The Cost of Maintaining Keys in Dynamic Groups with Applications to Multicast Encryption and Group Messaging

Michael Anastos[1], Benedikt Auerbach[*] [2], Mirza Ahad Baig[1], Miguel Cueto Noval[1], Matthew Kwan[†] [1], Guillermo Pascual-Perez[1], and Krzysztof Pietrzak[1]

[1]ISTA, Austria
{michael.anastos, mbaig, mcuetono, matthew.kwan, gpascual, pietrzak}@ista.ac.at
[2]PQShield
benedikt.auerbach@pqshield.com

December 9, 2024

Abstract

In this work we prove lower bounds on the (communication) cost of maintaining a shared key among a dynamic group of users. Being "dynamic" means one can add and remove users from the group. This captures important protocols like multicast encryption (ME) and continuous group-key agreement (CGKA), which is the primitive underlying many group messaging applications.

We prove our bounds in a combinatorial setting where the state of the protocol progresses in rounds. The state of the protocol in each round is captured by a set system, with each of its elements specifying a set of users who share a secret key. We show this combinatorial model implies bounds in symbolic models for ME and CGKA that capture, as building blocks, PRGs, PRFs, dual PRFs, secret sharing, and symmetric encryption in the setting of ME, and PRGs, PRFs, dual PRFs, secret sharing, public-key encryption, and key-updatable public-key encryption in the setting of CGKA. The models are related to the ones used by Micciancio and Panjwani (Eurocrypt'04) and Bienstock *et al.* (TCC'20) to analyze ME and CGKA, respectively.

We prove – using the Bollobás' Set Pairs Inequality – that the cost (number of uploaded ciphertexts) for replacing a set of d users in a group of size n is $\Omega(d \ln(n/d))$. Our lower bound is asymptotically tight and both improves on a bound of $\Omega(d)$ by Bienstock *et al.* (TCC'20), and generalizes a result by Micciancio and Panjwani (Eurocrypt'04), who proved a lower bound of $\Omega(\log(n))$ for $d = 1$.

[*]Benedikt Auerbach conducted part of this work at ISTA.

[†]Matthew Kwan was supported by ERC Starting Grant "RANDSTRUCT" No. 101076777

Contents

1 Introduction

1.1 Membership Changes in Multicast Encryption and Continuous Group-key Agreement

Multicast encryption and CGKA. A prevalent problem in many areas of cryptography involves the agreement on a common key by a group of protocol users. This underpins communication primitives which try to achieve scalability beyond what is offered by point-to-point communication. The problem is made more interesting (and practical) if we further consider a dynamically-changing group of users, i.e., where users get added and removed to and from the group, and which thus requires an ever-evolving common key. The two main primitives capturing this problem are, arguably, Multicast Encryption (ME) and Continuous Group-Key Agreement (CGKA). The former, with constructions dating back to the 1990s, considers the problem in the presence of a central authority (CA) that has access to all secrets and is in charge of sending protocol messages to users in order to effect group membership changes. In turn, CGKA is a much newer primitive, resulting from the development of end-to-end encrypted messaging systems, such as WhatsApp or Signal, and the recent IETF standard Message Layer Security (MLS). Given that the goal of these systems is the confidentiality of the messages exchanged, the reliance on a central authority to manage key material is naturally out of the question. Instead, group members themselves are the ones who refresh the key material when membership changes take place, with communication taking place over an untrusted server. CGKA has the additional security goal of post-compromise security (PCS), which roughly states users can "heal" from a compromise, so that future keys/messages can become again secure.

As mentioned before, the main goal of these protocols is to provide scalability to large groups, and so it is important that the protocols messages are small. In particular, these should be of size sub-linear in the group size n.

Key-Trees. The main technique employed by efficient ME and CGKA constructions are so-called *key-trees*, which were first used by [WHA99, WGL00] for building multicast protocols. A key-tree is a (usually, though not necessarily, binary) tree graph where each node is associated to a key. In the case of ME, keys typically correspond to a symmetric encryption scheme, whereas for CGKA, they correspond to a key-pair of a public-key encryption scheme. Leaves in the tree are associated to users, and the root of the tree corresponds to the group key. Further, the tree can be seen as a directed graph, with edges capturing the following hierarchical relationship between the keys: knowledge of the (in the CGKA case, secret) key of the source implies knowledge of the (secret) key of the target. It is easy to see that, if we consider the key at each leaf being known to exactly the user associated to it, users know exactly the keys on the path from their leaf to the root. This is known as the *tree invariant*, and the security of the protocol can be seen as ensuring this invariant holds throughout key-material changes.

The benefit of using key-trees is that, by making use of the auxiliary keys corresponding to the internal tree nodes, key material can be refreshed and shared to the rest of the group very efficiently. For the purpose of simplicity, in the following we focus on the communication cost of replacing users, i.e., substituting one user with another, so that the group size remains constant. In practice, this is equivalent to eliminating the keys known to the removed user, including the group key, and communicating the new (freshly sampled) group key to the new user and the remainder of the group.[1] It is clear that the cost of removing a user from the group, i.e., that of communicating a new key to the remainder of the group, is similar to that of a replacement. Thus, we will indistinctly use the term *replacement* throughout the paper except where a distinction is relevant. In particular, due to the tree invariant, key-trees allow to replace (or remove) a user with a cost equal to the length of the path of the replaced (removed) user times the in-degree of the nodes in said path. Indeed, each new key for a node along the path can be communicated to all users (leaves) below it by simply encrypting it to all of its children. If we consider a binary tree, this cost is approximately $2\log(n)$.[2]

[1] In CGKA, a so-called *update* operation, designed to provide security against a potential compromise of the issuer, can also be seen as such a replacement, where the old (potentially leaked) state is replaced by a new one.

[2] Computing the amortized cost of removing users is more convoluted, since by removing the key of a node one effectively increases the in-degree of its parent. Whereas replacing this key in ME can be easily done by the central authority, it becomes an

This was improved upon by multicast protocols using a pseudo-random generator (PRG) [CGI+99] to derive the new keys along a path, reducing the communication by a factor of 2. Most CGKA protocols incorporate this technique as well, thus also allowing for a replacement with $\log(n)$ cost.

Batching replacements for ME. A natural attempt to improve on the efficiency of the above constructions would be to consider batching replacements. Indeed, if we wanted to batch d replacements, we would need to replace only $d(1+\log(n/d))$ nodes in expectation [NNL01] (those on the intersection of the paths of d uniformly random leaves), as opposed to $d\log(n)$. One would hope that this would translate to protocols with an improved communication cost. And indeed, this is the case, as shown implicitly by [NNL01] and explicitly for ME in [LYGL01, SM03], which propose protocols, using only hashing and encryption as building blocks, that allow batching dynamic operations and, in the case where only replacements are performed, can replace d users in a single round of communication with a communication cost of $d(1+\log(n/d))$. A further motivation for these works was to alleviate out-of-sync-related issues that can arise in bigger groups where re-keying becomes more frequent.

Batching replacements for CGKA. Similar approaches can be seen in CGKA constructions, where the situation is more involved due to the absence of a central authority. Indeed the main example of this approach is TreeKEM, the CGKA underlying MLS [BBR+23]. Here, d users can be replaced, concurrently in 2 rounds of communication, with a communication cost of $d(1+\log(n/d))$ at the cost of increasing the cost of future group communication.[3] A first round where all d users announce a new key for the leaf will follow by another where one of the d users will sample the new group key. In TreeKEM the keys on the paths of the other $d-1$ users will get deleted and set to null, thus preventing their usage until they are replaced by future operations. Hence, this communication complexity is only achieved under so-called "fair-weather" conditions, i.e., under beneficial sequences of operations that, e.g., contribute towards quickly replacing the removed keys. All of the subsequent protocols based on TreeKEM (see Sec. 1.3) share the same or very similar issues.

Lower bounds. Given the upper bounds highlighted above, an interesting question is whether they are optimal. The first steps in this direction were taken in works by Canetti, Malkin and Nissim [CMN99] and Snoeyink, Suri and Varghese [SSV01], both, however, making restricting assumptions on the schemes and, in particular, not allowing for the use of pseudorandom generators. Regarding single (non-batched) replacements in ME, Micciancio and Panjwani [MP04] showed, in a symbolic model in the style of Dolev and Yao [DY83], that the protocol by [CGI+99] is optimal among those built using encryption and PRGs, proving a worst-case lower bound of $\log(n)+o(1)$.

As a result of the introduction of CGKA and the big amount of constructions proposed in the last years, a new line of work proving similar lower bounds in the symbolic model has been taking shape. Particularly interesting to our setting, Bienstock, Dodis and Rösler [BDR20], in a work on the communication complexity of concurrent recovery from corruption in CGKA, implicitly prove a lower bound of d for batched user replacements, which essentially says that every new leaf key in the group must be addressed separately. In the case $d=1$ Alwen *et al.* [AAB+21] lift the bound of [MP04] to an average case bound, and further extend it to CGKAs. This work also generalizes the bound to the case of several, potentially overlapping groups. The recent [ACPP23] generalizes [BDR20] to a setting where the condition of recovery from corruption is relaxed a larger number of rounds.

Going beyond bounds in the symbolic model, Bienstock *et al.* [BDG+22], by means of a black-box separation from public-key encryption, analyze the worst-case efficiency of CGKA. The work gives a sustained lower bound that is linear in the group size, i.e., $\Omega(n)$. This is done for a sequence of operations, in which a set of users of size $\Omega(n)$ is added to the group, followed by a sequence of removals and adds of a single

issue for CGKA protocols, where removing a sizeable number of users can result in subsequent communication costs degrading to linear in n.

[3] If considering only removals, the additive term d would not be present, as this corresponds to the individual encryptions to the new users.

user per round. However, the bound does not apply to ME and is worst-case, meaning that it relies on a particular, adversarially chosen sequence of additions and removals of users.

Despite both the ME constructions above and the lower bound of [MP04] existing for roughly 20 years, neither better constructions nor a matching lower bound considering batching have been proposed since, leaving upper bounds of order $d(1+\log(n/d))$ and the implicit lower bound of d [BDR20] as the state of the art regarding batched user replacements in ME.

1.2 Our Contributions

A tight lower bound on batched user replacements. In this work we close the gap between upper and lower bounds on the communication complexity incurred by batched user replacements in multicast encryption and continuous group-key agreement. On a high level, we prove the following statement.

> In the symbolic model, consider a secure and correct ME (or CGKA) scheme. If a set of d users chosen uniformly at random from a group with n members is replaced with different users, then the protocol messages must have contained at least $\frac{\ln(2)}{3} \cdot d \cdot \log(n/d)$ ciphertexts in expectation.

In the above we allow for symmetric encryption, pseudorandom generators, (dual) pseudorandom functions, and secret sharing as building blocks for ME, and for (key-updatable) public-key encryption, pseudorandom generators, (dual) pseudorandom functions, and secret sharing in the case of CGKA. As there exist constructions of ME [LYGL01, YLZL02, SM03] and CGKA [BBR^{+}23] (the latter however only with respect to fair-weather complexity, as discussed above) that achieve a communication cost of $d(1+\log(n/d))$ our bound is tight up to a small multiplicative factor.[4] Intuitively, this shows that existing ME and CGKA constructions are optimal and, in particular, suggests that the way the removal of users is handled in the MLS standard [BBR^{+}23] cannot be improved by simple means.

We point out that our bound is an average case bound, as the set of replaced users is chosen uniformly at random (as is the case for the single user bound of [AAB^{+}21]), as opposed to ones relying on an adversarially chosen sequences of operations [MP04, ACPP23, BDG^{+}22]. Our technical statements regarding ME and CGKA (Corollaries 4.6 and A.7) take *amortized* communication complexity into account. I.e., we consider a game running over $t_{\max}$ rounds where, in every round t, a set of d_t group members, chosen uniformly at random from the current group, is replaced by new users. Then, if we denote the set of ciphertexts and keys sent in round t by $\mathtt{M}_t$, we prove that

$$\mathbb{E}\left[\sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{\ln(2)}{3} \sum_{t=1}^{t_{\max}} d_t \log\left(\frac{n}{d_t}\right).$$

Note that we cannot guarantee that $|\mathtt{M}_t| \geq (\ln(2)/3) \cdot d_t \cdot \log(n/d_t)$ for all t. This is necessary as, in principle, some of the communication required to replace the users in round t might already have happened in prior rounds, as will be discussed in greater detail below.

Proof overview. Conceptually, we follow the approach of [ACPP23], who prove lower bounds on the cost incurred by CGKA schemes recovering from corruption(s) over several rounds. That is, we decouple the combinatorial problem at the core of minimizing the cost for batched user replacements from the more technical issues that arise when arguing within the confines of the symbolic model. More precisely, our proof consists of two major parts, the first of which is common to both the case of multicast encryption and CGKA simultaneously. First, in Section 3 we capture the problem of securely replacing a batch of users in ME and CGKA in a clean, self-contained combinatorial model, and prove our lower bound within this model. The second step consists of showing that bounds in the combinatorial model imply bounds in the symbolic model. This is done for ME in Section 4 and for CGKA in Appendix A, the proofs being very similar. We discuss these steps in greater detail below.

[4] We point out that for typical use cases we have $d \ll n$. Further, the case $\log(n/d) < 1$ in which our bound is not asymptotically optimal implies $d > n/2$. In this case the linear lower bound by Bienstock, Dodis, and Rösler [BDR20] is asymptotically optimal.

The combinatorial model. Our aim with the combinatorial model is to capture in an intuitive way how the sets of users that share a secure secret evolve over time. Here, 'secret' can be thought of as a symmetric key or secret key depending on whether we want to model ME or CGKA. More precisely, for $n_{\max}, t_{\max} \in \mathbb{N}$, we let $[n_{\max}]$ be the universe of users and, for rounds $t \in [t_{\max}]$, consider a sequence of groups $G_t \subseteq [n_{\max}]$ evolving round-by-round by replacing a set of group members in every round. We then capture the secrets shared by users in each round as a sequence of set systems $\mathcal{S}_t \subseteq 2^{[n_{\max}]}$. A set $S \subseteq [n_{\max}]$ being part of $\mathcal{S}_t$ intuitively means that there exists a secret $\mathtt{r}$ with the following two properties. On the one hand, the set of users in G_t that, in round t (or any round before), are able to recover $\mathtt{r}$ from their internal states and the protocol messages sent so far is exactly given by S; and, on the other hand, $\mathtt{r}$ is secure. The latter means that, even given the protocol messages as well as the current and all prior states of every user *not* in G_t, it is not possible to recover $\mathtt{r}$.

Correctness and security of the corresponding ME or CGKA scheme impose two restrictions on $\mathcal{S}_t$. Namely, for all t we have $G_t \in \mathcal{S}_t$, which corresponds to the existence of a secure group key shared by all the members in G_t. Further, as keys known to non-members of the group are being considered insecure, it must be the case that $S \subseteq G_t$ for all $S \in \mathcal{S}_t$.

The set system $\mathcal{S}_t$ evolves over time. Removing a user u from the group leads to all secrets they had at some point access to being considered insecure. This means that all sets in $\mathcal{S}_{t-1}$ that contained u can no longer be present in $\mathcal{S}_t$. On the other hand, by sending protocol messages, new secrets can be shared with users, meaning that sets can be added to $\mathcal{S}_t$. Adding sets to $\mathcal{S}_t$, however, comes at a communication cost, since the corresponding secrets cannot be simply sent in the clear, but instead must be encrypted under (potentially multiple) secure keys already present in the system. We capture this with a cost function $\mathrm{Cost}(t)$ that, for now, can be thought of as a lower bound on the ciphertexts needed to be sent in the rounds up to t, in order to communicate the secrets corresponding to $\mathcal{S}_t$. While the definition of $\mathcal{S}_t$ can be seen as a natural generalization of the set system introduced for static groups in [ACPP23] to the setting of dynamic groups, our definition of the cost function deviates substantially from prior lower bounds in the symbolic model, and we consider it to be one of the main conceptual contributions of this work, as we discuss below.

Defining the cost function. Prior works giving lower bounds for ME or CGKA schemes in the symbolic model follow one of two different approaches for counting the number of ciphertexts sent in order to achieve both correctness and security of the scheme. [MP04] and [AAB+21] for round t use as cost function the amount of ciphertexts that were used to communicate the group key of round $t-1$, and are no longer of use in round t as they are encryptions of keys that are known by users removed from the group in round t. Each of these ciphertexts can be identified with a particular secret (which is one of the encryption keys used in the ciphertext), and thus in our abstraction this cost metric can be seen as giving a cover of the set $G_{t-1} \setminus D_t$ using sets in $\mathcal{S}_{t-1}$, where D_t denotes the set of users removed from the group in round t. Moreover, it also admits another interpretation, namely, these ciphertexts are encryptions of secrets that are known by users in D_t and this means that the cost metric can be seen as counting some of the sets removed from the set system in round t.

On the other hand, [BDR20] and [ACPP23] consider the number of ciphertexts sent in a particular round t that are necessary to communicate a new secret $\mathtt{r}$ to (some of) the group members. Note that in order to communicate $\mathtt{r}$, it must have been encrypted under secret keys already established by the scheme. Seen through the abstraction of set system $\mathcal{S}_t$, this means if the set S (corresponding to $\mathtt{r}$) is added to $\mathcal{S}_t$, it must have been covered by the union of a collection of sets in $\mathcal{S}_{t-1}$. Accordingly, one can essentially use as cost function the size of a minimum cover of S with respect to $\mathcal{S}_{t-1}$, i.e., the smallest amount of sets in $\mathcal{S}_{t-1}$ the union of which covers S. To be a bit more precise, the cover may also include singletons $\{u\}$ for all users $u \in [n_{\max}]$ (corresponding to the users' personal keys). In particular, this is relevant regarding users being added to the group which, by the rules imposed by correctness and security, cannot be part of any set in $\mathcal{S}_{t-1}$.

In this work we define $\mathrm{Cost}(t)$ taking into account both the number of removed sets and the size of a minimum cover of G_t using sets in $\mathcal{S}_{t-1}$ and singletons for all users $u \in [n_{\max}]$. Unlike [MP04] and [AAB+21], which only take into account *some* of the destroyed sets, we generalize their approach and count *every* set removed from the set system. This is motivated by the following observation. When considering a scheme in the combinatorial model, we would like to exploit that, in every round t, the group G_t must be an element of $\mathcal{S}_t$. Following the second cost metric described above, we can argue that the cost of adding the corresponding key to the set system must be at least the size of a minimum cover of G_t with respect to the prior set system $\mathcal{S}_{t-1}$. However, we consider a security experiment running over multiple rounds and do not want to impose unnecessary restrictions on $\mathcal{S}_{t-1}$. In particular, it could be the case that $\mathcal{S}_{t-1} = 2^{G_{t-1}}$ is the full power set of the prior group. Thus, if we denote the d users removed from and added to G_{t-1} by D_t and A_t, respectively, we have that $S = (G_{t-1} \setminus D_t) \in \mathcal{S}_{t-1}$, and obtain a minimum cover of the new group as

$$G_t = S \cup \bigcup_{u \in A_t} \{u\}.$$

This means that removing the users from the group comes essentially for free, and the only contribution to the cost function stems from adding the users in A_t to the group. As a consequence, using this cost metric we would end up with a cost that is linear in d, in turn recovering the bound already implicitly given in [BDR20].

Note, however, that in the example above the set system $\mathcal{S}_{t-1}$ contains a number of sets that is exponential in the group size n. Further, every set in $\mathcal{S}_{t-1}$ containing at least one of the removed users would no longer be considered secure after round t, and there is an exponential number of sets of this type. Thus, in the first cost metric discussed above, i.e., counting sets removed from the system, maintaining such a huge system would be prohibitively expensive. For this reason, in this paper we use the sum of the two prior approaches as cost metric and define

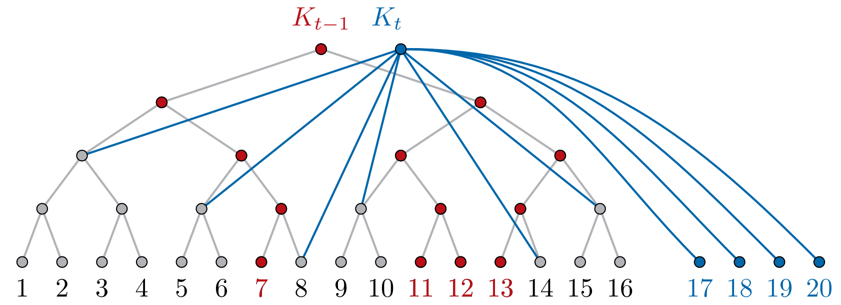
$$\mathrm{Cost}(t) = \underbrace{|\{S \in \mathcal{S}_{t-1} : S \cap D_t \neq \emptyset\}|}_{\text{sets removed from } \mathcal{S}_{t-1}} + \underbrace{\mathrm{SizeMinCov}(G_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t\})}_{\text{size of minimum cover of } G_t}.$$

For an illustration of our cost function for a concrete key-tree see Figure 1. We stress that $\mathrm{Cost}(t)$ is not to be understood as the amount of ciphertexts sent in round t, but instead as a lower bound on the ciphertexts sent up to, and including, that round. Further, our precise definition of the cost function (see Definition 3.1) also accounts for minor potential savings in terms of ciphertexts, stemming from the following two observations. On the one hand, that adding singletons to $\mathcal{S}_t$ does not necessarily require sending a ciphertext; on the other, that one ciphertext in the second summand of $\mathrm{Cost}(t)$ can be saved by deriving new keys from the output of a pseudorandom generator evaluated on the secret corresponding to one of the sets making up the minimum cover.

Lower bounding $\mathrm{Cost}(t)$ in the combinatorial model. The example discussed above suggests a trade-off between the two terms of $\mathrm{Cost}(t)$. Intuitively, the larger the set system $\mathcal{S}_{t-1}$, the cheaper it is to add G_t to $\mathcal{S}_t$. Here, the extreme case is given by the example discussed above, which essentially corresponds to preparing a key for the removal of every possible subset of G_{t-1}, and that leads to a large cost due to the first summand of $\mathrm{Cost}(t)$. The opposite extreme would be $\mathcal{S}_{t-1} = G_{t-1} \cup \{\{u\} : u \in G_{t-1}\}$ where, except for the group key, there is only a personal key for every group member. In this case any cover of G_t with respect to the previous set system would be made up of singletons and thus of size linear in $|G_t|$. Hence, to minimize the overall cost, intuitively it makes sense to balance the two components of $\mathrm{Cost}(t)$, which turns out to also be the case for the best known constructions of ME [LYGL01, YLZL02, SM03]. In these constructions, based on balanced binary trees, replacing d uniformly random group members requires in expectation to replace $\Theta(d(1 + \log(n/d)))$ keys in the system, each of which comes at the cost of one ciphertext. Further, the expected size of a minimum cover of G_t turns out to be of the same size. Accordingly, both summands of $\mathrm{Cost}(t)$ are of order $\Theta(d(1 + \log(n/d)))$.

