consider the category-specific trees and then the cross-class trees. The number of use cases per class, which have to be considered for the recommendation of a scheme, are listed in Table 16. We distinguish in this table whether only its performance or features are decisive for the recommendation of a scheme, or whether both are to be taken into account. Table 16 shows that we never have to distinguish between more than 70 use cases per class. To be able to classify the required number of use cases to distinguish, we consider the naive approach to propose a scheme. For centralized and hybrid schemes, the naive approach must distinguish 14!, 8!, and 22! many use cases, respectively, if the selection is performance-driven, feature-driven or performance and feature-driven, respectively. For distributed schemes, analogously, 15!, 8!, and 23! many use cases would have to be considered, which is due to the fact that, in contrast to the centralized and hybrid schemes, the number of rounds is not always constant for decentralized schemes and thus represent an additional selection criterion. Thus, we can draw the following conclusion: Based on the identified Pareto fronts, the number of use cases to be considered for recommending an SGC scheme can be limited to the extent that less than 1% of the use cases that the naive approach would have to consider are needed in our decision trees.

After discussing the class-specific decision trees, we next consider the cross-class decision trees. Our corresponding performance, feature and performance, and feature-driven decision trees must distinguish 42, 6, and 92 use cases in total, respectively. The naive approach would analogously have to distinguish 16!, 9!, and 24! many use cases, respectively. Thus, our cross-class decision trees must consider less than 1% of the use cases that the naive approach would have to consider in any case. Overall, we conclude the following for the quality of our decision trees: since our decision trees have to consider less than 1% of the use cases than the respective baseline we have achieved a significant reduction in the decision space, which speaks for the quality of the decision trees we build.

Having discussed the quality of the designed decision trees, we now discuss their extensibility. To evaluate the extensibility of our decision trees, we look more closely at

Table 16 Amount of use cases to be considered for recommending an SGC scheme per class and selection criteria

SGC class	Performance driven	Feature driven	Performance and feature-driven
Centralized	5	5	22
Distributed	23	4	62
Hybrid	8	2	43

Table 17 Delimitation form related work

	[5]	[15]	[25]	[26]	[27]	[28]	[29]	[10]	We
Systematic comparison	×	×	×	×	×	×	×	✓	✓
Computation cost per group operation	×	×	×	×	×	×	×	×	✓
Storage requirements	✓	✓	✓	✓	✓	✓	✓	✓	✓
Communication cost per group operation	×	×	×	×	×	×	×	×	✓
Forward, backward secrecy	✓	✓	✓	✓	✓	✓	✓	✓	✓
Key update frequency	✓	×	✓	×	×	×	×	✓	✓
Instant rekey	×	×	×	×	×	×	×	✓	✓
Member authentication	✓	✓	×	✓	×	×	×	✓	✓
Message confidentiality	✓	✓	×	×	×	×	×	✓	✓
Message integrity	✓	✓	×	×	×	×	×	✓	✓
Compromise robustness	✓	✓	×	×	×	×	✓	✓	✓
Group independence	×	✓	×	×	×	×	×	✓	✓
Performance-based selection guidelines	×	×	×	×	×	×	×	✓	✓
Feature-based selection guidelines	×	×	×	×	×	×	×	✓	✓
Performance- and Feature-based selection guidelines	×	×	×	×	×	×	×	×	✓

what would happen if an additional scheme is to be considered in the selection process. For this scenario, four cases can occur. (1) The new scheme is not part of the newly determined Pareto front. In this case, an already considered scheme is always better than the new scheme, so it does not matter for the design of the decision trees. Therefore, in this case, our decision trees remain valid. (2) The new scheme is part of the Pareto front, but coincides with an already existing point P in the front. In this case, we only need to update our decision trees so that when the schemes that form the point P are recommended, the new scheme is also recommended.

(3) The new scheme forms a new point in the Pareto front, does not coincide with any existing point, and does not remove any existing point in the front. In this case, we can also use our decision trees and extend them by performing an additional analysis on each leaf of the tree to determine whether the previously proposed scheme is better than the new scheme in the given situation.

(4) The new scheme creates a new point in the Pareto front and removes an existing point. This means that the new scheme is always better than the schemes of the displaced point. In this case, we can build back on our decision trees and can simply recommend the new scheme wherever the schemes of the removed point were recommended. For the leaves of the decision trees where no removed scheme was recommended, we have to analyze again whether in the respective situation the new scheme is better than the previously proposed scheme.

Thus, we conclude that our decision trees are not only relevant for the schemes we consider, but that they can also be extended to include new schemes. This underpins the extensibility of the decision trees we have created.

