RAD HRVATSKE AKADEMIJE ZNANOSTI I UMJETNOSTI
MATEMATIČKE ZNANOSTI

A. Klinger, S. Wüller, G. Traverso and U. Meyer

Hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer

Manuscript accepted for publication

HIERARCHICAL AND DYNAMIC THRESHOLD PAILLIER CRYPTOSYSTEM WITHOUT TRUSTED DEALER

Andreas Klinger, Stefan Wüller, Giulia Traverso and Ulrike Meyer

Abstract. We propose the first hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer and prove its security in the malicious adversary model. The new cryptosystem is fully distributed, i. e., public and private key generation is performed without a trusted dealer. The private key is shared with a hierarchical and dynamic secret sharing scheme over the integers. In such a scheme not only the amount of shareholders, but also their levels in the hierarchy decide whether or not they can reconstruct the secret and new shareholders can be added or removed without reconstruction of the secret.

1. Introduction

The use of homomorphic threshold cryptosystems for distributed computing on encrypted data has been proposed in various areas of application, including double auction (e. g., [5]), privacy preserving data mining (e. g., [28]), and data integration and sharing (e. g., [8]). In these cryptosystems, the access structure underlying the sharing of the private key between the shareholders is typically assumed to be a fixed flat access structure requiring a minimum number t of n equally powerful shareholders to cooperate during decryption. Many of these use cases could profit from supporting more complex access structures, such as hierarchical access structures, in which each shareholder is associated with a certain level in a hierarchy, as well as supporting dynamicity, such that shareholders may join or leave. A homomorphic cryptosystem with both properties is applicable in many scenarios, among others, online

2010 *Mathematics Subject Classification.* 94A60, 94A62.

Key words and phrases. Homomorphic Cryptosystem, Threshold Cryptosystem, Hierarchical Secret Sharing, Dynamic Secret Sharing, Paillier, SMPC, Birkhoff Interpolation.

auctions where bidders join at different points in time, or server aided secure multi-party computation. In particular, hierarchical access structures often better reflect the structure within an organization or between different cooperating organizations, and are also well suited for certain functionalities such as adding auditing to distributed computation on encrypted data. Dynamic systems on the other hand allow for reusing previously computed ciphertexts even if shareholders join or leave the system. For example, a simple auction consists of a seller and a number of bidders. As the involved parties typically do not know each other, there can be mutual distrust. Using a hierarchical access structure that enforces that the seller (highest level) and a certain number of bidders (lower level) are present, ensures that neither the seller nor the bidders itself can cheat, and thus creates trust. In addition, in online auctions it is common that not all bidders are present from the very beginning and some bidders might leave after a certain amount of time. Therefore, bidders need to be able to join and leave dynamically.

In the past, many (t, n) threshold encryption schemes have been proposed (e. g., [10, 35, 34, 11, 16, 29]). Some of these cryptosystems are homomorphic [9, 12, 22], others are dynamic (e. g., [18, 17]), and yet others support hierarchical access structures (e. g., [1, 32]). However, none of these schemes supports all three properties simultaneously. Straight forward constructions of a dynamic and hierarchical secret sharing scheme from Shamir's Secret Sharing scheme either require the threshold to be adapted continuously or would enable a subset of shareholders to prevent decryption. Recently, a dynamic and hierarchical secret sharing scheme based on Birkhoff interpolation has been proposed [39] that does not exhibit such disadvantages. However, a threshold cryptosystem that uses this secret sharing scheme has not been proposed yet.

In this paper we propose the first hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer and prove its security in the malicious adversary model. We tackle the problem in two steps. First, we develop a verifiable hierarchical and dynamic secret sharing scheme without trusted dealer that shares a secret over the integers. The new secret sharing scheme combines the verifiable (t, n) threshold sharing scheme over the integers in [29] with the dynamic and verifiable hierarchical secret sharing scheme in [39]. Second, we develop a hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer based on the threshold Paillier cryptosystem without trusted dealer in [29].

We start by briefly discussing related work in Section 2. In Section 3, we introduce the cryptographic background. In Section 4, we present the verifiable hierarchical and dynamic secret sharing over the integers and continue in Section 5 with presenting the hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer.

2. Related Work

Secret Sharing: Blakley [4] suggested the first secret sharing to securely store keys. Shortly afterwards, a much more efficient (t, n) threshold secret sharing scheme was introduced by Shamir [36], in which the secret is shared among n shareholders such that reconstructing the secret requires at least t shares, while with $t-1$ or fewer shares one learns nothing. Early secret sharing schemes supporting general access structures were introduced, e. g., in [25, 3] and require the shareholders to store exponentially many shares. In order to support hierarchical access structures, Shamir suggested that higher-level shareholders should receive more shares [36].[1] This approach is not optimal, as it overloads these higher-level shareholders with more shares to protect and manage. Furthermore, the resulting scheme is not ideal, meaning that the length of the shares is longer than the shared secret which can result in high storage consumption. Simmons [37] and Brickell [7] made steps forward to hierarchical secret sharing. However, as shown in [38] these schemes are not efficient as in the worst case the dealer has to preform exponentially many checks or find an irreducible polynomial. Later, Ghodosi et al. [19] introduced efficient schemes for specific access structures, but without the ability to add or delete shareholders. Tassa [38] provided the first polynomial-based hierarchical secret sharing schemes based on Birkhoff interpolation, i. e., a generalization of the Lagrange interpolation problem used in Shamir's threshold secret sharing scheme. The sharing phase of the scheme is based on evaluation of point on polynomials or derivatives of polynomials and an efficient reconstruction algorithm. Finally, Traverso et al. [39] extended Tassa's hierarchical secret sharing schemes to a hierarchical and dynamic secret sharing scheme, i. e., it allows adding and deleting shareholders, modifying the access structure and refreshing the shares periodically to enhance the security. Furthermore, Traverso et al. [40] provided algorithms to perform computations over hierarchically shared secrets. However, these secret sharing schemes in general do not easily allow reusing computations if shareholders join or leave during the protocol execution, e. g., in online algorithms.

Threshold Cryptosystems: The need for threshold cryptosystems was first discussed in [13]. After the seminal paper by Desmedt [14], several approaches to enable threshold cryptosystem have been proposed. Threshold cryptosystems for general access structure have been proposed by [26, 23] but are not ideal or require a trusted dealer, respectively. Shamir's secret sharing scheme was heavily used to develop (t, n) threshold versions for various cryptosystems [10, 35, 34, 11]. Fouque et al. [16] developed a threshold Paillier cryptosystem that uses a trusted dealer to generate the public and private key. After the key generation, the private key is shared with Shamir's secret sharing

[1] See Remark 3.6 in Section 3.4 why a (t, n)-threshold secret sharing scheme cannot easily be used to implement a hierarchical and dynamic secret sharing scheme.

scheme. Nishide and Sakurai [29] showed how to implement a fully distributed threshold Paillier cryptosystem, i. e., the key generation and the sharing of the private key is performed without a trusted dealer. However, these (t, n) threshold cryptosystems do neither support hierarchical access structures nor can they be dynamically modified. Dynamic (t, n) threshold cryptosystems that allow adding new shareholders either have large overhead [18], or use a trusted dealer and new shareholders get non-ideal shares [17]. Hierarchical threshold cryptosystems are less well studied than (t, n) threshold cryptosystem. An early approach generates keys by applying an one way function multiple times [1]. The final key can not be influenced in advance and thus is not suitable for general public key cryptosystems which typically require keys with special properties. The first distributed key generation for a discrete log based cryptosystem without a trusted dealer was introduced by [32] and uses the hierarchical secret sharing of [38]. However, the system is not dynamic and cannot be used for other cryptosystems like RSA or the Paillier cryptosystem. The latter cryptosystem is widely used [9, 12] in the context of secure multi-party computations, as it is additively homomorphic.

None of these approaches allows to implement a hierarchical and dynamic threshold cryptosystem that is homomorphic.

3. Preliminaries

In this section we first recap some basic notations and definitions. We continue with a brief description of secret sharing [36] in general, a hierarchical secret sharing scheme [38], and a hierarchical and dynamic secret sharing scheme [39]. We conclude with the Paillier cryptosystem [31], and a threshold version of Paillier's cryptosystem without trusted dealer [29].

3.1. *Notation.* Throughout the paper we will use the following notation: We denote the set of natural numbers with and without 0 as $\mathbb{N}$ and $\mathbb{N}_0$, respectively. The cardinality of a finite set A is denoted by $|A|$ and the power set is denoted by $\mathcal{P}(A)$. We write $r \leftarrow_{\$} A$ to denote that r is sampled uniformly at random from the set A. For a natural number $b \in \mathbb{N}$ we define the set $\mathbb{Z}_b := \{0, 1, \ldots, b-1\}$ and $\mathbb{Z}_b^* := \{a \in \mathbb{Z}_b \mid \gcd(a, b) = 1\}$. Euler's totient function is denoted by $\varphi(N) := |\mathbb{Z}_N^*|$ and the Carmichael function is denoted by $\lambda(N)$. For $N = pq$, where p and q are two distinct primes, it is $\varphi(N) = (p-1)(q-1)$ and $\lambda(N) := \operatorname{lcm}(p-1, q-1)$, where $\operatorname{lcm}(p-1, q-1)$ is the least common multiple of $p-1$ and $q-1$. We denote the ceiling function for a real value c by $\lceil c \rceil$, i. e., $\lceil c \rceil$ is the smallest integer greater than or equal to c.

3.2. *Secret Sharing.* A *secret sharing scheme* is a cryptographic primitive to share a *secret* among different parties referred to as *shareholders*. An *access structure* defines sets of shareholders who are allowed to reconstruct the secret.

DEFINITION 3.1 (Secret Sharing Scheme [39]). *Let $\mathcal{M}$ be a set of secrets and let Σ be a set of shares. Let $\mathcal{S} := \{s_1, \ldots, s_n\}$ be a set of n shareholders and let $\Gamma \subseteq \mathcal{P}(\mathcal{S})$ be an access structure over $\mathcal{S}$. A* secret sharing scheme *consists of two* probabilistic polynomial-time algorithms ***Share*** *and* ***Reconstruct****:*

Share: *The algorithm takes as input a secret $m \in \mathcal{M}$ and outputs n shares $\sigma_i \in \Sigma$ for $i \in \{1, \ldots, n\}$, where the share σ_i is sent to shareholder $s_i \in \mathcal{S}$.*

Reconstruct: *Let $\mathcal{R} := \{s_{i_1}, \ldots, s_{i_k}\} \subseteq S$ be a subset of shareholders, where $i_j \in \{1, \ldots, n\}$ for all $j \in \{1, \ldots, k\}$. The algorithm takes as input a set of shares $\sigma_{i_1}, \ldots, \sigma_{i_k} \in \Sigma$. If $\mathcal{R} \notin \Gamma$ the algorithm outputs $\perp$, otherwise the algorithm outputs the shared secret m.*

A (t, n) threshold scheme [36] is a secret sharing scheme where a secret $m \in \mathcal{M}$ is split into n shares $\sigma_1, \ldots, \sigma_n \in \Sigma$. Any t or more different shares σ_i can be used to reconstruct the secret m, but with any $t-1$ or fewer shares one gains no knowledge about m in an information theoretic sense. The most well known threshold secret sharing scheme is Shamir's secret sharing scheme [36].

3.3. *Hierarchical Secret Sharing.* A *hierarchical access structure* assumes a hierarchy between the shareholders. When devising a hierarchical threshold secret sharing scheme, the access structure needs to reflect the position of the shareholders in this hierarchy. This can be done in multiple different ways, e. g., with ideal weighted access structures where the total weight of the shareholders has to surpass a threshold [2]. More generally one may only require that any shareholder in an authorized set has to be replaceable by a hierarchically superior one [15].

We here focus on two very intuitive approaches to reflect hierarchies, namely a *disjunctive* and a *conjunctive* approach [38]. In a disjunctive approach each level in the hierarchy is associated with a strictly increasing threshold on the number of shares required for reconstruction. A set of shareholders can reconstruct the secret if it meets the threshold on at least one level, where higher level shareholders may replace lower level share holders. In a conjunctive access structure a minimum number of shareholders for each level is required to participate in the reconstruction where again lower level shareholders may be replaced by higher level shareholders. These two approaches are captured in the following definition [38]:

DEFINITION 3.2 (Conjunctive, Disjunctive $(\vec{t}, n)$-Hierarchical Threshold Access Structure [38]). *Let $\mathcal{S}$ be a set of n shareholders composed of $l+1$ pairwise disjoint levels $\mathcal{S}_h$ for $h \in \{0, \ldots, l\}$ s. t. $\mathcal{S} := \bigcup_{h=0}^{l} \mathcal{S}_h$. Level 0 is the* highest level *and level l is the* lowest level. *Let $\vec{t} := (t_0, t_1, \ldots, t_l) \in \mathbb{N}^{l+1}$ be a monotonically increasing vector of integers, i. e., $t_0 < t_1 < \ldots < t_l$.*

The conjunctive $(\vec{t}, n)$ hierarchical threshold access structure *is defined as*

$$\Gamma := \left\{ \mathcal{B} \subseteq \mathcal{S} : \forall h \in \{0, \ldots, l\} \text{ it is } \left| \mathcal{B} \cap \bigcup_{i=0}^{h} \mathcal{S}_i \right| \geq t_h \right\}.$$

The disjunctive $(\vec{t}, n)$ hierarchical threshold access structure *is defined as*

$$\Gamma := \left\{ \mathcal{B} \subseteq \mathcal{S} : \exists h \in \{0, \ldots, l\} \text{ s. t. } \left| \mathcal{B} \cap \bigcup_{i=0}^{h} \mathcal{S}_i \right| \geq t_h \right\}.$$

In the following, we will refer to these access structures also just by conjunctive *and* disjunctive access structure*.*

In [38] Tassa introduced a *hierarchical secret sharing scheme* based on Birkhoff interpolation that can be used for conjunctive as well as disjunctive access structures. The shares of the shareholders are points of the actual polynomial $f(x)$ and points of its derivatives $f^{(j)}(x)$, where $f^{(j)}(x)$ denotes the j-th derivative of $f(x)$. If the shareholders want to reconstruct the secret, they need to solve a Birkhoff interpolation problem. The conjunctive hierarchical secret sharing scheme is described in Protocol 3.1. The disjunctive version is given in brackets whenever it differs from the conjunctive scheme.