We show that these constructions are optimal (up to a small constant factor) by roughly proving in Theorem 3.4 that, for every choice of $(\mathcal{S}_t)_{t=0}^{t_{\max}}$ satisfying the requirements of the combinatorial model, it must hold that

$$\mathbb{E}\left[\sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)\right] \geq \sum_{t=1}^{t_{\max}} d_t \ln\left(\frac{n}{d_t}\right),$$



$\{1, .., 16\},$	
$\{1, .., 8\}, \{9, .., 16\},$	
$\{5, .., 8\}, \{9, .., 12\}, \{13, .., 16\}$	$\{1, .., 4\}$
$\{7, 8\}, \{11, 12\}, \{13, 14\}$	$\{5, 6\}, \{9, 10\}, \{15, 16\}$
$\{7\}, \{11\}, \{12\}, \{13\}$	$\{8\}, \{14\}, \{17\}, \{18\}, \{19\}, \{20\}$

$\mathrm{Cost}(t) =$ # destroyed sets (red nodes) $+$ $\mathrm{SizeMinCov}(G_t)$ (blue edges)

Figure 1: Example of our cost function on a balanced binary key-graph (top). The set associated to a node (key) is given by all leaves (users) whose path to the root contains the node in question. The figure depicts users 7, 11, 12, 13 in the group $G_{t-1} = \{1, \ldots, 16\}$ being replaced by users 17, 18, 19, and 20, producing group G_t. Gray and blue nodes were already part of the system at time $t-1$, blue nodes are added at time t. Our cost function $\mathrm{Cost}(t)$ counts the sets in $\mathcal{S}_{t-1}$ no longer secure after removing the users (bottom left, corresponding to the red nodes), as well as the size of a minimum cover of the new group with respect to the remaining secure sets and the added users' personal keys (bottom right, corresponding to the blue edges), i.e., the number of ciphertexts that have to be sent in order to establish the new group key K_t. Note that what is described here is a simplification ignoring possible optimizations and special cases (that our formal model does capture).

where d_t denotes the amount of users replaced in round t and the set D_t of users replaced is sampled uniformly at random in every round. In the proof we consider two families $(X_{D_t})_{D_t \subseteq G_{t-1}}$ and $(Y_{D_t})_{D_t \subseteq G_{t-1}}$ of set systems $X_{D_t}, Y_{D_t} \subseteq 2^{[n_{\max}]}$, parameterized by all possible choices of D_t. These essentially correspond to the two summands of the cost function. Accordingly, the elements of X_{D_t} capture the sets in $\mathcal{S}_{t-1}$ that are destroyed due to the removal of users in round t, and the elements of Y_{D_t} correspond to a minimum cover of the new group G_t with respect to $\mathcal{S}_{t-1}$. We then observe that the two families of set systems satisfy a disjointedness condition required for the Bollobás Set Pairs Inequality [Bol65]. Applying the inequality allows us to lower bound a term related to $\sum_{D_t \subseteq G_{t-1}} |X_{D_t}| + |Y_{D_t}|$ which, after some calculations, yields the desired bound on $\mathrm{Cost}(t)$.

Translation to the symbolic model. In the second conceptual step, we prove that lower bounds on $\sum_{t=1}^{t_{\max}} \mathrm{Cost}(t)$ in the combinatorial model imply lower bounds on the amount of ciphertexts sent by a secure and correct ME or CGKA scheme in the symbolic model, albeit at a potential loss of a factor of 1/3. In the symbolic model [DY83], one considers ME or CGKA schemes constructed from cryptographic primitives used as building blocks that are essentially modeled to satisfy ideal security. Our lower bound for ME allows for pseudorandom generators (PRGs), pseudorandom functions (PRFs), dual PRFs (dPRFs), secret sharing, and symmetric encryption (SE), and thus in particular covers all building blocks used for the corresponding upper bounds [LYGL01, YLZL02, SM03]. Compared to prior lower bounds, it covers more building blocks than the ones considered in [MP04] and [AAB+21], which do not cover dPRFs. Our lower bound for CGKA uses PRGs, PRFs, dPRFs, secret sharing, public-key encryption (PKE), and key-updatable public-key encryption (kuPKE) as building blocks, and in particular covers all primitives used in important schemes like MLS [BBR$^+$23]. Regarding a comparison to prior bounds, while it covers strictly more primitives than the one of [AAB$^+$21], it is incomparable to the bound of [BDR20], who do not allow for secret sharing, but additionally consider broadcast encryption (BE). We consider it an interesting open question, whether our bound can be extended to BE, but point out that it is a very powerful primitive, on the one hand, has to the best of our knowledge not been used in practical CGKA constructions, and, on the other hand, implies the existence of a multicast encryption scheme with constant communication complexity (see Appendix A.4). As a consequence any such bound would have to substantially differ from the techniques used in this work.

We now describe the translation of our combinatorial bound to the symbolic model in more detail. In every round t, to every secure secret $\mathtt{r}$ present in the symbolic model we associate the set of users that had access to $\mathtt{r}$ in a round up to and including t. Then, we prove that the resulting set systems satisfy the security and correctness properties imposed in the combinatorial model, and further that

$$\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq \frac{1}{3} \sum_{t=0}^{t_{\max}} \mathrm{Cost}(t),$$

where $\mathtt{M}_t$ denotes the protocol message sent in the symbolic model in round t. In fact, the inequality holds even if we only take into account the number of ciphertexts sent rather than all protocol messages. In combination with our lower bound in the combinatorial model, this immediately yields the desired bounds on ME and CGKA.

The loss of 1/3 in our bound is due to the following. Consider the cost function $\mathrm{Cost}(t)$, for some t, in the combinatorial model. Intuitively, each of the two components, i.e., amount of sets removed from the system and size of a minimum cover of G_t, is justified by the requirement that a corresponding amount of ciphertexts is sent to communicate the respective secrets. However, it might be the case that a ciphertext that corresponded to a part of the minimum cover of group G_t in a later round $t' > t$, might be the one being used to justify the cost of a set being removed from $\mathcal{S}_{t'-1}$ following the removal of some users. In this case, the same ciphertext is being counted twice in $\sum_{t=1}^{t_{\max}} \mathrm{Cost}(t)$.

For a minimal example of this, consider the universe of users $\{u_i : i = 0, \ldots, n_{\max}\}$, where each user holds a personal keys $\mathtt{k}_i$, and the sequence of groups is given by $G_t = \{u_0, u_t\}$. I.e., the second user of a group of size 2 is replaced in every round by encrypting the new group key to user u_t's personal key (it is possible to communicate the new group key to user 0 without the need of an additional ciphertext by making a clever use of a pseudorandom generator). Then, the ciphertext accounting for the minimum cover of G_t is the same as the one corresponding to the set $G_t = \{u_0, u_t\}$ that is being removed from the set system $\mathcal{S}_t$ when considering the cost of the following round $t + 1$. Accordingly we have that $\mathrm{Cost}(t) = 2$ for every round, while only one ciphertext is being sent per round.

However, we are able to show that this kind of double counting is essentially the only thing that can go wrong in our translation between combinatorial and symbolic models. The idea behind this is to derive separate bounds on the sum over t of each summand of $\mathrm{Cost}(t)$.

We start by studying $\sum_{t=0}^{t_{\max}}(\mathrm{SizeMinCov}(G_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1)$. In order to find a cover for the set G_t we first cover the subset of users in G_{t-1} that are not removed from the group at time t, i.e., $G_{t-1} \setminus D_t$. Every user in $G_{t-1} \setminus D_t$ must be able to derive the group key of round $t - 1$ and do so by decrypting some ciphertexts sent in or before round $t - 1$. If for each of these ciphertexts we consider the set of users who know the secret key needed for decryption we obtain a cover $\mathcal{C}_{t,1}$ of $G_{t-1} \setminus D_t$. Moreover, these ciphertext can be chosen so that they are an encryption of a secret that is no longer useful in round t. This guarantees that the ciphertexts used for $\mathcal{C}_{t,1}$ and $\mathcal{C}_{\tilde{t},1}$ are different for $t \neq \tilde{t}$. Therefore $\sum_{t=1}^{t_{\max}} |\mathcal{C}_{t,1}| \leq \sum_{t=0}^{t_{\max}-1} |\mathtt{M}_t|$.

Next we obtain a cover for G_t by considering the singletons $\{u\}$ for each user that is added in round t. Since the users being added at time t do not share any secrets with the users in G_{t-1}, we have $|\mathtt{M}_t| \geq |A_t| - 1$, where the subtraction comes from the possible use of PRGs. However this might introduced some double counting as these ciphertext might be used to obtain the inequality at the end of the previous paragraph. Thus

$$\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq \frac{1}{2} \sum_{t=0}^{t_{\max}} (\mathrm{SizeMinCov}(G_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1).$$

Regarding our bound on the first summand of $\mathrm{Cost}(t)$, if we consider a set S associated to some secret $\mathtt{r}$ in round $t-1$ that contains at least two users, it must be the case that at least one of them had to decrypt a ciphertext in order to learn $\mathtt{r}$. Now, the additional restriction that $S \cap D_t \neq \emptyset$ guarantees that this ciphertext is not used in future rounds and therefore we obtain

$$\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq \sum_{t=1}^{t_{\max}} |\{S \in \mathcal{S}_{t-1} : S \cap D_t \neq \emptyset \text{ and } |S| > 1\}|.$$

Combining the bounds on both components yields our bound on $\sum_{t=1}^{t_{\max}} \mathrm{Cost}(t)$.

1.3 Further Related Work

Multicast Encryption protocols can be traced back to [WHA99, WGL00]. Besides the already mentioned constructions, we point out the work of Canetti, Malking and Nissim [CMN99], who build schemes with trade-offs for communication complexity and storage on the user/CA side. In turn, Dondeti, Mukherjee and Samal [DMS00] build a scheme where intermediate nodes in the key graph need be trusted, which uses long-term keys and nested encryption to achieve this. However, as a downside, it is either not secure against collusion, or the long term key needs to be rekeyed, which comes at high cost. Regarding batched operations, the approach was already explicitly mentioned by [CEK+99]. They further build a scheme that beats our bound in terms of communication complexity, but does so in a considerably weaker security model where removed users are not allowed to collude. A recent work by Bienstock, Dodis and Tang [BDT22] propose an enhanced version of ME including PCS as a security goal as a suitable alternative to CGKAs and provide a construction using dual-PRFs and Updatable-SKE.

A work worth noting is that of Naor, Naor and Lotspiech [NNL01], who introduce the *Subset-Cover framework* in order to abstract revocation schemes as set systems satisfying certain properties, in an approach similar to ours. They focus on the related problem of a central authority wanting to send a message to a group of stateless receivers such that a subset of (revoked) receivers is not able to obtain the content of the message, and frame it as finding a set system that allows for partitioning the set of non-revoked users efficiently. Their work proposes two constructions, one achieving ciphertexts of size $d \log(n/d)$,[5] where d is the number of revoked users, matching our bound; and another achieving ciphertexts of size $2d$ at the cost of each user being part of a linear number of sets (thus having linear cost according to our metric). Further, they provide a lower bound based on the Sunflower Lemma that is weaker than ours.

The CGKA primitive was first defined in [ACDT20], in an attempt to capture the exact security of TreeKEM, the key-agreement protocol underlying the IETF MLS standard [BBR+23]. A variety of works have since looked at improving the efficiency of MLS, by providing alternative ways to handle removals [KPPW+21], exploiting the advantages given my multi-recipient public key encryption [HKP+21, AHKM22], or considering concurrent group operations [BDR20, AAN+22a, AAN+22b].

2 Preliminaries

2.1 Definitions and Results from Combinatorics

Definition 2.1 (Cover and (size of a) minimum cover)**.** *Let $n \in \mathbb{N}$ and $\mathcal{S} \subseteq 2^{[n]}$. Then for $X \subseteq [n]$, a* cover *of X with respect to $\mathcal{S}$ is a set $\mathcal{T} \subseteq \mathcal{S}$ satisfying $X = \bigcup_{T \in \mathcal{T}} T$. A cover of X with respect to $\mathcal{S}$ of minimal cardinality is referred to as a* minimum cover*. We will use the notation* $\mathrm{SizeMinCov}(X, \mathcal{S})$ *to denote the cardinality of a minimum cover of X with respect to $\mathcal{S}$.*

We now recall two results from combinatorics; the well-known inequality of arithmetic and geometric means and the *Bollobás Set Pairs Inequality.*

[5] The absence of the additive term d here is due to the fact that leaves of revoked users do not need to be replaced in this case.

Proposition 2.2 (Inequality of arithmetic and geometric means)**.** *For $k \in \mathbb{N}$ let $x_1, \ldots, x_k \in \mathbb{R}$ be non-negative. Then*

$$\prod_{i=1}^{k} x_i \leq \left(\frac{1}{k} \sum_{i=1}^{k} x_i \right)^k .$$

Lemma 2.3 (Bollobás Set Pairs Inequality [Bol65])**.** *Let $m \in \mathbb{N}$ and consider families of finite sets $\mathcal{X} = \{X_1, X_2, \ldots, X_m\}$ and $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_m\}$ such that $X_i \cap Y_j = \emptyset$ if and only if $i, j \in [m]$ are equal. Then*

$$\sum_{i=1}^{m} \binom{|X_i| + |Y_i|}{|X_i|}^{-1} \leq 1.$$

3 Lower Bounds in the Combinatorial Model

In this section we present a simple combinatorial model for the batched replacement of users in multicast encryption and continuous group-key agreement (in Section 3.1) and then use it to derive a lower bound on the communication required to replace sets of users picked uniformly at random from the group members (in Section 3.2).

3.1 The Combinatorial Model

In this section we present a simple, purely combinatorial model that aims to capture the communication cost of batched replacements of users in multicast-encryption (ME) and continuous group-key agreement (CGKA) schemes. In both settings, a group of users evolving through rounds wants to agree on a sequence of group-keys by sending and processing protocol messages. In the case of ME, these are generated and sent by a central authority, whereas in the case of CGKA they are generated by the users themselves and distributed via an untrusted server. Security essentially requires that, even given access to all protocol messages and the current and prior internal states of all users not currently in the group, it is not possible to gain any information on the current group key. Our model closely resembles the one of [ACPP23] but extends it to dynamic groups, and further differs in some aspects such as, for example, the cost metric (see Remark 3.2). Looking ahead, in Sections 4 and A we show that lower bounds in the combinatorial model imply lower bounds on the number of ciphertexts sent in ME or CGKA schemes in the symbolic model.

High-level structure and evolution of the group. An instantiation of the combinatorial model consists of two integers $n_{\max}, t_{\max} \in \mathbb{N}$, a set $G_0 \subseteq [n_{\max}]$, sequences of sets $(D_t, A_t)_{t=1}^{t_{\max}}$, and a sequence of collections of sets $(\mathcal{S}_t)_{t=0}^{t_{\max}}$. Here $[n_{\max}]$ represents the universe of users, $t_{\max}$ the number of rounds, and G_0 the initial group. For $t \in [t_{\max}]$, the sets D_t and A_t represent the users removed and added from and to the group, respectively. Accordingly, for $t \geq 1$ we inductively define the group in round t as

$$G_t := (G_{t-1} \cup A_t) \setminus D_t.$$

To make the additions and removals to and from the group meaningful we impose the requirement that $D_t \subseteq G_{t-1}$ and $A_t \subseteq [n_{\max}] \setminus G_{t-1}$ for all $t \geq 1$. Regarding the removed users, we will even impose the stronger requirement that they are never added back to the group, i.e, that for all $t \in [t_{\max}]$ we have that

$$A_t \subseteq [n_{\max}] \setminus \left(G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'} \right).$$

Looking ahead, this requirement will be necessary to formally justify our cost function. Consider the scenario where every even round user u replaces user v, and vice versa in odd rounds. Then, after the first couple of rounds, no more communication is required, as all users could simply switch between 2 previously established group keys, essentially allowing the repeated replacement of 2 users for free when considering the amortized cost over many rounds. Our restriction above, thus, allows us to get around artificial examples of this kind.

Associated set system and cost function. The final component of the combinatorial model is a sequence $(\mathcal{S}_t)_{t=0}^{t_{\max}}$, where every $\mathcal{S}_t \subseteq 2^{[n_{\max}]}$ is a collection of sets of users. We will refer to $\mathcal{S}_t$ as the *set system* at time t. The intuition behind a set S being part of $\mathcal{S}_t$ is that, in the corresponding ME or CGKA scheme, there exists a key that is secure in round t and known to exactly the users contained in S. More precisely, this means that the key is derivable from the (current or any prior) internal state of every user $u \in S$, but cannot be recovered from the sent protocol messages as well as the current and previous states of users not in S.

To capture the correctness and security of ME and CGKA schemes the set systems $\mathcal{S}_t$ in an instantiation of the combinatorial model must satisfy the following two properties.

(i) $G_t \in \mathcal{S}_t$ for all $t \in [t_{\max}]$. This corresponds to all group members in round t agreeing on a secure key.

(ii) $S \subseteq G_t$ for all sets $S \in \mathcal{S}_t$. This property represents that all keys known to users not in G_t are being considered insecure in round t.

Finally, we associate a cost to each round in an instantiation of the combinatorial model. The cost of round t is given by the sum of two terms; the first essentially being the size of a minimum cover of the new group G_t with respect to set system $\mathcal{S}_{t-1}$ of round $t-1$, and the second essentially being the number of sets in $\mathcal{S}_{t-1}$ no longer present in $\mathcal{S}_t$ due to the removal of the users in D_t. Intuitively, the first summand corresponds to a lower bound on the number of ciphertexts that have to be send in round t in order to establish the group key K_t and the second summand to keys established in previous rounds that are no longer secure, but were established at the cost of sending at least one ciphertext (see Remark 3.2).

Definition 3.1. *Let* $\left((n_{\max}, t_{\max}, G_0),\ (D_t, A_t)_{t=1}^{t_{\max}},\ (\mathcal{S}_t)_{t=0}^{t_{\max}}\right)$ *be an instantiation of the combinatorial model. We define the cost of round* 0 *as*

$$\mathrm{Cost}(0) = \mathrm{SizeMinCov}(G_0, \{\{u\} : u \in G_0\}) - 1 = |G_0| - 1,$$

and for $t \geq 1$ *we define the cost of round* t *as*

$$\begin{aligned}\mathrm{Cost}(t) = \mathrm{SizeMinCov}(G_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1 \\ + |\{S \in \mathcal{S}_{t-1} : S \cap D_t \neq \emptyset \text{ and } |S| > 1\}|.\end{aligned}$$

Looking ahead, we will show that $\sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)$ in the symbolic model corresponds to a lower bound on the number of ciphertexts sent as part of protocol messages over the whole execution of the experiment, albeit with a loss of a factor of 3. Before proving our lower bound in the combinatorial model we discuss some similarities and differences to the combinatorial model used in [ACPP23], which is used to derive lower bounds on the communication cost of recovering from state corruptions in CGKA by means of concurrent key updates.

Remark 3.2. *While the structure of our combinatorial model closely resembles the one of [ACPP23], it differs from it in the following aspects.*

(a) Our model allows for additions and removals of users to and from the group.

(b) We work with a different cost function. The cost metric used in [ACPP23] for set $S \in \mathcal{S}_t \setminus \mathcal{S}_{t-1}$ *and round* t *(following the minimum cover approach discussed in the introduction) essentially quantifies the communication cost required in round* t *to add* S *to the set system. The cost function we use in this work additionally takes the cost of sets being eliminated from* $\mathcal{S}_{t-1}$ *into account. Accordingly* $\mathrm{Cost}(t)$ *is not to be understood as the communication sent in the current round, but instead also takes communication that already occurred in prior rounds into account.*

(c) [ACPP23] also connects the cost of adding a set S *to* $\mathcal{S}_t$ *to a minimum cover with respect to the previous set system* $\mathcal{S}_{t-1}$*, However, [ACPP23] uses a relaxed definition of minimum cover, which requires* S *to be covered by a union of sets, not necessarily to be* equal *to the union. The intuition behind this is that,*

in this work, instead of security against an adversary corrupting users, we want to protect the group key against the users themselves as soon as they have been removed from the group, even if they stored all keys they previously had access to. Phrased differently, if a secret/symmetric key is communicated to a user at any point in time, we assume it remains known to that user for the remainder of the experiment. Accordingly, in this work we define the set system $\mathcal{S}_t$ by associating to a (secure) key the set of users in G_t which, at any point in time until round t, had access to the key. Since (except for the user generating the key from fresh randomness) all other users in the corresponding set must have learned it from a ciphertext encrypted under a secure key already known to them, the corresponding communication cost must be at least the size of a minimum cover of the set with respect to the previous set system (in the stronger sense of Definition 2.1).

On the other hand, [ACPP23] associated to a key the set of users which are able to recover the key from their states since the last corruption before round t, effectively allowing users to forget keys they knew in some prior round and leading to a relaxed definition of minimum cover.

3.2 Lower Bound for Batched Replacements of Users

We now prove a bound on the communication complexity of batched replacement of users in the combinatorial model. It essentially states that the prior multicast encryption schemes batching dynamic operations [LYGL01, SM03] and the MLS continuous group-key agreement standard [BBR+23] (the latter with respect to fair-weather communication complexity) are optimal up to a small constant factor.

On a technical level, we define two families of subsets of G_t, which essentially correspond to the two contributors to the cost function, i.e., the sets forming a minimum cover of the new group G_t with respect to the previous set system, and the sets containing at least one user removed in the current round, respectively. We then observe that said families satisfy the disjointedness condition required to apply the Bollobás Set Pairs Inequality. This allows us to lower bound their sizes, which after some calculations implies the desired bound.

We first prove an implication of the Bollobás Set Pairs Inequality (Lemma 2.3).

Lemma 3.3. *Let $\mathcal{X} = \{X_1, X_2, \ldots, X_m\}$ and $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_m\}$ be families of subsets of a finite set Z such that $X_i \cap Y_j \neq \emptyset$ if and only if $i, j \in [m]$ are distinct. Then*

$$\sum_{i=1}^{m} (|X_i| + |Y_i|) \geq m \ln m.$$

Proof. The condition of $X_i \cap Y_j \neq \emptyset$ for all $i \neq j$ implies $X_i, Y_i \neq \emptyset \; \forall i \in [m]$. Thus, from Lemma 2.3 we obtain

$$\sum_{i=1}^{m} \binom{|X_i| + |Y_i|}{|X_i|}^{-1} \leq 1,$$

which, using the bound $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$, implies

$$\begin{aligned} 1 \geq \sum_{i=1}^{m} \binom{|X_i| + |Y_i|}{|X_i|}^{-1} &\geq \sum_{i=1}^{m} \left(\frac{e(|X_i| + |Y_i|)}{|X_i|}\right)^{-|X_i|} \\ &= \sum_{i=1}^{m} e^{-|X_i|} \cdot \left(1 + \frac{|Y_i|}{|X_i|}\right)^{-|X_i|}. \end{aligned}$$

Now, using that $(1 + x/n)^{-n} \geq e^{-x}$ and by multiplying by $1/m$ we obtain that

$$\frac{1}{m} \geq \frac{1}{m} \sum_{i=1}^{m} e^{-|Y_i|} \cdot e^{-|X_i|} \geq \frac{1}{m} \sum_{i=1}^{m} e^{-(|Y_i| + |X_i|)}.$$

By the inequality of arithmetic and geometric means (Proposition 2.2) we have that

$$\frac{1}{m} \geq \Big(\prod_{i=1}^{m} e^{-(|X_i|+|Y_i|)}\Big)^{1/m} = \Big(\prod_{i=1}^{m} e^{(|X_i|+|Y_i|)}\Big)^{-1/m},$$

which by taking ln gives the desired result of $\sum_{i=1}^{m}(|X_i|+|Y_i|) \geq m \ln m$. □

We are now able to show that in the combinatorial model replacing a set of d users chosen uniformly at random in a group of size n has cost at least $d \cdot \ln(n/d)$ in expectation. The bound holds in an amortized sense, i.e., even if the experiment is repeated for several rounds. More formally, we obtain the following.

