Dynamic and Verifiable Hierarchical Secret Sharing

Giulia Traverso, Denise Demirel, Johannes Buchmann

Technische Universität Darmstadt*

Abstract. In this work we provide a framework for dynamic secret sharing and present the first dynamic and verifiable hierarchical secret sharing scheme based on Birkhoff interpolation. Since the scheme is dynamic it allows, without reconstructing the message distributed, to add and remove shareholders, to renew shares, and to modify the conditions for accessing the message. Furthermore, each shareholder can verify its share received during these algorithms protecting itself against malicious dealers and shareholders. While these algorithms were already available for classical Lagrange interpolation based secret sharing, corresponding techniques for Birkhoff interpolation based schemes were missing. Note that Birkhoff interpolation is currently the only technique available that allows to construct hierarchical secret sharing schemes that are efficient and allow to provide shares of equal size for all shareholder in the hierarchy. Thus, our scheme is an important contribution to hierarchical secret sharing.

Keywords: hierarchical secret sharing, distributed storage, cloud computing, long-term security, Birkhoff interpolation, proactive secret sharing

1 Introduction

1.1 Motivation and Contribution

Secret sharing is an important primitive that allows to store sensitive data in distributed fashion. In classical secret sharing schemes any subset of a certain amount of shareholders can reconstruct the message distributed. This is different for *hierarchical secret sharing* [20, 15, 21, 5, 11, 22]. Here the shares are generated, such that not only the amount of shareholders, but also the level in the hierarchy they are assigned to is crucial for message reconstruction. Assume, for instance, signature keys are distributed to employees of a company. Then, hierarchical secret sharing allows to introduce certain conditions to the signing process, e.g. that at least one department head or senior must attend for a valid signature.

However, compared to classical secret sharing schemes, the approaches concerning hierarchical secret sharing are less developed. For instance, *dynamic*

[*] This work was in part funded by the European Commission through grant agreement no 644962 (PRISMACLOUD). Furthermore, it received funding from the DFG as part of project S6 within the CRC 1119 CROSSING.

schemes allowing, without reconstructing the shared massage, to add or remove shareholders (e.g. to reboot or reinstall servers holding shares), to renew the shares, and to modify the conditions for accessing the message are available for classical secret sharing only, while solutions for dynamic hierarchical secret sharing schemes have not been provided yet. Furthermore, classic schemes allow for verifiability, i.e. each shareholder can verify the correctness of its share received. For hierarchical secret sharing such algorithms are only available for the very inefficient early approaches.

Contribution. *In this work we provide the first dynamic and verifiable secret sharing scheme that is hierarchical, efficient, and ideal with respect to the size of the shares.* More precisely, like in [22] our solution uses Birkhoff interpolation to reconstruct the shared message. This allows to compute shares of equal size for all shareholders independent of their ability to reconstruct the message. We show how to enhance Birkhoff interpolation based schemes, i.e. *disjunctive secret sharing* and *conjunctive secret sharing*, by algorithms that allow, without message reconstruction, to add and remove shareholders, to modify the conditions for accessing the message, and to renew shares. Furthermore, our scheme is verifiable and therefore protects against malicious dealers and shareholders. Moreover, we provide the first definition of *dynamic secret sharing* and prove our scheme secure.

Organization. After providing preliminaries in Section 2, we introduce a framework for dynamic secret sharing in Section 3. Afterwards, we provide an introduction to hierarchical secret sharing in Section 4, present our verifiable and dynamic hierarchical secret sharing scheme in Section 5, and conclude with a summary and possible future work in Section 6.

1.2 Related Work

Hierarchical Secret Sharing. The first solutions for hierarchical secret sharing have been proposed by Shamir in [20] and Kothari in [15]. In Shamir's approach the higher a shareholder is in the hierarchy, the more shares it gets, overloading the most powerful shareholders. In Kothari's solution, shareholders are grouped in sets and for each set an independent secret sharing scheme is instantiated. This requires managing multiple secret sharing schemes and does not allow for cooperation among sets during the reconstruction. *Disjunctive secret sharing* as introduced by Simmons in [21], is the first approach using only one secret sharing scheme and supporting cooperations of shareholders assigned to different sets, or rather levels in a hierarchy. However, his approach is not ideal meaning that the higher a shareholder in the hierarchy the larger the share to be stored. Brickell in [5] improved this by providing a disjunctive secret sharing scheme that is ideal with respect to the size of the shares, but apart from that rather inefficient. Later, Ghodosi et al. showed in [11] how to achieve efficient schemes for specific access structures. Finally, in [22] Tassa further improved this line of research by providing an efficient disjunctive secret sharing scheme for general access structures. Furthermore, he introduced *conjunctive secret sharing* that does not only allow concurrency among levels, but strictly requires the presence of a minimum amount of shareholders from the highest levels. Both conjunctive and disjunctive secret sharing are good solutions for hierarchical secret sharing and our contribution builds on Tassa's work. None of these approaches provide verifiability, nor do they allow, without reconstructing the shared message, to add or remove shareholders, to modify the conditions for accessing the message, nor to renew shares.