Protocol 3.1 (Hierarchical Secret Sharing Scheme [38]). *Let $\mathcal{S}$ be a set of shareholders composed of $l+1$ pairwise disjoint levels. Let $n_{-1} := 0$ and let $n_h \in \mathbb{N}$ be such that $n_h \geq n_{h-1} + |\mathcal{S}_h|$ for $h \in \{0, \ldots, l\}$. Let $\Gamma \subseteq \mathcal{P}(\mathcal{S})$ be a conjunctive (disjunctive) $(\vec{t}, n_l)$ hierarchical threshold access structure with $\vec{t} := (t_0, \ldots, t_l) \in \mathbb{N}^{l+1}$ and let $t_{-1} := 0$. Let (i, j) denote the unique ID of shareholder $s_{i,j} \in \mathcal{S}_h$ for $h \in \{0, \ldots, l\}$, where $i \in \{n_{h-1}+1, \ldots, n_h\}$ and $j := t_{h-1}$ ($j := t_l - t_h$). Let $p > 2^{-(t_l-2)} \cdot (t_l-1)^{\frac{t_l-1}{2}} \cdot (t_l-1)! \cdot n_l^{\frac{(t_l-2)(t_l-1)}{2}}$ be a prime. Let $m \in \mathbb{N}_0$ be the secret. The* ***Share*** *algorithm proceeds as follows:*

1. *Set $a_0 := m \bmod p$ ($a_{t_l-1} := m \bmod p$) and select $a_1, \ldots, a_{t_l-1} \leftarrow_{\$} \mathbb{Z}_p$ ($a_0, \ldots, a_{t_l-2} \leftarrow_{\$} \mathbb{Z}_p$).*
2. *Construct $f(x) := \sum_{\omega=0}^{t_l-1} a_\omega x^\omega \bmod p$.*
3. *Distribute the share $\sigma_{i,j} := f^{(j)}(i) \bmod p$ to shareholder $s_{i,j}$, for all $s_{i,j} \in \mathcal{S}$.*

The ***Reconstruct*** *algorithm proceeds as follows to reconstruct a shared secret: Let $\mathcal{R} \subseteq \mathcal{S}$ be a minimal authorized set.*

1. *If $\mathcal{R} \notin \Gamma$, the algorithm outputs $\perp$.*
2. *If $\mathcal{R} \in \Gamma$, the algorithm solves the Birkhoff interpolation problem and outputs $m \equiv a_0 \bmod p$ ($m \equiv a_{t_l-1} \bmod p$).*

The condition $n_h \geq n_{h-1} + |\mathcal{S}_h|$ for $h \in \{0, \ldots, l\}$ in Protocol 3.1 is required for extending the hierarchical secret sharing scheme to a dynamic

secret sharing scheme in Section 3.4. It ensures that new shareholders obtain a unique ID, which guarantees correctness. Thus, the values n_h for $h \in \{0, \dots, l\}$ must be chosen in advance, as it restricts level h to a maximal number of $n_h - n_{h-1}$ shareholders. Note that the maximal number of shareholder that are supported is then $n_l \geq n = |\mathcal{S}|$.

REMARK 3.3. The condition $p > 2^{-(t_l-2)} \cdot (t_l-1)^{\frac{t_l-1}{2}} \cdot (t_l-1)! \cdot n_l^{\frac{(t_l-2)(t_l-1)}{2}}$ yields an upper bound for the size of the *reconstruction matrix* and thus guarantees that its inverse exists [38].

The reconstruction matrix is used to compute a solution for the Birkhoff interpolation problem and is defined as follows:

DEFINITION 3.4 (Reconstruction Matrix [27]). *Let $E := (e_{i,j})_{i=1\,j=0}^{k\quad r}$ be an interpolation matrix, i. e., a binary matrix where $e_{i,j} \in \{0,1\}$ for $i \in \{1, \dots, k\}$ and $j \in \{0, \dots, r\}$, and in addition $\sum_{i=1}^{k} \sum_{j=0}^{r} e_{i,j} = t$. Let $X := \{x_1, \dots, x_k\} \subset \mathbb{N}_0$ and $\mathcal{G} := \{g_0, \dots, g_{t-1}\}$ be a system of linearly independent and r times continuously differentiable real-valued functions. Let $g_i^{(j)}$ denote the j-th derivative of g_i. Let $\mathrm{I}(E) := \{(i,j) : i \in \{1, \dots, k\}, j \in \{0, \dots, r\}, e_{i,j} = 1\}$ be the ordered set of indices of the interpolation matrix where the entries are 1. Specifically, $\mathrm{I}(E) := \{(i_0, j_0), \dots, (i_{t-1}, j_{t-1})\}$ is ordered such that (i,j) precedes (i',j') if and only if $i < i'$ or $i = i' \wedge j < j'$. The* reconstruction matrix $\mathrm{A}(E, X, \mathcal{G})$ *is defined as follows:*

$$\mathrm{A}(E, X, \mathcal{G}) := \begin{pmatrix} g_0^{(j_0)}(x_{i_0}) & \cdots & g_{t-1}^{(j_0)}(x_{i_0}) \\ \vdots & \ddots & \vdots \\ g_0^{(j_{t-1})}(x_{i_{t-1}}) & \cdots & g_{t-1}^{(j_{t-1})}(x_{i_{t-1}}) \end{pmatrix}$$

The reconstruction matrix is indexed starting from 0, i. e., $\mathrm{A}(E, X, \mathcal{G})$ is a $t \times t$ matrix.

In the conjunctive hierarchical secret sharing scheme it holds that $X := \{x_1, \dots, x_k\} = \{i \mid s_{i,j} \in \mathcal{S}\}$ and $\mathcal{G} := \{g_0, \dots, g_{t-1}\} = \{1, x, x^2, \dots, x^{t_l-1}\}$, i. e., $g_\omega(x) := x^\omega$. The set $\mathrm{I}(E) := \{(i,j) \mid s_{i,j} \in \mathcal{R}\}$ is the set of the IDs of the shareholders reconstructing the secret.

The original polynomial can be reconstructed with

$$f(x) \equiv \sum_{\omega=0}^{t-1} a_\omega g_\omega(x) \bmod p$$

where

$$a_\omega := \frac{\det(\mathrm{A}(E, X, \mathcal{G}_\omega))}{\det(\mathrm{A}(E, X, \mathcal{G}))}.$$

The matrix $\mathrm{A}(E, X, \mathcal{G}_\omega)$ is obtained from $\mathrm{A}(E, X, \mathcal{G})$ by replacing its ω-th column with the shares $\sigma_{i,j}$ in lexicographic order [39]. We want to stress the

fact that a coefficient a_ω can be computed in a distributed fashion [39] with Laplace's expansion formula: It is $a_\omega \equiv \sum_{u=0}^{t-1} a_{u,\omega} \bmod p$, where

$$a_{u,\omega} := \sigma_{i_u,j_u}(-1)^{u+\omega} \cdot \frac{\det(\mathrm{A}_{u,\omega}(E,X,\mathcal{G}))}{\det(\mathrm{A}(E,X,\mathcal{G}))}$$

and $\mathrm{A}_{u,\omega}(E,X,\mathcal{G})$ is the matrix that results from $\mathrm{A}(E,X,\mathcal{G})$ by removing the u-th row and the ω-th column. We will write A for $\det(\mathrm{A}(E,X,\mathcal{G}))$ and $\mathrm{A}_{u,\omega}$ for $\det(\mathrm{A}_{u,\omega}(E,X,\mathcal{G}))$. In addition, the j-th derivative $f^{(j)}(x)$ of the interpolation polynomial $f(x)$ can be computed in a distributed fashion [39]: It holds that $f^{(j)}(x) \equiv \sum_{u=0}^{t-1} f_u^{(j)}(x) \bmod p$ where

$$f_u^{(j)}(x) := \sum_{\omega=0}^{t-1} a_{u,\omega} g_\omega^{(j)}(x) \bmod p. \tag{3.1}$$

3.4. *Dynamic Secret Sharing.* In the previous sections we considered fixed access structures, i. e., the access structure is not altered after the **Share** algorithm. However, there are many cases where the access structure needs to be altered, e. g., a shareholder shall be added or removed. Next, we describe the flavor of dynamic secret sharing scheme coined in [39] and present how the authors of [39] turned the hierarchical secret sharing scheme of [38] into a *dynamic secret sharing scheme.*

Definition 3.5 (Dynamic Secret Sharing Scheme [39]). *A* dynamic secret sharing scheme *consists of two additional probabilistic polynomial-time algorithms* ***Add*** *and* ***Reset*** *compared to Theorem 3.1:*

Add: *Let $\mathcal{R} := \{s_{i_1}, \ldots, s_{i_k}\} \subseteq S$ be a subset of shareholders, where $i_j, k \in \{1, \ldots, n\}$ for all $j \in \{1, \ldots, k\}$. Let $s_{n+1} \notin \mathcal{S}$ be a new shareholder. The algorithm takes as input a set of shares $\sigma_{i_1}, \ldots, \sigma_{i_k} \in \Sigma$. If $\mathcal{R} \notin \Gamma$ it outputs $\perp$. If $\mathcal{R} \in \Gamma$ it outputs without secret reconstruction a new share $\sigma_{n+1} \in \Sigma$, where the share σ_{n+1} is sent to the new shareholder s_{n+1}.*

Reset: *Let $\mathcal{R} = \{s_{i_1}, \ldots, s_{i_k}\} \subseteq S$ be a subset of shareholders, where $i_j, k \in \{1, \ldots, n\}$ for all $j \in \{1, \ldots, k\}$. Let $\mathcal{S}' = \{s'_1, \ldots, s'_{n'}\}$ be a new set of shareholders. Let $\Gamma' \subseteq \mathcal{P}(\mathcal{S}')$ be a new access structure. The algorithm takes as input a set of shares $\sigma_{i_1}, \ldots, \sigma_{i_k} \in \Sigma$. If $\mathcal{R} \notin \Gamma$ it outputs $\perp$. If $\mathcal{R} \in \Gamma$ it outputs without secret reconstruction n' new shares $\sigma'_i \in \Sigma$ for $i \in \{1, \ldots, n'\}$, where the share σ'_i is sent to the shareholder $s'_i \in \mathcal{S}'$ and the old shares $\sigma_1, \ldots, \sigma_n \in \Sigma$ are deleted.*

The **Reset** algorithm has no constraints on the new set of shareholders $\mathcal{S}'$, e. g., $\mathcal{S}$ and $\mathcal{S}'$ can but do not have to be disjoint. Note that Theorem 3.5 allows that a previously unauthorized set $\mathcal{R} \notin \Gamma$ can become an authorized set $\mathcal{R} \in \Gamma'$ after the execution of the **Reset** algorithm. This is a desired property, as the purpose of the **Reset** algorithm is to create a completely new access structure Γ'. Nevertheless, the **Reset** algorithm provides security

in the presence of mobile adversaries [39], i. e., the adversary can alter the set of corrupted shareholders: The shares before and after executing the **Reset** algorithm cannot be combined. Thus, if the **Reset** algorithm is executed often enough, then the mobile adversary will never obtain enough shares to reconstruct the secret.

The hierarchical secret sharing scheme as introduced in [38] (see Protocol 3.1) can be extended to a *dynamic and hierarchical secret sharing scheme* [39]. The **Add** and **Reset** algorithm of this scheme are described in Protocols 3.3 and 3.4 and make use of the sub-algorithm **Birkhoff Setup** descried in Protocol 3.2. The **Birkhoff Setup** sub-algorithm is also used in the hierarchical threshold Paillier cryptosystems in Section 5.

PROTOCOL 3.2 (**Birkhoff Setup**). *Let $\mathcal{R} \subseteq \mathcal{S}$ be a subset of shareholders.*

1. *If $\mathcal{R} \notin \Gamma$, the algorithm outputs $\perp$. If $\mathcal{R} \in \Gamma$, we assume that $\mathcal{R}$ is a minimal authorized set.*
2. *Set $X := \{x_1, \ldots, x_k\} = \{i : s_{i,j} \in \mathcal{R}\}$ s. t. $x_1 < \ldots < x_k$, set $d := \min\{j : s_{i,j} \in \mathcal{R}\}$, set $r := \max\{j : s_{i,j} \in \mathcal{R}\} - d$, and set $E := (e_{i,j})_{i=1}^{k}{}_{j=0}^{r}$ s. t. $e_{i,j} = 1$ if $s_{x_i, j+d} \in \mathcal{R}$ and otherwise $e_{i,j} = 0$.*
3. *Set $\mathcal{G} := \left\{ d!, \frac{(d+1)!}{1!}x, \ldots, \frac{(t_l-1)!}{(t_l-1-d)!}x^{t_l-1-d} \right\}$.*
4. *Let $\mathrm{I}(E) := \{(i_0, j_0), (i_1, j_1), \ldots, (i_{k-1}, j_{k-1})\}$.*
5. *Let $\mathrm{A} := \det(\mathrm{A}(E, X, \mathcal{G}))$ and let $b := 0$ ($b := t_l - 1$).*

The **Add** algorithm is implemented by computing a point or derivative of the function $f(x)$ in a distributed fashion [39]. The **Reset** algorithm can be implemented by computing the shares of a new polynomial $f'(x)$ in a distributed fashion such that $f'(0) \equiv m \equiv f(0) \bmod p$ holds [39].