Theorem 3.4. *Let $n, t_{\max}, n_{\max} \in \mathbb{N}$ and $(d_t)_{t=1}^{t_{\max}}$ such that $d_t \in \mathbb{N}$ with $d_t \leq n$ for all t. Consider an instantiation of the combinatorial model with respect to $(n_{\max}, t_{\max}, G_0)$ and $(D_t, A_t)_{t=1}^{t_{\max}}$ where $|G_0| = n$ and, for all t, the set D_t of removed users is sampled uniformly at random from the set $\{D \subseteq G_{t-1} \mid |D| = d_t\}$, and A_t can be arbitrary according to the restrictions $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'})$ and $|A_t| = |D_t|$. Then it holds that*

$$\mathbb{E}\Bigg[\sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)\Bigg] \geq \sum_{t=1}^{t_{\max}} \bigg(d_t \ln\Big(\frac{n}{d_t}\Big) - 1\bigg),$$

where the expectation is taken over the choice of $(D_t)_t$. In particular, if $d_t = d$ for all t, then

$$\mathbb{E}\Bigg[\sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)\Bigg] \geq t_{\max} \cdot \bigg(d \cdot \log\Big(\frac{n}{d}\Big) - 1\bigg).$$

Proof. Since $|D_t| = |A_t| = d_t$ for all $t > 0$, we have $|G_t| = n$ for all $t \geq 0$. We first consider the cost of a single round $t \in [t_{\max}]$. By definition $G_t = (G_{t-1} \cup A_t) \setminus D_t$, where $D_t \subset G_{t-1}$, and $G_{t-1} \cap A_t = \emptyset$. Therefore we get that $G_t \cap G_{t-1} = G_{t-1} \setminus D_t = G_t \setminus A_t$ and

$$\begin{aligned}
&\mathrm{SizeMinCov}(G_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t\}) \\
&= \mathrm{SizeMinCov}(G_t \setminus A_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t \setminus A_t\}) + |\{\{u\} : u \in A_t\}| \\
&= \mathrm{SizeMinCov}(G_t \setminus A_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t \setminus A_t\}) + d_t \\
&= \mathrm{SizeMinCov}(G_{t-1} \setminus D_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_{t-1} \setminus D_t\}) + d_t \qquad (1)
\end{aligned}$$

as $S \cap A_t = \emptyset$ for all $S \in \mathcal{S}_{t-1}$ (as $S \subseteq G_{t-1}$ by Property (ii) of the combinatorial model).

Now for each possible subset $D_t \subseteq G_{t-1}$ such that $|D_t| = d_t$, consider the sets

$$\begin{aligned}
X_{D_t} &= \{S \in \mathcal{S}_{t-1} \mid S \cap D_t \neq \emptyset\} \cup \{\{u\} \colon u \in D_t\} \\
&= \{S \in \mathcal{S}_{t-1} \mid S \cap D_t \neq \emptyset \text{ and } |S| > 1\} \cup \{\{u\} \colon u \in D_t\}. \qquad (2)
\end{aligned}$$

Further, let Y_{D_t} denote any minimum cover of $G_{t-1} \setminus D_t$ with respect to $\mathcal{S}_{t-1} \cup \{\{u\} \colon u \in G_{t-1} \setminus D_t\}$. Such a minimum cover always exists since $G_{t-1} \setminus D_t$ is actually covered by sets in $\mathcal{S}_{t-1} \cup \{\{u\} \colon u \in G_{t-1} \setminus D_t\}$. Note that Y_{D_t} also is a minimum cover of $G_{t-1} \setminus D_t$ with respect to $(\mathcal{S}_{t-1} \cup \{\{u\} \colon u \in G_{t-1} \setminus D_t\}) \setminus X_{D_t}$, and that by Equation 1 we have that

$$\mathrm{SizeMinCov}(G_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in G_t\}) = |Y_{D_t}| + d_t. \qquad (3)$$

We claim that $X_{D_t} \cap Y_{D'_t} = \emptyset$ if and only if $D_t = D'_t$. Take the case of $D_t = D'_t$; if $S \in Y_{D_t}$, then by definition of Y_{D_t}, $S \notin X_{D_t}$, thus $X_{D_t} \cap Y_{D_t} = \emptyset$. Now for the case of $D_t \neq D'_t$, there must be a user u such that $u \in D_t, u \notin D'_t$. Since $Y_{D'_t}$ covers $G_{t-1} \setminus D'_t$, there must exist $S \in Y_{D'_t}$ such that $u \in S$. Thus $S \in X_{D_t}$. Hence $X_{D_t} \cap Y_{D'_t} \neq \emptyset$.

Using Lemma 3.3 we obtain

$$\frac{1}{\binom{n}{d_t}} \sum_{D_t \subseteq G_{t-1}, |D_t| = d_t} |X_{D_t}| + |Y_{D_t}| \geq \frac{1}{\binom{n}{d_t}} \binom{n}{d_t} \ln \binom{n}{d_t} \geq d_t \ln \frac{n}{d_t}. \qquad (4)$$

Note that Equation 4 gives a lower bound on the expectation of $|X_{D_t}|+|Y_{D_t}|$ if the set D_t is chosen uniformly at random. To make this formal, given n, $n_{\max}$, $t_{\max}$, $(d_t)_{t=1}^{t_{\max}}$, and G_0, we define a sequence of random variables $(\mathbf{D}_t, \mathbf{A}_t, \mathbf{G}_t)_{t=1}^{t_{\max}}$ all taking values in $2^{[n_{\max}]}$ where $\mathbf{G}_1 := (G_0 \cup \mathbf{A}_1) \setminus \mathbf{D}_1$ and for $t \geq 2$

$$\mathbf{G}_t := (\mathbf{G}_{t-1} \cup \mathbf{A}_t) \setminus \mathbf{D}_t.$$

The sequence is distributed as follows. The set $\mathbf{D}_1$ of users removed in the first round is distributed uniformly over $\{D_1 \subseteq G_0 : |D_1| = d_1\}$ and $\mathbf{A}_1$ can distributed arbitrarily over $\{A_1 \subseteq [n_{\max}] \setminus G_0 : |A_1| = d_1\}$. Now, conditioned on $\mathbf{D}_{t'} = D_{t'}$, $\mathbf{A}_{t'} = A_{t'}$, and $\mathbf{G}_{t'} = G_{t'}$ for $t' \in [t-1]$, the random variables $\mathbf{D}_t$ is distributed uniformly over $\{D_t \subseteq G_{t-1} : |D_t| = d_t\}$ and $\mathbf{A}_t$ can be distributed arbitrarily over $\{A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'}) : |A_t| = d_t\}$.

If we consider the expected cost of round t (see Definition 3.1) with respect to the sequence of adds and removes given by $(\mathbf{D}_t, \mathbf{A}_t)_{t=1}^{t_{\max}}$ we obtain by Equations 2, 3, and 4 that

$$\begin{aligned}
\mathbb{E}[\mathrm{Cost}(t)] &= \mathbb{E}[|\{S \in \mathcal{S}_{t-1} : S \cap \mathbf{D}_t \neq \emptyset \text{ and } |S| > 1\}|] \\
&\quad + \mathbb{E}[\mathrm{SizeMinCov}(\mathbf{G}_t, \mathcal{S}_{t-1} \cup \{\{u\} : u \in \mathbf{G}_t\})] - 1 \\
&\geq \mathbb{E}[|X_{\mathbf{D}_t}| - d_t + |Y_{\mathbf{D}_t}| + d_t] - 1 \\
&\geq d_t \ln \frac{n}{d_t} - 1.
\end{aligned}$$

Now, the theorem's statement follows by linearity of expectation. □

4 Lower Bound for Batched Replacements in Multicast Encryption

In this section we define multicast encryption in the symbolic model and show that the lower bound on batched replacement of users in the combinatorial model of Section 3 carries over. Section 4.1 specifies the symbolic model and provides syntax for multicast encryption, Section 4.2 proves the corresponding bound.

4.1 Multicast Encryption in the symbolic model

Considered building blocks. We now define syntax for *multicast encryption* (ME) in a symbolic model in the style of Dolev and Yao [DY83]. In models of this type, keys and ciphertexts of cryptographic primitives are seen as symbolic variables, which are generated according to grammar rules, and can be derived from sets of other symbolic variables according to an entailment relation $\vdash$, which itself models ideal security notions of the used cryptographic building blocks. Throughout this section we will denote symbolic variables in typewriter font to distinguish them from non-symbolic inputs and outputs of algorithms. Further, single variables are depicted using lower case letters, sets of variables using upper case letters.

In our symbolic treatment of multicast encryption we consider symbolic variables of the following two types; (pseudo)random strings denoted by $\mathtt{r}$ and messages $\mathtt{m}$. The former will also serve as keys of symmetric encryption schemes and, in this context, we will often denote them by $\mathtt{k}$. Similarly, ciphertexts of symmetric encryption are of message type and we will often denote them by $\mathtt{c}$. We consider ME schemes constructed from symmetric encryption schemes (SE), pseudorandom generators (PRG), pseudorandom functions (PRF), dual pseudorandom functions (dPRF) and secret sharing defined according to the following syntax.

- A symmetric encryption scheme $\mathsf{SE} = (\mathsf{SE.Enc}, \mathsf{SE.Dec})$ specifies an encryption algorithm $\mathsf{SE.Enc}(\mathtt{k}, \mathtt{m})$ that, on input symmetric key $\mathtt{k}$ of type $\mathtt{r}$ and message $\mathtt{m}$, returns a ciphertext $\mathtt{c}$ that is of message type. Deterministic decryption algorithm $\mathsf{SE.Dec}(\mathtt{k}, \mathtt{c})$, on input symmetric key $\mathtt{k}$ and ciphertext $\mathtt{c}$, returns a message $\mathtt{m}$.

  We require perfect correctness, i.e., $\mathsf{SE.Dec}(\mathtt{k}, \mathsf{SE.Enc}(\mathtt{k}, \mathtt{m})) = \mathtt{m}$ for all $\mathtt{k}$ and $\mathtt{m}$.

- A pseudorandom generator $\mathsf{PRG}(\mathtt{r})$, on input random string $\mathtt{r}$, returns a value $(\mathtt{r}_1, \mathtt{r}_2)$, consisting of two pseudorandom strings. For simplicity we restrict ourselves to PRGs with stretch 2. Note that PRGs with larger stretch can easily be built from these using standard methods.
- A pseudorandom function $\mathsf{PRF}(\mathtt{r}, ad)$, on input random string $\mathtt{r}$ and non-symbolic associated data ad, returns a pseudorandom string $\mathtt{r}'$.
- A dual pseudorandom function $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2)$, takes two random strings $\mathtt{r}_1, \mathtt{r}_2$ as input and returns a pseudorandom string $\mathtt{r}' = \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathsf{dPRF}(\mathtt{r}_2, \mathtt{r}_1)$.
- A secret sharing scheme given by two algorithms S and R. On input a message $\mathtt{m}$, S outputs a set of s many shares $\mathsf{S}(\mathtt{m}) = \{\mathsf{S}_i(\mathtt{m})\}_{i \in [s]}$ of type message and the original message can be recovered given some subset of shares as determined by an access structure $\Gamma \subseteq 2^{[s]}$, namely, for every $I \in \Gamma$, $\mathsf{R}(I, \{\mathsf{S}_i(\mathtt{m})\}_{i \in I}) = \mathtt{m}$.

We now describe the grammar rules and entailment relation.

variable type		grammar rule
$\mathtt{r}$	$\leftarrow$	terminal type, $\mathsf{PRG}(\mathtt{r}), \mathsf{PRF}(\mathtt{r}), \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2)$
$\mathtt{m}$	$\leftarrow$	$\mathtt{r}$, $\mathsf{SE.Enc}(\mathtt{k}, \mathtt{m}), \mathsf{S}_i(\mathtt{m})$
entailment relation		
$\mathtt{m} \in \mathtt{M}$	$\Rightarrow$	$\mathtt{M} \vdash \mathtt{m}$
$\mathtt{M} \vdash \mathtt{r}$	$\Rightarrow$	$\mathtt{M} \vdash \mathsf{PRG}(\mathtt{r}) = (\mathtt{r}_1, \mathtt{r}_2)$
$\mathtt{M} \vdash \mathtt{r}$	$\Rightarrow$	$\forall ad \colon \mathtt{M} \vdash \mathsf{PRF}(\mathtt{r}, ad)$
$\mathtt{M} \vdash \mathtt{r}_1, \mathtt{r}_2$	$\Rightarrow$	$\mathtt{M} \vdash \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2)$
$\mathtt{M} \vdash \mathtt{r}, \mathtt{m}$	$\Rightarrow$	$\mathtt{M} \vdash \mathsf{SE.Enc}(\mathtt{r}, \mathtt{m})$
$\mathtt{M} \vdash (\mathtt{r}, \mathsf{c}) : \mathsf{c} = \mathsf{SE.Enc}(\mathtt{r}, \mathtt{m})$	$\Rightarrow$	$\mathtt{M} \vdash \mathtt{m}$
$\exists I \in \Gamma \colon \mathtt{M} \vdash \{\mathsf{S}_i(\mathtt{m})\}_{i \in I}$	$\Rightarrow$	$\mathtt{M} \vdash \mathtt{m}$

The grammar rules state that (pseudo)random coins can either be directly sampled or generated using a PRG or be obtained as the image of PRF or dPRF; that the encryption algorithm of SE, on input a key of type $\mathtt{r}$ and message $\mathtt{m}$, generates a ciphertext; and that messages can be of arbitrary type. The entailment relation states that every symbolic variable contained in a set $\mathtt{M}$ can be recovered from the set. Further, it models ideal PRG security, stating that outputs of a PRG can only be recovered if given access to the respective input. PRF security is also modeled in the same way, which means that there is no significant difference between PRGs and PRFs in the symbolic model. The security of a dPRF is modeled by requiring that outputs of a dPRF can only be recovered given access to both inputs. Similarly, ideal SE security, i.e., that ciphertexts can only be decrypted if given access to the corresponding key. For a more detailed explanation and examples of the symbolic model we refer to [MP04]. The security of the secret sharing scheme corresponds to the requirement that the original message can be recovered from a set of shares as determined by the access structure. Given a set $\mathtt{M}$ of symbolic variables we denote the set of all variables derivable from it using the entailment relation by $\mathsf{Der}(\mathtt{M})$, i.e.,

$$\mathtt{m} \in \mathsf{Der}(\mathtt{M}) \text{ exactly if } \mathtt{M} \vdash \mathtt{m}.$$

If $\mathtt{M}_1, \mathtt{M}_2$ are two sets of symbolic variables, we use the notation $\mathsf{Der}(\mathtt{M}_1, \mathtt{M}_2) = \mathsf{Der}(\mathtt{M}_1 \cup \mathtt{M}_2)$.

Multicast encryption syntax. A multicast encryption scheme essentially allows a central authority to provide a dynamically changing group of users with a group key by sending protocol messages via a broadcast channel. The main goal being to use protocol messages that are as small as possible while still achieving correctness and security, i.e., that group members in every round agree on a group key that, however, cannot be recovered from the sent protocol messages even if given access to all previous states of non-members of the group.

As our goal in this work is to prove lower bounds and these are easier to state by keeping the group size constant over all rounds, we work with a simplified syntax only allowing for replacements of users, but not arbitrary removes and adds. We essentially follow [MP04, AAB+21], who analyzed the communication cost of multicast encryption for replacing a single user per round in the setting of a single group and of a system of potentially overlapping groups, respectively. The main difference in this work is that we allow for batched operations, i.e, replacing a set of more than one user at a time. Note that such a replacement can always be implemented by both removing and adding parties and thus our bounds in particular also hold for schemes allowing these operations.

In the following we split the inputs and outputs of algorithms into a symbolic part, i.e., sets of symbolic variables, and a non-symbolic part containing, e.g., user identifiers. As already stated above, the former variables are depicted in typewriter font, the latter in italics.

A multicast encryption scheme ME specifies algorithms ME.Setup, ME.Init, ME.Repl, ME.Proc, ME.Key with the following syntax.[6]

- $\mathsf{ME.Setup}(n_{\max}; \mathtt{R})$ takes as input $n_{\max}$, the universe of users, and the set of random coins $\mathtt{R}$. It sets up the initial state $(\mathtt{ST}_u^{-1}, st_u^{-1})$ for every user $u \in [n_{\max}]$. The symbolic part of the initial state, namely, $\mathtt{ST}_u^{-1}$ is subject to the requirements that $\mathtt{ST}_u^{-1} = \{\mathtt{k}_u^{-1}\}$ where $\mathtt{k}_u^{-1}$ is of type random string and $\mathtt{ST}_u^{-1} \cap \mathsf{Der}(\bigcup_{v \in [n_{\max}] \setminus \{u\}} \mathtt{ST}_v^{-1}) = \emptyset$. Similar assumptions are also made in [AAB+21, MP04]. For instance, in [MP04] it is assumed that each user is assigned exactly one key that cannot be derived from those assigned to other users, while in [AAB+21] users are additionally assigned a key for each subgroup they belong to with the property that they cannot be derived from keys of users that do not belong to the corresponding subgroup. Making this kind of assumption is justified. Otherwise one could consider schemes in which communication is artificially reduced. For instance, generating two keys $k_{S,1}, k_{S,2}$ for each possible subset $S \in 2^{[n_{\max}]}$ during setup and instead of giving users these keys, they would get a ciphertext $c_S = \mathsf{Enc}(k_{S,1}, k_{S,2})$ for each set they are a member of and then the key $k_{S,1}$ would be sent in the clear in a later round.[7]
- $\mathsf{ME.Init}(G_0; \mathtt{R})$, on input the first group G_0 and a set of random coins $\mathtt{R}$, outputs a control message $(\mathtt{M}, M)$. Further, it implicitly sets up the initial group key $\mathtt{k}^0$.
- $\mathsf{ME.Repl}(A_t, D_t; \mathtt{R})$ allows replacing a set D_t of group members by a set A_t of new users. In round t it takes as input the set $A_t \in [n_{\max}] \setminus G_{t-1}$ of users to be added to the group, $D_t \subseteq G_{t-1}$, the set of users to be removed, and a set of random coins $\mathtt{R}$. We require that $|A_t| = |D_t|$. The output of the algorithm is a control message $(\mathtt{M}, M)$. Further, the algorithm implicitly sets up the t^{th} group key $\mathtt{k}_t$.
- Deterministic algorithm $\mathsf{ME.Proc}((\mathtt{ST}_u^{t-1}, st_u^{t-1}), (\mathtt{M}, M))$ takes as input, in round t, a user's internal state $(\mathtt{ST}_u^{t-1}, st_u^{t-1})$ as well as a control message $(\mathtt{M}, M)$ (either output by ME.Init or by ME.Repl). It returns the user's updated state $(\mathtt{ST}_u^t, st_u^t)$.
- Deterministic algorithm $\mathsf{ME.Key}(\mathtt{ST}_u^t, st_u^t)$, on input user u's state at the end of round t, returns the t^{th} group key $\mathtt{k}^t$.

The algorithms ME.Setup, ME.Init, ME.Repl are run by the central authority and it is understood that they also take as input all users' states and all messages despite this not being explicitly indicated. We require that symbolic outputs of algorithms are derivable from the symbolic part of their inputs, e.g. if $(\mathtt{ST}_u^t, st_u^t) \leftarrow \mathsf{ME.Proc}((\mathtt{ST}_u^{t-1}, st_u^{t-1}), (\mathtt{M}, M))$ then it must hold that $(\mathtt{ST}_u^{t-1}, \mathtt{M}) \vdash \mathtt{ST}_u^t$. Moreover, we also require that only a finite number of derivation steps is needed. Further and for brevity, while in the following we will make the users removed from, and added to, the group explicit, we will often drop the non-symbolic parts of protocol messages and users' states, and simply write $\mathtt{ST}_u^t \leftarrow \mathsf{ME.Proc}(\mathtt{ST}_u^{t-1}, \mathtt{M})$.

[6] One can consider the possibility that some of these algorithms be randomized by also including non-symbolic randomness as an input and the results would hold for any choice of non-symbolic randomness.

[7] The restriction that $\mathtt{ST}_u^{-1}$ consists of just one element can be weakened if one requires that it only consists of random coins and for all coins $\mathtt{r} \in \mathtt{ST}_u^{-1}$ it holds that $\mathtt{r} \notin \mathsf{Der}((\mathtt{ST}_u^{-1} \setminus \{\mathtt{r}\}) \cup \cup_{v \in [n_{\max}] \setminus \{u\}} \mathtt{ST}_v^{-1})$. This is done in Appendix A in the case of CGKA schemes and it applies, mutatis mutandis, to the case of ME. But it comes at the cost of an additional step in the proof of the lower bound, so we leave it for the appendix.

```
Game $\mathrm{SEC}^{\mathsf{ME}}((n_{\max}, t_{\max}, G_0), (A_t, D_t)_{t=1}^{t_{\max}})$
00 sample $\mathtt{R}_{-1}, \mathtt{R}_0$
01 $(\mathtt{ST}_u^{-1})_{u \in [n_{\max}]} \leftarrow \mathsf{ME.Setup}(n_{\max}; \mathtt{R}_{-1})$
02 $\mathtt{M}_0 \leftarrow \mathsf{ME.Init}(G_0; \mathtt{R}_0)$
03 for $u \in G_0$:
04     $\mathtt{ST}_u^0 \leftarrow \mathsf{ME.Proc}(\mathtt{ST}_u^{-1}, \mathtt{M}_0)$
05     $\mathtt{k}_u^0 \leftarrow \mathsf{ME.Key}(\mathtt{ST}_u^0)$
06     $\mathtt{k}^0 \leftarrow \mathtt{k}_u^0$
07 for $u \in [n_{\max}] \setminus G_0$:
08     $\mathtt{ST}_u^0 \leftarrow \mathtt{ST}_u^{-1}$
09 if $\exists u \in G_0 : \mathtt{k}_u^0 \neq \mathtt{k}^0$:
10     return 0                          \\ disagreement on key
11 if $\mathtt{k}^0 \in \mathsf{Der}(\mathtt{M}_0, ((\mathtt{ST}_u^{t'})_{t'=-1}^{0})_{u \notin G_0})$:
12     return 0                          \\ group key insecure
13 for $t = 1, \ldots, t_{\max}$:
14     $\mathsf{ROUND}(A_t, D_t)$
15 return 1

Oracle $\mathsf{ROUND}(A_t, D_t)$
16 require $D_t \subseteq G_{t-1} \wedge A_t \subseteq [n_{\max}] \setminus G_{t-1}$
17 $G_t \leftarrow (G_{t-1} \cup A_t) \setminus D_t$
18 sample $\mathtt{R}_t$
19 $\mathtt{M}_t \leftarrow \mathsf{ME.Repl}(A_t, D_t; \mathtt{R}_t)$
20 for $u \in G_t$:
21     $\mathtt{ST}_u^t \leftarrow \mathsf{ME.Proc}(\mathtt{ST}_u^{t-1}, \mathtt{M}_t)$
22     $\mathtt{k}_u^t \leftarrow \mathsf{ME.Key}(\mathtt{ST}_u^t)$
23     $\mathtt{k}^t \leftarrow \mathtt{k}_u^t$
24 for $u \in [n_{\max}] \setminus G_t$:
25     $\mathtt{ST}_u^t \leftarrow \mathtt{ST}_u^{t-1}$
26 if $\exists u \in G_t : \mathtt{k}_u^t \neq \mathtt{k}^t$:
27     return 0                          \\ disagreement on key
28 if $\mathtt{k}^t \in \mathsf{Der}((\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t})_{u \notin G_t})$:
29     return 0                          \\ group key insecure
```

Figure 2: Symbolic security and correctness game for multicast encryption scheme ME. In Line 16 if the condition after **require** is not met the game aborts and outputs 1, meaning that the execution of the game is considered to have been secure.