Dynamical and verifiable hierarchical secret sharing. Notions of dynamic secret sharing have been already proposed, yet with different meanings and less functionalities with respect to our definition. More precisely, in the one hand, in [4] it is the dealer that decides which shareholders reconstruct which secret. On the other hand, in [3] it is not possible to add shareholders without changing all the shares already distributed. Moreover, none of these approaches are suitable for hierarchical secret sharing nor do they provide verifiability. The only step towards a dynamic Birkhoff interpolation-based secret sharing scheme has been made by Pakniat et al. in [17]. It is shown how to renew shares, but, again, this process does not allow to add or remove shareholders and to modify the conditions for accessing the message nor does it provide verifiability or addresses conjunctive secret sharing. At the same time, for classical secret sharing schemes dynamic and verifiable solutions have been developed. For instance, in [16] it is shown how to add shareholders, in [13] it is shown how shares can be renewed, and in [12] it is shown how even the entire set of shareholders and the conditions for accessing the message can be changed. In addition, all these algorithms come with verifiability. Note that classical secret sharing is based on Lagrange interpolation and the protocols [16], [13], and [12] allowing for dynamism are defined accordingly. However, these approaches cannot be used for secret sharing schemes based on Birkhoff interpolation and solutions introducing dynamism also for these schemes need to be found.

Thus, our work is the first to provide dynamic and verifiable secret sharing based on Birkhoff interpolation.

2 Preliminaries

Secret sharing is a cryptographic primitive enabling a *dealer* to distribute a message among a set of *shareholders*, each of whom is allocated a *share* of the message. More precisely, to distribute a message $m \in \mathcal{M}$ to a set of shareholders $S = \{s_1, \ldots, s_n\}$ the dealer computes shares $\sigma_1, \ldots, \sigma_n \in \mathit{\Sigma}$, where $\mathcal{M}$ is the message space and $\mathit{\Sigma}$ the space of all possible shares. The message can be reconstructed only when an *authorized* subset $A \subset S$ of these shareholders combine their shares while *unauthorized* subsets $U \subset S$ are prevented from doing it. The *access structure* $\mathit{\Gamma} \in \mathcal{P}(S)$[1] determines both sets, i.e. $A \in \mathit{\Gamma}$ and $U \notin \mathit{\Gamma}$. From now on, the number of shareholders of a subset $R \subset S$ is denoted as $r := |R|$. Note that for security we assume that all data communicated by a dealer to a shareholder and between the shareholders is sent using private channels to prevent attackers from eavesdropping.

[1] $\mathcal{P}(S)$ denotes the partition of the set S.

Definition 1. *For a message space $\mathcal{M}$, a space of shares Σ, a set of shareholders $S = \{s_1, \ldots, s_n\}$, where $i \in \mathcal{I}$ is the unique ID of shareholder $s_i \in S$, and an access structure $\Gamma \subset \mathcal{P}(S)$, a* secret sharing scheme *is a pair of PPT algorithms* Share *and* Reconstruct*.*

Share *It takes as input a message $m \in \mathcal{M}$ and it outputs n shares $\sigma_1, \ldots, \sigma_n \in \Sigma$, where share σ_i is to be sent to shareholder s_i, for $i = 1, \ldots, n$.*

Reconstruct *It takes as input a set of shares $\sigma_1, \ldots, \sigma_r$ held by a subset $R \subset S$ of shareholders. It outputs $m \in \mathcal{M}$ if $R \in \Gamma$, and $\perp$ otherwise.*

A secret sharing scheme is *perfectly secure* if any unauthorized subset of participants learns nothing about the message in an information-theoretic sense, while any authorized subset of participants is able to reconstruct the secret (*accessibility*). Since our improvements rely on the scheme proposed by Tassa in [22] we recall here his definition, which uses the *Shannon's entropy H*.