PROTOCOL 3.3 (**Add** [39]). *Let $\mathcal{R} \subseteq \mathcal{S}$ be an authorized set, then a new shareholder is **Added** as follows:*

1. *Let (i', j') be the ID of the new shareholder $s_{i',j'}$.*
2. *Execute Protocol 3.2 (**Birkhoff Setup**).*
3. *Let $s_u := s_{i_u, j_u}$ and $\sigma_u := \sigma_{i_u, j_u}$ for $(i_u, j_u) \in \mathrm{I}(E)$.*
4. *Each shareholder $s_u \in \mathcal{R}$ computes and splits*

$$f_u^{j'}(i') := \sigma_u \sum_{\omega=j'}^{t_l-1} \frac{\omega!(-1)^{u+\omega}}{(\omega - j')!} \cdot \frac{\mathrm{A}_{u,\omega-d}}{\mathrm{A}} i'^{\omega - j'} \bmod p$$

   s. t. $f_u^{j'}(i') \equiv \lambda_{0,u} + \ldots + \lambda_{k-1,u} \bmod p$ and sends $\lambda_{u',u}$ to shareholder $s_{u'} \in \mathcal{R}$.
5. *Each Shareholder $s_u \in \mathcal{R}$ sends $\delta_u := \sum_{u'=0}^{k-1} \lambda_{u,u'} \bmod p$ to shareholder $s_{i',j'}$.*
6. *The new share is then $\sigma_{i',j'} := \sum_{u=0}^{k-1} \delta_u \bmod p$.*

PROTOCOL 3.4 (**Reset** [39]). *Let Γ' be the new access structure for new shareholders $s'_{i',j'} \in \mathcal{S}'$. Let $\mathcal{R} \subseteq \mathcal{S}$ be an authorized set, then Γ is* ***Reseted*** *as follows:*

1. *Execute Protocol 3.2 (**Birkhoff Setup**).*
2. *Let $s_u := s_{i_u,j_u}$ and $\sigma_u := \sigma_{i_u,j_u}$ for $(i_u, j_u) \in \mathrm{I}(E)$.*
3. *Each shareholder $s_u \in \mathcal{R}$ computes*

$$a_{u,b} := \sigma_u (-1)^{u+b-d} \frac{\mathrm{A}_{u,b-d}}{\mathrm{A}} \mod p$$

   and shares $a_{u,b}$ with Protocol 3.1 according to Γ'. Let $\lambda^u_{i',j'}$ be the share for shareholder $s'_{i',j'} \in \mathcal{S}'$.
4. *Old shares $\sigma_{i,j}$ of $s_{i,j} \in \mathcal{S}$ are deleted and the new shares of $s'_{i',j'} \in \mathcal{S}'$ is $\sigma'_{i',j'} := \sum_{u=0}^{k-1} \lambda^u_{i',j'} \mod p$.*

REMARK 3.6 (Why a (t, n)-threshold secret sharing scheme is not enough). To the best of our knowledge, there are two main techniques using a (t, n)-threshold secret sharing scheme to realize a conjunctive hierarchical access structure:

1. Shareholders obtain different amounts of shares corresponding to their level.
2. The secret is split in multiple parts (one for each level). Each part is then shared with a threshold secret sharing scheme.

The first approach does not allow to model any disjunctive hierarchical access structures with support for arbitrary number of shareholders: Assume the following $((1, 3), n)$ hierarchical access structure, i. e., reconstruction can be done by three shareholders where at least one is from level 0, and the other two are from level 0 or level 1. Let t be the threshold of Shamir's secret sharing scheme, and let A and B denote the number of shares a shareholder in level 0 and level 1 obtains, respectively. Assume we want to support x shareholds in level 1 and at least 3 in level 0. Then it has to hold that $X \cdot B < t$ and $2A < t$. However, according to the access structure it must be that $t \leq A + 2B$. Combining these inequalities lead to $(X - 2)B < A$ and $A < 2B$, and thus $(X - 2)B < 2B \Leftrightarrow X < 4$. Hence, this access structure can support at most 3 shareholders in level 1. Note, the hierarchical secret sharing scheme based on Birkhoff interpolation has also a limit on the number of shareholders per levels once the shares are distributed. However, this limit can be chosen arbitrarily large in advance during the setup, and thus it is only a minor restriction w. r. t. practical applications.

In the second approach, a subset of shareholders can prevent secret reconstruction, namely if all shareholders of one part of the secret refuse to participate. This cannot happen in the hierarchical secret sharing scheme based on Birkhoff interpolation, as higher level shareholder are able to replace lower level shareholders.

3.5. *Security Model.* A threshold cryptosystem is a cryptosystem where t parties can jointly decrypt a ciphertext [29]. In a public key cryptosystem the private key is typically shared among the parties. Each shareholder is in the possession of a secret share σ_i. An authorized set of shareholders can jointly decrypt a ciphertext c by combing their shares: Each shareholder s_i uses their secret share σ_i and the ciphertext c to compute a *partial decryption* c_i. The partial decryptions c_i are sent to the *combiner*, i. e., the party who shall learn the decryption of c. The combiner uses the partial decryptions to obtain the plaintext.

We differentiate between two main adversary types: A *semi-honest* adversary and a *malicious* adversary [24]. A semi-honest adversary behaves like an honest party and follows the protocol as intended. However, a semi-honest adversary might store all messages received and carry out additional computations in order to deduce more information than the intended output. The malicious adversary can additionally arbitrarily deviate from the protocol, i. e., send inconsistent inputs, send different messages deviating from the protocol specifications or even refuse to send messages and abort the protocol at any point in time.

Semi-honest or malicious adversaries can additional be *mobile* adversaries [21], i. e., the set of corrupted parties can be altered, but the total amount of corrupted parties at any point in time is limited. In the real world such a mobile adversary can occur if the adversary can break into different servers storing the shares or is able to bribe the shareholders [39]. At the same time, the administrators of the corresponding servers can slowly block the adversary or the bribed shareholders reveal their shares only once, and thus the set of corrupted parties can change over time while the total number of parties is still limited. The counter part is called a *static* adversary, i. e., the set of corrupted parties cannot be altered once it is chosen by the adversary.

Our hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer (see Section 5) is *robust* and *threshold semantic secure* in the malicious adversary model. We therefore introduce these two notions of security and adapted the definitions of [16, 29] to general access structures: A robust threshold cryptosystem guarantees that the *combiner*, i. e., the party that shall learn the plaintext, obtains a correct decryption.

Definition 3.7 (Robustness [16, 29]). *Let $\mathcal{S}$ be the set of shareholders and let Γ be the access structure. Let $\mathcal{R} \subseteq \mathcal{S}$ be the set of shareholders participating in the decryption. Let $\mathcal{U} \subset \mathcal{R}$ be a set of unauthorized shareholders ($\mathcal{U} \notin \Gamma$) chosen and controlled by a malicious adversary such that $\mathcal{R} \setminus \mathcal{U} \in \Gamma$.*

A threshold cryptosystem is said to be robust *if the combiner is able to correctly decrypt any ciphertext, even in the presence of a malicious adversary who corrupts all shareholders in $\mathcal{U}$.*

The definition of threshold semantic security is based on a game between the adversary and a challenger. Again, we adapted the definitions in [16, 29] and provide a version for general access structures:

DEFINITION 3.8 (Threshold Semantic Security [16, 29]). *Let $\mathcal{S}$ be the set of shareholders and let Γ be the access structure. Let $\mathcal{R} \subseteq \mathcal{S}$ be the set of shareholders participating in the decryption. Let $k \in \mathbb{N}$ be a security parameter and let $K : \mathbb{N} \to \mathbb{N}$ be a function s. t. $1/\mathrm{K}(\cdot)$ is negligible. The game between the challenger and the malicious adversary proceeds as follows:*

1. *The adversary selects a set $\mathcal{U} \subset \mathcal{R}$ of unauthorized shareholders ($\mathcal{U} \notin \Gamma$) such that $\mathcal{R} \setminus \mathcal{U} \in \Gamma$. The challenger controls the honest parties.*
2. *The key generation is performed and the adversary learns the public key, all verification shares and all secret information generated or obtained by any of the corrupted shareholders in $\mathcal{U}$.*
3. *The adversary chooses a message m and sends it to a partial decryption oracle that returns the encryption c of m and the partial decryptions c_i for $s_i \in \mathcal{R} \setminus \mathcal{U}$ of the honest parties to the adversary. The adversary may repeat this step a polynomial number of times in k.*
4. *The adversary sends two messages m_0, m_1 to the challenger. The challenger picks a random bit $b \in \{0, 1\}$ and sends the ciphertext c, which is the encryption of m_b, to the adversary.*
5. *The adversary repeats Step 3..*
6. *The adversary outputs a guess $b' \in \{0, 1\}$.*

A threshold cryptosystem is said to be threshold semantic secure *if it holds that $\left|\Pr(b = b') - \frac{1}{2}\right| \leq 1/\mathrm{K}(k)$.*

In order for the fully distributed protocols to be robust, the participating parties need to follow the protocol specification. For this purpose we make use of *Pedersen commitments* [33] and *zero-knowledge proofs* [20], where the *prover* can prove knowledge or a certain relation. The interfaces of the used zero-knowledge proofs are given in Protocols 3.5 to 3.7.

PROTOCOL 3.5 (Discrete Logarithm Equality [16]). *For public g_1, g_2, y_1, y_2 and private x the prover can show that $\log_{g_1}(y_1) = \log_{g_2}(y_2) = x$.*

PROTOCOL 3.6 (Zero-knowledge Range Proof [30]). *For public v, g, h, p and private s, r the prover can show that $v \equiv g^s h^r \mod p$ and that s is from a certain range.*

PROTOCOL 3.7 (Zero-knowledge Proof for Verification [29, Appendix C]). *For public g, h, p, v, N and public commitments $g^\sigma h^{\sigma'} \mod p, v^\sigma$ and private σ, σ' the prover can show that $\log_v\left(v^\sigma \mod N^2\right) = \sigma$ holds.*

3.6. *Threshold Paillier Cryptosystem without Trusted Dealer.* The *Paillier cryptosystem* [31] is an asymmetric and probabilistic encryption scheme, i. e.,

one message can encrypt to multiple ciphertexts. In addition, it is also additive homomorphic, i. e., the multiplication of two ciphertexts results in the addition of the corresponding plaintexts. The **Encryption** and **Decryption** algorithm are given in Protocol 3.8.

PROTOCOL 3.8 (Paillier Cryptosystem [31]). *Let $N := pq$, where p and q are two distinct primes such that $\gcd(N, \varphi(N)) = 1$. Let g be a generator of $W_N := \{a \in \mathbb{Z}^*_{N^2} : a^N \equiv 1 \bmod N^2\}$. The value N and the generator g are public.*

*The **Encryption** is performed as follows:*

1. *Let $m \in \mathbb{Z}_N$ be the message to be encrypted.*
2. *Select $r \leftarrow_\$ \mathbb{Z}^*_N$.*
3. *The ciphertext is then $c := g^m r^N \bmod N^2$.*

*The **Decryption** is performed as follows:*

1. *Let $c \in \mathbb{Z}^*_{N^2}$ be the ciphertext to be decrypted.*
2. *Message $m \in \mathbb{Z}_N$ is obtained by computing $m := \frac{\mathrm{L}\left(c^{\lambda(N)} \bmod N^2\right)}{\mathrm{L}\left(g^{\lambda(N)} \bmod N^2\right)} \bmod N$ where $\mathrm{L}(w) := \frac{w-1}{N}$.*

Nishide and Sakurai implemented a fully distributed Paillier cryptosystem without trusted dealer [29], i. e., the primes, the modulus, the private key and corresponding shares are generated without a trusted dealer. In the following we give a brief description of their threshold Paillier cryptosystem without trusted dealer [29], as we will reuse their key generation for our *hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer* in Section 5.

We first present their verifiable secret sharing over the integers as introduced in [29]. This scheme is required to share the private key: Recall that $\lambda(N)$ is the private key of the Paillier cryptosystem (cf. Protocol 3.8). The private key cannot be shared with Shamir's secret sharing scheme (or a variant with different modulus), as for correctness it would have to be shared modulo a multiple of $\lambda(N)$, which then allows decryption by knowing only the modulus. Thus, the private key is shared over the integers to prevent unauthorized shareholders obtaining the private key. The scheme is based on Shamir's secret sharing scheme, but the coefficients of the polynomial are chosen from a larger range. The actual shared secret is $m \cdot n!$ instead of m, where n is the number of shareholders. This is required to prevent information leakage, as otherwise an adversary with $t-1$ shares can guess a secret m' and check if the coefficients of the reconstructed polynomial are all integers. If a fraction occurs (caused by the Lagrange interpolation), then the guessed secret m' can be excluded from the possible secrets. The **Share** and **Reconstruct** algorithm are described in Protocol 3.9. Verifiability can be achieved with standard techniques as shown in the full version in [29].

PROTOCOL 3.9 (Secret Sharing over the integers [29]). *Let $z \in \mathbb{N}$ and let the secret be $m \in [-z, z] \subset \mathbb{Z}$. Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ be a set of n shareholders. Let $t \in \{1, \ldots, n\}$ be the threshold and let $\Delta := n!$. Let $k \in \mathbb{N}$ be a security parameter and let $K : \mathbb{N} \to \mathbb{N}$ be a function s. t. $1/\mathrm{K}(\cdot)$ is negligible. The* ***Share*** *algorithm proceeds as follows:*

1. *Set $a_0 := m\Delta$.*
2. *Select coefficients $a_1, \ldots, a_{t-1} \leftarrow_{\$} [-\mathrm{K}(k)z\Delta^2, \mathrm{K}(k)z\Delta^2] \subset \mathbb{Z}$.*
3. *Construct $f(x) := \sum_{\omega=0}^{t-1} a_\omega x^\omega$.*
4. *Send the share $\sigma_i := f(i)$ to shareholder $s_i \in \mathcal{S}$.*

An authorized set of shareholders $\mathcal{R}$, i. e., $|\mathcal{R}| \geq t$, can ***Reconstruct*** *the secret m by computing*

$$m = \frac{1}{\Delta} \sum_{s_i \in \mathcal{R}} \sigma_i \prod_{s_j \in \mathcal{R}, j \neq i} \frac{j}{j - i}.$$

The **Key Generation** of the threshold Paillier cryptosystem without trusted dealer [29] (see Protocol 3.10) is performed by n shareholders in a distributed fashion.