Correctness and security. We capture security and correctness of multicast-encryption schemes in the symbolic model simultaneously with the game in Figure 2. Similar to the experiment in the combinatorial model, the game is parameterized by a tuple $(n_{\max}, t_{\max}, G_0)$ which specifies the initialization of the group and a sequence $(A_t, D_t)_{t=1}^{t_{\max}}$ of users added to, and removed from, the group. In round 0 the states of all users in $[n_{\max}]$ are set up using $\mathsf{ME.Setup}$ and the group G_0 is initialized using $\mathsf{ME.Init}$ and $\mathsf{ME.Proc}$. Then security and correctness are verified for the first round, meaning that (a) all users in G_0 have access to the (unique) group key $\mathtt{k}^0$, and (b) the non-members of G_0 are not able to derive $\mathtt{k}^0$ from their internal states and the protocol message $\mathtt{M}^0$ sent in round 0 even if colluding. If both checks succeed, the game proceeds in rounds t. In each of them the users in A_t are added to the group and the users in D_t removed from it using $\mathsf{ME.Repl}(A_t, D_t)$, and all current group members are made to process the resulting protocol message $\mathtt{M}^t$ with $\mathsf{ME.Proc}$. Again it is checked that the round satisfies correctness and security. The former means that all users in G_t derive the same group key for round t, which can essentially be seen as the requirement that

$$\exists \mathtt{k}^t : \mathtt{k}^t = \mathsf{ME.Key}(\mathtt{ST}_u^t) \text{ for all } u \in G_t.$$

Note that this in particular implies $\mathtt{k}^t \in \mathsf{Der}(\mathtt{ST}_u^t)$ for all $u \in G_t$.

The latter means that, even if all non group-members never deleted their old states and collude, they are not able to recover the current group key, i.e.,

$$\mathtt{k}^t \notin \mathsf{Der}\Big((\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t})_{u \in [n_{\max}] \setminus G_t}\Big).$$

This notion of security only asks for post-compromise security and not forward-secrecy since we are not considering the possibility that the group key at time t can be derived from future exposures. This only strengthens our lower-bound. If one of the checks fails, the game aborts and returns 0, else it returns 1. We say that a ME scheme ME is correct and secure, if game SEC with respect to any input $(n_{\max}, t_{\max}, G_0), (A_t, D_t)_{t=1}^{t_{\max}}$ returns 1.

Useful keys and associated set system. Consider an execution of game $\mathrm{SEC}^{\mathsf{ME}}$ with respect to $(n_{\max}, t_{\max}, G_0)$ and $(A_t, D_t)_{t=1}^{t_{\max}}$. Let $t \in [t_{\max}]_0 := [t_{\max}] \cup \{0\}$ and consider a random coin $\mathtt{r}$ that was generated in some round up to and including t. We say $\mathtt{r}$ is *useful* at time t, if

$$\mathtt{r} \notin \mathsf{Der}\Big((\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t})_{u\in[n_{\max}]\setminus G_t}\Big),$$

which means that it cannot be derived from all protocol messages sent so far and all prior and current states of users that are not members of the group at time t. Following [ACPP23], we associate to a secure coin $\mathtt{r}$ a set of users, with the important difference, however, that in this work the set contains all users that had access to $\mathtt{r}$ *at any point in time.*

Definition 4.1. *Consider an execution of security game* $\mathrm{SEC}^{\mathsf{ME}}$ *with respect to input* $(n_{\max}, t_{\max}, G_0)$ *and* $(A_t, D_t)_{t=1}^{t_{\max}}$*. Let* $t \in [t_{\max}]_0$ *and* $\mathtt{r}$ *be a random coin. We define*

$$S(t, \mathtt{r}) := \{u \in [n_{\max}] \mid \mathtt{r} \in \mathsf{Der}(\mathtt{ST}_u^{-1}, (\mathtt{M}_{t'})_{t'\leq t\colon u\in G_{t'}})\}$$

It should be noted that $\mathtt{ST}_u^{t'} \subseteq \mathsf{Der}(\mathtt{ST}_u^{-1}, (\mathtt{M}_{t'})_{t'\leq t\colon u\in G_{t'}})$ *for* $t' \leq t$*. Further, we define the set system at time* t *as*

$$\mathcal{S}_t := \{S \subseteq [n_{\max}] \mid \exists \textit{ useful coin } \mathtt{r} : S = S(t, \mathtt{r})\}.$$

We prove two Lemmas that capture how derivation works in the symbolic model and connects it to the sets defined in Definition 4.1.

Lemma 4.2. *Let* $\mathtt{r}$ *be of type random coin and useful at time* $t \in [t_{\max}]_0$*, and* u *a user such that* $u \in S(t, \mathtt{r})$*. Then (at least) one of the following cases holds.*

1. *There exist* $\mathtt{r}'$ *with* $\mathsf{PRG}(\mathtt{r}') = (\mathtt{r}_1, \mathtt{r}_2)$ *and* $i \in \{1, 2\}$ *such that* $\mathtt{r} = \mathtt{r}_i$*. Further,* $\mathtt{r}'$ *is useful at time* t *and* $u \in S(t, \mathtt{r}')$*.*
2. *There exists* $\mathtt{r}'$ *and associated data ad such that* $\mathsf{PRF}(\mathtt{r}', ad) = \mathtt{r}$*. Further,* $\mathtt{r}'$ *is useful at time* t *and* $u \in S(t, \mathtt{r}')$*.*
3. *There exist* $\mathtt{r}_1$ *and* $\mathtt{r}_2$ *such that* $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$*, at least one of* $\mathtt{r}_1$ *and* $\mathtt{r}_2$ *is useful at time* t*, and* $u \in S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2)$*.*
4. $\mathtt{r} \in \mathtt{ST}_u^{-1}$
5. *There exists* $\mathsf{c} = e_0(\cdot) \circ \ldots \circ e_g(\cdot) \circ \ldots \circ e_h(\mathtt{r})$ *where* $e_i = \mathsf{SE.Enc}(\mathtt{r}_i, \cdot)$ *or* $e_i = \mathsf{S}_{i_j}(\cdot)$ *and*
   (a) $\mathsf{c} \in \bigcup_{\tilde{t}\leq t\colon u\in G_{\tilde{t}}} \mathtt{M}_{\tilde{t}}$,
   (b) *if* $e_i = \mathsf{SE.Enc}(\mathtt{r}_i, \cdot)$ *and* $i \geq g + 1$*,* $\mathtt{r}_i$ *is not useful at time* t*,*
   (c) *there exists* $i \in \{0, \ldots, h\}$ *such that* $e_i = \mathsf{SE.Enc}(\mathtt{r}_i, \cdot)$*,*
   (d) $e_g = \mathsf{SE.Enc}(\mathtt{r}_g, \cdot)$ *and* $\mathtt{r}_g$ *is useful at time* t*, and*
   (e) *for all encryptions* $e_i = \mathsf{SE.Enc}(\mathtt{r}_i, \cdot)$*, it holds that* $u \in S(t, \mathtt{r}_i)$*.*

Proof. If $\mathtt{r}$ admits a PRG pre-image $\mathtt{r}'$, $\mathtt{r}'$ must be useful at time t since $\mathtt{r}$ is. Therefore we have two possible cases depending on whether $u \in S(t, \mathtt{r}')$. If $u \in S(t, \mathtt{r}')$ we are in Case 1. If $u \notin S(t, \mathtt{r}')$, then one of the following holds:

- $\mathtt{r} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'\leq t\colon u\in G_{t'}} \mathtt{M}_{t'}$ and the fact that $\mathtt{r}$ is useful implies that $\mathtt{r} \in \mathtt{ST}_u^{-1}$.
- Or, by repeatedly applying the last two rules of the entailment relation, there exists a ciphertext $\mathsf{c} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'\leq t\colon u\in G_{t'}} \mathtt{M}_{t'}$ of the form $e_0(\cdot) \circ \ldots \circ e_g(\cdot) \circ \ldots \circ e_h(\mathtt{r})$ where each e_i is an application of S or $\mathsf{SE.Enc}$ such that condition *(e)* holds. By assumption $\mathtt{ST}_u^{-1}$ only contains symbols of type random coins, so $\mathsf{c} \in \bigcup_{t'\leq t\colon u\in G_{t'}} \mathtt{M}_{t'}$. Therefore there must exist at least one encryption in c under a useful key since $\mathtt{r}$ is useful. This shows *(c)* and *(d)*. Condition *(b)* is just a matter of choice.

The two options above correspond to Cases 4 or 5, respectively.

If $\mathtt{r}$ does not admit a PRG pre-image, we consider whether it admits a PRF pre-image $\mathtt{r}'$, which must be useful at time t since $\mathtt{r}$ is. If $u \in S(t, \mathtt{r}')$ we are in Case 2, else we are in Cases 4 or 5. If $\mathtt{r}$ does not admit a PRF pre-image, we study whether there exist $\mathtt{r}_1$ and $\mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$. In this case at least one of $\mathtt{r}_1$ and $\mathtt{r}_2$ must be useful at time t since $\mathtt{r}$ is. If $u \in S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2)$, then we are in Case 3. Else we are in Cases 4 or 5. □

Repeatedly applying Lemma 4.2 one can obtain the following result:

Lemma 4.3. *Let $\mathtt{r}$ be of type random coin and useful at time $t \in [t_{\max}]_0$, and u a user such that $u \in S(t, \mathtt{r})$. Then there exists a sequence $\{\mathtt{r}_{1,u,t}, \ldots, \mathtt{r}_{\ell_u,u,t}\}$ such that*

6. *for all i the secret $\mathtt{r}_{i,u,t}$ is useful at time t and $u \in S(t, \mathtt{r}_{i,u,t})$,*
7. *$\mathtt{r}_{\ell_u,u,t} = \mathtt{r}$,*
8. *$\mathtt{r}_{1,u,t} \in \mathsf{ST}_u^{-1}$, and*
9. *for all $i \in \{1, \ldots, \ell_u - 1\}$ one of the following is true*
   (a) *$\mathsf{PRG}(\mathtt{r}_{i,u,t}) = (\mathtt{r}_1, \mathtt{r}_2)$ for some $\mathtt{r}_1$, $\mathtt{r}_2$ such that either $\mathtt{r}_{i+1,u,t} = \mathtt{r}_1$ or $\mathtt{r}_{i+1,u,t} = \mathtt{r}_2$, or*
   (b) *there exists ad such that $\mathsf{PRF}(\mathtt{r}_{i,u,t}, ad) = \mathtt{r}_{i+1,u,t}$, or*
   (c) *there exists $\mathtt{r}'_{i,u,t}$ such that $u \in S(t, \mathtt{r}'_{i,u,t})$ and $\mathsf{dPRF}(\mathtt{r}_{i,u,t}, \mathtt{r}'_{i,u,t}) = \mathtt{r}_{i+1,u,t}$, or*
   (d) *there exists a ciphertext $\mathtt{c}_{i,u,t} \in \bigcup_{\tilde{t} \le t:\ u \in G_{\tilde{t}}} \mathtt{M}_{\tilde{t}}$ such that*

$$\mathtt{c}_{i,u,t} = e_0(\cdot) \circ \ldots \circ e_g(\cdot) \circ \ldots \circ e_h(\mathtt{r}_{i+1,u,t})$$

where all properties of Case 5 are satisfied and $\mathtt{r}_{i,u,t} = \mathtt{r}_g$ the secret used in $e_g = \mathsf{SE.Enc}(\mathtt{r}_g, \cdot)$.

Observe that ℓ_u depends on u, t and $\mathtt{r}$, so in some cases we make this explicit and write $\ell_{u,t,\mathtt{r}}$ or just $\ell_{u,t}$ if the random coin $\mathtt{r}$ is clear from context.

Proof. Let $\mathtt{r} \leftarrow \mathtt{r}$ and $\mathsf{Seq} \leftarrow \emptyset$. Repeat $(\mathtt{r}, \mathsf{Seq}) \leftarrow f(\mathtt{r}, \mathsf{Seq})$ until $\mathtt{r} = \text{STOP}$ where:

$$f(\mathtt{r}, \mathsf{Seq}) = \begin{cases} \text{if we are in Case 1, do } (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}', \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 2, do } (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}', \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 3 and } \mathtt{r}_i \text{ is useful, do } (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}_i, \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 4, do} (\mathtt{r}, \mathsf{Seq}) \leftarrow (\text{STOP}, \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 5, do} (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}_g, \{\mathtt{r}\} \cup \mathsf{Seq}). \end{cases}$$

By construction, Properties 6, 7, 8, as well as one of Properties 9a to 9d are clearly satisfied by Seq at every point in time. This process must end since we require that only a finite number of derivation steps is made by the ME algorithms. □

Now we follow the approach of [MP04] in order to construct a graph for each round and use it to establish a connection between the sets in $\mathcal{S}_t$ and those in $\mathcal{S}_{t-1}$ obtaining a similar result to the one in [ACPP23].

The sequences constructed in Lemma 4.3 suggest considering the following graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ for $t \in [t_{\max} - 1]_0$. The set of nodes $\mathcal{V}_t$ is a subset of useful random coins at time t which corresponds to the elements of the sequences $\{\mathtt{k}^t_{1,u,t}, \ldots, \mathtt{k}^t_{\ell_u,u,t}\}$ associated to the group key $\mathtt{k}^t = \mathtt{k}^t_{\ell_u,u,t}$ and each user $u \in S(t, \mathtt{k}^t)$. The set of edges $\mathcal{E}_t$ consists of all pairs of the form $(\mathtt{k}^t_{i,u,t}, \mathtt{k}^t_{i+1,u,t})$.

In the case that an edge $(\mathtt{k}^t_{i,u,t}, \mathtt{k}^t_{i+1,u,t})$ is obtained from Property 9c, i.e., there exists $\mathtt{r}'_{i,u,t}$ such that $\mathsf{dPRF}(\mathtt{r}_{i,u,t}, \mathtt{r}'_{i,u,t}) = \mathtt{r}_{i+1,u,t}$ and $u \in S(t, \mathtt{r}'_{i,u,t})$, one can construct the sequence from Lemma 4.3 using the secret $\mathtt{r}'_{i,u,t}$ instead of $\mathtt{r}_{i,u,t}$ when both $\mathtt{r}_{i,u,t}$ and $\mathtt{r}'_{i,u,t}$ are useful at time t. If this happens, we make the same choice for all users in order to guarantee that one dPRF pre-image (Case 9c) does not result in two edges in $\mathcal{E}_t$. This is possible since $u \in S(t, \mathtt{r}_{i,u,t}) \cap S(t, \mathtt{r}'_{i,u,t})$.

If an edge $(\mathtt{k}^t_{i,u,t}, \mathtt{k}^t_{i+1,u,t})$ satisfies Properties 9a, 9b or 9c we refer to it as a trivial edge, while we refer to an edge that satisfies Property 9d as a communication edge. The graph $\mathcal{G}_t$ has some basic properties which we state in the following result.

Lemma 4.4. *Let ME be a correct and secure ME scheme. Consider an execution of game $\mathrm{SEC}^{\mathsf{ME}}$ on input $(n_{\max}, t_{\max}, G_0)$ and $(A_t, D_t)_{t=1}^{t_{\max}}$ such that $D_t \subseteq G_{t-1}$ and $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_t)$ for all $t \in [t_{\max}]$. Let $t \in \{0, \ldots, t_{\max} - 1\}$ and $\mathtt{k}^t$ denote the group key at time t output by $\mathsf{ME.Key}$ in Line 22. Then the following properties of the graph $\mathcal{G}_t$ are true.*

10. *For every $u \in S(t, \mathtt{k}^t)$, the node $\mathtt{k}^t_{1,u,t}$ has no incoming edges and $\mathtt{k}^t_{1,u,t} \neq \mathtt{k}^t_{1,v,t}$ for all $u \neq v$. Actually it holds that $S(t, \mathtt{k}^{-1}_u) = \{u\}$.*
11. *For every node $\mathtt{k}^t_{i,u,t}$ there exists at most one node $\mathtt{r}$ such that $\mathsf{PRG}(\mathtt{r}) = (\mathtt{r}'_1, \mathtt{r}'_2)$ and $\mathtt{r}'_j = \mathtt{k}^t_{i,u,t}$ for some $j \in \{1, 2\}$, or that $\mathsf{PRF}(\mathtt{r}, ad) = \mathtt{k}^t_{i,u,t}$ for some ad, or that $\mathsf{dPRF}(\mathtt{r}, \mathtt{r}') = \mathtt{k}^t_{i,u,t}$ or $\mathsf{dPRF}(\mathtt{r}', \mathtt{r}) = \mathtt{k}^t_{i,u,t}$ for some $\mathtt{r}'$ (where $\mathtt{r}'$ may not be in $\mathcal{V}_t$).*
12. *There exists at most one user u in $S(t, \mathtt{k}^t)$ such that for all $1 \leq i \leq \ell_u - 1$ the edge $(\mathtt{r}_{i,u,t}, \mathtt{r}_{i+1,u,t})$ is a trivial edge.*
13. *If $D_{t+1} \neq \emptyset$, then for every $u \in S(t, \mathtt{k}^t) \setminus D_{t+1}$, there exists $j_{u,t}$ such that $1 \leq j_{u,t} < \ell_{u,t}$ and for the corresponding edge $(\mathtt{k}^t_{j_{u,t},u,t}, \mathtt{k}^t_{j_{u,t}+1,u,t}) \in \mathcal{E}_t$ there exists a user $v \in D_{t+1}$ such that $v \in S(t, \mathtt{k}^t_{j_{u,t}+1,u,t})$ and for all $w \in D_{t+1}$ we have $w \notin S(t, \mathtt{k}^t_{j_{u,t},u,t})$. Moreover, $j_{u,t}$ will denote the least integer in $\{1, \dots, \ell_{u,t} - 1\}$ with this property.*

Proof. Since $\mathtt{ST}^{-1}_u = \{\mathtt{k}^{-1}_u\}$, it follows from Property 8 that $\mathtt{k}^t_{1,u,t} = \mathtt{k}^{-1}_u$. If there exists $(\mathtt{r}, \mathtt{k}^{-1}_u) \in \mathcal{E}_t$, then there exists a user $v \in S(t, \mathtt{k}^t)$ such that $u \neq v$ and $\mathtt{k}^t_{i,v,t} = \mathtt{r}$ and $\mathtt{k}^t_{i+1,v,t} = \mathtt{k}^{-1}_u$ by definition of $\mathcal{G}_t$. By Property 6 applied to $\mathtt{k}^t_{i+1,v,t}$, we obtain $v \in S(t, \mathtt{k}^{-1}_u)$. This would imply that it would not be secure to remove user v in round $t+1$ while maintaining u in the group. Indeed,

$$\mathtt{k}^{t+1} \in \mathsf{Der}(\mathtt{ST}^{-1}_u, (\mathtt{M}_{t'})_{t' \leq t+1 :\, u \in G_{t'}}) \subseteq \mathsf{Der}\Big((\mathtt{M}_{t'})_{t'=0}^{t+1}, ((\mathtt{ST}^{t'}_u)_{t'=-1}^{t+1})_{u \in [n_{\max}] \setminus G_t}\Big).$$

We have actually shown that $v \in S(t, \mathtt{k}^{-1}_u)$ implies $v = u$. Therefore $S(t, \mathtt{k}^{-1}_u) = \{u\}$. This completes the proof of Property 10.

Property 11 follows directly from the properties of the symbolic model and the fact that when constructing $\mathcal{G}_t$ we choose only one edge of the two possible for dPRF evaluations.

Property 12 is a direct consequence of the two previous properties.

Property 13 follows from the observation that the node $\mathtt{k}_{\ell_u,u,t} = \mathtt{k}^t$ satisfies the first condition for all users in D_{t+1} and the node $\mathtt{k}_{1,u,t} = \mathtt{k}^{-1}_u$ satisfies the second condition (by Property 10). Since $D_{t+1} \neq \emptyset$ by assumption, there must exist an edge with the required property. □

4.2 Lower Bound on Batched Replacements

We now show that a subset $\tilde{\mathcal{S}}_t$ of the set system $\mathcal{S}_t$ defined above satisfies the properties of the combinatorial model regarding correctness and security and, additionally, that the amount of ciphertexts sent in the symbolic model matches the cost function of Section 3.1 (with respect to the set system $\tilde{\mathcal{S}}_t$) up to a multiplicative loss of 3. As a consequence, the lower bound derived in Section 3.2 applies to batched replacements in multicast in the symbolic model.