9 Related work

In this section, we review related work and highlight the novelty of our contributions. To this end, we consider Table 17, which provides an overview of related work. These works consist of surveys that present an overview of SGC schemes. From this table, it is immediately apparent that only our previous survey [10] even addresses the question of how to support or structure the selection process of an SGC scheme. However, the table also shows that our previous work [10] only provided guidelines for cases when the selection of a scheme depends only on performance, or only on features. In addition, our previous work only captured the computational costs in total and did not break them down by the particular group operation. Similarly, the communication costs were not broken down by group operation and only the total data to

be transmitted was considered, without analyzing it in more detail in terms of message volume and message size. Compared to our previous work [10], we now have (1) a guideline that considers both performance and features in addition to the purely performance- or feature-oriented guidelines and (2) a more detailed breakdown of the computation and communication costs with respect to each group operation and the amount and size of messages, which is also be taken into account accordingly in the creation process of the guidelines.

The rest of the related work from Table 17 focuses on providing overviews of the performance and features of SGC schemes and does not provide any selection aids. Table 17 shows that our work not only provides appropriate selection aids, but also contains much more detailed overviews. For example, our work includes information on forward and backward secrecy, key updates frequency, instant rekey, member authentication, message confidentiality, message integrity, compromise robustness, and group independence for all schemes. The only other work that provides all these pieces of information for each of the schemes is our previous work [10]. The aspects of computation and communication costs are also already considered in the related work, but only our work breaks down these costs for all schemes with respect to the group operations, and only our work includes for the communication costs not only the pure amount of data that has to be transmitted, but also the number and size of the messages needed for each group operation.

10 Conclusion

In 2016, an Internet-of-Things (IoT) device-based DDoS attack brought down various services such as Netflix and Spotify. This attack was made possible because of the lack of security in developing these devices and the lack of consideration for security during their installation. As the number of installed IoT devices is expected to increase worldwide, so does the potential threat and the importance of securing these devices and their communications. In contrast to traditional 1-to-1 communication encryption, the n-to-n communication used by IoT devices is more challenging to encrypt. To overcome this, secure group communication (SGC) schemes can be used to encrypt messages for a group of recipients efficiently. The choice of SGC method for a particular use case is critical, as there are a large number of proposed methods in the literature. They differ in architecture, workflow, security features, and performance, making it difficult to determine the best procedure for a particular use case. In this paper, we study 34 SCG methods in terms of their computational and communication costs and their security features, resulting in 24 performance and security characteristics. Based on this information, we provide an overview and guidelines for selecting an SGC scheme by modeling the selection process as a multi-objective problem and using decision trees to prioritize the objectives.

As future work, we plan to extend our work to attribute-based encryption schemes. For this purpose, we first want to provide corresponding overviews of the performance and properties of attribute-based schemes in order to design corresponding selection aids. After analyzing attribute-based schemes, we plan to compare SGC and attribute-based schemes and to design corresponding selection aids that consider both categories.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Research Data Policy and Data Availability Statements All data are either included in the paper or can be found in the sources given.

References

1. Perrone, G., Vecchio, M., Pecori, R., Giaffreda, R.: The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices (2017). https://doi.org/10.5220/0006287302460253
2. von Gravrock, E.: How 5G, AI and IoT are set to Accelerate digital Transformation (2019). https://www.forbes.com/sites/forbeslacouncil/2019/05/23/how-5g-ai-and-iot-are-set-to-accelerate-digital-transformation/68ed1eef183a
3. Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., Stiller, B.: Landscape of IoT security, vol. 44 (Elsevier, 2022). Computer Science Review
4. Cisco's annual internet report (2018–2023) forecasts huge growth for iot and m2m; tepid growth for mobile. Tech. rep., ComSoc (2020). https://techblog.comsoc.org/2020/02/20/ciscos-annual-internet-report-2018-2023-forecasts-huge-growth-for-iot-and-m2m-tepid-growth-for-mobile/