PROTOCOL 3.10 (Threshold Paillier cryptosystem without Trusted Dealer - **Key Generation** [29]). *Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ be a set of n shareholders. Let $t \in \mathbb{N}, t \leq (n+2)/3$ be the threshold and let $\Delta = n!$. Let $k \in \mathbb{N}$ be a security parameter, i. e., the required bit size for the generated prime candidates and let $K : \mathbb{N} \to \mathbb{N}$ be a function s. t. $1/\mathrm{K}(\cdot)$ is negligible. Let $P' > 2(2n\Delta\,(\mathrm{K}(k))^2\, N_{\max}^2(1 + tn^t\Delta\mathrm{K}(k)) + \theta_{\max})$ be a prime where $\theta_{\max} := 2n\Delta\mathrm{K}(k)(1 + \mathrm{K}(k))N_{\max}^2$, $N_{\max} := p_{\max}^2$ and $p_{\max} := 3n \cdot 2^{k-1}$. In order to create a modulus N and a private key in a distributed fashion the shareholders proceed as follows:*

1. *The shareholders perform the* distributed computation of RSA modulus *and thus obtain $N := pq$, shares $\tilde{p}_i, \tilde{q}_i$ of two prime candidates p and q with bit size k, and shares $\tilde{\varphi}_i$ of $\varphi(N)$ over $\mathbb{Z}_{P'}$.*
2. *The shareholders test p, $\frac{p-1}{2}$, q and $\frac{q-1}{2}$ for small divisors with trial division. They also test whether N is the product of two primes with the biprimality test. If a test fails, restart at Step 1..*
3. *Each shareholder s_i selects $\beta_i \leftarrow_{\$} [0, N-1]$, $r_i \leftarrow_{\$} [0, \mathrm{K}(k)N]$ and shares β_i over $\mathbb{Z}_{P'}$ and r_i over the integers with Protocol 3.9. They compute $\tilde{\theta}_i := \Delta\tilde{\varphi}_i\tilde{\beta}_i + N\Delta\tilde{r}_i \bmod P'$.*
4. *Let $\sigma_i := N\tilde{r}_i - \theta$. The corresponding verification key is $v_{\sigma_i} := g^{\sigma_i} h^{\sigma_i'} \bmod P$. The shareholders have now a sharing of $\sigma := -\Delta\varphi(N)\beta$ over the integers.*
5. *The shareholders set $v := H(N)^2 \bmod N^2$ (H is a hash function modeled as random oracle), publish verification keys $v_i := v^{\Delta\sigma_i} \bmod N^2$, and prove that $\log_{v^\Delta}(v_i) = \sigma_i$ (Protocol 3.7).*

The public key is (N, θ) *and the private key is* σ*. Note that* $\theta \equiv \Delta\varphi(N)\beta + N\Delta r \equiv \Delta\varphi(N)\beta \bmod N$*.*

The shareholders first perform a *distributed computation of an RSA modulus* (see Step 1. in Protocol 3.10), i. e., the generation of two prime candidates p and q s. t. $p \equiv q \equiv 3 \bmod 4$ (needed for the biprimality test later). In order to generate a k bit prime candidate, each shareholder $s_i \in \mathcal{S}$ selects a random value r_i from an appropriate range. Then the first shareholder computes $p_1 := 4r_1 + 3$ and the other shareholders compute $p_i := 4r_i$ for $i \in \{2, \ldots, n\}$. The final prime candidate is then $p := \sum_{i=1}^{n} p_i$. The modulus $N := pq$ is computed in a distributed fashion by multiplying the corresponding shares and reconstruction of the result [29]. In addition, the shareholders now have shares $\tilde{p}_i, \tilde{q}_i$ of the primes p, q such that they can locally compute shares $\tilde{\varphi}_i := N - \tilde{p}_i - \tilde{q}_i + 1$ of $\varphi(N)$. Note that with high probability p and q are not safe primes. However, the condition on p and q being safe primes can be relaxed when $\frac{p-1}{2}$ and $\frac{q-1}{2}$ are not divisible by any prime less than n (see [11, 29]). This is tested with trial division (see Step 2. in Protocol 3.10), i. e., testing whether p and q are dividable by small numbers. After the trial division, the shareholders perform the biprimality test [6], i. e., they test whether N is the composite of two primes. If all checks succeed, they accept the primes.

The private key is computed as follows [29] (see Step 3. and 4. in Protocol 3.10): Each shareholder s_i selects $\beta_i \leftarrow_{\$} [0, N-1]$ and shares it over $\mathbb{Z}_{P'}$ where P' is a large prime. In addition, they agree on a function $\mathrm{K}(k)$ such that $1/\mathrm{K}(k)$ is negligible in k, select $r_i \leftarrow_{\$} [0, \mathrm{K}(k)N]$ and share it over the integers with the secret sharing scheme over the integers (see Protocol 3.9). The shareholders are now in possession of shares $\tilde{\beta}_i$ and $\tilde{r}_i$ of $\beta := \sum_{i=1}^{n} \beta_i \bmod P'$ and $r := \sum_{i=1}^{n} r_i \in [0, n\mathrm{K}(k)N]$, respectively. They locally compute $\tilde{\theta}_i := \Delta\tilde{\varphi}_i\tilde{\beta}_i + N\Delta\tilde{r}_i \bmod P'$. The shareholders now have shares σ_i of the private key $\sigma := Nr - \theta = -\Delta\varphi(N)\beta$ over the integers.

The **Encryption** algorithm is the same as for the original Paillier cryptosystem (Protocol 3.8) with $g := 1 + N$. The **Decryption** algorithm is described in Protocol 3.11. The resulting system is robust and threshold semantically secure in the presence of a malicious adversary corrupting at most $t - 1$ shareholders [29].

Protocol 3.11 (Threshold Paillier cryptosystem without Trusted Dealer - **Decryption** [29]). *Let* c *be the ciphertext to be decrypted. Let* $\mathcal{R} \subseteq \mathcal{S}$ *denote the shareholders who participate in the decryption process. Each shareholder* $s_i \in \mathcal{R}$ *proceeds as follows:*

1. *Compute* $c_i := c^{2\Delta\sigma_i} \bmod N^2$*.*
2. *Prove that* $\log_{c^{4\Delta}}\left(c_i^2\right) = \log_v(v_i)$ *(Protocol 3.5).*
3. *Send* c_i *and the proof to the combiner.*

The combiner performs the following steps:

1. *Let $\mathcal{R}' \subseteq \mathcal{R}$ be the set of shareholders with correct proofs. If $|\mathcal{R}'| < t$, then the decryption fails.*
2. *Otherwise compute*

$$m := \mathrm{L}\left(\prod_{s_i \in \mathcal{R}'} c_i^{2\mu_i} \ mod\ N^2 \right) \cdot \frac{1}{-4\Delta^2\theta} \ mod\ N$$

where

$$\mu_i := \Delta \cdot \prod_{\substack{s_j \in \mathcal{R}' \\ j \neq i}} \frac{j}{j-i} \in \mathbb{Z}.$$

4. Verifiable Hierarchical and Dynamic Secret Sharing over the Integers

Our new hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer (see Section 5) needs to share a secret with a hierarchical access structure in a verifiable way over the integers. The hierarchical secret sharing scheme as presented in Sections 3.3 and 3.4 cannot be used for this, as the shareholders do not know the correct modulus or cannot publish it. In addition, the secret reconstruction is based on the quotient $\frac{\det(\mathrm{A}(E,X,\mathcal{G}_\omega))}{\det(\mathrm{A}(E,X,\mathcal{G}))}$. This quotient is not guaranteed to be an integer if the hierarchical secret sharing from [38, 39] is used over the integers directly. If an adversary corrupts an unauthorized set of shareholders such that only one shareholder is missing to form an authorized set, then the adversary could guess a possible secret and compute the quotient. If the quotient is not an integer, then the adversary can exclude the guessed secret.

Thus, we devise a novel hierarchical and dynamic secret sharing scheme that shares a secret over the integers and prove its security. In order to prevent information leakage we share $m\Lambda$ instead of only the secret m, where $\Lambda := \mathrm{lcm}(2, \ldots, \mathrm{A}_{\max})$ and $\mathrm{A}_{\max} := \left\lceil 2^{-(t_l-2)} \cdot (t_l-1)^{\frac{t_l-1}{2}} \cdot (t_l-1)! \cdot n_l^{\frac{(t_l-2)(t_l-1)}{2}} \right\rceil$. The value of $\mathrm{A}_{\max}$ is an upper bound for the largest possible determinant $\mathrm{A}_{\max} := \max_{E,X,\mathcal{G}} \det(\mathrm{A}(E,X,\mathcal{G}))$ that can occur (cf. Theorem 3.3 in Section 3.3).

Verifiability is achieved with Pedersen commitments [33]. Our newly developed **Share** algorithm of the actual scheme is described in Protocol 4.1. We will focus on a conjunctive access structure. The version for a disjunctive access structure is given in brackets.

Protocol 4.1 (Hierarchical Secret Sharing over the integers). *Let $k \in \mathbb{N}$ be a security parameter and let $K : \mathbb{N} \to \mathbb{N}$ be a function s. t. $1/\mathrm{K}(\cdot)$ is negligible. Let $z \in \mathbb{N}$ and let the secret be $m \in [-z, z] \subset \mathbb{Z}$. Let $\mathcal{S}$ be a set of shareholders composed of $l+1$ pairwise disjoint levels. Let $n_{-1} := 0$ and let $n_h \in \mathbb{N}$ be such that $n_h \geq n_{h-1} + |\mathcal{S}_h|$ for $h \in \{0, \ldots, l\}$. Let*

*$\Gamma \subseteq \mathcal{P}(\mathcal{S})$ be a conjunctive (disjunctive) $(\vec{t}, n_l)$ hierarchical threshold access structure with $\vec{t} := (t_0, \ldots, t_l) \in \mathbb{N}^{l+1}$ and let $t_{-1} := 0$. Let (i,j) denote the unique ID of shareholder $s_{i,j} \in \mathcal{S}_h$ for $h \in \{0, \ldots, l\}$, where $i \in \{n_{h-1}+1, \ldots, n_h\}$ and $j := t_{h-1}$ $(j := t_l - t_h)$. Let $\Lambda := \mathrm{lcm}(2, 3, \ldots, \mathrm{A}_{\max})$, where $\mathrm{A}_{\max} := \left\lceil 2^{-(t_l-2)} \cdot (t_l-1)^{\frac{t_l-1}{2}} \cdot (t_l-1)! \cdot n_l^{\frac{(t_l-2)(t_l-1)}{2}} \right\rceil$. The **Share** algorithm proceeds as follows:*

1. *Set $a_0 := m\Lambda$ $(a_{t_l-1} := m\Lambda)$.*
2. *Select $a_1, \ldots, a_{t_l-1} \leftarrow_{\$} [-\mathrm{K}(k)z\Lambda^2, \mathrm{K}(k)z\Lambda^2]$ $(a_0, \ldots, a_{t_l-2} \leftarrow_{\$} [-\mathrm{K}(k)z\Lambda^2, \mathrm{K}(k)z\Lambda^2])$.*
3. *Construct $f(x) := \sum_{\omega=0}^{t_l-1} a_\omega x^\omega$.*
4. *Distribute the share $\sigma_{i,j} := f^{(j)}(i)$ to shareholder $s_{i,j}$, for all $s_{i,j} \in \mathcal{S}$.*

*The **Verification** is performed as follows: Let P and P' be primes such that $P' \mid (P-1)$ and $P' > 2\left(2z\Lambda + 2(t_l-1)n_l^{t_l-1}\mathrm{K}(k)z\Lambda^2\right)$. Let $g, h \in \mathbb{Z}_P$ such that the order of g, h is P' and $\log_g(h)$ is unknown.*

1. *Select $m' \leftarrow_{\$} \mathbb{Z}_{P'}$, set $a'_0 := m'\Lambda$ $(a'_{t_l-1} := m'\Lambda)$ and select coefficients $a'_1, \ldots, a'_{t_l-1} \leftarrow_{\$} \mathbb{Z}_{P'}$ $(a'_0, \ldots, a'_{t_l-2} \leftarrow_{\$} \mathbb{Z}_{P'})$.*
2. *Construct $f'(x) := \sum_{\omega=0}^{t_l-1} a'_\omega x^\omega \mod P'$.*
3. *Distribute the share $\sigma'_{i,j} := f'^{(j)}(i)$ to shareholder $s_{i,j}$, for all $s_{i,j} \in \mathcal{S}$.*
4. *The dealer publishes $v_\omega := g^{a_\omega} h^{a'_\omega} \mod P$ for $\omega \in \{0, \ldots, t_l - 1\}$.*
5. *The dealer publishes zero-knowledge range proofs (Protocol 3.6) that $a_\omega \in [-\mathrm{K}(k)z\Lambda^2, \mathrm{K}(k)z\Lambda^2] \subset \mathbb{Z}$ for $\omega \in \{1, \ldots, t_l - 1\}$ and that $m \in [-z, z] \subset \mathbb{Z}$ (they first compute $g^m h^{m'} \mod P$, prove the range, and then compute $v_0 = \left(g^m h^{m'}\right)^\Lambda \mod P$).*

Each shareholder verifies their share as follows:

1. *Shareholder $s_{i,j}$ checks if $g^{\sigma_{i,j}} h^{\sigma'_{i,j}} \equiv \prod_{\omega=j}^{t_l-1} (v_\omega)^{\frac{\omega!}{(\omega-j)!} i^{\omega-j}} \mod P$. The dealer is disqualified if more than $t_l - 1$ checks fail.*
2. *Shareholder $s_{i,j}$ checks if $|\sigma_{i,j}| \leq \left(z\Lambda + (t_l-1)n_l^{t_l-1}\mathrm{K}(k)z\Lambda^2\right)$, otherwise publishes $\sigma_{i,j}$ and the dealer is disqualified.*

The **Reconstruct**, **Add** and **Reset** algorithm are analogous to Protocols 3.1, 3.3 and 3.4 in Sections 3.3 and 3.4 and thus not detailed again. Note that during the **Add** or **Reset** algorithm the shareholders obtain rational numbers, as they need to divide by $\mathrm{A} := \det(\mathrm{A}(E, X, \mathcal{G}))$. This is not a problem, as it is easy to see that the final shares are guaranteed to be integers (the shared secret is an integer). In addition, the relevant computations for verification are performed modulo $P' > \mathrm{A}_{\max}$ and thus the inverse of A exists.