Lemma 4.5. *Let $n_{\max}$ and $t_{\max}$ be in $\mathbb{N}$ and $(d_t)_{t=1}^{t_{\max}}$ such that $d_t \leq n_{\max}$ for all t. Let ME be a correct and secure ME scheme. Consider an execution of game $\mathrm{SEC}^{\mathsf{ME}}$ on input $(n_{\max}, t_{\max}, G_0)$ and $(A_t, D_t)_{t=1}^{t_{\max}}$ such that $D_t \subseteq G_{t-1}$ and $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'})$ for all $t \in [t_{\max}]$. Let $(\mathcal{S}_t)_{t=0}^{t_{\max}}$ be the associated set system as defined in Definition 4.1. Further, for $t \in [t_{\max}]_0$ let*

$$\tilde{\mathcal{S}}_t = \left\{ S \in \mathcal{S}_t \;\middle|\; \begin{array}{l} \exists \mathtt{r} \text{ such that } S = S(t, \mathtt{r}),\ \mathtt{r} \text{ is useful at time } t \text{ and} \\ \nexists \mathtt{r}_1, \mathtt{r}_2 \text{ such that } \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r} \text{ and } S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{r}) \end{array} \right\}.$$

Then it holds that

- *(i)* *$G_t \in \tilde{\mathcal{S}}_t$ for all $t \in [t_{\max}]_0$,*
- *(ii)* *$S \subseteq G_t$ for all $S \in \mathcal{S}_t$ and, in particular, $S \subseteq G_t$ for all $S \in \tilde{\mathcal{S}}_t$*
- *(iii)* *$\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq 1/3 \cdot \sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)$, where $\mathrm{Cost}(t)$ is the cost function defined in Section 3.1 with respect to $\tilde{\mathcal{S}}_t$, namely:*

$$\mathrm{Cost}(t) = (\mathrm{SizeMinCov}(G_t, \tilde{\mathcal{S}}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1) + |\{S \in \tilde{\mathcal{S}}_{t-1} : S \cap D_t \neq \emptyset \text{ and } |S| > 1\}|.$$

The reason for introducing $\tilde{\mathcal{S}}_t$ is that allowing the use of dPRFs means that the original set $\mathcal{S}_t$ also contains the intersection of any pair of sets such that one of the associated secrets is useful and this does not require any additional communication. Before turning to the lemma's proof we state our bound on the communication complexity of batched replacements in multicast encryption, which follows directly by applying Theorem 3.4 to set system $(\tilde{\mathcal{S}}_t)_{t=0}^{t_{\max}}$ which is possible due to Lemma 4.5.

Corollary 4.6. *Let $n \leq n_{\max}$ and $t_{\max}$ be in $\mathbb{N}$ and $(d_t)_{t=1}^{t_{\max}}$ such that $d_t \leq n$ for all t. Let ME be a correct and secure ME scheme. Consider an execution of game $\mathrm{SEC}^{\mathsf{ME}}$ on input $(n_{\max}, t_{\max}, G_0)$ and $(A_t, D_t)_{t=1}^{t_{\max}}$ where $|G_0| = n$ and, for all t, the set D_t of removed users is sampled uniformly at random from the set $\{D \subseteq G_{t-1} \mid |D| = d_t\}$ and A_t can be arbitrary according to the restrictions $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'})$ and $|A_t| = |D_t|$. Then, it holds that*

$$\mathbb{E}\left[\sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{\ln(2)}{3} \sum_{t=1}^{t_{\max}} d_t \log\left(\frac{n}{d_t}\right),$$

where the expectation is taken over the choice of $(D_t)_t$. In particular, if $d_t = d$ for all t, then

$$\mathbb{E}\left[\sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{\ln(2)}{3} t_{\max} \cdot d \cdot \log\left(\frac{n}{d}\right).$$

Proof of Lemma 4.5. We start proving Property (ii). Let $S = S(t, \mathtt{r}) \in \mathcal{S}_t$ and $u \in S$. By definition of $S(t, \mathtt{r})$ we have that $\mathtt{r} \in \mathsf{Der}(\mathtt{ST}_u^{-1}, (\mathtt{M}_{t'})_{t' \leq t \colon u \in G_{t'}})$ and since $\mathtt{r}$ is useful at time t it holds that $\mathtt{r} \notin \mathsf{Der}((\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t})_{u \notin G_t})$. Thus $u \in G_t$ as claimed in Property (ii).

Now we proceed to show that Property (i) is true. Recall that $\mathtt{ST}_u^t \subseteq \mathsf{Der}(\mathtt{ST}_u^{-1}, (\mathtt{M}_{t'})_{t' \leq t \colon u \in G_{t'}})$. By correctness there exists a key $\mathtt{k}^t$ such that $\mathtt{k}^t = \mathsf{ME.Key}(\mathtt{ST}_u^t)$ for all users $u \in G_t$ and by security we have that $\mathtt{k}^t \notin \mathsf{Der}((\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t})_{u \notin G_t})$, so $S(t, \mathtt{k}^t) = G_t$ and $S(t, \mathtt{k}^t) \in \mathcal{S}_t$. Moreover, assume that there exist $\mathtt{r}_1, \mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{k}^t$ and $S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{k}^t) = G_t$. Since $\mathtt{k}^t$ is useful at time t, there exists $i \in \{1, 2\}$ such that $\mathtt{r}_i$ is useful at time t. By Property (ii) it must hold that $S(t, \mathtt{r}_i) \subseteq G_t$. The fact that $S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{k}^t) = G_t$ implies that we also have $G_t \subseteq S(t, \mathtt{r}_i)$. Therefore $S(t, \mathtt{r}_i) = G_t$. By repeating this process we can find a secret $\mathtt{r}$ that is useful at time t such that $S(t, \mathtt{r}) = G_t$ and that satisfies the property that $\nexists \mathtt{r}_1, \mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$ and $S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{r})$. This shows that $G_t \in \tilde{\mathcal{S}}_t$ as claimed in Property (i). Observe that we have shown $S(t, \mathtt{k}^t) = G_t$ and not just $G_t \in \tilde{\mathcal{S}}_t$.

We now proceed to prove Property (iii). We divide the proof into showing each of the following two equations separately:

$$\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq \frac{1}{2} \sum_{t=0}^{t_{\max}} (\mathrm{SizeMinCov}(G_t, \tilde{\mathcal{S}}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1) \tag{5}$$

$$\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq \sum_{t=1}^{t_{\max}} |\{S \in \tilde{\mathcal{S}}_{t-1} : S \cap D_t \neq \emptyset \text{ and } |S| > 1\}|. \tag{6}$$

Let $t \in \{1, \ldots, t_{\max}\}$ and denote by $\mathtt{k}^t$ the group key of round t. If $D_t = \emptyset$, $\mathrm{SizeMinCov}(G_t, \tilde{\mathcal{S}}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1 = 0$, so we assume that $D_t \neq \emptyset$. In order to construct a cover of G_t we give a cover of $G_t \setminus A_t$ and a cover of A_t. Observe that $u \in G_t \setminus A_t$ if and only if $u \in G_{t-1} \setminus D_t$.

For each $u \in G_{t-1} \setminus D_t$, we consider the index $j_{u,t-1}$ from Property 13. We claim that

$$\mathcal{C}_{t,1} = \{S(t-1, \mathtt{k}_{j_{u,t-1},u,t-1}^{t-1}) \mid u \in G_{t-1} \setminus D_t\}$$

is a cover of $G_{t-1} \setminus D_t$ and $\mathcal{C}_{t,1} \subseteq \mathcal{S}_{t-1}$. The fact that $u \in S(t-1, \mathtt{k}_{j_{u,t-1},u,t-1}^{t-1})$ and $S(t-1, \mathtt{k}_{j_{u,t-1},u,t-1}^{t-1}) \in \mathcal{S}_{t-1}$ is a consequence of Property 6. It also holds that $S(t-1, \mathtt{k}_{j_{u,t-1},u,t-1}^{t-1}) \subseteq G_{t-1} \setminus D_t$ by Properties 13 and (ii).

Let $d_{j_{u,t-1}+1}$ denote the in-degree of the node $\mathtt{k}_{j_{u,t-1}+1,u,t-1}^{t-1}$ in $\mathcal{G}_{t-1}$ and let $u_1 = u, u_2, \ldots, u_h$ be users such that $\mathtt{k}_{j_{u,t-1}+1,u,t-1}^{t-1} = \mathtt{k}_{j_{u_i,t-1}+1,u_i,t-1}^{t-1}$ for $i = 1, \ldots, h$ and for any user v with $\mathtt{k}_{j_{v,t-1}+1,v,t-1}^{t-1} =$

$\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$, there exists a unique $i \in \{1,\dots,h\}$ such that $\mathtt{k}^{t-1}_{j_{v,t-1},v,t-1} = \mathtt{k}^{t-1}_{j_{u_i,t-1},u_i,t-1}$. I.e., we choose exactly one user u_i for each of the incoming edges of the node $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ that satisfy Property 13. Therefore $d_{j_{u,t-1}+1} \geq h$. Each node $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ has at most one incoming trivial edge (Property 11). All the other $d_{j_{u,t-1}+1} - 1$ incoming edges correspond to ciphertexts in $\bigcup_{t'=0}^{t-1} \mathtt{M}_{t'}$. If $d_{j_{u,t-1}+1} \geq h+1$, then the node $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ contributes to $\mathcal{C}_{t,1}$ with at most $h \leq d_{j_{u,t-1}+1} - 1$ sets. If $d_{j_{u,t-1}+1} = h$, then we can consider the graph we would obtain for the secret $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ rather than $\mathtt{k}^{t-1}$. Let's denote it $\mathcal{H}$. The set of nodes corresponds to the elements of the sequences constructed in Lemma 4.3 for the secret $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ and the set of edges consists of all pairs of consecutive elements in those sequences. Since the node $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ in $\mathcal{H}$ has at least one additional incoming edge that corresponds to the user $v \in D_t$ guaranteed to exist by Property 13 for $\mathcal{G}_{t-1}$, which does not contribute to $\mathcal{C}_{t,1}$, the node $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ contributes to $\mathcal{C}_{t,1}$ with at most $h = d_{j_{u,t-1}+1}$ sets and we have at least $d_{j_{u,t-1}+1}$ ciphertexts in $\bigcup_{t'=0}^{t-1} \mathtt{M}_{t'}$. Thus $|\mathcal{C}_{t,1}| \leq \sum_{t'=0}^{t-1} |\mathtt{M}_{t'}|$.

We observe that $S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}) \cap D_t \neq \emptyset$. Therefore, $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ is not a useful random coin at time t. This guarantees that the node $\mathtt{k}^{t-1}_{j_{u,t-1}+1,u,t-1}$ will not be in $\mathcal{G}_{\tilde{t}}$ for $\tilde{t} \geq t$. Therefore $\sum_{t=1}^{t_{\max}} |\mathcal{C}_{t,1}| \leq \sum_{t=0}^{t_{\max}-1} |\mathtt{M}_t|$.

Moreover, we can find covers $\mathcal{C}'_{t,1} \subseteq \tilde{\mathcal{S}}_{t-1}$ such that $\sum_{t=1}^{t_{\max}} |\mathcal{C}'_{t,1}| \leq \sum_{t=0}^{t_{\max}-1} |\mathtt{M}_t|$. We obtain $\mathcal{C}'_{t,1}$ from $\mathcal{C}_{t,1}$ by substituting the sets that are in $\mathcal{S}_{t-1} \setminus \tilde{\mathcal{S}}_{t-1}$ for sets in $\tilde{\mathcal{S}}_{t-1}$. Let's assume that there exist $\mathtt{r}_1, \mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{k}^{t-1}_{j_{u,t-1},u,t-1}$ and $S(t-1, \mathtt{r}_1) \cap S(t-1, \mathtt{r}_2) = S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1},u,t-1})$. Since $\mathtt{k}_{j_{u,t-1},u,t-1}$ is useful at time $t-1$, there exists $i \in \{1,2\}$ such that $\mathtt{r}_i$ is useful at time $t-1$. From $S(t-1, \mathtt{r}_1) \cap S(t-1, \mathtt{r}_2) = S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1},u,t-1})$ and the way sequences are constructed in Lemma 4.3, it follows that $j_{u,t-1} > 1$, $\mathtt{k}^{t-1}_{j_{u,t-1}-1,u,t-1} = \mathtt{r}_i$. From $S(t-1, \mathtt{r}_1) \cap S(t-1, \mathtt{r}_2) = S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1},u,t-1})$, we obtain $S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1},u,t-1}) \subseteq S(t-1, \mathtt{r}_i) = S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1}-1,u,t-1})$. The minimality condition imposed on $j_{u,t}$ by Property 13 guarantees that $S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1}-1,u,t-1}) \cap D_t = \emptyset$. This shows that $\mathcal{C}'_{t,1} = (\mathcal{C}_{t,1} \setminus \{S(t-1, \mathtt{k}^{t-1}_{j_{u,t-1},u,t-1})\}) \cup \{S(t-1, \mathtt{r}_i)\}$ is also a cover of $G_{t-1} \setminus D_t$ and it has the same size as $\mathcal{C}_{t,1}$. Therefore by repeating this process we obtain a cover of $G_{t-1} \setminus D_t$ with respect to $\tilde{\mathcal{S}}_{t-1}$ and it has at most as many sets as the original $\mathcal{C}_{t,1}$. We denote it $\mathcal{C}'_{t,1}$ and it holds that

$$\sum_{t=1}^{t_{\max}} |\mathcal{C}'_{t,1}| \leq \sum_{t=0}^{t_{\max}-1} |\mathtt{M}_t|. \tag{7}$$

Now we give a cover of A_t. The argument also considers the case where $t = 0$ if we define $A_0 = G_0$. For each $u \in A_t$, let $i_{u,t} \in \{1,\dots,\ell_{u,t}\}$ be maximal such that for all $1 \leq j < i_{u,t}$, $(\mathtt{k}^t_{j,u,t}, \mathtt{k}^t_{j+1,u,t})$ is a trivial edge. By Property 12, there exists at most one user in A_t such that $i_{u,t} = \ell_{u,t}$. For every user $u \in A_t$ such that $i_{u,t} < \ell_{u,t}$, there exists a ciphertext $\mathtt{c}_{i_{u,t},u,t}$ as proven in Property 9d. All these ciphertexts must be different or else there would exist users $u, v \in A_t$ such that $\mathtt{k}^t_{i_{u,t},u,t} = \mathtt{k}^t_{i_{v,t},v,t}$, which would contradict Properties 11 and 10.

Each of the ciphertexts $\mathtt{c}_{i_{u,t},u,t}$ belongs to $\bigcup_{\tilde{t} \leq t:\, u \in G_{\tilde{t}}} \mathtt{M}_{\tilde{t}}$. From the condition $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_t)$, it follows that $\mathtt{c}_{i_{u,t},u,t} \in \mathtt{M}_t$. Thus we have at least $|A_t| - 1$ many ciphertexts sent in round t. The cover of A_t, $\mathcal{C}_{t,2} = \{\{u\} \mid u \in A_t\}$ satisfies the inequality $|\mathtt{M}_t| \geq |\mathcal{C}_{t,2}| - 1$. From this inequality and Equation 7, we obtain

$$\frac{1}{2}\sum_{t=1}^{t_{\max}} |\mathcal{C}_{t,1}| + \frac{1}{2}\sum_{t=0}^{t_{\max}} (|\mathcal{C}_{t,2}| - 1) \leq \frac{1}{2}\sum_{t=0}^{t_{\max}-1} |\mathtt{M}_t| + \frac{1}{2}\sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \leq \sum_{t=0}^{t_{\max}} |\mathtt{M}_t|.$$

This shows Equation 5 since $|\mathcal{C}'_{t,1}| \leq |\mathcal{C}_{t,1}|$ and $\mathcal{C}'_{t,1} \cup \mathcal{C}_{t,2}$ is a cover of G_t for all $t \in [t_{\max}]_0$ (we take $\mathcal{C}'_{0,1} = \emptyset$).

Now we show Equation 6. Let $S = S(t-1, \mathtt{r}) \in \tilde{\mathcal{S}}_{t-1}$ such that $S \cap D_t \neq \emptyset$, $|S| > 1$, and $\nexists \mathtt{r}_1, \mathtt{r}_2$ that satisfy the following two properties: $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$ and $S(t-1, \mathtt{r}_1) \cap S(t-1, \mathtt{r}_2) = S(t-1, \mathtt{r})$. We proceed to consider the graph $\mathcal{G}_{t-1,\mathtt{r}} = (\mathcal{V}_{t-1,\mathtt{r}}, \mathcal{E}_{t-1,\mathtt{r}})$ where $\mathcal{V}_{t-1,\mathtt{r}}$ is a subset of useful random coins at time $t-1$

which corresponds to the elements of the sequences $\{\mathtt{r}_{1,u,t-1},\ldots,\mathtt{r}_{\ell_u,u,t-1}\}$ constructed in Lemma 4.3 for each user $u \in S(t-1,\mathtt{r})$. The set of edges $\mathcal{E}_{t-1,\mathtt{r}}$ consists of all pairs $(\mathtt{r}_{i,u,t-1},\mathtt{r}_{i+1,u,t-1})$. From Property 12[8] and the fact that $|S|>1$, it follows that not all edges in $\mathcal{E}_{t-1,\mathtt{r}}$ are trivial edges. If the node $\mathtt{r}$ only has only one incoming edge of the form $(\mathtt{r}_{\ell_u-1,u,t-1},\mathtt{r}_{\ell_u,u,t-1}=\mathtt{r})$ for some $u \in S(t-1,\mathtt{r})$ and it is a trivial edge, then $S(t-1,\mathtt{r}_{\ell_u-1,u,t-1}) = S(t-1,\mathtt{r})$. In order to show this we consider two cases:

- if $\nexists \mathtt{r}_1,\mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1,\mathtt{r}_2)=\mathtt{r}$, then $(\mathtt{r}_{\ell_u-1,u,t-1},\mathtt{r}_{\ell_u,u,t-1}=\mathtt{r})$ must correspond to a PRG or a PRF and $S(t-1,\mathtt{r}_{\ell_u-1,u,t-1}) = S(t-1,\mathtt{r})$,
- if $\exists \mathtt{r}_1,\mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1,\mathtt{r}_2)=\mathtt{r}$, but $S(t-1,\mathtt{r}_1)\cap S(t-1,\mathtt{r}_2) \subsetneq S(t-1,\mathtt{r})$, then there exists a user $v \in S(t-1,\mathtt{r})\setminus S(t-1,\mathtt{r}_1)\cap S(t-1,\mathtt{r}_2)$ and by Property 9c $\mathtt{r}_{\ell_v-1,v,t-1} \neq \mathtt{r}_1$ and $\mathtt{r}_{\ell_v-1,v,t-1} \neq \mathtt{r}_2$. This contradicts the assumption that $\mathtt{r}$ had only one incoming edge.

As argued in the previous paragraph we may assume without loss of generality that for some user $u \in S(t-1,\mathtt{r}) = S$ the edge $(\mathtt{r}_{\ell_u-1,u,t-1},\mathtt{r}_{\ell_u,u,t-1}=\mathtt{r})$ corresponds to a ciphertext $\mathtt{c}_{\ell_u-1,u,t} \in \bigcup_{t'=0}^{t-1}\mathtt{M}_{t'}$. This shows that $\sum_{t'=0}^{t-1}|\mathtt{M}_{t'}| \geq |\{S \in \tilde{\mathcal{S}}_{t-1} : S \cap D_t \neq \emptyset \text{ and } |S|>1\}|$. Moreover, the fact that $S \cap D_t \neq \emptyset$ guarantees that $\mathtt{r}$ does not appear in $\mathcal{G}_{\tilde{t}-1,\tilde{\mathtt{r}}}$ for any $\tilde{t} > t$ and useful $\tilde{\mathtt{r}}$ at time $\tilde{t}$ by Property (ii). Thus,

$$\sum_{t=0}^{t_{\max}-1}|\mathtt{M}_t| \geq \sum_{t=1}^{t_{\max}}|\{S \in \tilde{\mathcal{S}}_{t-1} : S \cap D_t \neq \emptyset \text{ and } |S|>1\}|.$$

Finally we multiply Equation 5 by 2/3 and Equation 6 by 1/3 and add them together to obtain $\sum_{t=0}^{t_{\max}}|\mathtt{M}_t| \geq 1/3 \cdot \sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)$, as desired.

□

[8] Property 12 was shown for the graph $\mathcal{G}_t$ and the same argument shows that this property also holds for $\mathcal{G}_{t-1,\mathtt{r}}$.

References

[AAB+21] Joël Alwen, Benedikt Auerbach, Mirza Ahad Baig, Miguel Cueto Noval, Karen Klein, Guillermo Pascual-Perez, Krzysztof Pietrzak, and Michael Walter. Grafting key trees: Efficient key management for overlapping groups. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part III*, volume 13044 of *LNCS*, pages 222–253. Springer, Cham, November 2021.

[AAN+22a] Joël Alwen, Benedikt Auerbach, Miguel Cueto Noval, Karen Klein, Guillermo Pascual-Perez, Krzysztof Pietrzak, and Michael Walter. CoCoA: Concurrent continuous group key agreement. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 815–844. Springer, Cham, May / June 2022.

[AAN+22b] Joël Alwen, Benedikt Auerbach, Miguel Cueto Noval, Karen Klein, Guillermo Pascual-Perez, and Krzysztof Pietrzak. Decaf: Decentralizable continuous group key agreement with fast healing. Cryptology ePrint Archive, Paper 2022/559, 2022. https://eprint.iacr.org/2022/559.

[ACDT20] Joël Alwen, Sandro Coretti, Yevgeniy Dodis, and Yiannis Tselekounis. Security analysis and improvements for the IETF MLS standard for group messaging. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 248–277. Springer, Cham, August 2020.

[ACPP23] Benedikt Auerbach, Miguel Cueto Noval, Guillermo Pascual-Perez, and Krzysztof Pietrzak. On the cost of post-compromise security in concurrent continuous group-key agreement. In Guy Rothblum and Hoeteck Wee, editors, *TCC 2023, Part III*, volume 14371 of *LNCS*, pages 271–300. Springer, Heidelberg, Nov 2023.

[AHKM22] Joël Alwen, Dominik Hartmann, Eike Kiltz, and Marta Mularczyk. Server-aided continuous group key agreement. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 69–82. ACM Press, November 2022.

[BBR+23] Richard Barnes, Benjamin Beurdouche, Raphael Robert, Jon Millican, Emad Omara, and Katriel Cohn-Gordon. The Messaging Layer Security (MLS) Protocol. RFC 9420, July 2023.

[BDG+22] Alexander Bienstock, Yevgeniy Dodis, Sanjam Garg, Garrison Grogan, Mohammad Hajiabadi, and Paul Rösler. On the worst-case inefficiency of CGKA. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part II*, volume 13748 of *LNCS*, pages 213–243. Springer, Cham, November 2022.

[BDR20] Alexander Bienstock, Yevgeniy Dodis, and Paul Rösler. On the price of concurrency in group ratcheting protocols. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 198–228. Springer, Cham, November 2020.

[BDT22] Alexander Bienstock, Yevgeniy Dodis, and Yi Tang. Multicast key agreement, revisited. In Steven D. Galbraith, editor, *CT-RSA 2022*, volume 13161 of *LNCS*, pages 1–25. Springer, Cham, March 2022.

[Bol65] Béla Bollobás. On generalized graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 16(3):447–452, 1965.

[CEK+99] Isabella Chang, Robert Engel, Dilip Kandlur, Dimitrios Pendarakis, and Debanjan Saha. Key management for secure lnternet multicast using boolean function minimization techniques. 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)*, volume 2, pages 689–698. IEEE, 1999.

[CGI$^+$99] Ran Canetti, Juan Garay, Gene Itkis, Daniele Micciancio, Moni Naor, and Benny Pinkas. Multicast security: A taxonomy and some efficient constructions. In *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, volume 2, pages 708–716. IEEE, 1999.