5. Cheikhrouhou, O.: Secure group communication in wireless sensor networks: A survey. J. Netw. Comput. Appl. **61**, 115 (2016). https://doi.org/10.1016/j.jnca.2015.10.011
6. Alohali, B.A., Vassilakis, V.G., Moscholios, I.D., Logothetis, M.D., A secure scheme for group communication of wireless iot devices, In: 11th International Symposium on Communication Systems. Networks & Digital Signal Processing (CSNDSP) **2018**, 1–6 (2018)
7. Prantl, T., Ten, P., Iffländer, L., Dmitrenko, A., Kounev, S., Krupitzer, C.: 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. https://doi.org/10.1109/MASCOTS50786.2020.9285948
8. Rodeh, O., Birman, K., Dolev, D.: Optimized group rekey for group communication systems (2000)
9. Waldvogel, M., Caronni, G., Sun, D., Weiler, N., Plattner, B.: The versakey framework: versatile group key management. IEEE J. Sel. Areas Commun. **17**(9), 1 (1999). https://doi.org/10.1109/49.790485
10. Prantl, T., Zeck, T., Bauer, A., Ten, P., Prantl, D., Yahya, A.E.B., Ifflaender, L., Dmitrienko, A., Krupitzer, C., Kounev, S.: A survey on secure group communication schemes with focus on iot communication. IEEE Access pp. 1–1 (2022). https://doi.org/10.1109/ACCESS.2022.3206451
11. Saleh, M., Jhanjhi, N., Abdullah, A., Saher, R.: Proposing encryption selection model for iot devices based on iot device design. In: 2021 23rd International Conference on Advanced Communication Technology (ICACT) (2021), pp. 210–219. https://doi.org/10.23919/ICACT51234.2021.9370721
12. Khan, I.H., Javaid, M.: Role of internet of things (iot) in adoption of industry 4.0. J. Ind. Integr. Manag. **7**(04), 515 (2022)
13. Saravanan, G., Parkhe, S.S., Thakar, C.M., Kulkarni, V.V., Mishra, H.G., Gulothungan, G.: Implementation of iot in production and manufacturing: An industry 4.0 approach. Mater. Today Proc. **51**, 2427 (2022)
14. Noordin, N.A.: In Sustainable Development Through Data Analytics and Innovation: Techniques, Processes, Models, Tools, and Practices, pp. 97–111. Springer, Berlin (2022)
15. Sakarindr, P., Ansari, N.: Security services in group communications over wireless infrastructure, mobile ad hoc, and wireless sensor networks. IEEE Wirel. Commun. **14**(5), 8 (2007). https://doi.org/10.1109/MWC.2007.4396938
16. Prantl, T., Zeck, T., Bauer, A., Ten, P., Prantl, D., Yahya, A.E.B., Ifflaender, L., Dmitrienko, A., Krupitzer, C., Kounev, S.: A survey on secure group communication schemes with focus on iot communication. IEEE Access **10**, 99944 (2022). https://doi.org/10.1109/ACCESS.2022.3206451
17. Garg, S.K.: Cryptography using xor cipher, Research. J. Sci. Technol. **9**(1), 25 (2017)
18. Kowalczyk, C.: Crypto-it: Symmetric ciphers. http://www.crypto-it.net/eng/symmetric/index.html. Last accessed: 2020-09-02
19. Goldreich, O.: Foundations of Cryptography, vol. 2. Cambridge University Press Cambridge, Cambridge (2004)
20. Li, N.: Research on diffie-hellman key exchange protocol. In: 2010 2nd International Conference on Computer Engineering and Technology, vol. 4 (IEEE, 2010), vol. **4**, pp. V4–634
21. Bos, J.W., Halderman, J.A., Heninger, N., Moore, J., Naehrig, M., Wustrow, E.: Elliptic curve cryptography in practice. In: International Conference on Financial Cryptography and Data Security (Springer, 2014), pp. 157–175