Definition 2. *Let us assume that $m \in \mathcal{M}$ is the message distributed by a secret sharing scheme among a set S of shareholders according to access structure Γ. For an authorized subset $A \in S$, i.e. $A \in \Gamma$, let us denote by σ_A the set of shares owned by the shareholders $s_i \in A$, i.e. $\sigma_A := \{\sigma_i \text{ such that } s_i \in A\}$. The* accessibility *of a secret sharing scheme is the property such that: $H(m|\sigma_A) = 0$, $\forall\ A \in \Gamma$. In contrast, any unauthorized subset $U \in S$, i.e. $U \notin \Gamma$, should not be able to reconstruct the secret. If in addition no information about $m \in \mathcal{M}$ is leaked to the shareholders in U, then the secret sharing scheme is* perfectly secure*: $H(m|\sigma_U) = H(m)$, $\forall\ U \notin \Gamma$.*

Another interesting primitive is *verifiable secret sharing* (VSS) [6]: each algorithm within a secret sharing scheme outputs some audit data allowing to check whether the algorithms themselves were performed correctly. Formally, a VSS scheme is a secret sharing scheme with the following additional requirements.

Definition 3 ([18]). *The algorithms in which shares are computed are extended by an additional verification protocol executed between the dealer and the shareholders $S = \{s_1 \ldots s_n\}$, such that the following properties are fulfilled.*

Completeness *If the parties computing the shares, e.g. dealers and shareholders, follow the algorithms correctly, then each shareholder accepts the new share with probability 1.*

Committing *If for any two authorized subsets $A_1 \subset S$ and $A_2 \subset S$, i.e. $A_1, A_2 \in \Gamma$, the shareholders of A_1 and A_2 accept their shares, then the following holds except with negligible probability: if m_i is the message reconstructed by the shareholders in A_i (for $i = 1, 2$), then $m_1 = m_2$.*

Note that the committing property of Definition 3 holds except with negligible probability, because this definition covers solutions using Pedersen commitments that are unconditionally hiding, but only computationally binding. If Feldmann commitments are used the verification protocol provides completeness even with probability 1. However, these commitments are only computationally hiding and do not ensure confidentiality in the long-term.

3 Dynamic Secret Sharing

The standard secret sharing definition only covers the algorithms Share and Reconstruct. However, in practice it is desirable that secret sharing schemes provide algorithms allowing to Add new shareholders and to Reset the entire access structure (i.e. the conditions for accessing the message and the set of shareholders). Note that algorithm Reset can be run to refresh the shares only, without modifying the access structure nor the set of shareholders. The algorithm Add differs from Reset in the sense that the access structure remains unchanged and old shareholders keep their shares. This is of practical interest since renewing shares could be a quite demanding and expensive procedure, e.g. in case shares are distributed on smartcards. Note that the algorithm Reset allows to remove shareholders, since the set S of shareholders can be replaced by a subset $S' \subset S$. In the framework of dynamic secret sharing, we assume that all communication channels used guarantee reliable delivery of messages, any two shareholders can communicate via a private channel, all shareholders can receive messages sent over a broadcast channel, any shareholder can declare and no shareholder can spoof its identity, and a majority of the shareholders participating in each algorithm is trustworthy such that wrongly generated shares can be detected. Note that these are standard assumption for classical secret sharing schemes that provide verifiability and dynamism and that the latter assumption can be weakened using the complaint mechanism proposed in [12]. Furthermore, our algorithms assume a synchronous network, but can easily be adapted to asynchronous networks, for instance, by using the techniques proposed in [19]. In the following, we formally introduce *dynamic secret sharing schemes* as secret sharing schemes that in addition allow to perform Add and Reset in distributed fashion.

Definition 4. *For a message space $\mathcal{M}$, a space of shares Σ, a set of shareholders $S = \{s_1, \ldots, s_n\}$ where $i \in \mathcal{I}$ is the unique ID of shareholder $s_i \in S$, and an access structure $\Gamma \subset \mathcal{P}(S)$, a* dynamic secret sharing scheme *is a tuple of PPT algorithms* Share, Add, Reset, *and* Reconstruct.

Share *It takes as input a message $m \in \mathcal{M}$. It outputs n shares $\sigma_1, \ldots, \sigma_n \in \Sigma$, where share σ_i is to be sent to shareholder $s_i \in S$, for $i = 1, \ldots, n$.*

Add *It takes as input a set of shares $\sigma_1, \ldots, \sigma_r$ held by a subset $R \subset S$ of shareholders and the ID i, i.e. $i = n + 1$, of the new shareholder. If R is unauthorized, i.e. $R \notin \Gamma$, it outputs $\perp$. Otherwise, $R \in \Gamma$ and without message reconstruction, it outputs a corresponding share $\sigma_i \in \Sigma$ for the new shareholder s_i.*