[CMN99] Ran Canetti, Tal Malkin, and Kobbi Nissim. Efficient communication-storage tradeoffs for multicast encryption. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 459–474. Springer, Berlin, Heidelberg, May 1999.

[DMS00] Lakshminath R Dondeti, Sarit Mukherjee, and Ashok Samal. Scalable secure one-to-many group communication using dual encryption. *Computer Communications*, 23(17):1681–1701, 2000.

[DY83] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[HKP$^+$21] Keitaro Hashimoto, Shuichi Katsumata, Eamonn Postlethwaite, Thomas Prest, and Bas Westerbaan. A concrete treatment of efficient continuous group key agreement via multi-recipient PKEs. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 1441–1462. ACM Press, November 2021.

[KPPW$^+$21] Karen Klein, Guillermo Pascual-Perez, Michael Walter, Chethan Kamath, Margarita Capretto, Miguel Cueto, Ilia Markov, Michelle Yeo, Joël Alwen, and Krzysztof Pietrzak. Keep the dirt: Tainted TreeKEM, adaptively and actively secure continuous group key agreement. In *2021 IEEE Symposium on Security and Privacy*, pages 268–284. IEEE Computer Society Press, May 2021.

[LYGL01] Xiaozhou Steve Li, Yang Richard Yang, Mohamed G. Gouda, and Simon S. Lam. Batch rekeying for secure group communications. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 525–534, New York, NY, USA, 2001. Association for Computing Machinery.

[MP04] Daniele Micciancio and Saurabh Panjwani. Optimal communication complexity of generic multicast key distribution. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 153–170. Springer, Berlin, Heidelberg, May 2004.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, Berlin, Heidelberg, August 2001.

[SM03] Alan T Sherman and David A McGrew. Key establishment in large dynamic groups using one-way function trees. *IEEE transactions on Software Engineering*, 29(5):444–458, 2003.

[SSV01] J. Snoeyink, S. Suri, and G. Varghese. A lower bound for multicast key distribution. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, volume 1, pages 422–431 vol.1, 2001.

[WGL00] Chung Kei Wong, Mohamed Gouda, and Simon S Lam. Secure group communications using key graphs. *IEEE/ACM transactions on networking*, 8(1):16–30, 2000.

[WHA99] Debby Wallner, Eric Harder, and Ryan Agee. Key management for multicast: Issues and architectures. Request for Comments 2627, Internet Engineering Task Force, June 1999.

[YLZL02] Yang Yang, X. Li, X. Zhang, and Simon Lam. Reliable group rekeying: A performance analysis. *ACM SIGCOMM Computer Communication Review*, 01 2002.

A Lower Bound for Batched Replacements in Continuous Group-key Agreement

In this section we show that the lower bound of Section 3 in the combinatorial model carries over to the setting of batched replacements of users in continuous group-key agreement schemes in the symbolic model. In Section A.1 we define the symbolic model and provide syntax for continuous group-key agreement, in Section A.2 we prove our lower bound.

A.1 Continuous Group-key Agreement in the symbolic model

Building blocks and entailment relation. We now give syntax for continuous group-key agreement (CGKA) in the symbolic model, focusing on batched replacement of users. As in Section 4, we first discuss the covered building blocks, i.e., public-key encryption (PKE), pseudorandom generators (PRG), pseudorandom functions (PRF) and dual pseudorandom functions (dPRF). Further, our bound also extends to the setting allowing for key-updatable public-key encryption (kuPKE) as an additional building block. As it makes for a cleaner presentation, instead of adding kuPKE to the considered building blocks in this section we prove this in Section A.3. We consider symbolic variables of the following three types; (pseudo)random coins denoted by $\texttt{r}$, public keys $\texttt{pk}$, and messages $\texttt{m}$. Random coins of type $\texttt{r}$ also serve as secret keys for PKE schemes, and we will typically denote them by $\texttt{sk}$ if they are used in that context. Similarly, ciphertexts are considered to be of message type $\texttt{m}$ and we will denote them by $\texttt{c}$ if used in this context. As in Section 3, symbolic variables are depicted using typewriter font to distinguish them from their non-symbolic counterparts. Upper case typewriter letters are used for sets of symbolic variables.

- A public-key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ specifies the following. Key generation algorithm $\mathsf{PKE.Gen}(\texttt{sk})$ receives as input a secret key $\texttt{sk}$ (of type $\texttt{r}$) and returns the corresponding public key $\texttt{pk}$. Encryption algorithm $\mathsf{PKE.Enc}(\texttt{pk}, \texttt{m})$, on input public key $\texttt{pk}$ and message $\texttt{m}$, returns a ciphertext $\texttt{c}$ (of message type). Decryption algorithm $\mathsf{PKE.Dec}(\texttt{sk}, \texttt{c})$, on input secret key $\texttt{sk}$ and ciphertext $\texttt{c}$, returns a message $\texttt{m}$. We require perfect correctness, i.e., $\mathsf{PKE.Dec}(\texttt{sk}, \mathsf{PKE.Enc}(\texttt{pk}, \texttt{m})) = \texttt{m}$ for all $\texttt{m}$ and all $\texttt{sk}, \texttt{pk}$ with $\texttt{pk} = \mathsf{PKE.Gen}(\texttt{sk})$.
- A pseudorandom generator PRG of stretch 2 is defined as in Section 4, i.e., $\mathsf{PRG}(\texttt{r})$ on input random string $\texttt{r}$ returns $(\texttt{r}_1, \texttt{r}_2)$ consisting of two pseudorandom strings.
- A pseudorandom function $\mathsf{PRF}(\texttt{r}, ad)$ takes a random string $\texttt{r}$ and some non-symbolic associated data ad as input and returns a pseudorandom string $\texttt{r}'$.
- On input two random string $\texttt{r}_1, \texttt{r}_2$, a dual pseudorandom function $\mathsf{dPRF}(\texttt{r}_1, \texttt{r}_2)$ returns a pseudorandom string $\texttt{r}' = \mathsf{dPRF}(\texttt{r}_1, \texttt{r}_2) = \mathsf{dPRF}(\texttt{r}_2, \texttt{r}_1)$.
- A secret sharing scheme is given by two algorithms S and R. S takes as input a message $\texttt{m}$ and generates a set of shares $\mathsf{S}(\texttt{m}) = \{\mathsf{S}_i(\texttt{m})\}_{i \in [s]}$ of type message. The original message can be recovered given some subset of shares as determined by an access structure $\Gamma \subseteq 2^{[s]}$, namely, for every $I \in \Gamma$, $\mathsf{R}(I, \{\mathsf{S}_i(\texttt{m})\}_{i \in I}) = \texttt{m}$.

The grammar rules, and the rules for the entailment relation $\vdash$ for PKE, PRGs, PRFs, dPRFs and the secret sharing scheme are very similar to the ones used in Section 4.

variable type		grammar rule
$\mathtt{r}$	$\leftarrow$	terminal type, $\mathsf{PRG}(\mathtt{r}), \mathsf{PRF}(\mathtt{r}), \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2)$
$\mathtt{pk}$	$\leftarrow$	$\mathsf{PKE.Gen}(\mathtt{r})$
$\mathtt{m}$	$\leftarrow$	$\mathtt{r}$, $\mathtt{pk}$, $\mathsf{PKE.Enc}(\mathtt{r}, \mathtt{m})$

entailment relation		
$\mathtt{m} \in \mathtt{M}$	$\Rightarrow$	$\mathtt{M} \vdash \mathtt{m}$
$\mathtt{M} \vdash \mathtt{r}$	$\Rightarrow$	$\mathtt{M} \vdash \mathsf{PRG}(\mathtt{r}) = (\mathtt{r}_1, \mathtt{r}_2)$
$\mathtt{M} \vdash \mathtt{r}$	$\Rightarrow$	$\forall ad \colon \mathtt{M} \vdash \mathsf{PRF}(\mathtt{r}, ad)$
$\mathtt{M} \vdash \mathtt{r}_1, \mathtt{r}_2$	$\Rightarrow$	$\mathtt{M} \vdash \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2)$
$\mathtt{M} \vdash \mathtt{pk}, \mathtt{m}$	$\Rightarrow$	$\mathtt{M} \vdash \mathsf{PKE.Enc}(\mathtt{pk}, \mathtt{m})$
$\mathtt{M} \vdash (\mathtt{r}, \mathtt{c}) \colon \mathtt{c} = \mathsf{PKE.Enc}(\mathtt{pk}, \mathtt{m})$ and $\mathtt{pk} = \mathsf{PKE.Gen}(\mathtt{r})$	$\Rightarrow$	$\mathtt{M} \vdash \mathtt{m}$
$\exists I \in \Gamma \colon \mathtt{M} \vdash \{\mathsf{S}_i(\mathtt{m})\}_{i \in I}$	$\Rightarrow$	$\mathtt{M} \vdash \mathtt{m}$

Thus, (pseudo)random coins can be sampled freshly or derived with a PRG, a PRF or a dPRF, public keys are constructed from random coins using $\mathsf{PKE.Gen}$, and messages can be of arbitrary type and, in particular, include encryptions of other messages under public keys. To a set $\mathtt{M}$ of symbolic variables we associate $\mathsf{Der}(\mathtt{M})$, the set of variables derivable from it, i.e., for $\mathtt{m}$ we have

$$\mathtt{m} \in \mathsf{Der}(\mathtt{M}) \text{ exactly if } \mathtt{M} \vdash \mathtt{m}.$$

CGKA in the symbolic model. A *continuous group-key agreement scheme* allows a group of users to maintain a shared group key evolving over rounds. Opposed to the setting of multicast encryption, however, CGKA does not rely on a trusted central authority that has access to all secrets. Instead, every user keeps track of a personal state and can send protocol messages that are distributed to the other users via an untrusted server. The security goals being (a) that the scheme provides post-compromise security (PCS) by users issuing update operations, which roughly corresponds to the group being able to achieve security even in the presence of past exposures of users' states[9]; and (b) enabling group members to both add new users and remove group members from the group, while maintaining the expected confidentiality guarantees with respect to these added or removed users. The goal of this section is to prove lower bounds on the communication complexity of operations of the latter kind, essentially showing that it is not possible to improve over the "fair-weather" complexity of the add/remove mechanism employed by the MLS standard. Again, for an easier presentation of our bound, we keep the group size constant over the experiment by working with a simplified syntax that, similarly to Section 4.2, instead of allowing for arbitrary updates, adds, and removes, uses a replacement algorithm that can be called to replace a batch of group members with a set of non-members of the same size.

Formally, a CGKA scheme specifies 5 algorithms $\mathsf{CGKA.Setup}$, $\mathsf{CGKA.Init}$, $\mathsf{CGKA.Repl}$, $\mathsf{CGKA.Proc}$, and $\mathsf{CGKA.Key}$.[10] The scheme proceeds in rounds t, each of which establishes a group $G_t \subseteq [n_{\max}]$ and corresponding group key $\mathtt{k}^t$ of type $\mathtt{r}$. Here $[n_{\max}]$, for $n_{\max} \in \mathbb{N}$, is the universe of users. After initialization of the group G_0, in every round, some user u replaces a set of group members with a set of new users of the same size. Afterwards, all users process these operations, resulting in a new group and group key. More precisely, for $t \geq 1$, let A_t and D_t be the sets describing the users added to and removed from the group round t, respectively. Then, the group G_t established at the end of round t is computed from the one of the previous round as

$$G_t = (G_{t-1} \cup A_t) \setminus D_t. \tag{8}$$

We now describe the syntax of the algorithms more formally.

[9] There are also other notions of security like post-compromise forward secrecy (PCFS) that require security even in the presence of both past and future exposures of users' states. By only asking for PCS we obtain a stronger result since we are proving a lower bound.

[10] Our syntax is an adaptation of the one from [ACPP23] to account for dynamic operations. Compared to that of [BDR20], ours separates the setup from group creation (needed to extend the syntax to the dynamic setting), and includes an explicit algorithm outputting the group key, as opposed to it be part of the output of $\mathsf{CGKA.Proc}$.

```
**Game** $\mathrm{SEC}^{\mathsf{CGKA}}((n_{\max}, t_{\max}, G_0, u_0), (u_t, A_t, D_t)_{t=1}^{t_{\max}})$
**sample** $\mathtt{R}_{u_0}^0, \mathtt{R}^{-1}$
**for** $u \in [n_{\max}] \setminus \{u_0\}$:
    $\mathtt{R}_u^0 \leftarrow \emptyset$
$(\mathtt{PUB}, (\mathtt{ST}_u^{-1})_{u \in [n_{\max}]}) \leftarrow \mathsf{CGKA.Setup}(n_{\max}; \mathtt{R}^{-1})$
$(\mathtt{M}_0, \mathtt{ST}_{u_0}^{-1}) \leftarrow \mathsf{CGKA.Init}(\mathtt{ST}_{u_0}^{-1}, \mathtt{PUB}, G_0; \mathtt{R}_{u_0}^0)$
**for** $u \in G_0$:
    $\mathtt{ST}_u^0 \leftarrow \mathsf{CGKA.Proc}(\mathtt{ST}_u^{-1}, \mathtt{PUB}, \mathtt{M}_0)$
    $\mathtt{k}_u^0 \leftarrow \mathsf{CGKA.Key}(\mathtt{ST}_u^0)$
    $\mathtt{k}^0 \leftarrow \mathtt{k}_u^0$
**for** $u \in [n_{\max}] \setminus G_0$:
    $\mathtt{ST}_u^0 \leftarrow \mathtt{ST}_u^{-1}$
**if** $\exists u \in G_0 : \mathtt{k}_u^0 \neq \mathtt{k}^0$:
    **return** 0                                  \\ disagreement on key
**if** $\mathtt{k}^0 \in \mathsf{Der}(\mathtt{PUB}, \mathtt{M}_0, ((\mathtt{ST}_u^{t'})_{t'=-1}^0, \mathtt{R}_u^0)_{u \notin G_0})$:
    **return** 0                                  \\ group key insecure
**for** $t = 1, \ldots, t_{\max}$:
    $\mathsf{ROUND}(A_t, D_t)$
**return** 1

**Oracle** $\mathsf{ROUND}(u_t, A_t, D_t)$
**require** $u_t \in G_{t-1} \setminus D_t$
**require** $D_t \subseteq G_{t-1} \land A_t \subseteq [n_{\max}] \setminus G_{t-1}$
$G_t \leftarrow (G_{t-1} \cup A_t) \setminus D_t$
**sample** $\mathtt{R}_{u_t}^t$
**for** $u \in [n_{\max}] \setminus \{u_t\}$:
    $\mathtt{R}_u^t \leftarrow \emptyset$
$(\mathtt{M}_t, \mathtt{ST}_{u_t}^{t-1}) \leftarrow \mathsf{CGKA.Repl}(\mathtt{ST}_{u_t}^{t-1}, \mathtt{PUB}, A_t, D_t; \mathtt{R}_{u_t}^t)$
**for** $u \in G_t$:
    $\mathtt{ST}_u^t \leftarrow \mathsf{CGKA.Proc}(\mathtt{ST}_u^{t-1}, \mathtt{PUB}, \mathtt{M}_t)$
    $\mathtt{k}_u^t \leftarrow \mathsf{CGKA.Key}(\mathtt{ST}_u^t)$
    $\mathtt{k}^t \leftarrow \mathtt{k}_u^t$
**for** $u \in [n_{\max}] \setminus G_t$:
    $\mathtt{ST}_u^t \leftarrow \mathtt{ST}_u^{t-1}$
**if** $\exists u \in G_t : \mathtt{k}_u^t \neq \mathtt{k}^t$::
    **return** 0                                  \\ disagreement on key
**if** $\mathtt{k}^t \in \mathsf{Der}(\mathtt{PUB}, (\mathtt{M}_{t'})_{t'=0}^t, ((\mathtt{ST}_u^{t'})_{t'=-1}^t, (\mathtt{R}_u^{t'})_{t'=0}^t)_{u \notin G_t})$:
    **return** 0                                  \\ group key insecure
```

Figure 3: Symbolic security and correctness game for continuous group-key agreement scheme CGKA.

- $\mathsf{CGKA.Setup}(n_{\max}; \mathtt{R})$, on input the universe of users and random coins $\mathtt{R}$, outputs public information $(\mathtt{PUB}, pub)$ (e.g., containing every user's initial public key), as well as an initial state $(\mathtt{ST}_u, st_u)$ for every user $u \in [n_{\max}]$. Note that the public information, as well as the users' states, consist of a symbolic and a non-symbolic part.
- $\mathsf{CGKA.Init}((\mathtt{ST}_u, st_u), (\mathtt{PUB}, pub), G_0; \mathtt{R})$ receives as input the user's (initial) state, the public information, the description of a group $G_0 \subseteq [n_{\max}]$, and random coins $\mathtt{R}$. Its output $((\mathtt{ST}'_u, st'_u), (\mathtt{M}_0, M_0))$ consists of the updated state and a control message $(\mathtt{M}_0, M_0)$.
- $\mathsf{CGKA.Repl}((\mathtt{ST}_u, st_u), (\mathtt{PUB}, pub), A_t, D_t; \mathtt{R})$ in round t takes as input a user u's state, where $u \in G_{t-1}$, the public information, a set $A_t \subseteq G_{t-1}$ of users to be added to the group, a set $D_t \subseteq [n_{\max}] \setminus G_{t-1}$ of the same size that collects the users to be added to the group, and random coins $\mathtt{R}$. It returns u's updated state $(\mathtt{ST}'_u, st'_u)$ and a control message $(\mathtt{M}_t, M_t)$.[11]
- Deterministic algorithm $\mathsf{CGKA.Proc}((\mathtt{ST}_u, st_u), (\mathtt{PUB}, pub), (\mathtt{M}, M))$ gets as input a user u's state, the public information, and a protocol message. Its output is an updated state $(\mathtt{ST}'_u, st'_u)$ and a group description $G \subseteq [n_{\max}]$.
- Deterministic algorithm $\mathsf{CGKA.Key}(\mathtt{ST}_u, st_u)$, on input a user's state, returns u's view of current group key $\mathtt{k}$.

Analogously to the setting of multicast encryption, for brevity in the following we will typically drop the non-symbolic parts of the public information $(\mathtt{PUB}, pub)$, control messages $(\mathtt{M}, M)$, and users' states $(\mathtt{ST}_u, st_u)$; and simply write $\mathtt{PUB}$, $\mathtt{M}$, and $\mathtt{ST}_u$. We also impose the requirement that symbolic outputs of the algorithms are derivable from their symbolic inputs and that only a finite number of symbolic derivations is done.

Correctness and security. We define correctness and security of CGKA schemes in the symbolic model according to the game $\mathrm{SEC}^{\mathsf{CGKA}}$ in Figure 3. The game is defined with respect to $(n_{\max}, t_{\max}, G_0, u_0)$ and

[11] As it makes for a cleaner presentation, in our syntax a single user generates the control message replacing the users in A_t. However, our lower bound also applies in straightforward manner to CGKA in which users in any given round are able to concurrently issue the replacement of sets of users, and the resulting control messages are in turn processed by all group members. In this setting our lower bound holds with respect to the overall number of users removed per round.

$(u_t, A_t, D_t)_{t=1}^{t_{\max}}$. Similarly to the corresponding game for ME, $[n_{\max}]$ is the universe of users, $t_{\max}$ the number of rounds the game is run for, and G_0 the initial group. Following the initial set-up, user u_0 generates G_0 using algorithm $\mathsf{CGKA.Init}$. Afterwards the game proceeds in rounds t determined by (u_t, A_t, D_t), where user u_t uses $\mathsf{CGKA.Repl}(\mathtt{ST}_{u_t}^{t-1}, A_t, D_t)$ to replace the set $A_t \subseteq G_{t-1}$ by the set $D_t \subseteq [n_{\max}] \setminus G_{t-1}$. Here we require that $u_t \in G_{t-1} \setminus D_t$, i.e., users executing the replacement must be group members and users are not able to remove themselves from the group. In every round the game verifies that (a) all group members have access to the current group key, which essentially means

$$\exists \mathtt{k}^t : \mathtt{k}^t = \mathsf{CGKA.Key}(\mathtt{ST}_u^t) \text{ for all } u \in G_t;$$

and that (b) non-members of the group, even when colluding and having stored all their previous states $\mathtt{ST}_u^{t'}$ as well as all random coins $\mathtt{R}_u^{t'}$ sampled while issuing replacement operations, are not able to derive the current group key, i.e.,

$$\mathtt{k}^t \notin \mathsf{Der}\Big(\mathtt{PUB}, (\mathtt{M}_{t'})_{t'=0}^t, ((\mathtt{ST}_u^{t'})_{t'=-1}^t, (\mathtt{R}_u^{t'})_{t'=0}^t)_{u \in [n_{\max}] \setminus G_t}\Big).$$

Similarly to the case of multicast encryption, we assume that the symbolic part of the initial state, namely, $\mathtt{ST}_u^{-1}$ only contains symbols of type random coins. We also require that for all $\mathtt{r} \in \mathtt{ST}_u^{-1} \bigcup_{0 \le t' \le t_{\max}} \mathtt{R}_u^{t'}$:

$$\mathtt{r} \notin \mathsf{Der}\Big(((\mathtt{ST}_u^{-1} \cup \cup_{0 \le t' \le t_{\max}} \mathtt{R}_u^{t'}) \setminus \{\mathtt{r}\}) \cup \cup_{v \in [n_{\max}] \setminus \{u\}} (\mathtt{ST}_v^{-1} \cup \cup_{0 \le t' \le t_{\max}} \mathtt{R}_v^{t'})\Big). \quad (9)$$

A similar assumption is made in [BDR20, ACPP23].

Useful secrets and associated set system. Analogously to Section 4.2 we first define useful secrets generated during the execution of the security game. Thus, consider an execution of $\mathrm{SEC}^{\mathsf{CGKA}}$ with respect to CGKA scheme CGKA, $(n_{\max}, t_{\max}, G_0, u_0)$, and $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$. For $t \in [t_{\max}]_0$ let $\mathtt{r}$ be a symbolic random coin (recall that $\mathtt{r}$ can also serve as secret key of a PKE scheme) generated up to and including round t. We say that $\mathtt{r}$ is an *useful secret in round* t, if

$$\mathtt{r} \notin \mathsf{Der}\Big(\mathtt{PUB}, (\mathtt{M}_{t'})_{t'=0}^t, ((\mathtt{ST}_u^{t'})_{t'=-1}^t, (\mathtt{R}_u^{t'})_{t'=0}^t)_{u \in [n_{\max}] \setminus G_t}\Big),$$

i.e., if it cannot be derived even if all non-group members at time t are colluding. To useful secrets we associate a set of users as follows.