PROPOSITION 4.1. *Let a semi-honest/malicious adversary corrupt one unauthorized set $\mathcal{U} \subset \mathcal{S}$ of shareholders, i. e., $\mathcal{U} \notin \Gamma$. The view of the adversary of the secret shares generated by Protocol 4.1 is statistically independent*

of the sharing of a random secret by an appropriate polynomial with coefficients taken from the same range.

PROOF. W.l.o.g. assume the adversary corrupts a maximal unauthorized set of shareholders $\mathcal{U}$ such that only one share is missing to be able to reconstruct the secret, i. e., there is a shareholder $s \in \mathcal{S}$ such that $\mathcal{U} \cup \{s\} \in \Gamma$. This assumption is made to cover the conjunctive and the disjunctive case. We present the proof for the conjunctive version and give the disjunctive version in brackets.

We show that for any $m' \in [-z, z]$ there is a polynomial $h(x)$ over the integers that shares the secret $m'\Lambda$ and $f^{(j)}(i) = h^{(j)}(i)$ for all shareholders $s_{i,j} \in \mathcal{U}$. This can be seen by first constructing a polynomial $h'(x)$ that shares the secret $(m - m')\Lambda$ and $\sigma'_{i,j} := h'^{(j)}(i) = 0$ for all the shareholders $s_{i,j} \in \mathcal{U}$. We define $\mathcal{U}' := \mathcal{U} \cup \{s_{0,0}\}$ ($\mathcal{U}' := \mathcal{U} \cup \{s_{0,t_l-1}\}$) and the corresponding share as $\sigma'_{0,0} := h'(0) = (m - m')\Lambda$ ($\sigma'_{0,t_l-1} := h'^{(t_l-1)}(0) = (m - m')\Lambda \cdot (t_l - 1)!$). It is $\mathcal{U}' \in \Gamma$. Let E, X and $\mathcal{G}$ be defined according to Protocol 3.2. Let $\mathrm{A} := \det(\mathrm{A}(E, X, \mathcal{G}))$ and $\mathrm{A}_{x,y} := \det(\mathrm{A}_{x,y}(E, X, \mathcal{G}))$. For $h'(x)$ it holds

$$\begin{aligned} h'(x) &= \sum_{\omega=0}^{|\mathcal{U}'|-1} \frac{\det(\mathrm{A}(E, X, \mathcal{G}_\omega))}{\det(\mathrm{A}(E, X, \mathcal{G}))} x^\omega \\ &= \sum_{\omega=0}^{|\mathcal{U}'|-1} \left(\sum_{u=0}^{|\mathcal{U}'|-1} \sigma'_{i_u,j_u} (-1)^{u+\omega} \cdot \frac{\mathrm{A}_{u,\omega}}{\mathrm{A}} x^\omega \right). \end{aligned}$$

Furthermore, $\sigma'_{i,j} = 0$ for all shareholders $s_{i,j} \in \mathcal{U}$ and

$$h'(x) = \sum_{\omega=0}^{|\mathcal{U}'|-1} \sigma'_{i_0,j_0} (-1)^\omega \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}} x^\omega.$$

For the conjunctive case it holds that

$$\begin{aligned} h'(x) &= \sum_{\omega=0}^{|\mathcal{U}'|-1} \sigma'_{0,0} (-1)^\omega \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}} x^\omega \\ &= \sum_{\omega=0}^{|\mathcal{U}'|-1} (m - m')\Lambda (-1)^\omega \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}} x^\omega \end{aligned}$$

and for the disjunctive case it holds that

$$\begin{aligned} h'(x) &= \sum_{\omega=0}^{|\mathcal{U}'|-1} \sigma'_{0,t_l-1}(-1)^{\omega} \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}} x^{\omega} \\ &= \sum_{\omega=0}^{|\mathcal{U}'|-1} (m-m')\Lambda(t_l-1)!(-1)^{\omega} \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}} x^{\omega}. \end{aligned}$$

As Λ is chosen as the least common multiple of all numbers between 2 and the largest possible determinant of $\mathrm{A}(E, X, \mathcal{G})$ that can occur, it holds that $\det(\mathrm{A}(E, X, \mathcal{G})) \mid \Lambda$. Hence, $h'(x)$ is a polynomial over the integers. Define $h(x) := f(x) - h'(x)$. Since $f(x) \neq h'(x)$ it holds that $h(0) = m\Lambda - (m - m')\Lambda = m'\Lambda$ and $h^{(j)}(i) = f^{(j)}(i)$ for all $s_{i,j} \in \mathcal{U}$.

Let a_ω, a'_ω and b_ω for $\omega \in \{0, \ldots, t_l - 1\}$ denote the coefficients of $f(x)$, $h'(x)$ and $h(x)$, respectively. The coefficients a'_ω are bounded absolutely by

$$\begin{aligned} |a'_\omega| &\leq \left|(m-m')\Lambda(-1)^{\omega} \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}}\right| = \left|(m-m')\Lambda \cdot \frac{\mathrm{A}_{0,\omega}}{\mathrm{A}}\right| \\ &\leq |(m-m')\Lambda \cdot \mathrm{A}_{0,\omega}| \leq |(m-m')\Lambda \cdot \Lambda| = \left|(m-m')\Lambda^2\right| \\ &\leq \left|2z\Lambda^2\right| \end{aligned}$$

.

As $|a_\omega| \leq \mathrm{K}(k)z\Lambda^2$ the coefficients b_ω are bounded by

$$|a_\omega - a'_\omega| \leq |a_\omega| + |a'_\omega| \leq \mathrm{K}(k)z\Lambda^2 + 2z\Lambda^2 \leq (\mathrm{K}(k)+2)z\Lambda^2.$$

Hence, the probability that any $b_\omega \notin [-\mathrm{K}(k)z\Lambda^2, \mathrm{K}(k)z\Lambda^2]$ is $t_l \cdot \frac{2 \cdot 2z\Lambda^2}{2(\mathrm{K}(k)+2)z\Lambda^2} = \frac{2t_l}{\mathrm{K}(k)+2}$. Thus, with high probability the coefficients are in the correct range. As $1/\mathrm{K}(k)$ is negligible in k, the view of the adversary is statistically independent of the sharing of a random secret. □

5. Hierarchical and Dynamic Threshold Paillier Cryptosystem without Trusted Dealer

In this section, we devise the first hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer.[2] Our new scheme uses the techniques of the threshold Paillier cryptosystem without trusted dealer [29]. We reuse the key generation of the threshold Paillier cryptosystem without trusted dealer (see Protocol 3.10) and obtain a private key shared with the verifiable secret sharing scheme over the integers (see Protocol 3.9). Next, we use the

[2]We first developed a hierarchical threshold Paillier cryptosystem with trusted dealer that is secure in the semi-honest adversary model. However, as it is a rather straight forward modification of the threshold Paillier cryptosystem in [16], we include it only for completeness in Appendix A.

Reset algorithm of the hierarchical secret sharing scheme and obtain a hierarchical sharing of the private key:[3] The verifiable secret sharing scheme over the integer is a special case of our hierarchical secret sharing scheme over the integers (see Protocol 4.1). Thus, the **Reset** algorithm allows changing a (t, n) threshold access structure into a $(\vec{t}, n_l)$ hierarchical access structure. However, the decryption needs to be adapted to cover the hierarchical structure.

In the overall setup one needs to be careful when choosing the threshold t and $\vec{t}$ for the corresponding access structure before and after the **Reset** algorithm. In order to prevent unauthorized access to the private key, we need to assume that the adversary can corrupt only an unauthorized set of shareholders and that during the key generation the adversary is a static adversary. If the threshold t of Protocol 3.10 and $\vec{t} := (t_0, \ldots, t_l)$ is chosen such that $t \leq t_l$ ($t \leq t_0$), then any corrupted set of $t-1$ shareholders is before and after the **Reset** algorithm unable to retrieve the private key. After the key generation is finished, the system can cope with mobile adversaries if the **Reset** algorithm is executed in regular intervals.

This new hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer allows to dynamically add and remove shareholders while providing a hierarchical access structure. The new cryptosystem is robust and threshold semantic secure in the malicious adversary model. We will focus on a conjunctive access structure. The version for a disjunctive access structure is given in brackets. The **Key Generation** algorithm is described in Protocol 5.1. The **Encryption** algorithm is the same as for the original Paillier cryptosystem (Protocol 3.8) with $g := 1 + N$. The **Decryption** algorithm is described in Protocol 5.2.

Protocol 5.1 (Hierarchical and Dynamic Threshold Paillier Cryptosystem without Trusted Dealer - **Key Generation**). *Let $\mathcal{S}$ be a set of shareholders composed of $l+1$ pairwise disjoint levels. Let $n_{-1} := 0$ and let $n_h \in \mathbb{N}$ be such that $n_h \geq n_{h-1} + |\mathcal{S}_h|$ for $h \in \{0, \ldots, l\}$. Let $\Gamma \subseteq \mathcal{P}(\mathcal{S})$ be a conjunctive (disjunctive) $(\vec{t}, n_l)$ hierarchical threshold access structure with $\vec{t} := (t_0, \ldots, t_l) \in \mathbb{N}^{l+1}$ and let $t_{-1} := 0$. Let (i, j) denote the unique ID of shareholder $s_{i,j} \in \mathcal{S}_h$ for $h \in \{0, \ldots, l\}$, where $i \in \{n_{h-1}+1, \ldots, n_h\}$ and $j := t_{h-1}$ ($j := t_l - t_h$). Let $\Lambda := \operatorname{lcm}(2, 3, \ldots, \mathrm{A}_{\max})$, where $\mathrm{A}_{\max} := \left\lceil 2^{-(t_l-2)} \cdot (t_l-1)^{\frac{t_l-1}{2}} \cdot (t_l-1)! \cdot n_l^{\frac{(t_l-2)(t_l-1)}{2}} \right\rceil$. Let $\tilde{k} \in \mathbb{N}$ be a security parameter, i. e., the required bit size for the generated prime candidates. In*

[3] It is also possible to exchange Shamir's secret sharing scheme and the verifiable secret sharing scheme over the integers in Protocol 3.10 with the hierarchical secret sharing schemes in Sections 3.3 and 4: The primes are shared with the hierarchical secret sharing scheme and the multiplication is performed analogously as described in [40]. We decided against this approach, as the multiplication in the hierarchical secret sharing scheme is more expensive and does not provide additional security.

order to create a modulus N and a private key in a distributed fashion Protocol 3.10 (Section 3.6) is executed with the following modifications:

1. *Select a security parameter $k \geq \tilde{k}$ and $k > \log_2(\mathrm{A}_{\max})$ to assure that $p > \mathrm{A}_{\max}$.*
2. *Choose the threshold t of Protocol 3.10 and $\vec{t} := (t_0, \ldots, t_l)$ such that $t \leq t_l$ ($t \leq t_0$).*
3. *In Step 3. when Protocol 3.9 is executed, we use $\Delta := \Lambda$ (instead of $\Delta := n!$).*
4. *At the end of Step 4. of Protocol 3.10 execute the **Reset** algorithm (Theorem 3.5) such that the new access structure after the execution is Γ. The private key $\sigma := -\Lambda\varphi(N)\beta$ is now shared with the hierarchical secret sharing scheme over the integers (Protocol 4.1) with secret shares $\sigma_{i,j}$ for $s_{i,j} \in \mathcal{S}$.*
5. *In Step 5. of Protocol 3.10 the shareholders set $v := H(N)^2 \bmod N^2$ (H is a hash function modeled as random oracle). Each shareholder $s_{i,j}$ publishes a verification key $v_{i,j} := v^{\Lambda\sigma_{i,j}} \bmod N^2$ and proves that $\log_{v^\Lambda}(v_{i,j}) = \sigma_{i,j}$ (Protocol 3.7).*

The public key is (N, θ) and the private key is σ. Note that $\theta \equiv \Lambda\varphi(N)\beta + N\Lambda r \equiv \Lambda\varphi(N)\beta \bmod N$.

Protocol 5.2 (Hierarchical and Dynamic Threshold Paillier Cryptosystem without Trusted Dealer - **Decryption**). *Let c be the ciphertext to be decrypted. Let $\mathcal{R} \subseteq \mathcal{S}$ denote the shareholders who participate in the decryption process. The shareholders proceed as follows:*

1. *Execute Protocol 3.2 (**Birkhoff Setup**).*
2. *Let $s_u := s_{i_u,j_u}$ and $\sigma_u := \sigma_{i_u,j_u}$ for $(i_u, j_u) \in \mathrm{I}(E)$.*
3. *Compute $c_u := c^{2\mathrm{A}\sigma_u} \bmod N^2$.*
4. *Prove that $\log_{c^{4\mathrm{A}}}\left(c_u^2\right) = \log_v(v_u)$ (Protocol 3.5).*
5. *Send c_u and its proof to the combiner.*

The combiner performs the following steps:

1. *Let $\mathcal{R}' \subseteq \mathcal{R}$ be the set of shareholders with correct proofs. If $\mathcal{R} \neq \mathcal{R}'$, then restart the protocol with $\mathcal{R} := \mathcal{R}'$. If $\mathcal{R}' \notin \Gamma$, then the decryption fails.*
2. *Otherwise compute*

$$m := \mathrm{L}\left(\prod_{s_u \in \mathcal{R}'} c_u^{2\psi_u} \bmod N^2\right) \frac{1}{-4\mathrm{A}^2\theta} \bmod N$$

where

$$\psi_u := (-1)^{u+b-d} \cdot \det(\mathrm{A}_{u,b-d}(E, X, \mathcal{G})).$$

Proposition 5.1 (Correctness). *The combiner obtains a correct decryption of a ciphertext with the execution of Protocol 5.2 if it holds that $\mathcal{R} = \mathcal{R}'$ and $\mathcal{R}' \in \Gamma$.*

PROOF. We prove that the decryption is correct. Therefore, we first prove that the private key is shared correctly, i. e., the private key can be reconstructed. Second, we show that the proposed decryption will yield the correct plaintext.