22. Kowalczyk, C.: Crypto-it: Pseudorandom generator (prg). http://www.crypto-it.net/eng/theory/pseudorandom-generator.html. Accessed on 1 Aug 2020
23. Kowalczyk, C.: Crypto-it: Pseudorandom functions and permutations. http://www.crypto-it.net/eng/theory/prf-and-prp.html. Accessed 1 Aug 2020
24. Tušar, T., Filipič, B.: Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. IEEE Trans. Evol. Comput. **19**(2), 225 (2014)
25. Rafaeli, S., Hutchison, D.: A survey of key management for secure group communication. ACM Comput. Surv. **35**, 309 (2003). https://doi.org/10.1145/937503.937506
26. Xiao, Y., Rayi, V.K., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor networks. Comput. Commun. **30**(11), 2314 (2007). https://doi.org/10.1016/j.comcom.2007.04.009. www.sciencedirect.com/science/article/pii/S0140366407001752
27. Mapoka, T.T.: Group key management protocols for secure mobile multicast communication: A comprehensive survey. Int. J. Comput. Appl. **84**, 28 (2013). https://doi.org/10.5120/14629-2985
28. Jiang, B., Hu, X.: A survey of group key management. In: 2008 International Conference on Computer Science and Software Engineering, vol. 3 (2008), vol. 3, pp. 994–1002
29. Li, S., Wu, Y.: A survey on key management for multicast. In: 2010 Second International Conference on Information Technology and Computer Science (2010), pp. 309–312
30. Ballardie, T., Crowcroft, J.: Multicast-specific security threats and counter-measures, In: Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95) (IEEE Computer Society, USA, 1995), SNDSS '95, p. 2. https://doi.org/10.5555/526950.830436
31. Harney, H., Muckenhirn, C.: Rfc2094: Group key management protocol (gkmp) architecture. RFC Editor. USA (1997). https://doi.org/10.17487/RFC2094
32. Harney, H., Muckenhirn, C.: Rfc2093: Group key management protocol (gkmp) specification (1997). https://doi.org/10.17487/RFC2093
33. Kim, Y., Perrig, A., Tsudik, G.: Simple and fault-tolerant key agreement for dynamic collaborative groups. In: Proceedings of the 7th ACM Conference on Computer and Communications Security (Association for Computing Machinery, New York, NY, USA (2000), CCS '00, pp. 235–244. https://doi.org/10.1145/352600.352638
34. Kim, Y., Perrig, A., Tsudik, G.: Tree-based group key agreement. ACM Trans. Inf. Syst. Secur. **7**(1), 60 (2004). https://doi.org/10.1145/984334.984337
35. DeCleene, B., Dondeti, L., Griffin, S., Hardjono, T., Kiwior, D., Kurose, J., Towsley, D., Vasudevan, S., Zhang, C.: 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), vol. 1, pp. 113–117 vol.1
36. Wong, C.K., Gouda, M., Lam, S.S.: Secure group communications using key graphs. IEEE/ACM Trans. Netw. **8**(1), 16 (2000)
37. Dondeti, L.R., Mukherjee, S., Samal, A.: A distributed group key management scheme for secure many-to-many communication (1999)
38. Mittra, S.: Iolus: A framework for scalable secure multicasting. In: Proceedings of the ACM SIGCOMM '97 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (Association for Computing Machinery, New York, NY, USA, 1997), SIGCOMM '97, pp. 277–288. https://doi.org/10.1145/263105.263179
39. Tubaishat, M., Yin, J., Panja, B., Madria, S.: A secure hierarchical model for sensor network. SIGMOD Rec. **33**(1), 7 (2004). https://doi.org/10.1145/974121.974123