Reset *It takes as input a set of shares $\sigma_1, \ldots, \sigma_r$ held by a subset $R \subset S$ of shareholders, a new set of shareholders $S' = \{s'_1, \ldots, s'_{n'}\}$ (that need not be disjoint to S), and an access structure $\Gamma' \subset \mathcal{P}(S')$. If R is unauthorized, i.e. $R \notin \Gamma$, it outputs $\perp$. Otherwise, $R \in \Gamma$ and without message reconstruction, it outputs n' shares $\sigma'_1, \ldots, \sigma'_{n'}$, where share σ'_i is to be sent to each new* *shareholder $s'_i \in S'$, for $i = 1, \ldots, n'$. The shares $\sigma_1, \ldots, \sigma_n \in \Sigma$ held by the old shareholders are deleted.* [2]

Reconstruct *It takes as input a set of shares $\sigma_1, \ldots, \sigma_r$ held by a subset $R \subset S$ of shareholders. It outputs $m \in \mathcal{M}$ if $R \in \Gamma$, and $\perp$ otherwise.*

In addition to the algorithms Share, Add, and Reset, a *Verifiable and Dynamic Secret Sharing Scheme* provides audit data for verification according to Definition 3.

4 Secret Sharing Based on Birkhoff Interpolation

Simmons introduced in [21] hierarchical secret sharing as a secret sharing scheme where shareholders are divided into disjoint levels $L_0, \ldots, L_\ell$ and the power of a shareholder to reconstruct the message depends on the level it is assigned to. The union of all shareholders from all levels constitutes the set of shareholders $S = \{s_1, \ldots, s_n\}$, i.e.

$$S = \bigcup_{h=0}^{\ell} L_h, \text{ such that } L_h \cap L_k = \emptyset \text{ for } h \neq k.$$

If n_h is the number of shareholders assigned to level L_h, then $n = |S| = \sum_{h=0}^{\ell} n_h$. Furthermore, assume that L_0 is the highest level and L_ℓ the lowest level. Clearly, it is expected that less shares are needed to reconstruct the message at the higher levels, i.e. shareholders assigned to the highest level have a larger ability to reconstruct the message. Therefore, denoted by t_h the threshold associated to level L_h, for $h = 0, \ldots, \ell$, it is plausible to assume that the lower a level the higher the threshold, i.e. $0 < t_0 < \cdots < t_\ell$.

For legibility, in the following we concentrate on conjunctive secret sharing as introduced by Tassa in [22]. The corresponding solution for disjunctive secret sharing can be found in brackets.

Definition 5. *Assume the existence of a message space $\mathcal{M}$, a space of shares Σ, and an access structure $\Gamma \subset \mathcal{P}(S)$ where t_h is the threshold for level L_h, for $h = 0, \ldots, \ell$ with $t := t_\ell$ and $t_{-1} := 0$. Furthermore, assume a set of n shareholders S where the pair $(i, j) \in \mathcal{I} \times \mathcal{I}$ is the unique ID of shareholder $s_{i,j} \in L_h$ and $j := t_{h-1} (j := t_\ell - t_h)$, for $i = 1, \ldots, n_h$ and $h = 0, \ldots, \ell$. Then a* conjunctive (disjunctive) secret sharing scheme *is a pair of PPT algorithms* Share *and* Reconstruct*, defined as follows.*

Share *It takes as input a message $m \in \mathcal{M}$ and generates a polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$ where $a_0 := m$ $(a_{t-1} := m)$ and the coefficients $a_1, \ldots, a_{t-1} \in \mathbb{F}_q$ $(a_0, \ldots, a_{t-2} \in \mathbb{F}_q)$ are chosen uniformly at random. It outputs n shares $\sigma_{i,j} \in \Sigma$, where share $\sigma_{i,j} := f^j(i)$ is to be sent to shareholder $s_{i,j} \in L_h$, for $i = 1, \ldots, n_h$ and $h = 0, \ldots, \ell$ and $f^j(x)$ is the j-th derivative of the polynomial $f(x)$.*

[2] To renew the shares, the algorithm Reset is run with the old set of shareholder S and the old access structure Γ as input.

Reconstruct *It takes as input a set of shares held by a subset $R \subset S$ of shareholders. It outputs $m \in \mathcal{M}$ if $R \in \Gamma$, where $m = a_0$ ($m = a_{t-1}$) is retrieved using Birkhoff interpolation. It outputs $\perp$ otherwise.*

In the following, it is described in details how Birkhoff interpolation is performed such that Reconstruct outputs the message $m \in \mathcal{M}$.