Let $s_u := s_{i_u,j_u} \in \mathcal{R}$ and let $\sigma_u := \sigma_{i_u,j_u}$ for $(i_u, j_u) \in \mathrm{I}(E)$. Let $\mathrm{A} := \det(\mathrm{A}(E, X, \mathcal{G}))$ and $\mathrm{A}_{x,y} := \det(\mathrm{A}_{x,y}(E, X, \mathcal{G}))$. It holds that $\mathcal{R} = \mathcal{R}'$ and $\mathcal{R}' \in \Gamma$. This means that the partial decryptions are correct and that $\mathcal{R}$ is an authorized set of shareholders.

Using the shares of the shareholders, the private key can be reconstructed, as it holds that

$$\begin{aligned}\sum_{s_u\in\mathcal{R}} \psi_u\sigma_u &= \sum_{s_u\in\mathcal{R}} \sigma_u \cdot (-1)^{u+b-d} \cdot \mathrm{A}_{u,b-d} \\ &= \mathrm{A} \sum_{s_u\in\mathcal{R}} \sigma_u \cdot (-1)^{u+b-d} \frac{\mathrm{A}_{u,b-d}}{\mathrm{A}} \\ &= \mathrm{A}a_b = -\mathrm{A}\Lambda\varphi(N)\beta.\end{aligned}$$

Hence, for the decryption of a ciphertext $c := (1+N)^m r^N \bmod N^2$ we get

$$\begin{aligned}\prod_{s_u\in\mathcal{R}} c_u^{2\psi_u} &\equiv \prod_{s_u\in\mathcal{R}} \left(c^{2\mathrm{A}\sigma_u}\right)^{2\psi_u} \equiv c^{4\mathrm{A}\sum_{s_u\in\mathcal{R}}\sigma_u\psi_u} \\ &\equiv c^{-4\mathrm{A}^2\Lambda\varphi(N)\beta} \equiv \left((1+N)^m r^N\right)^{-4\mathrm{A}^2\Lambda\varphi(N)\beta} \\ &\equiv (1+N)^{-4\mathrm{A}^2\Lambda\varphi(N)\beta m} \cdot \underbrace{\left(r^{\varphi(N)}\right)}_{r^{\varphi(N)}\equiv 1 \bmod N}^{-4\mathrm{A}^2\Lambda\beta N} \\ &\equiv (1+N)^{-4\mathrm{A}^2\Lambda\varphi(N)\beta m} \bmod N^2.\end{aligned}$$

Thus, we have

$$\begin{aligned}&\mathrm{L}\left(\prod_{s_u\in\mathcal{R}'} c_u^{2\psi_u} \bmod N^2\right) \cdot \frac{1}{-4\mathrm{A}^2\theta} \\ \equiv\ &\mathrm{L}\left((1+N)^{-4\mathrm{A}^2\Lambda\varphi(N)\beta m} \bmod N^2\right) \cdot \frac{1}{-4\mathrm{A}^2\theta} \\ \equiv\ &-4\mathrm{A}^2 \underbrace{\Lambda\varphi(N)\beta}_{\equiv\theta \bmod N} m \cdot \frac{1}{-4\mathrm{A}^2\theta} \equiv m \bmod N.\end{aligned}$$

□

PROPOSITION 5.2. *The key generation of the hierarchical and dynamic Paillier cryptosystem in Protocol 5.1 is robust and threshold semantic secure in the presence of a malicious adversary corrupting at most $t-1$ shareholders.*

PROOF. As shown in [29], the key generation of the threshold Paillier cryptosystem without trusted dealer (Protocol 3.10) is robust and threshold

semantic secure. Thus, we need to show that the modifications of Protocol 3.10 as described in Protocol 5.1 do not affect robustness or threshold semantic security.

Modification 1. has no influence at all, as it is just a lower bound for the needed bit security.

Modification 2. is only a condition on the threshold t which ensures that unauthorized sets during the key generation with Protocol 3.10 are also unauthorized according to the hierarchical access structure Γ after the **Reset** algorithm in Step 4. of Protocol 5.1. Hence, it does no influence robustness or threshold semantic security.

Modification 3.: The secret sharing scheme over the integers [29] (Protocol 3.9) is a special case of the hierarchical secret sharing scheme over the integers (Protocol 4.1), namely if only one level of shareholders exists. Both schemes provide the same security (cf. [29] and Theorem 4.1).

Modification 4.: The **Reset** algorithm does not alter security as shown in [39].

Modification 5. does not affect robustness, as the zero-knowledge proof is basically identical to the original protocol: The verification keys v_i and $v_{i,j}$ are both Pedersen commitments. The hierarchical secret sharing scheme over the integers guarantees that the new verification key $v_{i,j}$ corresponds to the new share $\sigma_{i,j}$. As Shamir's secret sharing scheme is a special case of the hierarchical secret sharing scheme, it follows that if the zero-knowledge proof in Protocol 5.1 leaks information, then it leaks information in Protocol 3.10, too. Hence, computing the zero-knowledge proof does not affect robustness or threshold semantic security.

None of the modifications influence robustness or threshold semantic security of the key generation of the threshold Paillier cryptosystem without trusted dealer. Hence, the key generation of the hierarchical and dynamic Paillier cryptosystem is robust and threshold semantic secure. □

PROPOSITION 5.3. *The hierarchical and dynamic threshold Paillier cryptosystem as described in Protocols 5.1 and 5.2 is robust in the presence of a malicious adversary corrupting at most $t-1$ shareholders.*

PROOF. Theorem 5.1 guarantees a correct decryption and thus robustness if the combiner is able to distinguish between corrupted shareholders and honest shareholders, and the combiner finally has received only correct partial decryptions.

Let $\mathcal{R} \subseteq \mathcal{S}$ be the set of shareholders participating in the decryption. Let $\mathcal{U} \subset \mathcal{R}$ be a set of unauthorized shareholders ($\mathcal{U} \notin \Gamma$) chosen and controlled by a malicious adversary such that $\mathcal{R} \setminus \mathcal{U} \in \Gamma$. The partial decryptions c_u are either correct or incorrect. The combiner can distinguish between them by verifying the corresponding proof and exclude the corresponding shareholders from the decryption process. With each round the amount of

corrupted shareholders providing wrong partial decryptions decreases, i. e., $\mathcal{R}' \subseteq \mathcal{R}$. As $\mathcal{R} \setminus \mathcal{U} \in \Gamma$ it has to hold that at some point in time it is either that only honest shareholders are left or that all partial decryptions are correct, i. e., it holds that $\mathcal{R} = \mathcal{R}'$. Hence, the combiner can correctly decrypt any ciphertext and the protocol is robust. □

THEOREM 5.4. *The hierarchical and dynamic threshold Paillier cryptosystem as described in Protocols 5.1 and 5.2 is threshold semantically secure in the random oracle model with a malicious adversary corrupting at most $t-1$ shareholders.*

PROOF. We prove that the hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer is threshold semantic secure by showing that if there exists an adversary that can break threshold semantic security, then we can construct a simulator that uses the power of the adversary to break the semantic security of the original Paillier cryptosystem.

The key generation of the new hierarchical and dynamic threshold Paillier cryptosystem (Protocol 5.1) is robust and threshold semantic secure as proven in Theorem 5.2. Theorem 5.3 guarantees robustness for the **Decryption** algorithm. Thus, it is left to show that the decryption is threshold semantic secure. We start with the generation of the verification keys in Protocol 5.1. We prove the conjunctive case and give changes for the disjunctive case in brackets.

Let $\mathcal{S}$ be the set of shareholders and let $\mathcal{R} \subseteq \mathcal{S}$ be the set of shareholders participating in the decryption. Let $\mathcal{U} \subset \mathcal{R}$ be a set of unauthorized shareholders ($\mathcal{U} \notin \Gamma$) chosen by the adversary. Let $\mathcal{U}' := \mathcal{U} \cup \{s_{0,0}\}$ ($\mathcal{U}' := \mathcal{U} \cup \{s_{0,t_l-1}\}$) and w.l.o.g. assume that the adversary has chosen the corrupted parties such that $\mathcal{U}' \in \Gamma$, where $s_{0,0}$ (s_{0,t_l-1}) is a non existing shareholder with corresponding share $\sigma_{0,0} := -\Lambda\varphi(N)\beta$ ($\sigma_{0,t_l-1} := -\Lambda\varphi(N)\beta \cdot (t_l-1)!$). Note that $\sigma_{0,0}$ (σ_{0,t_l-1}) is unknown to the simulator.

For verification v is generated as $H(N)^2 \bmod N^2$ with a random oracle H where $H(N) := (1+N)^{m_v} r_v^N \bmod N^2$ and $m_v, r_v \in \mathbb{Z}_N$ are chosen by the simulator. Thus, we have $v \equiv (1+N)^{2m_v} r_v^{2N} \bmod N^2$. During the key generation in Protocol 5.1 the simulator has learned the values $\sigma_{i,j}$ for $s_{i,j} \in \mathcal{U}$ as shown in [29].

The simulator has to create the verification keys for all shareholders in $\mathcal{R}$: The simulator can easily compute the verification keys $v_{i,j} := v^{\Lambda\sigma_{i,j}}$ for $s_{i,j} \in \mathcal{U}$ from the knowledge of $\sigma_{i,j}$. It holds that

$$\begin{aligned} v^{-\Lambda\varphi(N)\beta} &\equiv \left((1+N)^{2m_v} r_v^{2N}\right)^{-\Lambda\varphi(N)\beta} \\ &\equiv (1+N)^{-2m_v\Lambda\varphi(N)\beta} \\ &\equiv 1 - 2m_v\Lambda\varphi(N)\beta N \bmod N^2. \end{aligned}$$

As $\theta = \Lambda\varphi(N)\beta + N\Lambda r$ it holds that

$$\begin{aligned} 1 - 2m_v\theta N &\equiv 1 - 2m_v(\Lambda\varphi(N)\beta + N\Lambda r)N \\ &\equiv 1 - 2m_v\Lambda\varphi(N)\beta N - 2m_v\Lambda r N^2 \\ &\equiv 1 - 2m_v\Lambda\varphi(N)\beta N \bmod N^2. \end{aligned}$$

Hence, it is $v^{-\Lambda\varphi(N)\beta} \equiv 1 - 2m_v\theta N \bmod N^2$.

As θ is public, the simulator can compute a correct verification key for $s_{0,0}$ (s_{0,t_l-1}) by setting $v_{0,0} := 1 - 2m_v\Lambda\theta N \bmod N^2$ ($v_{0,t_l-1} := 1 - 2m_v\Lambda\theta N(t_l - 1)! \bmod N^2$). Let E, X and $\mathcal{G}$ be defined according to Protocol 3.2. Now, the simulator can compute the verification shares of the honest parties by using Birkhoff interpolation in the exponent: Let $\mathrm{A}_{x,y} := \det(\mathrm{A}_{x,y}(E, X, \mathcal{G}))$. The shareholders compute

$$f_u^{(j')}(i') := \sigma_{i_u,j_u} \sum_{\omega=j'}^{t_l-1} \frac{\omega!}{(\omega - j')!}(-1)^{u+\omega} \cdot \frac{\mathrm{A}_{u,\omega-d}}{\mathrm{A}} i'^{\omega-j'}$$

and then $\sum_{u=0}^{k-1} f_u^{(j')}(i') = \sigma_{i',j'}$ (cf. Equation (3.1)). Let $\mathcal{U}' = \{s_{i_0,j_0}, \ldots, s_{i_{k-1},j_{k-1}}\}$ be lexicographically ordered by their unique index. Observe that $s_{i_0,j_0} = s_{0,0}$ ($s_{i_0,j_0} = s_{0,t_l-1}$) and that $\mathcal{U} = \{s_{i_u,j_u} \in \mathcal{U}' : u \geq 1\}$. Then, it holds that

$$\begin{aligned} v_{i',j'} &\equiv v^{\Lambda\sigma_{i',j'}} \equiv v^{\Lambda\sum_{u=0}^{k-1} f_u^{(j')}(i')} \equiv \prod_{u=0}^{k-1} v^{\Lambda f_u^{(j')}(i')} \\ &\equiv v^{\Lambda f_0^{(j')}(i')} \cdot \prod_{u=1}^{k-1} v^{\Lambda f_u^{(j')}(i')} \bmod N^2. \end{aligned}$$

If the simulator can compute $v^{\Lambda f_0^{(j')}(i')}$ and $\prod_{u=1}^{k-1} v^{\Lambda f_u^{(j')}(i')}$, then they can compute a verification share for every honest shareholder. The simulator can easily compute $\prod_{u=1}^{k-1} v^{\Lambda f_u^{(j')}(i')}$ as they are in possession of σ_{i_u,j_u}. Next, we show how to compute $v^{\Lambda f_0^{(j')}(i')}$. The simulator can compute $v^{\Lambda f_0^{(j')}(i')}$ as it holds that

$$\begin{aligned} v^{\Lambda f_0^{(j')}(i')} &\equiv v^{\Lambda\sigma_{i_0,j_0}\sum_{\omega=j'}^{t_l-1} \frac{\omega!}{(\omega-j')!}(-1)^{\omega}\frac{\mathrm{A}_{0,\omega-d}}{\mathrm{A}} i'^{\omega-j'}} \\ &\equiv v^{\frac{\Lambda}{\mathrm{A}}\sigma_{i_0,j_0}\sum_{\omega=j'}^{t_l-1} \frac{\omega!}{(\omega-j')!}(-1)^{\omega}\mathrm{A}_{0,\omega-d}\cdot i'^{\omega-j'}} \\ &\equiv v^{(-\Lambda\varphi(N)\beta)\frac{\Lambda}{\mathrm{A}}\sum_{\omega=j'}^{t_l-1} \frac{\omega!}{(\omega-j')!}(-1)^{\omega}\mathrm{A}_{0,\omega-d}\cdot i'^{\omega-j'}} \\ &\equiv \left(v^{-\Lambda\varphi(N)\beta}\right)^{\frac{\Lambda}{\mathrm{A}}\sum_{\omega=j'}^{t_l-1} \frac{\omega!}{(\omega-j')!}(-1)^{\omega}\mathrm{A}_{0,\omega-d}\cdot i'^{\omega-j'}} \\ &\equiv (1 - 2m_v\theta N)^{\frac{\Lambda}{\mathrm{A}}\sum_{\omega=j'}^{t_l-1}\frac{\omega!}{(\omega-j')!}(-1)^{\omega}\mathrm{A}_{0,\omega-d}\cdot i'^{\omega-j'}} \bmod N^2. \end{aligned}$$

The disjunctive case is done analogously by exchanging $1 - 2m_v \Lambda\theta N$ with $1 - 2m_v \Lambda\theta N(t_l - 1)!$. Consequently, the simulator can for all honest shareholders compute correct verification key.