40. Briscoe, B.: 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 (Springer, Berlin, Heidelberg, 1999), vol. 1736. https://doi.org/10.1007/978-3-540-46703-8_19
41. Sherman, A.T., McGrew, D.A.: Key establishment in large dynamic groups using one-way function trees. IEEE Trans. Software Eng. **29**(5), 444 (2003)
42. Zhang, W., Cao, G.: 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), vol. 1, pp. 503–514 vol. 1
43. Setia, S., Koussih, S., Jajodia, S., Harder, E.: Kronos: A scalable group re-keying approach for secure multicast, In: Proceedings of the 2000 IEEE Symposium on Security and Privacy (IEEE Computer Society, USA, 2000), SP '00, pp. 215–228. https://doi.org/10.5555/882494.884414
44. Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., Pinkas, B.: 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, vol. 2, pp. 708–716 vol.2 (1999)
45. Guo, S., Shen, A.N.: A compromise-resilient pair-wise rekeying protocol in hierarchical wireless sensor networks. Comput. Syst. Sci. Eng. **25** (2010)
46. Huang, Jyh-How, Buckingham, J., Han, R.: 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. https://doi.org/10.1109/SECURECOMM.2005.3
47. Dini, G., Savino, I.M.: 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. https://doi.org/10.1109/MOBHOC.2006.278586
48. Wen, M., Zheng, Y.F., Ye, W.j., Chen, K.F., Qiu, W.D.: A key management protocol with robust continuity for sensor networks. Comput. Stand. Interfaces **31**(4), 642 (2009). https://doi.org/10.1016/j.csi.2008.06.005. http://www.sciencedirect.com/science/article/pii/S0920548908000937
49. Cheikhrouhou, O., Koubâa, A., Dini, G., Abid, M.: Riseg: a ring based secure group communication protocol for resource-constrained wireless sensor networks. Pers. Ubiquit. Comput. **15**(8), 783 (2011)
50. Dini, G., Savino, I.M.: Lark: A lightweight authenticated rekeying scheme for clustered wireless sensor networks. ACM Trans. Embed. Comput. Syst. **10**(4) (2011). https://doi.org/10.1145/2043662.2043665
51. Diop, A., Qi, Y., Wang, Q.: Efficient group key management using symmetric key and threshold cryptography for cluster based wireless sensor networks. Int. J. Comput. Netw. Inf. Security **6**, 9 (2014)
52. Cheikhrouhou, O., et al.: Lnt: A logical neighbor tree secure group communication scheme for wireless sensor networks. Ad Hoc Netw. **10**(7), 1419 (2012). https://doi.org/10.1016/j.adhoc.2012.03.019
53. Son, J.H., et al.: Topological key hierarchy for energy-efficient group key management in wireless sensor networks. Wireless Pers. Commun. **52**(2), 359 (2010). https://doi.org/10.1007/s11277-008-9653-4
54. Bilal, M., Kang, S.G.: A secure key agreement protocol for dynamic group. Clust. Comput. **20**(3), 2779 (2017). https://doi.org/10.1007/s10586-017-0853-0
55. Dondeti, L., Mukherjee, S., Samal, A.: Scalable secure one-to-many group communication using dual encryption. Comput. Commun. **23**(17), 1681 (2000). https://doi.org/10.1016/S0140-3664(00)00255-3
56. Burmester, M., Desmedt, Y.: A secure and efficient conference key distribution system. In: Santis, A.D. (ed.) Advances in cryptology–EUROCRYPT '94, Lecture Notes in Computer Science, vol. 950 (Springer, Berlin, 1994), vol. 950, pp. 275–286
57. Molva, R., Pannetrat, A.: Scalable multicast security in dynamic groups. In: Proceedings of the 6th ACM Conference on Computer and Communications Security (Association for Computing Machinery, New York, NY, USA (1999), CCS '99, pp. 101–112. https://doi.org/10.1145/319709.319723
58. Tygar, J., Perrig, A., Song, D., Elk, a new protocol for efficient large-group key distribution. In: IEEE Symposium on Security and Privacy (IEEE Computer Society, Los Alamitos, CA, USA **2001**, 0247 (2012). https://doi.org/10.1109/SECPRI.2001.924302
59. Steiner, M., Tsudik, G., Waidner, M.: Diffie-hellman key distribution extended to group communication. In: Proceedings of the 3rd ACM Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA (1996), CCS '96, pp. 31–37. https://doi.org/10.1145/238168.238182
60. Rafaeli, S., Hutchison, D.: 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
61. Kausar, F., Hussain, S., Park, J.H., Masood, A.: 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, Springer, Berlin, Heidelberg (2007), MSN'07, pp. 737–748
62. Becker, K., Wille, U.: Communication complexity of group key distribution. In: Proceedings of the 5th ACM Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA (1998), CCS '98, pp. 1–6. https://doi.org/10.1145/288090.288094
63. Yang, Y., Zhou, J., Deng, R.H., Bao, F.: Hierarchical self-healing key distribution for heterogeneous wireless sensor networks. In: Chen Y, Dimitriou TD, Zhou J (eds), Springer, Berlin (2009), pp. 285–295. https://doi.org/10.1007/978-3-642-05284-2_16
64. Boyd, C.: 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 (1997), vol. 1270
65. Zhu, S., Setia, S., Jajodia, S.: Leap+: Efficient security mechanisms for large-scale distributed sensor networks. ACM Trans. Sen. Netw. **2**(4), 500 (2006). https://doi.org/10.1145/1218556.1218559
66. Eltoweissy, M., Heydari, M.H., Morales, L., Sudborough, I.H.: Combinatorial optimization of group key management. J. Netw. Syst. Manage. **12**(1), 33 (2004). https://doi.org/10.1023/B:JONS.0000015697.38671.ec
67. Eltoweissy, M., Wadaa, A., Olariu, S., Wilson, L.: Group key management scheme for large-scale sensor networks. Ad Hoc Netw. **3**(5), 668 (2005). https://doi.org/10.1016/j.adhoc.2004.08.012
68. Gaddour, O., Koubâa, A., Abid, M.: 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. https://doi.org/10.1109/COMNET.2009.5373554
69. Ghafoor, A., Sher, M., Imran, M., Saleem, K.: A lightweight key freshness scheme for wireless sensor networks. In: 2015 12th International Conference on Information Technology - New Generations (2015), pp. 169–173

70. Szalachowski, P., Kim, T.H.J.: Secure broadcast in distributed networks with strong adversaries. Secur. Commun. Netw. **8**(18), 3739 (2015). https://doi.org/10.1002/sec.1296
71. Bao, X., Liu, J., She, L., Zhang, S.: 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. https://doi.org/10.1109/WCICA.2014.7053290
72. Seo, S.H., Won, J., Sultana, S., Bertino, E.: Effective key management in dynamic wireless sensor networks. IEEE Trans. Inf. Forensics Secur. **10**, 371 (2015). https://doi.org/10.1109/TIFS.2014.2375555