Let us assume a subset $R \subset S$ of $r := |R|$ shareholders participating in the reconstruction such that $R \in \Gamma$. The *interpolation matrix* associated to set R is a binary matrix E where entry $e_{i,j}$ is set to '1' if shareholder $s_{i,j}$ participates with share $\sigma_{i,j}$ (that is the j-th derivative of f on position i) and '0' otherwise. The *Birkhoff interpolation problem* is the problem of finding a polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1} \in \mathbb{R}_{t-1}[x]$ satisfying the equalities $f^j(i) = \sigma_{i,j}$, where $\mathbb{R}_{t-1}[x]$ is the ring of the polynomials with degree at most $t-1$.

In the following, $I(E) = \{(i, j) \text{ such that } e_{i,j} = 1\}$ is the set containing the entries of E in lexicographic order, i.e. the pair (i, j) precedes the pair (i', j') if and only if $i < i'$ or $i = i'$ and $j < j'$. The elements of $I(E)$ are denoted by $(i_1, j_1), (i_2, j_2), \ldots, (i_r, j_r)$. Furthermore, we set $\varphi := \{\phi_0, \phi_1, \phi_2, \ldots, \phi_{t-1}\} = \{1, x, x^2, \ldots, x^t\}$ and denote by ϕ_k^j the j-the derivative of ϕ_k, for $k = 0, \ldots, t-1$. Then the matrix $A(E, X, \varphi)$ is defined as follows:

$$A(E,X,\varphi) = \begin{pmatrix} \phi_0^{j_1}(i_1) & \phi_1^{j_1}(i_1) & \phi_2^{j_1}(i_1) & \cdots & \phi_{t-1}^{j_1}(i_1) \\ \phi_0^{j_2}(i_2) & \phi_1^{j_2}(i_2) & \phi_2^{j_2}(i_2) & \cdots & \phi_{t-1}^{j_2}(i_2) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \phi_0^{j_r}(i_r) & \phi_1^{j_r}(i_r) & \phi_2^{j_r}(i_r) & \cdots & \phi_{t-1}^{j_r}(i_r) \end{pmatrix}.$$

Then polynomial $f(x) \in \mathbb{R}_{t-1}[x]$ can be reconstructed by computing

$$f(x) = \sum_{k=0}^{t-1} \frac{\det(A(E,X,\varphi_k))}{\det(A(E,X,\varphi))} x^k,$$

where $A(E, X, \varphi_k)$ is obtained from $A(E, X, \varphi)$ by replacing its $(k+1)$-th column with the shares $\sigma_{i,j}$ in lexicographic order.

Note that it depends on the interpolation matrix E whether the Birkhoff interpolation problem has a unique solution and, consequently, the secret sharing scheme is accessible (see Appendix A for the necessary and sufficient conditions). In the following, it is assumed that the access structure Γ is chosen such that the matrix E leads to a well posed Birkhoff interpolation problem, as already discussed by Tassa in [22].

5 Providing a Dynamic and Verifiable Hierarchical Secret Sharing Scheme

In this section, we show how Tassa's conjunctive and disjunctive hierarchical secret sharing schemes can be enhanced by introducing the algorithms Add and

Reset to the existing algorithms Share and Reconstruct. This leads to dynamic secret sharing, as defined in Definition 4. Note that with respect to algorithm Reset that renews the shares our construction is more efficient compared to the protocol proposed in [17]. More precisely, they demand the shareholders to reconstruct the entire function in distributed fashion while in our scheme one coefficient of the function is sufficient Furthermore, we show how the algorithms can be enhanced such that verifiability is provided. In fact, this ensures that the distributed message cannot be changed by malicious shareholders when these algorithms are run. From now on we simplify the notation referring to the shareholders within subset $R \subset S$ as s_l and no longer as $s_{(i,j)}$. However, we stress that shareholders in R are not equal from the hierarchical point of view.

5.1 Distributed Computation of Determinants

To fulfill Definition 4, the algorithms Add and Reset have to be performed without reconstructing the message $m \in \mathcal{M}$. This is possible since determinants $\det(A(E, X, \varphi_k))$, for $k = 0, \dots, t-1$, can be computed in distributed fashion.

Theorem 1. *The polynomial* $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{t-1}x^{t-1} \in \mathbb{R}_{t-1}[x]$ *can be computed by*

$$f(x) = \sum_{k=0}^{t-1} a_k x^k = \sum_{k=0}^{t-1}\sum_{l=1}^{r} a_{l,k} x^k,$$

where $a_{l,k}$ *is computed by shareholder* $s_l \in R$, *for* $l = 1, \dots, r$ *and* $R \in \mathit{\Gamma}$ *is an authorized subset of* S, *with* $r =: |R|$.