In the disjunctive case, the simulator cannot compute the verification key for every shareholder in general, i. e., the simulator cannot compute the verification key for a lower level shareholder than the lowest corrupted shareholder, as higher level shares contain no information about the lower coefficients. However, the simulator can avoid this problem by selecting an appropriate access structure.

If the adversary sends a message m to the partial decryption oracle, the simulator has to respond with an encryption c, the partial decryptions and corresponding proofs. The encryption is just selecting a random $r \leftarrow_\$ \mathbb{Z}_N$ and then $c := (1 + N)^m r^N \bmod N^2$. The partial decryptions are created analogously to the verification keys. However, as the set of shareholders participating in the decryption protocol is known, we do not have to use Λ to avoid calculating the inverse in the exponent. Instead, we can just use A, which is covered by the definition of ψ_u. The simulator can easily generate the corresponding zero-knowledge proofs, as the simulator controls the random oracle H.

If there is an adversary that can break the threshold semantic security of the hierarchical and dynamic threshold Paillier cryptosystem, then the simulator can use the adversary's power to break the semantic security of the original Paillier cryptosystem as shown above. This is a contradiction to the original Paillier cryptosystem being semantically secure [31]. This implies threshold semantic security of the hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer. □

5.1. *Computation and Communication Complexity.* The key generation of the threshold Paillier cryptosystem [29] and our novel system are both probabilistic, as they first generate two prime candidates, check their primality, and if the check fails restart, e. g., x times. The total complexity w. r. t. computation (number of modular exponentiations) and communication (number of messages exchanged) of the key generations of both systems is $\mathcal{O}\left(xn^2\right)$ where n is the number of parties participating.

For security parameter k (the bit size of the primes) the share size in bits is bound by $4k+6+\log_2(t)+(t+7)\log_2(n)+2\log_2(\Delta)+3\log_2(\mathrm{K}(k))$, with $\Delta := n!$ for the threshold Paillier cryptosystem [29], and $\Delta := \mathrm{lcm}(2, 3, \ldots, \mathrm{A}_{\max})$ for our system where additionally k has to be chosen such that $k > \log_2(\mathrm{A}_{\max})$.

A malicious adversary controlling $t-1$ parties can prevent the decryption at most $t - 1$ times, as in each round at least one malicious party is detected and excluded from participation. Therefore, the decryption consists of at most t rounds. In each round a shareholder computes a partial decryption

and corresponding proof of correctness which have computation and communication complexity $\mathcal{O}(1)$ with a message size in the order of the share size. The combiner has to check at most $\frac{t}{2}(2n - t + 1)$ proofs. If all proofs are correct, then the plaintext can be restored with computational complexity $\mathcal{O}(n)$.

6. Conclusion

In this paper, we introduced a hierarchical and dynamic threshold Paillier cryptosystems without a trusted dealer. As a building block we also introduced a verifiable hierarchical and dynamic secret sharing scheme over the integers. Our new hierarchical and dynamic Paillier cryptosystems without trusted dealer allows using hierarchical access structures while being dynamic at the same time. This allows dynamically adding or removing shareholders.

In the future, we plan to explore the applicability of our novel cryptosystem in the context of secure multi-party computations, as our system allows reusing computations for different party constellations, even if shareholders join or leave during the protocol execution.

Acknowledgements.

This work is supported by the German research council (DFG) Research Training Group 2236 UnRAVeL.

References

[1] S. G. Akl and P. D. Taylor. Cryptographic Solution to a Problem of Access Control in a Hierarchy. *ACM Trans. Comput. Syst.*, 1(3):239–248, 1983.

[2] A. Beimel, T. Tassa, and E. Weinreb. Characterizing Ideal Weighted Threshold Secret Sharing. In *Theory of Cryptography*, Lecture Notes in Computer Science, pages 600–619. Springer Berlin Heidelberg, 2005.

[3] J. Benaloh and J. Leichter. Generalized Secret Sharing and Monotone Functions. In *Advances in Cryptology — CRYPTO' 88*, Lecture Notes in Computer Science, pages 27–35. Springer New York, 1990.

[4] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the national computer conference*, pages 313–317, 1979.

[5] P. Bogetoft, D. L. Christensen, I. Damgård, M. Geisler, T. Jakobsen, M. Krøigaard, J. D. Nielsen, J. B. Nielsen, K. Nielsen, J. Pagter, M. Schwartzbach, and T. Toft. Secure Multiparty Computation Goes Live. In *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 325–343. Springer Berlin Heidelberg, 2009.

[6] D. Boneh and M. Franklin. Efficient Generation of Shared RSA Keys. In *Annual International Cryptology Conference*, Lecture Notes in Computer Science, pages 425–439. Springer Berlin Heidelberg, 1997.

[7] E. F. Brickell. Some Ideal Secret Sharing Schemes. In *Advances in Cryptology — EUROCRYPT '89*, Lecture Notes in Computer Science, pages 468–475. Springer Berlin Heidelberg, 1990.

[8] C. Clifton, M. Kantarcioğlu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, and D. Suciu. Privacy-preserving data integration and sharing. In *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD '04*, page 19, Paris, France, 2004. ACM Press.

[9] R. Cramer, I. Damgård, and J. B. Nielsen. Multiparty Computation from Threshold Homomorphic Encryption. In *Advances in Cryptology — EUROCRYPT 2001*, Lecture Notes in Computer Science, pages 280–300. Springer Berlin Heidelberg, 2001.

[10] I. Damgård and M. Jurik. A Length-Flexible Threshold Cryptosystem with Applications. In *Information Security and Privacy*, Lecture Notes in Computer Science, pages 350–364. Springer Berlin Heidelberg, 2003.

[11] I. Damgård and M. Koprowski. Practical Threshold RSA Signatures without a Trusted Dealer. In *Advances in Cryptology – EUROCRYPT 2001*, Lecture Notes in Computer Science, pages 152–165. Springer Berlin Heidelberg, 2001.

[12] I. Damgård and J. B. Nielsen. Universally Composable Efficient Multiparty Computation from Threshold Homomorphic Encryption. In *Advances in Cryptology - CRYPTO 2003*, Lecture Notes in Computer Science, pages 247–264. Springer Berlin Heidelberg, 2003.

[13] Y. Desmedt. Society and Group Oriented Cryptography: a New Concept. In *Advances in Cryptology — CRYPTO '87*, Lecture Notes in Computer Science, pages 120–127. Springer Berlin Heidelberg, 1988.

[14] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Advances in Cryptology — CRYPTO' 89 Proceedings*, Lecture Notes in Computer Science, pages 307–315. Springer New York, 1990.

[15] O. Farràs and C. Padró. Ideal Hierarchical Secret Sharing Schemes. *IEEE Transactions on Information Theory*, 58(5):3273–3286, 2012.

[16] P.-A. Fouque, G. Poupard, and J. Stern. Sharing Decryption in the Context of Voting or Lotteries. In *Financial Cryptography*, Lecture Notes in Computer Science, pages 90–104. Springer Berlin Heidelberg, 2001.

[17] R. Gennaro, S. Halevi, H. Krawczyk, and T. Rabin. Threshold RSA for Dynamic and Ad-Hoc Groups. In *Advances in Cryptology – EUROCRYPT 2008*, Lecture Notes in Computer Science, pages 88–107. Springer Berlin Heidelberg, 2008.

[18] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini. Dynamic Threshold Cryptosystems (A New Scheme in Group Oriented Cryptography). *Proceedings of PRAGOCRYPT*, 96:370–379, 1996.

[19] H. Ghodosi, J. Pieprzyk, and R. Safavi-Naini. Secret sharing in multilevel and compartmented groups. In *Information Security and Privacy*, Lecture Notes in Computer Science, pages 367–378. Springer Berlin Heidelberg, 1998.

[20] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2003.

[21] O. Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[22] D. Grigoriev and I. Ponomarenko. Homomorphic Public-Key Cryptosystems and Encrypting Boolean Circuits. *Applicable Algebra in Engineering, Communication and Computing*, 17(3):239–255, 2006.

[23] L. Harn, H. Lin, and S. Yang. Threshold cryptosystem with multiple secret sharing policies. *IEE Proceedings - Computers and Digital Techniques*, 141(2):142–144, 1994.

[24] C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols*. Springer Berlin Heidelberg, 2010.

[25] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(9):56–64, 1989.

[26] C. S. Laih and L. Harn. Generalized threshold cryptosystems. In *Advances in Cryptology — ASIACRYPT '91*, Lecture Notes in Computer Science, pages 159–166. Springer Berlin Heidelberg, 1993.

[27] G. G. Lorentz, K. Jetter, and S. Riemenschneider. *Birkhoff Interpolation*. Number 19 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1983.
[28] R. Mendes and J. P. Vilela. Privacy-Preserving Data Mining: Methods, Metrics, and Applications. *IEEE Access*, 5:10562–10582, 2017.
[29] T. Nishide and K. Sakurai. Distributed Paillier Cryptosystem without Trusted Dealer. In *Information Security Applications*, Lecture Notes in Computer Science, pages 44–60. Springer Berlin Heidelberg, 2010.
[30] T. Okamoto. An Efficient Divisible Electronic Cash Scheme. In *Advances in Cryptology — CRYPTO '95*, Lecture Notes in Computer Science, pages 438–451. Springer Berlin Heidelberg, 1995.
[31] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology – EUROCRYPT '99*, Lecture Notes in Computer Science, pages 223–238. Springer Berlin Heidelberg, 1999.
[32] N. Pakniat, M. Noroozi, and Z. Eslami. Distributed key generation protocol with hierarchical threshold access structure. *IET Information Security*, 9(4):248–255, 2015.
[33] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology – CRYPTO '91*, Lecture Notes in Computer Science, pages 129–140. Springer Berlin Heidelberg, 1991.
[34] T. P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Advances in Cryptology — EUROCRYPT '91*, Lecture Notes in Computer Science, pages 522–526. Springer Berlin Heidelberg, 1991.
[35] T. Rabin. A Simplified Approach to Threshold and Proactive RSA. In *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science, pages 89–104. Springer Berlin Heidelberg, 1998.
[36] A. Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.
[37] G. J. Simmons. How to (Really) Share a Secret. In *Advances in Cryptology — CRYPTO' 88*, Lecture Notes in Computer Science, pages 390–448. Springer New York, 1990.
[38] T. Tassa. Hierarchical Threshold Secret Sharing. *Journal of Cryptology*, 20(2):237–264, 2007.
[39] G. Traverso, D. Demirel, and J. Buchmann. Dynamic and Verifiable Hierarchical Secret Sharing. In *Information Theoretic Security*, Lecture Notes in Computer Science, pages 24–43. Springer, Cham, 2016.
[40] G. Traverso, D. Demirel, and J. Buchmann. Performing Computations on Hierarchically Shared Secrets. In *Progress in Cryptology – AFRICACRYPT 2018*, Lecture Notes in Computer Science, pages 141–161. Springer International Publishing, 2018.

Appendix A. Hierarchical Threshold Paillier Cryptosystem with Trusted Dealer

The hierarchical threshold Paillier cryptosystem with trusted dealer is based on the threshold Paillier cryptosystem of [16] which uses a trusted dealer to perform the setup, i. e., computing the private and public key, as well as sharing the private key with Shamir's secret sharing scheme. In their scheme, the trusted dealer also generates verification keys for each party, in order to provide verifiability. Their system is robust and threshold semantically secure in the presence of a malicious adversary corrupting at most $t-1$ shareholders excluding the dealer [16], i. e., the dealer is honest. The **Key Generation**

algorithm is described in Protocol A.1. The **Encryption** algorithm is the same as for the original Paillier cryptosystem (Protocol 3.8) with $g := 1 + N$. The **Decryption** algorithm is described in Protocol A.2.

PROTOCOL A.1 (Threshold Paillier Cryptosystem - **Key Generation** [16]). *Let $\mathcal{S} := \{s_1, \ldots, s_n\}$ be a set of shareholders and let $\Delta := n!$. Let $t \in \{1, \ldots, n\}$ be the threshold. Let k be a security parameter. The following steps are performed by the trusted dealer:*

1. *Select uniformly at random two distinct safe primes*[4] *p and q and set $N := pq$ s. t.*
   (a) *$p, q > n$, p, q have bit size k, and $\gcd(N, \varphi(N)) = 1$*
   (b) *and $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes*
2. *Select $\beta \leftarrow_\$ \mathbb{Z}_N^*$, set $N' := p'q'$, set $\theta := \beta N' \bmod N$ and set $a_0 := \beta N'$.*
3. *Select $a_1, \ldots, a_{t-1} \leftarrow_\$ \{0, \ldots, NN' - 1\}$ nad*
4. *Construct $f(x) := \sum_{\omega=0}^{t-1} a_\omega x^\omega \bmod NN'$.*
5. *Distribute the share $\sigma_i := f(i) \bmod NN'$ to shareholder s_i, for $i \in \{1, \ldots, n\}$.*
6. *Select $w \leftarrow_\$ \mathbb{Z}_{N^2}^*$ and set $v := w^2 \bmod N^2$.*
7. *Compute verification keys $v_i := v^{\Delta\sigma_i} \bmod N^2$ for $i \in \{1, \ldots, n\}$.*

The public key is (N, θ) and the private key is $\beta N'$. In addition, all verification keys v_i and v are published for $i \in \{1, \ldots, n\}$.