Definition A.1. *Consider an execution of security game* $\mathrm{SEC}^{\mathsf{CGKA}}$ *with respect to CGKA scheme* CGKA*, setup* $(n_{\max}, t_{\max}, G_0, u_0)$*, and sequence of operations* $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$*. Let* $t \in [t_{\max}]_0$ *and* $\mathtt{r}$ *be a random coin. We associate to* $(t, \mathtt{r})$ *the set* $S(t, \mathtt{r})$ *of users that at any point in time up to round* t*, had access to* $\mathtt{r}$*, i.e.,*

$$S(t, \mathtt{r}) := \Big\{u \in [n_{\max}] \mid \mathtt{r} \in \mathsf{Der}(\mathtt{PUB}, \mathtt{ST}_u^{-1}, (\mathtt{M}_{t'})_{0 \le t' \le t:\, u \in G_{t'}}, (\mathtt{R}_u^{t'})_{t'=0}^t)\Big\}.$$

Further, we define the associated set system $\mathcal{S}_t$ *in round* t *as*

$$\mathcal{S}_t := \{S \subseteq [n_{\max}] \mid \textit{there exists useful coin } \mathtt{r} : S = S(t, \mathtt{r})\}.$$

In the following we will show that $(\mathcal{S}_t)_t$ (or rather a subset that we will define later) satisfies the properties required in the combinatorial model of Section 3. Before, though, we give a brief comparison to the associated set system used in [ACPP23].

Remark A.2. *[ACPP23] also consider a symbolic model for CGKA and associate a set system to useful secrets, with the aim of proving lower bounds on the communication complexity of the group recovering from corruption over a certain number of rounds. In that work, a set* $S(t, \mathtt{r})$ *associated to some useful secret* $\mathtt{r}$ *intuitively correspond to the users who have access to* $\mathtt{r}$ *in round* t*. In this work, in turn, we work with*

$S(t, \mathtt{r})$ corresponding to the set of users who in any round up to t had access to $\mathtt{r}$, be it from their internal state or because it was one of the random coins they sampled while generating a replacement operation.

Our definitional choice has two effects. On the one hand, we are no longer able to connect the evolution of the set system from $\mathcal{S}_{t-1}$ to $\mathcal{S}_t$ to the number of ciphertexts that were sent in round t, but have to work with the weaker concept of ciphertexts sent from the beginning of the experiment until round t (which, however, is still strong enough to obtain meaningful bounds). On the other hand, while the approach of [ACPP23] was only able to guarantee that a set S added to $\mathcal{S}_t$ in round t must have been covered by a union of sets in $\mathcal{S}_{t-1}$ and that the cost of this operation in terms of ciphertexts essentially matched the size of the union, in this work we can guarantee the existence of sets in $\mathcal{S}_{t-1}$ whose union exactly matches S. This allows us to connect the number of ciphertexts sent in order to add a set to the set system, to a cover with respect to the set system of the previous round. The latter is necessary to be able to apply the Bollobás Set Pairs Inequality.

The connection between the derivation rules of the symbolic model and the sets introduced in Definition A.1 is similar to the one we proved for the case of multicast encryption in Lemmas 4.2 and 4.3. In the case of CGKA we state the results and defer the proofs to Appendix B.

Lemma A.3. *Let $\mathtt{r}$ be of type random coin and useful at time $t \in [t_{\max}]_0$, and u a user such that $u \in S(t, \mathtt{r})$. Then*

1. *there exist $\mathtt{r}'$ and $i \in \{1, 2\}$ such that $\mathsf{PRG}(\mathtt{r}')_i = \mathtt{r}$, $\mathtt{r}'$ is useful at time t and $u \in S(t, \mathtt{r}')$, or*
2. *there exist $\mathtt{r}'$ and associated data ad such that $\mathsf{PRF}(\mathtt{r}', ad) = \mathtt{r}$, $\mathtt{r}'$ is useful at time t and $u \in S(t, \mathtt{r}')$, or*
3. *there exist $\mathtt{r}_1$ and $\mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$, at least one of $\mathtt{r}_1$ and $\mathtt{r}_2$ is useful at time t, and $u \in S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2)$, or*
4. *$\mathtt{r} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$, or*
5. *there exists $\mathtt{c} = e_0(\cdot) \circ \ldots \circ e_g(\cdot) \circ \ldots \circ e_h(\mathtt{r})$ where $e_i = \mathsf{PKE.Enc}(\mathtt{r}_i, \cdot)$ or $e_i = \mathsf{S}_{i_j}(\cdot)$ such that*
    - (a) *$\mathtt{c} \in \mathtt{PUB} \cup \bigcup_{\tilde{t} \le t\colon u \in G_{\tilde{t}}} \mathtt{M}_{\tilde{t}}$,*
    - (b) *for all i such that $e_i = \mathsf{PKE.Enc}(\mathtt{pk}_i, \cdot)$ there exist random coins $\mathtt{r}_i$ such that $\mathtt{pk}_i = \mathsf{PKE.Gen}(\mathtt{r}_i)$,*
    - (c) *if $e_i = \mathsf{PKE.Enc}(\mathtt{pk}_i, \cdot)$ and $i \ge g + 1$, then $\mathtt{r}_i$ is not useful at time t,*
    - (d) *there exists $i \in \{0, \ldots, h\}$ such that $e_i = \mathsf{PKE.Enc}(\mathtt{pk}_i, \cdot)$,*
    - (e) *$e_g = \mathsf{PKE.Enc}(\mathtt{pk}_g, \cdot)$ and $\mathtt{r}_g$ is useful at time t, and*
    - (f) *for all i such that $e_i = \mathsf{PKE.Enc}(\mathtt{pk}_i, \cdot)$ it holds that $u \in S(t, \mathtt{r}_i)$.*

Lemma A.4. *Let $\mathtt{r}$ be of type random coin and useful at time $t \in [t_{\max}]_0$, and u a user such that $u \in S(t, \mathtt{r})$. Then there exists a sequence $\{\mathtt{r}_{1,u,t}, \ldots, \mathtt{r}_{\ell_u,u,t}\}$ such that*

6. *for all i the secret $\mathtt{r}_{i,u,t}$ is useful at time t and $u \in S(t, \mathtt{r}_{i,u,t})$,*
7. *$\mathtt{r}_{\ell_u,u,t} = \mathtt{r}$,*
8. *$\mathtt{r}_{1,u,t} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$, and*
9. *for all $i \in \{1, \ldots, \ell_u - 1\}$ one of the following is true*
    - (a) *$\mathsf{PRG}(\mathtt{r}_{i,u,t}) = (\mathtt{r}_1, \mathtt{r}_2)$ for some $\mathtt{r}_1$, $\mathtt{r}_2$ such that either $\mathtt{r}_{i+1,u,t} = \mathtt{r}_1$ or $\mathtt{r}_{i+1,u,t} = \mathtt{r}_2$, or*
    - (b) *there exists ad such that $\mathsf{PRF}(\mathtt{r}_{i,u,t}, ad) = \mathtt{r}_{i+1,u,t}$, or*
    - (c) *there exists $\mathtt{r}'_{i,u,t}$ such that $u \in S(t, \mathtt{r}'_{i,u,t})$ and $\mathsf{dPRF}(\mathtt{r}_{i,u,t}, \mathtt{r}'_{i,u,t}) = \mathtt{r}_{i+1,u,t}$, or*
    - (d) *there exists a ciphertext $\mathtt{c}_{i,u,t} \in \mathtt{PUB} \cup \bigcup_{\tilde{t} \le t\colon u \in G_{\tilde{t}}} \mathtt{M}_{\tilde{t}}$ such that*

$$\mathtt{c}_{i,u,t} = e_0(\cdot) \circ \ldots \circ e_g(\cdot) \circ \ldots \circ e_h(\mathtt{r}_{i+1,u,t})$$

where all properties of Case 5 are satisfied and $\mathtt{r}_{i,u,t} = \mathtt{r}_g$ satisfies that $e_g = \mathsf{PKE.Enc}(\mathtt{pk}_g, \cdot)$ and $\mathtt{pk}_g = \mathsf{PKE.Gen}(\mathtt{r}_g)$.

Observe that ℓ_u depends on u, t and $\mathtt{r}$, so in some cases we make this explicit and write $\ell_{u,t,\mathtt{r}}$ or just $\ell_{u,t}$ if $\mathtt{r}$ is clear from context.

Following the same approach as in Section 4.1, we define a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ for $t \in [t_{\max}-1]_0$ using the sequences from Lemma A.4 for the group key $\mathtt{k}^t$. Namely, the set of nodes $\mathcal{V}_t$ corresponds to the elements of the sequences $\{\mathtt{k}^t_{1,u,t}, \ldots, \mathtt{k}^t_{\ell_u,u,t}\}$ associated to the group key $\mathtt{k}^t$ and each user $u \in S(t, \mathtt{k}^t)$, and the set of edges $\mathcal{E}_t$ consists of all pairs of the form $(\mathtt{k}^t_{i,u,t}, \mathtt{k}^t_{i+1,u,t})$.

Property 9c can lead to two edges in the graph $\mathcal{G}_t$ if both inputs $\mathtt{r}_{i,u,t}$ and $\mathtt{r}'_{i,u,t}$ are useful at time t. If this happens, then we make the same choice for all users' sequences in order to make sure that one dPRF pre-image does not result in two edges in $\mathcal{E}_t$. This is possible since $u \in S(t, \mathtt{r}_{i,u,t}) \cap S(t, \mathtt{r}'_{i,u,t})$.

Before studying the properties of this graph, we introduce a modification of any CGKA protocol such that the graph associated to the modified protocol has properties analogous to the ones of Lemma 4.4. This approach was already used in [AAB+21]. If CGKA is a CGKA scheme we define a new CGKA scheme CGKA' by substituting any random coin in $\mathtt{r} \in \bigcup_{u\in[n_{\max}]}(\mathtt{ST}_u^{-1} \cup \bigcup_{0\le t'\le t_{\max}} \mathtt{R}_u^{t'})$ by $\mathtt{r}' = \mathsf{PRG}_1(\tilde{\mathtt{r}})$ for some new random coin $\tilde{\mathtt{r}}$. The modified protocol CGKA' preserves the communication cost of CGKA. If $\mathtt{r}$ is useful at time t in an execution of CGKA, then $\mathtt{r}'$ is useful at time t with respect to the same execution of CGKA'. Therefore CGKA' preserves correctness and security. Moreover, the additional random coins $\tilde{\mathtt{r}}$ used in CGKA' satisfy the property that there exist unique $u \in [n_{\max}]$ and $-1 \le t \le t_{\max}$ such that for all $\tilde{t} \ge t$ it holds $S(\tilde{t}, \tilde{\mathtt{r}}) = \{u\}$. This shows that the value of the function $\mathrm{Cost}(t)$ defined in Section 3.1 is also preserved by CGKA'.

We refer to the edges $(\mathtt{k}^t_{i,u,t}, \mathtt{k}^t_{i+1,u,t})$ that satisfy Properties 9a, 9b or 9c as trivial edges, whereas those that satisfy Property 9d are called communication edges. Now we discuss the properties of the graph $\mathcal{G}_t$ captured in the result below. For a proof we refer the reader to Appendix B.

Lemma A.5. *Let CGKA be a correct and secure CGKA scheme. Consider an execution of game $\mathrm{SEC}^{\mathsf{CGKA}}$ on input $(n_{\max}, t_{\max}, G_0, u_0)$ and $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$ such that $D_t \subseteq G_{t-1}$, $u_t \in G_{t-1} \setminus D_t$, and $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_t)$ for all $t \in [t_{\max}]$. Let $t \in \{0, \ldots, t_{\max}-1\}$ and $\mathtt{k}^t$ denote the group key at time t output by CGKA. Then the following properties of the graph $\mathcal{G}'_t$ associated to the same execution of the modified protocol CGKA' are true:*

10. *For every $u \in S(t, \mathtt{k}^t)$, the node $\mathtt{k}^t_{1,u,t}$ has no incoming edges and $\mathtt{k}^t_{1,u,t} \neq \mathtt{k}^t_{1,v,t}$ for all $u \neq v$. Actually it holds that $S(t, \mathtt{k}^t_{1,u,t}) = \{u\}$.*
11. *For every node $\mathtt{k}^t_{i,u,t}$ there exists at most one node $\mathtt{r}$ such that $\mathsf{PRG}_j(\mathtt{r}) = \mathtt{k}^t_{i,u,t}$ for some $j \in \{1,2\}$, or that $\mathsf{PRF}(\mathtt{r}, ad) = \mathtt{k}^t_{i,u,t}$ for some ad, or that $\mathsf{dPRF}(\mathtt{r}, \mathtt{r}') = \mathtt{k}^t_{i,u,t}$ for some $\mathtt{r}'$ (where $\mathtt{r}'$ may not be in $\mathcal{V}_t$).*
12. *There exists at most one user u in $S(t, \mathtt{k}^t)$ such that for every $1 \le i \le \ell_u - 1$ the edge $(\mathtt{r}_{i,u,t}, \mathtt{r}_{i+1,u,t})$ is trivial.*
13. *If $D_{t+1} \neq \emptyset$, then for every $u \in S(t, \mathtt{k}^t) \setminus D_{t+1}$, there exists $j_{u,t}$ such that $1 \le j_{u,t} < \ell_{u,t}$ and for the corresponding edge $(\mathtt{k}^t_{j_{u,t},u,t}, \mathtt{k}^t_{j_{u,t}+1,u,t}) \in \mathcal{E}_t$ there exists a user $v \in D_{t+1}$ such that $v \in S(t, \mathtt{k}^t_{j_{u,t}+1,u,t})$ and for all $w \in D_{t+1}$ we have $w \notin S(t, \mathtt{k}^t_{j_{u,t},u,t})$. Moreover, $j_{u,t}$ will denote the least integer in $\{1, \ldots, \ell_{u,t}-1\}$ with this property.*

A.2 Lower Bound on Batched Replacements

Lemma A.6. *Consider an execution of game $\mathrm{SEC}^{\mathsf{CGKA}}$ with respect to a correct and secure continuous-group-key agreement scheme CGKA on input $(n_{\max}, t_{\max}, G_0, u_0)$, $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$ such that $D_t \subseteq G_{t-1}$, $u_t \in G_{t-1} \setminus D_t$, and $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'})$ for all $t \in [t_{\max}]$. Let $(\mathcal{S}_t)_{t=0}^{t_{\max}}$ be the associated set system as defined in Definition A.1 and*

$$\tilde{\mathcal{S}}_t = \left\{ S \in \mathcal{S}_t \,\middle|\, \begin{array}{l} \exists \mathtt{r} \text{ such that } S = S(t,\mathtt{r}),\ \mathtt{r} \text{ is useful at time } t \text{ and} \\ \nexists \mathtt{r}_1, \mathtt{r}_2 \text{ such that } \mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r} \text{ and } S(t,\mathtt{r}_1) \cap S(t,\mathtt{r}_2) = S(t,\mathtt{r}) \end{array} \right\}.$$

Then it holds that

(i) $G_t \in \tilde{\mathcal{S}}_t$ *for all* $t \in [t_{\max}]_0$*,*

(ii) $S \subseteq G_t$ *for all* $S \in \mathcal{S}_t$ *and, in particular,* $S \subseteq G_t$ *for all* $S \in \tilde{\mathcal{S}}_t$*,*

(iii) $|\mathtt{PUB}| + \sum_{t=0}^{t_{\max}} |\mathtt{M}_t| \geq 1/3 \cdot \sum_{t=0}^{t_{\max}} \mathrm{Cost}(t)$*, where* $\mathrm{Cost}(t)$ *is the cost function defined in Section 3.1 with respect to* $\tilde{\mathcal{S}}_t$*, namely:*

$$\mathrm{Cost}(t) = (\mathrm{SizeMinCov}(G_t, \tilde{\mathcal{S}}_{t-1} \cup \{\{u\} : u \in G_t\}) - 1) + |\{S \in \tilde{\mathcal{S}}_{t-1} : S \cap D_t \neq \emptyset \textit{ and } |S| > 1\}|.$$

For the proof we refer the reader to Appendix B, but we observe that the lower bound in Property (iii) only relies on counting ciphertexts not public keys. Lemma A.6 shows that it is possible to apply Theorem 3.4 to set system $(\tilde{\mathcal{S}}_t)_{t=0}^{t_{\max}}$ and therefore we obtain the following bound on the communication complexity of batched replacements in CGKA schemes.

Corollary A.7. *Let* $n \leq n_{\max}$ *and* $t_{\max}$ *be in* $\mathbb{N}$ *and* $(d_t)_{t=1}^{t_{\max}}$ *such that* $d_t < n$ *for all* t*. Consider an execution of game* $\mathrm{SEC}^{\mathsf{CGKA}}$ *with respect to a correct and secure CGKA scheme* CGKA *on input* $(n_{\max}, t_{\max}, G_0, u_0)$ *and* $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$ *with* $|G_0| = n$*, where for all* t *the set* D_t *of removed users is sampled uniformly at random from the set* $\{D \subseteq G_{t-1} \mid |D| = d_t\}$*, and* u_t *and* A_t *can be arbitrary according to the restrictions* $u_t \in G_{t-1} \setminus D_t$ *and* $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_t)$ *respectively. Then it holds that*

$$\mathbb{E}\left[|\mathtt{PUB}| + \sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{1}{3} \sum_{t=1}^{t_{\max}} d_t \ln\left(\frac{n}{d_t}\right),$$

where the expectation is taken over the choice of $(D_t)_t$*. In particular, if* $d_t = d$ *for all* t*, then*

$$\mathbb{E}\left[|\mathtt{PUB}| + \sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{\ln(2)}{3} t_{\max} \cdot d \cdot \log\left(\frac{n}{d}\right).$$

A.3 Extending the bound to Key-updatable PKE.

In [BDR20] the authors in their symbolic model additionally include key-updatable public-key encryption (kuPKE)[12] and broadcast encryption (BE) in the considered building blocks. In the following we will argue that the lower bound for CGKA proven in Section A.2 also applies to schemes additionally taking kuPKE into account.

kuPKE. A kuPKE scheme $\mathsf{kuPKE} = (\mathsf{kuPKE.Gen}, \mathsf{kuPKE.Enc}, \mathsf{kuPKE.Dec}, \mathsf{kuPKE.Upd})$ specifies the following. As is the case for PKE, $\mathsf{kuPKE.Gen}$ receives as input a secret key $\mathtt{sk}$ (of type $\mathtt{r}$) and returns the corresponding public key $\mathtt{pk}$. Encryption algorithm $\mathsf{kuPKE.Enc}(\mathtt{pk}, \mathtt{m})$, on input a public key $\mathtt{pk}$ and message $\mathtt{m}$, returns a ciphertext $\mathtt{c}$ (of message type). Decryption algorithm $\mathsf{kuPKE.Dec}(\mathtt{sk}, \mathtt{c})$, on input a secret key $\mathtt{sk}$ and ciphertext $\mathtt{c}$, returns a message $\mathtt{m}$. The update algorithm $\mathsf{kuPKE.Upd}$ receives as input associated data ad as well as either a public key $\mathtt{pk}$ or a secret key $\mathtt{sk}$ and returns an updated public key $\mathtt{pk}'$ or secret key $\mathtt{sk}'$, respectively. We require that the updated keys are distinct symbolic variables from the ones input to $\mathsf{kuPKE.Upd}$. The rules for the entailment relation with respect to kuPKE are

$$\begin{aligned} &\mathtt{M} \vdash \mathtt{pk} \Rightarrow \forall ad : \mathtt{M} \vdash \mathsf{kuPKE.Upd}(\mathtt{pk}, ad) \\ &\mathtt{M} \vdash \mathtt{sk} \Rightarrow \forall ad : \mathtt{M} \vdash \mathsf{kuPKE.Upd}(\mathtt{sk}, ad) \text{ and } \mathtt{M} \vdash \mathsf{kuPKE.Gen}(\mathtt{sk}) \end{aligned} \tag{10}$$

[12] This primitive implies hierarchical identity based encryption (HIBE) in the random oracle model if we use as associated data a hash of the identities. By allowing the use of kuPKE we strengthen our lower bound.

i.e., from a key one can obtain its update under every associated data, and regarding security and correctness

$$\begin{aligned}\exists & \mathtt{sk}_0, ad_o, \ldots, ad_{s-1} \text{ and } \mathtt{M} \vdash (\mathtt{c}, \mathtt{sk}_r) : \\ & \mathtt{c} = \mathsf{kuPKE.Enc}(\mathtt{pk}_s, \mathtt{m}), \\ & \mathtt{pk}_0 = \mathsf{kuPKE.Gen}(\mathtt{sk}_0), s \geq r \geq 0, \\ & \forall i \in [s-1]_0 : \mathtt{pk}_{i+1} = \mathsf{kuPKE.Upd}(\mathtt{pk}_i, ad_i), \\ & \forall i \in [r-1]_0 : \mathtt{sk}_{i+1} = \mathsf{kuPKE.Upd}(\mathtt{sk}_i, ad_i) \\ \Rightarrow & \mathtt{M} \vdash \mathtt{m},\end{aligned} \tag{11}$$

saying that a message encrypted under a public key that was updated several times can be recovered from its ciphertext if one is able to derive at least one of the preceding corresponding secret keys in the update chain.

Extending the lower bound to kuPKE. In the following we will argue that the lower bound of Corollary A.7 also holds if one additionally allows for kuPKE one of the considered building blocks. The intuition behind this is that the update mechanism of secret keys in kuPKE can be emulated by applying a PRF to the secret key. Note, however, that using this update mechanism update public keys can only be derived if one also has access to the corresponding secret key. Thus our argument proceeds in two steps. In the first for a particular execution of the security game we essentially move every update operation $\mathsf{kuPKE.Upd}$ applied to a public key $\mathtt{pk}$ to the point in time where the public key was generated from its corresponding secret key. In the second step we are now able to replace kuPKE by PKE and PRF applications as every algorithm using $\mathsf{kuPKE.Upd}$ on a public key has access to the corresponding secret key. To prove that our bound carries over we have to argue that these modifications preserve correctness, security, and leave the cost, i.e., the number of ciphertexts, unchanged. Formally, we obtain the following.