Proof. Let us first recall that *Laplace's expansion formula* computes the determinant $\det(A)$ of an $n \times n$ matrix A as the weighted sum of the determinants of n sub-matrices of A, each of size $(n-1) \times (n-1)$. More precisely $\det(A) = \sum_{j'=1}^{n} a_{i,j'}(-1)^{i+j'} \det(A_{i,j'}) = \sum_{i'=1}^{n} a_{i',j}(-1)^{i'+j} \det(A_{i',j})$, where $A_{i,j}$ results from A by deleting the i-th row and j-th column.

The fact that $A(E, X, \varphi)$ can be computed by each shareholder from public information together with Laplace's expansion formula implies that each shareholder $s_l \in R$, for $l = 1, \dots, r$, can compute the partial information $a_{l,k}$ for coefficient $a_k = \frac{\det(A(E,X,\varphi_k))}{\det(A(E,X,\varphi))}$, by $a_{l,k} := \sigma_{i,j}(-1)^{l-1+k}\frac{\det(A_{l-1,k}(E,X,\varphi))}{\det(A(E,X,\varphi))}$, where $\sigma_{i,j}$ is the share held by shareholder s_l, and $A_{l-1,k}(E, X, \varphi)$ is the matrix that results from $A(E, X, \varphi)$ by removing the l-th row and the $(k+1)$-th column. From Laplace's expansion formula it follows that:

$$\sum_{l=1}^{r} a_{l,k} = \sum_{l=1}^{r} \sigma_{i,j}(-1)^{l-1+k}\frac{\det(A_{l-1,k}(E,X,\varphi))}{\det(A(E,X,\varphi))} = \frac{\det(A(E,X,\varphi_k))}{\det(A(E,X,\varphi))} = a_k.$$

In conclusion, the coefficients a_k, for $k = 0, \dots, t-1$, of polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{t-1}x^{t-1}$ are computed as the sum of the partial coefficients $a_{l,k}$, where $a_{l,k}$ is computed by shareholder $s_l \in R$ and $R \in \mathit{\Gamma}$ is an authorized set. Importantly, this also implies that $f(x) = \sum_{l=1}^{r} f_l(x)$, where $f(x) = \sum_{l=1}^{r} f_l(x) = \sum_{l=1}^{r}\sum_{k=0}^{t-1} a_{l,k}x^k$.

In the following, the notation defined above holds. That is, $a_{l,k}$ is the partial information held by shareholder s_l about the coefficient a_k of polynomial $f(x)$ and $f_l(x) = \sum_{k=0}^{t-1} a_{l,k}x^k$ is the partial Birkhoff interpolation polynomial of shareholder s_l. Note that Theorem 1 implies that also derivatives of polynomial $f(x)$ can be computed in a distributed fashion.

Theorem 2. *The j-th derivative $f^j(x)$ of polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ can be computed in distributed fashion as*

$$f^j(x) = \sum_{l=1}^{r} f_l^j(x),$$

where $f_l^j(x)$ is computed by shareholder $s_l \in R$, for $l = 1, \ldots, r$ and $R \in \Gamma$ is an authorized subset of S, with $r =: |R|$.

Proof. To compute the derivative of polynomial $f(x)$ each shareholder $s_l \in R$ first computes its partial Birhkoff interpolation polynomial $f_l(x) = \sum_{k=0}^{t-1} a_{l,k}x^k$. Then it computes the j-th derivative $f_l^j(x) = \sum_{k=j}^{t-1} \frac{k!}{(k-j)!} a_{l,k}x^{k-j}$. Note that due to the sum rule for derivatives, i.e. $(f(x) + g(x))' = f(x)' + g(x)'$, and $f(x) = \sum_{l=1}^{r} f_l(x)$ the j-th derivative $f^j(x)$ of polynomial $f(x)$ can be computed by adding all partial derivatives, i.e. $f^j(x) = \sum_{l=1}^{r} f_l^j(x)$.

5.2 Verifiable Algorithms for Dynamic Hierarchical Secret Sharing

In this section, we provide a verifiable dynamic conjunctive and a verifiable dynamic disjunctive secret sharing scheme using Birkhoff interpolation. The verification process is described using Feldman commitments [8]. However, it can easily be adapted to Pedersen commitments [18] to achieve information-theoretic confidentiality.[3] Like in Section 4, we focus on conjunctive secret sharing and show the differences to disjunctive secret sharing in brackets.