PROTOCOL A.2 (Threshold Paillier Cryptosystem - **Decryption** [16]). *Let c be the ciphertext to be decrypted. Let $\mathcal{R} \subseteq \mathcal{S}$ denote the shareholders who participate in the decryption process. Each shareholder $s_i \in \mathcal{R}$ proceeds as follows:*

1. *Compute $c_i := c^{2\Delta\sigma_i} \bmod N^2$.*
2. *Prove that $\log_{c^{4\Delta}}(c_i^2) = \log_v(v_i)$ (Protocol 3.5).*
3. *Send c_i and its proof to the combiner.*

The combiner performs the following steps:

1. *Let $\mathcal{R}' \subseteq \mathcal{R}$ be the set of shareholders with correct proofs. If $|\mathcal{R}'| < t$, then the decryption fails.*
2. *Otherwise compute*

$$m := \mathrm{L}\left(\prod_{s_i \in \mathcal{R}'} c_i^{2\mu_i} \bmod N^2\right) \cdot \frac{1}{4\Delta^2\theta} \bmod N$$

where

$$\mu_i := \Delta \cdot \prod_{\substack{s_j \in \mathcal{R}' \\ j \neq i}} \frac{j}{j - i} \in \mathbb{Z}.$$

[4] A prime p is a safe prime if $\frac{p-1}{2}$ is also a prime.

The hierarchical threshold Paillier cryptosystem with trusted dealer is created by replacing Shamir's secret sharing scheme with a hierarchical secret sharing scheme to share the private key. We consider a semi-honest adversary as this will yield an efficient protocol. We will focus on a conjunctive access structure. The version for a disjunctive access structure is given in brackets. The **Key Generation** algorithm is described in Protocol A.3. The **Encryption** algorithm is the same as for the original Paillier cryptosystem (Protocol 3.8) with $g := 1+N$. The **Decryption** algorithm solves the Birkhoff interpolation problem in the exponent and is described in Protocol A.4.

PROTOCOL A.3 (Hierarchical Paillier Cryptosystem - **Key Generation**). *Let $\mathcal{S}$ be a set of shareholders composed of $l+1$ pairwise disjoint levels. Let $n_{-1} := 0$ and let $n_h \in \mathbb{N}$ be such that $n_h \geq n_{h-1} + |\mathcal{S}_h|$ for $h \in \{0, \ldots, l\}$. Let $\Gamma \subseteq \mathcal{P}(\mathcal{S})$ be a conjunctive (disjunctive) $(\vec{t}, n_l)$ hierarchical threshold access structure with $\vec{t} := (t_0, \ldots, t_l) \in \mathbb{N}^{l+1}$ and let $t_{-1} := 0$. Let (i,j) denote the unique ID of shareholder $s_{i,j} \in \mathcal{S}_h$ for $h \in \{0, \ldots, l\}$, where $i \in \{n_{h-1} + 1, \ldots, n_h\}$ and $j := t_{h-1}$ ($j := t_l - t_h$). Let k be a security parameter.*

The following steps are performed by the trusted dealer:

1. *Select uniformly at random two distinct safe primes p and q and set $N := pq$ s. t.*
   (a) *p and q have bit size k, and $\gcd(N, \varphi(N)) = 1$.*
   (b) *$p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also primes*
   (c) *$p', q' > 2^{-(t_l-2)} \cdot (t_l - 1)^{\frac{t_l-1}{2}} \cdot (t_l - 1)! \cdot n_l^{\frac{(t_l-2)(t_l-1)}{2}}$*
2. *Select $\beta \leftarrow_{\$} \mathbb{Z}_N^*$, set $N' := p'q'$, $\theta := \beta N'$ mod N and $a_0 := \beta N'$ ($a_{t_l-1} := \beta N'$).*
3. *Select $a_1, \ldots, a_{t_l-1} \leftarrow_{\$} \{0, \ldots, NN'-1\}$ ($a_0, \ldots, a_{t_l-2} \leftarrow_{\$} \{0, \ldots, NN'-1\}$).*
4. *Construct $f(x) := \sum_{\omega=0}^{t_l-1} a_\omega x^\omega$ mod NN'.*
5. *Distribute the share $\sigma_{i,j} := f^{(j)}(i)$ mod NN' to shareholder $s_{i,j}$, for all $s_{i,j} \in \mathcal{S}$.*

The public key is (N, θ) and the private key is $\beta N'$.

PROTOCOL A.4 (Hierarchical Paillier Cryptosystem - **Decryption**). *Let c be the ciphertext to be decrypted. Let $\mathcal{R} \subseteq \mathcal{S}$ denote the shareholders who participate in the decryption process. The shareholders proceed as follows:*

1. *Execute Protocol 3.2 (**Birkhoff Setup**).*
2. *Let $s_u := s_{i_u,j_u}$ and $\sigma_u := \sigma_{i_u,j_u}$ for $(i_u, j_u) \in \mathrm{I}(E)$.*
3. *Compute $c_u := c^{2\mathrm{A}\sigma_u}$ mod N^2.*
4. *Send the partial decryption c_u to the combiner.*

If $\mathcal{R} \in \Gamma$, then the combiner computes

$$m := \mathrm{L}\left(\prod_{s_u \in \mathcal{R}'} c_u^{\psi_u} \bmod N^2\right) \frac{1}{2\mathrm{A}^2\theta} \bmod N$$

where

$$\psi_u := (-1)^{u+b-d} \cdot \det(\mathrm{A}_{u,b-d}(E, X, \mathcal{G})).$$

Due to the fact that the modulus NN' is unknown to the shareholders, Protocol A.3 is not dynamic, as the required computations cannot be performed in the correct field. However, the dealer can be used to generate new shares or reset the structure.

PROPOSITION A.1 (Correctness). *The combiner obtains a correct decryption of a ciphertext with the execution of Protocol A.4 if it holds that* $\mathcal{R} \in \Gamma$.

PROOF. We prove that the decryption is correct. Therefore, we first prove that the private key is shared correctly, i. e., the private key can be reconstructed. Second, we show that the proposed decryption will yield the correct plaintext.

Let $s_u := s_{i_u,j_u} \in \mathcal{R}$ and let $\sigma_u := \sigma_{i_u,j_u}$ for $(i_u, j_u) \in \mathrm{I}(E)$. Let $\mathrm{A} := \det(\mathrm{A}(E, X, \mathcal{G}))$ and $\mathrm{A}_{x,y} := \det(\mathrm{A}_{x,y}(E, X, \mathcal{G}))$. Using the shares of the shareholders, the private key can be reconstructed, as it holds that

$$\begin{aligned}\sum_{s_u \in \mathcal{R}} \psi_u \sigma_u &\equiv \sum_{s_u \in \mathcal{R}} \sigma_u \cdot (-1)^{u+b-d} \cdot \mathrm{A}_{u,b-d} \\ &\equiv \mathrm{A} \sum_{s_u \in \mathcal{R}} \sigma_u \cdot (-1)^{u+b-d} \cdot \frac{\mathrm{A}_{u,b-d}}{\mathrm{A}} \\ &\equiv \mathrm{A} a_b \equiv \mathrm{A}\beta N' \bmod NN'.\end{aligned}$$

The order of c^2 divides NN' [16]. Hence, for a ciphertext $c := (1+N)^m r^N \bmod N^2$ it holds that

$$\begin{aligned}\prod_{s_u \in \mathcal{R}} c_u^{\psi_u} &\equiv \prod_{s_u \in \mathcal{R}} \left(c^{2A_{\mathcal{R}}\sigma_u}\right)^{\psi_u} \equiv c^{2A_{\mathcal{R}} \sum_{s_u \in \mathcal{R}} \sigma_u \psi_u} \\ &\equiv c^{2\mathrm{A}^2\beta N'} \equiv \left((1+N)^m r^N\right)^{2\mathrm{A}^2\beta N'} \\ &\equiv (1+N)^{m2\mathrm{A}^2\beta N'} \cdot \underbrace{\left(r^{2NN'}\right)}_{\equiv r^{\lambda(N^2)} \equiv 1 \bmod N}^{\mathrm{A}^2\beta} \\ &\equiv (1+N)^{m2\mathrm{A}^2\beta N'} \bmod N^2.\end{aligned}$$

Thus, we have

$$\begin{aligned}\mathrm{L}\left(\prod_{s_u \in \mathcal{R}} c_u^{\psi_u} \bmod N^2\right) \cdot \frac{1}{2\mathrm{A}^2\theta} &\equiv \mathrm{L}\left((1+N)^{m2\mathrm{A}^2\beta N'} \bmod N^2\right) \cdot \frac{1}{2\mathrm{A}^2\theta} \\ &\equiv m2\mathrm{A}^2 \underbrace{\beta N'}_{\equiv \theta \bmod N} \cdot \frac{1}{2\mathrm{A}^2\theta} \equiv m \bmod N.\end{aligned}$$

□

THEOREM A.2. *The hierarchical threshold Paillier cryptosystem as described in Protocols A.3 and A.4 is threshold semantically secure in the random oracle model with a semi-honest adversary corrupting an unauthorized set $\mathcal{U} \notin \Gamma$ of shareholders.*

PROOF. The proof of Theorem A.2 is analogous to the proof of Theorem 5.4 of the hierarchical and dynamic threshold Paillier cryptosystem without trusted dealer in Section 5. The main difference is that we don't have to generate verification keys. The partial decryptions are computed as follows: The simulator chooses random shares for the corrupted shareholders and then computes the partial decryption shares analogous as in the proof of Theorem 5.4. □

REMARK A.3 (Creating a hierarchical and dynamic threshold RSA cryptosystem). The techniques used for the hierarchical and dynamic threshold Paillier cryptosystem can also be applied to the RSA cryptosystem.[5] In the case of a trusted dealer, the dealer generates the public and private key and shares the private key with the hierarchical and dynamic secret sharing scheme from Sections 3.3 and 3.4. If no trusted dealer is used, the key generation of Protocol 5.1 can be reused to create the modulus and shares of the primes. The private key can be generated analogously to the threshold RSA cryptosystem in [6]. As the decryption in the RSA cryptosystem is also based on exponentiation, a similar technique as in Protocol A.4 and Protocol 5.2 can be applied to realize the decryption of a corresponding hierarchical RSA cryptosystem, respectively.

A.1. *Evaluation.* In order to give an impression of the performance, we measure the run time of the hierarchical threshold Paillier cryptosystem with trusted dealer as described in Appendix A. We consider the conjunctive case for different combinations of threshold vectors and number of shareholders. One test run consists of the dealer generating the keys, the shareholders computing the partial decryptions and the combiner recovering the plain text. The reconstruction of the plaintext is performed with the minimal amount of shareholders required from each level. The thresholds and the total number of shareholders are chosen arbitrarily, as the run time measurements shall only provide an idea of the performance.

The tests were performed in a single run on a Thinkpad T420 notebook with an Intel Core i5-2520M CPU with 2.50GHz (turbo: 3.20GHz) and 8GB of memory. Each test was executed in a single thread with Java 8. Each test run was executed ten times, and we measured the average run time. The actual run times are given in Table 1.

[5]The hierarchical and dynamic secret sharing scheme over the integers from Section 4 can be used to share the private key of any cryptosystem where the private key can be shared additively.

TABLE 1. The average time of ten runs needed by the participating parties for different combinations of thresholds and number of shareholders in the hierarchical threshold Paillier cryptosystem (Appendix A). The times are measured for the key generation performed by the dealer, the computation of a partial decryption of a shareholder and the message recovery performed by the combiner. The key size is 2048 bit, i. e., $\log_2(N) = 2048$.

$\vec{t}$	$(\lvert\mathcal{S}_0\rvert, \lvert\mathcal{S}_1\rvert, \ldots, \lvert\mathcal{S}_l\rvert)$	$\lvert\mathcal{S}\rvert$	Dealer	Shareholder	Combiner
$(1, 4, 8)$	$(20, 30, 50)$	100	54.5 s	46 ms	13 ms
$(1, 7, 14)$	$(20, 30, 50)$	100	79.1 s	772 ms	40.7 s
$(1, 7, 11, 14)$	$(20, 30, 50, 100)$	200	49.2 s	86 ms	3.9 s

The dealer requires most of the time for the prime generation. The different timings for a shareholder or the combiner for different threshold vectors $\vec{t}$ are caused by the corresponding reconstruction matrices. The most time-consuming part of the decryption is the computation of the determinant of the reconstruction matrix. The reconstruction matrix contains more 0's in the case $\vec{t} = (1, 7, 11, 14)$ than in the case $\vec{t} = (1, 7, 14)$ and thus, the determinant can be computed faster. As the reconstruction matrix depends on the threshold vector, the time needed for decryption can vary largely for different threshold vectors.

Naslov

Prvi autor, drugi autor i treći autor

SAŽETAK. Hrvatski prijevod sažetka.

Andreas Klinger
RWTH Aachen University
Templergraben 55
52062 Aachen, Germany
E-mail: klinger@itsec.rwth-aachen.de

Stefan Wüller
RWTH Aachen University
Templergraben 55
52062 Aachen, Germany
E-mail: wueller@itsec.rwth-aachen.de

Giulia Traverso
CYSEC SA
EPFL Innovation Park Builgin A
1015 Lausanne, Switzerland
E-mail: giulia.traverso@cysec.systems

Ulrike Meyer
RWTH Aachen University
Templergraben 55
52062 Aachen, Germany
E-mail: meyer@itsec.rwth-aachen.de