Corollary A.8. *Let $n \leq n_{\max}$ and $t_{\max}$ be in $\mathbb{N}$ and $(d_t)_{t=1}^{t_{\max}}$ such that $d_t < n$ for all t. Let CGKA be a correct and secure CGKA scheme using PKE, kuPKE, PRG, PRF, dPRF, and secret sharing as building blocks. Consider an execution of game $\mathrm{SEC}^{\mathsf{CGKA}}$ with respect to CGKA on input $(n_{\max}, t_{\max}, G_0, u_0)$ and $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$ with $|G_0| = n$, where for all t the set D_t of removed users is sampled uniformly at random from the set $\{D \subseteq G_{t-1} \mid |D| = d_t\}$, and u_t and A_t can be arbitrary according to the restrictions $u_t \in G_{t-1} \backslash D_t$ and $A_t \subseteq [n_{\max}] \setminus (G_{t-1} \cup \bigcup_{t'=1}^{t-1} D_{t'})$ respectively. Then it holds that*

$$\mathbb{E}\left[|\mathtt{PUB}| + \sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{1}{3} \sum_{t=1}^{t_{\max}} d_t \ln\left(\frac{n}{d_t}\right),$$

where the expectation is taken over the choice of $(D_t)_t$. In particular, if $d_t = d$ for all t, then

$$\mathbb{E}\left[|\mathtt{PUB}| + \sum_{t=0}^{t_{\max}} |\mathtt{M}_t|\right] \geq \frac{\ln(2)}{3} t_{\max} \cdot d \cdot \log\left(\frac{n}{d}\right).$$

Proof. Consider the modification to the derivation of symbolic variables in security game $\mathrm{SEC}^{\mathsf{CGKA}}$ of Figure 3 with respect to CGKA, $(n_{\max}, t_{\max}, G_0, u_0)$, and $(u_t, A_t, D_t)_{t=1}^{t_{\max}}$ defined as follows. Whenever one of the algorithms $\mathsf{CGKA.Setup}$, $\mathsf{CGKA.Init}$, $\mathsf{CGKA.Repl}$, $\mathsf{CGKA.Proc}$, or $\mathsf{CGKA.Key}$ makes a call to $\mathtt{pk}_0 \leftarrow \mathsf{kuPKE.Gen}(\mathtt{sk}_0)$ for some secret key $\mathtt{sk}_0$ do the following:

- Parse the remainder of the security experiment for all chains of associated data used to update $\mathtt{pk}_0$, i.e., all $ad_{0,j}, \ldots, ad_{\ell_j,j}$ such that there are symbolic operations $\mathtt{pk}_{i+1,j} \leftarrow \mathsf{kuPKE.Upd}(\mathtt{pk}_{i,j})$ for $i \in [\ell_j]_0$ where we define $\mathtt{pk}_{0,j} = \mathtt{pk}_0$ for all such j.
- In the code move the execution of all update operations $\mathsf{kuPKE.Upd}(\mathtt{pk}_{i,j}, ad_{i,j})$ to the lines immediately after the call to $\mathsf{kuPKE.Gen}(\mathtt{sk})$.

- Add all public keys to the outputs of the CGKA operation that the call to kuPKE.Gen occurred in, i.e., to $\mathtt{PUB}$ (but not $\mathtt{ST}_u$) for CGKA.Setup, to $\mathtt{M}_0$ and $\mathtt{ST}_{u_0}$ for CGKA.Init, to $\mathtt{ST}_u$ for CGKA.Proc, and to $\mathtt{M}_t$ and $\mathtt{ST}_{u_t}$ for CGKA.Repl. Moreover all public keys in $(\mathtt{M}_{t'})_{t'=0}^{t-1}$ are included again in $\mathtt{M}_t$ to guarantee that users added to the group later also have access to them.

Note that this change does not affect the number of ciphertexts sent throughout the security game and that it preserves correctness. The lower bound in Property (iii) only relies on counting ciphertexts not public keys, so if it holds for the modified protocol and the number of ciphertexts is preserved. Indeed, all algorithms are still able to derive all the required symbolic variables, in particular the group keys, since they only learn about public keys at an earlier point in time, either as it was included in the public parameters, a public message, or because the particular user added it to their state at an earlier time (we may assume that public keys never get deleted from users' states since this does not affect the set of secrets one is able to derive as we will argue below). Further, as the modification did not change any secret symbolic variables, all users must still derive the same group key as required in lines 11 and 31 of the security game. We now argue that including additional kuPKE public keys in the set

$$\overline{\mathtt{M}}_t := (\mathtt{PUB}, (\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t}, (\mathtt{R}_u^{t'})_{t'=0}^{t})_{u\in[n_{\max}]\setminus G_t})$$

leaves the set of symbolic secrets that can be derived from $\overline{\mathtt{M}}_t$ unchanged. This, in particular, implies that the modification preserves security, namely we have $\mathtt{k}^t \notin \mathsf{Der}(\overline{\mathtt{M}}_t)$ for all t as required in Lines 13 and 33. To see this, note that a public key $\mathtt{pk}$ can only be updated to other public keys, be used as a message and encrypted, or be used to encrypt a messag. In the latter case the encrypted message has to be derivable from $\overline{\mathtt{M}}_t$ as well, and in turn does not yield additional secrets if later decrypted (in case the corresponding secret key $\mathtt{sk}$ is derivable from $\overline{\mathtt{M}}_t$). Finally note that by definition of the change users' initial states still only contain variables of type random coins. Concluding our argument, that the modification does preserve correctness and security of the execution of the security game.

We now argue that after this modification we may see the use of kuPKE in the security as a variant that uses a modified syntax which simply sees all updates to public keys as being part of kuPKE.Gen. Accordingly, this modified kuPKE syntax uses algorithms $\mathsf{kuPKE.Enc}_{\mathrm{mod}}$ and $\mathsf{kuPKE.Dec}_{\mathrm{mod}}$ that behave as in normal kuPKE, an update algorithm $\mathsf{kuPKE.Upd}_{\mathrm{mod}}$ that on input a secret key $\mathtt{sk}$ and additional data ad returns updated secret key $\mathtt{sk}'$ (but no longer accepts public keys as input), and a key generation algorithm $\mathsf{kuPKE.Gen}_{\mathrm{mod}}$ that on input $\mathtt{sk}_0$ and chains of associated data $(ad_{i,j})_{i\in[\ell_j]_0,j\in[k]}$ returns corresponding public keys $\mathtt{pk}_0$ and $(\mathtt{pk}_{i,j})_{i\in[\ell_j+1],j\in[k]}$. Plugging his into the rules for the entailment relation given in Equations 10 and 11 they now read as

$$\begin{aligned} &\mathtt{M} \vdash \mathtt{sk} \Rightarrow \forall (ad_{i,j})_{i\in[\ell_j]_0,j\in[k]} : \mathtt{M} \vdash \mathsf{kuPKE.Gen}_{\mathrm{mod}}(\mathtt{sk}, (ad_{i,j})_{i\in[\ell_j]_0,j\in[k]}), \\ &\mathtt{M} \vdash \mathtt{sk} \Rightarrow \forall ad : \mathtt{M} \vdash \mathsf{kuPKE.Upd}_{\mathrm{mod}}(\mathtt{sk}, ad) \end{aligned} \tag{12}$$

and

$$\begin{aligned} &\exists \mathtt{sk}_0, (ad_{i,j})_{i\in[\ell_j]_0,j\in[k]} \text{ and } \mathtt{M} \vdash (\mathtt{c}, \mathtt{sk}_{r,j}) : \\ &\quad (\mathtt{pk}_0, (\mathtt{pk}_{i,j})_{i\in[\ell_j+1],j\in[k]}) = \mathsf{kuPKE.Gen}_{\mathrm{mod}}(\mathtt{sk}_0, (ad_{i,j})_{i\in[\ell_j]_0,j\in[k]}), \ell_j + 1 \geq s \geq r \geq 0, \\ &\quad \mathtt{c} = \mathsf{kuPKE.Enc}_{\mathrm{mod}}(\mathtt{pk}_{s,j}, \mathtt{m}) \text{ for some } j \in [k], \\ &\quad \forall i \in [r-1]_0 : \mathtt{sk}_{i+1,j} = \mathsf{kuPKE.Upd}_{\mathrm{mod}}(\mathtt{sk}_i, ad_{i,j}) \\ &\Rightarrow \mathtt{M} \vdash \mathtt{m}. \end{aligned} \tag{13}$$

In a final step, we now argue that the kuPKE with modified syntax can be realized in the symbolic model from PKE and PRF as follows. Algorithms $\mathsf{kuPKE.Enc}_{\mathrm{mod}}$ and $\mathsf{kuPKE.Dec}_{\mathrm{mod}}$ are replaced by their counterparts PKE.Enc and PKE.Dec, secret keys are updated as $\mathsf{kuPKE.Upd}_{\mathrm{mod}}(\mathtt{sk}, ad) = \mathsf{PRF}(\mathtt{sk}, ad)$, and $\mathsf{kuPKE.Gen}_{\mathrm{mod}}(\mathtt{sk}, (ad_{i,j})_{i\in[\ell_j]_0,j\in[k]})$ first sets $\mathtt{pk}_0 = \mathsf{PKE.Gen}(\mathtt{sk}_0)$, then for all $j \in [k]$ and all $i \in [\ell_j]_0$ iteratively computes $\mathtt{sk}_{i+1,j} = \mathsf{PRF}(\mathtt{sk}_{i,j}, ad_{i,j})$ and $\mathtt{pk}_{i+1,j} = \mathsf{PKE.Gen}(\mathtt{sk}_{i+1,j})$, where we set $\mathtt{pk}_{0,j} = \mathtt{pk}_0$ for all j. Its output is $(\mathtt{pk}_0, (\mathtt{pk}_{i,j})_{i\in[\ell_j],j\in[k]})$. Note, that due to the application of the PRF updated keys differ from their not updated counterparts, as required. Further we will now argue that the construction satisfies the three rules regarding the entailment relation. The rules in Equation 12 follow immediately from the syntax of PKE and PRFs in the symbolic model. Regarding the third rule, assume that $\mathtt{M} \vdash (\mathtt{c}, \mathtt{sk}_{r,j})$ with the properties listed in Equation 13. Then by construction it must be the case that

$$\mathtt{sk}_{r,j} = \mathsf{PRF}(\cdot, ad_{r-1,j}) \circ \cdots \circ \mathsf{PRF}(\cdot, ad_{0,j})(\mathtt{sk}_{0,j}),$$

and

$$\mathtt{pk}_{s,j} = \mathsf{PKE.Gen} \circ \mathsf{PRF}(\cdot, ad_{s-1,j}) \circ \cdots \circ \mathsf{PRF}(\cdot, ad_{0,j})(\mathtt{sk}_{0,j}).$$

Thus given $\mathtt{sk}_{r,j}$ one can compute $\mathtt{sk}_{s,j}$ from $\mathtt{sk}_{r,j}$ by applying $\mathsf{PRF}(\cdot, ad_{i,j})$ for $i \in [s, \ldots, r-1]$ (if $r = s$ no PRF applications are necessary) and it holds that $\mathtt{pk}_{s,j} = \mathsf{PKE.Gen}(\mathtt{sk}_{s,j})$. In turn, one obtains $\mathtt{m}$ from $\mathtt{c}$ by using the derivation rule modeling correctness and security of PKE, which concludes the argument that modified kuPKE can be constructed from PKE and PRFs.

This in particular implies that replacing the modified kuPKE scheme in the security game by PKE and PRF preserves correctness and security. Now, the corollary follows by applying Corollary A.7 and observing that the modification using only PKE and PRFs leaves the number of ciphertexts sent during the game unchanged. □

A.4 Regarding the Use of Broadcast Encryption as Additional Building Block.

The bound by Bienstock, Dodis, and Rösler [BDR20] additionally allows for the use of broadcast encryption (BE), which essentially allows to register from a master secret-key so called user secret-keys in a way that allows the creation of ciphertexts with respect to a master public-key and a set R which in turn can be decrypted by all registered users that are not in R. In this section we recall the definition of BE and then briefly argue that it allows for the construction of ME with constant communication complexity. Thus, while we consider it an interesting open question whether our bound for CGKA can be extended to also include BE as a building block, the proof would require an approach substantially differing from ours.

A broadcast encryption scheme $\mathsf{BE} = (\mathsf{BE.Gen}, \mathsf{BE.Reg}, \mathsf{BE.Enc}, \mathsf{BE.Dec})$ specifies the following. Algorithm $\mathsf{BE.Gen}$ on input a master secret-key $\mathtt{msk}$ of type $\mathtt{r}$ returns the corresponding master public-key $\mathtt{mpk}$ of public-key type. Registration algorithm $\mathsf{BE.Reg}$ can be used to register a secret key for a particular identity that is modeled by a non-symbolic identifier $u \in \mathbb{N}$. On input $(\mathtt{msk}, u)$ it returns the user's secret key $\mathtt{sk}_u$, which we require to not be equal to $\mathtt{msk}$. The encryption algorithm $\mathsf{BE.Enc}(\mathtt{mpk}, R, \mathtt{m})$ on input the master public-key, a set of excluded users $R \subseteq \mathbb{N}$, and a message $\mathtt{m}$ returns a ciphertext $\mathtt{c}$ of message type. Finally, decryption algorithm $\mathsf{BE.Dec}$ on input $(\mathtt{sk}, \mathtt{c})$ returns a message $\mathtt{m}$. The rules for the entailment relation are that from a master secret key one can derive all users' secret keys, i.e.,

$$\mathtt{M} \vdash \mathtt{msk} \Rightarrow \forall u \in \mathbb{N} : \mathtt{M} \vdash \mathsf{BE.Reg}(\mathtt{msk}, u),$$

and that from a ciphertext encrypted under a master public key one can recover the underlying message given access to the secret key of at least one user that was not excluded, i.e.,

$$\begin{aligned} &\exists u \notin R \wedge \mathtt{M} \vdash (\mathtt{c}, \mathtt{sk}_u) : \\ &\qquad \mathtt{c} = \mathsf{BE.Enc}(\mathtt{mpk}, R, \mathtt{m}), \\ &\qquad \mathtt{mpk} = \mathsf{BE.Gen}(\mathtt{msk}), \\ &\qquad \mathtt{sk}_u = \mathsf{BE.Reg}(\mathtt{msk}, u) \\ &\Rightarrow \mathtt{M} \vdash \mathtt{m}. \end{aligned}$$

We now briefly describe how one can use BE to construct a multicast encryption scheme ME that allows for batched user replacement with constant per-round communication complexity. $\mathsf{ME.Setup}(n_{\max}; \mathtt{r})$ generates a master secret key $\mathtt{msk} = \mathsf{BE.Gen}(\mathtt{r})$ and then sets the initial state of each user $u \in [n_{\max}]$ to $sk_u = \mathsf{BE.Reg}(\mathtt{msk}, u)$. To initialize a group G_0, algorithm $\mathsf{ME.Init}$ samples a random secret $\mathtt{k}_0$ and returns the control message $\mathsf{BE.Enc}(\mathtt{mpk}, [n_{\max}] \setminus G_0, \mathtt{k}_0)$, and similarly replacing a set D_t of users by users A_t is done by sampling key $\mathtt{k}_t$ and sending the control message $\mathsf{BE.Enc}(\mathtt{mpk}, [n_{\max}] \setminus G_t, \mathtt{k}_t)$. Note that all group members at time t have access to the group key, since they were not excluded from the recipient set, and that further, that even if all non-group members collide they are not able to derive $\mathtt{k}_t$, as they are not able to derive at least one of the $\mathtt{sk}_u$ for $u \in G_t$ (here we use that removed users are never added back to the group). Finally, note that each round a single ciphertext is sent.

B Proofs Appendix A

B.1 Proofs of Appendix A.1 (Lemmas A.3, A.4 and A.5)

The approach we follow is essentially the same as in the case of multicast encryption (Lemmas 4.2, 4.3 and 4.4) and account for the differences between the two primitives.

Proof of Lemma A.3. If $\mathtt{r}$ admits a PRG pre-image $\mathtt{r}'$, $\mathtt{r}'$ must be useful at time t since $\mathtt{r}$ is useful at time t. Therefore we have two possible cases depending on whether $u \in S(t, \mathtt{r}')$. If $u \in S(t, \mathtt{r}')$, we are in Case 1. If $u \notin S(t, \mathtt{r}')$, one of the following holds:

- $\mathtt{r} \in \mathtt{PUB} \cup \mathtt{ST}_u^{-1} \cup \bigcup_{0 \le t' \le t:\ u \in G_{t'}} \mathtt{M}_{t'} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$ and the fact that $\mathtt{r}$ is useful implies that $\mathtt{r} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$.
- There exists a ciphertext $\mathtt{c} \in \mathtt{PUB} \cup \mathtt{ST}_u^{-1} \cup \bigcup_{0 \le t' \le t:\ u \in G_{t'}} \mathtt{M}_{t'} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$ of the form described in Case 5 such that condition *(f)* holds. By assumption $\mathtt{ST}_u^{-1}$ only contains symbols of type random coins, so $\mathtt{c} \in \mathtt{PUB} \cup \bigcup_{0 \le t' \le t:\ u \in G_{t'}} \mathtt{M}_{t'}$. The usefulness of $\mathtt{r}$ implies that there must exist $i \in \{0, \ldots, h\}$ such that e_i is an encryption under a public key with an associated secret key that is useful. This shows *(d)* and *(e)*. Condition *(c)* is just a matter of choice.

If $\mathtt{r}$ does not admit a PRG pre-image, we proceed to consider whether there exist $\mathtt{r}'$ and associated data ad such that $\mathsf{PRF}(\mathtt{r}', ad) = \mathtt{r}$. If this is the case, $\mathtt{r}'$ must be useful at time t since $\mathtt{r}$ is. Depending on whether $u \in S(t, \mathtt{r}')$ we have two possible cases. If $u \in S(t, \mathtt{r}')$, we are in Case 2. If $u \notin S(t, \mathtt{r}')$, then we are in Cases 4 or 5 by the same argument as in the case of PRG pre-images. If $\mathtt{r}$ does not admit neither a PRG pre-image nor a PRF pre-image, we proceed to consider whether there exist $\mathtt{r}_1$ and $\mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$. In this case at least one of $\mathtt{r}_1$ and $\mathtt{r}_2$ must be useful at time t since $\mathtt{r}$ is. If $u \in S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2)$, then we are in Case 3. Else we are in Cases 4 or 5 by arguing as in the case of PRG pre-images. □

Proof of Lemma A.4. Let $\mathtt{r} \leftarrow \mathtt{r}$ and $\mathsf{Seq} \leftarrow \emptyset$. Repeat $(\mathtt{r}, \mathsf{Seq}) \leftarrow f(\mathtt{r}, \mathsf{Seq})$ until $\mathtt{r} = \text{STOP}$ where:

$$f(\mathtt{r}, \mathsf{Seq}) = \begin{cases} \text{if we are in Case 1, do } (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}', \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 2, do } (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}', \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 3 and } \mathtt{r}_i \text{ is useful, do } (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}_i, \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 4, do} (\mathtt{r}, \mathsf{Seq}) \leftarrow (\text{STOP}, \{\mathtt{r}\} \cup \mathsf{Seq}), \\ \text{if we are in Case 5, do} (\mathtt{r}, \mathsf{Seq}) \leftarrow (\mathtt{r}_g, \{\mathtt{r}\} \cup \mathsf{Seq}). \end{cases}$$

This process terminates since it is required that only a finite number of symbolic derivations is needed in order to derive the symbolic outputs of the CGKA algorithms from their symbolic inputs. By construction, all properties are clearly satisfied. □

Proof of Lemma A.5. First of all, we observe that in the modified protocol CGKA' a random coin $\mathtt{r} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$ sampled in some round can only be derived by the user u and it remains a useful secret until $u \in D_t$. Moreover, it cannot have any incoming edges in the graph $\mathcal{G}_t$ by construction of the modified protocol CGKA'. By Property 8, $\mathtt{k}_{1,u,t}^{t} \in \mathtt{ST}_u^{-1} \cup \bigcup_{t'=0}^{t} \mathtt{R}_u^{t'}$, so this proves Property 10.

Property 11 follows directly from the properties of the symbolic model.

Property 12 is a direct consequence of the two previous properties.

Property 13 follows from the observation that the node $\mathtt{k}_{\ell_u,u,t} = \mathtt{k}^t$ satisfies the condition that $v \in S(t, \mathtt{k}^t)$ for all users $v \in D_{t+1}$ and the node $\mathtt{k}_{1,u,t}$ satisfies the second condition (by Property 10). Since $D_{t+1} \neq \emptyset$ by assumption, there must exist an edge with the required property. □

B.2 Proofs of Appendix A.2

The proof is essentially the same as the one given in the case of multicast encryption for Lemma 4.5, but adapted to the CGKA setting.

Proof of Lemma A.6. We start proving Property (ii). Let $S = S(t, \mathtt{r}) \in \mathcal{S}_t$ and $u \in S$. By definition of $S(t, \mathtt{r})$,

$$\mathtt{r} \in \mathsf{Der}(\mathtt{PUB}, \mathtt{ST}_u^{-1}, (\mathtt{M}_{t'})_{0 \leq t' \leq t:\ u \in G_{t'}}, (\mathtt{R}_u^{t'})_{t'=0}^{t})$$

and since $\mathtt{r}$ is useful at time t,

$$\mathtt{r} \notin \mathsf{Der}\Big(\mathtt{PUB}, (\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t}, (\mathtt{R}_u^{t'})_{t'=0}^{t})_{u \in [n_{\max}] \setminus G_t}\Big),$$

Thus $u \in G_t$ as claimed in Property (ii).

Now we prove Property (i). By correctness there exists a group key at time t, $\mathtt{k}^t$, such that $\mathtt{k}^t = \mathsf{CGKA.Key}(\mathtt{ST}_u^t)$ for all users $u \in G_t$. Therefore $G_t \subseteq S(t, \mathtt{k}^t)$. By security

$$\mathtt{k}^t \notin \mathsf{Der}\Big(\mathtt{PUB}, (\mathtt{M}_{t'})_{t'=0}^{t}, ((\mathtt{ST}_u^{t'})_{t'=-1}^{t}, (\mathtt{R}_u^{t'})_{t'=0}^{t})_{u \in [n_{\max}] \setminus G_t}\Big),$$

so $S(t, \mathtt{k}^t) \subseteq G_t$ and $\mathtt{k}^t$ is useful at time t. This shows that $S(t, \mathtt{k}^t) = G_t \in \mathcal{S}_t$. It remains to prove that $G_t \in \tilde{\mathcal{S}}_t$ as claimed in Property (i). Let's assume that there exist $\mathtt{r}_1, \mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{k}^t$ and $S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{k}^t) = G_t$. The fact that $\mathtt{k}^t$ is useful at time t implies that there exists $i \in \{1, 2\}$ such that $\mathtt{r}_i$ is useful at time t. By Property (ii) it must hold that $S(t, \mathtt{r}_i) \subseteq G_t$. By assumption $S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{k}^t) = G_t$, so we also have $G_t \subseteq S(t, \mathtt{r}_i)$. Therefore $S(t, \mathtt{r}_i) = G_t$. By repeating this process we can find a secret $\mathtt{r}$ that is useful at time t and that satisfies the properties that $S(t, \mathtt{r}) = G_t$ and $\nexists \mathtt{r}_1, \mathtt{r}_2$ such that $\mathsf{dPRF}(\mathtt{r}_1, \mathtt{r}_2) = \mathtt{r}$ and $S(t, \mathtt{r}_1) \cap S(t, \mathtt{r}_2) = S(t, \mathtt{r})$. This completes the proof of Property (i), i.e., $G_t \in \tilde{\mathcal{S}}_t$.

Finally, in order to prove Property (iii), it suffices to observe that the graph $\mathcal{G}_t'$ from the modified protocol CGKA' satisfies the same properties according to Lemma A.5 than the ones we needed in the multicast encryption setting from Lemma 4.4. This shows the cost inequality for CGKA' and this protocol preserves the cost of the original protocol CGKA as well as the value of the function $\mathrm{Cost}(t)$, which in turn proves inequality(iii) for CGKA. □