Let Γ be an access structure arranged in disjoint levels $L_0, \ldots, L_\ell$, where t_h is the threshold of level L_h for $h = 0, \ldots, \ell$. Let us assume a message space $\mathcal{M}$, a space of shares Σ, and a set of shareholders S where the pair $(i, j) \in \mathcal{I} \times \mathcal{I}$ is the unique ID of shareholder $s_{i,j} \in S$, such that $j = t_{h-1}$ ($j = t_\ell - t_h$) and $t_{-1} = 0$. Then the algorithms Share, Add, Reset, and Reconstruct for *verifiable dynamic conjunctive (disjunctive) secret sharing* are defined as follows.

Share It takes as input a message $m \in \mathcal{M}$. This algorithm works like the one in Definition 5 except that some additional audit data is computed and distributed. More precisely, the algorithm randomly chooses two large primes p, q, such that $q|(p-1)$. Let g be a generator of the q-th order subgroup $\mathbb{F}_q$

[3] There exists solutions [10], [9], [14], [2] for VSS providing both information-theoretic confidentiality and bindingness. However, they are not secure against a mobile adversary that is able to collect over time enough share to retrieve the message. The solution proposed in [2] is an interactive protocol while we only consider non-interactive protocol having less communication complexity.

of $\mathbb{F}_p^*$ and set $\mathcal{M} := \mathbb{F}_q$. After defining the polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$, where $a_0 := m$ ($a_{t-1} := m$) and $a_1, \ldots, a_{t-1} \in \mathbb{F}_q$ ($a_0, \ldots, a_{t-2} \in \mathbb{F}_q$) are chosen uniformly at random, the dealer commits to each coefficient a_k by computing $c_k := g^{a_k} \mod p$, for $k = 0, \ldots, t-1$. It broadcasts the commitments and sends each share $\sigma_{i,j}$ to shareholder $s_{i,j} \in L_h$, for $i = 1, \ldots, n_h$ and $h = 0, \ldots, \ell$ using a private channel. Shareholder $s_{i,j}$ accepts $\sigma_{i,j}$ as its valid share, if and only if

$$g^{\sigma_{i,j}} \equiv \prod_{k=j}^{t-1} c_k^{\frac{k!}{(k-j)!} i^{k-j}} = g^{f^j(i)}.$$

Add It takes as input a set of shares $\sigma_1, \ldots, \sigma_r$ held by a subset $R \subset S$ of shareholders and the ID (i', j') of the new shareholder. If R is unauthorized, i.e. $R \notin \Gamma$, it outputs $\perp$. Otherwise, $R \in \Gamma$ and the shareholders compute $\sigma_{i',j'} := f^{j'}(i')$ in distributed fashion. More precisely, each shareholder $s_l \in R$ performs the following steps, for $l = 1, \ldots, r$.

1. It computes the j'-th derivative of its partial Birkhoff interpolation polynomial at $x = i'$, i.e.

$$\lambda_l := \sigma_l \sum_{k=j'}^{t-1} \frac{k!}{(k-j')!} (-1)^{l-1+k} \frac{\det(A_{l-1,k}(E, X, \varphi))}{\det(A(E, X, \varphi))} i'^{k-j'}.$$

2. It randomly splits the result into r values, i.e. $\lambda_l = \delta_{1,l} + \cdots + \delta_{r,l}$ and sends $\delta_{m,l}$ to shareholder $s_{m,j} \in R$, for $m = 1, \ldots, r$ and $m \neq l$ using a private channel.
3. It collects all values $\delta_{l,m}$ received and computes $\delta_l := \sum_{m=1}^{r} \delta_{l,m}$.
4. It sends δ_l to the new shareholder $s_{i',j'}$ using a private channel and broadcasts the audit data $c_0, \ldots, c_{t-1}$ received during the share algorithm.

The new shareholder $s_{i',j'}$ computes its share $\sigma_{i',j'}$ by adding all values δ_l received, i.e. $\sigma_{i',j'} := \sum_{l=1}^{r} \delta_l$. It can verify the correctness of its share by checking whether

$$g^{\sigma_{i',j'}} \equiv \prod_{k=j'}^{t-1} c_k^{\frac{k!}{(k-j')!} i'^{k-j'}} = g^{f^{(j')}(i')},$$

using the audit data received from the shareholders.

Reset It takes as input a set of shares $\sigma_1, \ldots, \sigma_r$ held by a subset $R \subset S$ of shareholders a new set of shareholders $S' = \{s'_1, \ldots, s'_{n'}\}$, each accompanied with a unique ID (i', j'), and an access structure $\Gamma' \subset \mathcal{P}(S')$ with maximal threshold t'. If R is unauthorized, i.e. $R \notin \Gamma$, it outputs $\perp$. Otherwise, $R \in \Gamma$ and the subset of old shareholders jointly computes shares for the new shareholders in S'. More precisely, each old shareholder $s_l \in R$ performs the following steps, for $l = 1, \ldots, r$.

1. It computes its partial Birkhoff interpolation coefficient

$$a_{l,0} := \sigma_l(-1)^{l-1}\frac{\det(A_{l-1,0}(E,X,\varphi))}{\det(A(E,X,\varphi))}$$

   $\left(a_{l,t-1} = \sigma_l(-1)^{l+t-2}\frac{\det(A_{l-1,t-1}(E,X,\varphi))}{\det(A(E,X,\varphi))}\right)$.
2. It chooses a polynomial $f'_l(x) = a'_{l,0} + a'_{l,1}x + a'_{l,2}x^2 + \cdots + a'_{l,t'-1}x^{t'-1}$ of degree $t'-1$, where $a'_{l,0} = a_{l,0}$ ($a'_{l,t-1} = a_{l,t-1}$) is the partial Birkhoff interpolation coefficient and coefficients $a'_{l,1}, \ldots, a'_{l,t'-1} \in \mathbb{F}_q$ ($a'_{l,0}, \ldots, a'_{l,t'-2} \in \mathbb{F}_q$) are chosen uniformly at random.
3. It computes subshare $\sigma_{l,i',j'}$ for shareholder $s'_{i',j'} \in S'$ as $\sigma_{l,i',j'} := {f'}_l^{j'}(i')$.
4. It sends subshare $\sigma_{l,i',j'}$ to shareholder $s'_{i',j'} \in S$ using a private channel and broadcasts the audit data, composed of commitments to each coefficient of polynomial $f'_l(x)$, i.e. $c'_{l,k} := g^{a'_{l,k}}$, for $k = 0, \ldots, t'-1$, and commitment $c_0 = g^m$ ($c_{t-1} = g^m$) of the old polynomial $f(x)$.
5. It deletes its share.

Each new shareholder $s_{i',j'} \in S'$ computes its share $\sigma'_{i',j'}$ adding all subshares $\sigma_{l,i',j'}$ received, i.e. $\sigma'_{i',j'} := \sum_{l=1}^{r} \sigma_{l,i',j'}$. To verify the correctness of share $\sigma_{l,i',j'}$, each new shareholder $s_{i',j'} \in S'$ performs the following steps.

1. It checks the function value of each polynomial, i.e.

$$g^{\sigma_{l,i',j'}} \equiv \prod_{k=j'}^{t'-1} {c'_{l,k}}^{\frac{k!}{(k-j')!}{i'}^{k-j'}} = g^{f_l'^{(j')}(i')},\ \text{for } l = 1, \ldots, r.$$

2. It checks whether the free coefficient (last coefficient) of all polynomials $f'_l(i')$ leads to the original message $m \in \mathcal{M}$, i.e.

$$c_0 \equiv \sum_{l=1}^{r} c'_{l,0}$$

   ($c_{t-1} \equiv \sum_{l=1}^{r} c'_{l,t'-1}$).
3. If both equations are satisfied, it accept $\sigma'_{i',j'}$ as its valid share.

Reconstruct It takes as input shares held by a subset $R \subset S$ of shareholders. If $R \in \mathit{\Gamma}$, it outputs $m \in \mathcal{M}$ reconstructed using Birkhoff interpolation. It outputs $\perp$ otherwise. Having access to the original audit data $c_0 = g^{a_0}$ ($c_{t-1} = g^{a_{t-1}}$) it is possible to verify whether the reconstructed message $m \in \mathcal{M}$ is a correct opening value for commitment c_0 (c_{t-1}), i.e. $g^m \equiv c_0$ ($g^m \equiv c_{t-1}$).

5.3 Security and Efficiency

In this work, our achievement is enhancing Tassa's protocols by the algorithms Add and Reset. What we need to show is that even after performing these algorithms no information is leaked and the message can still be reconstructed, i.e.

perfect security and accessibility are provided. However, merging dynamic secret sharing with the verification protocol leads to an overall scheme that is either unconditionally binding or unconditionally hiding. A rigorous analysis can be found in Appendix B.

With respect to the algorithm Add, to compute a share for a new shareholder $s_{i',j'}$ each shareholder $s_l \in A$ of an authorized subset $A \in \Gamma$ computes $f_l^{j'}(i')$. Since this subshare leaks information about the own share, each shareholder randomly splits and distributes this value to the other shareholders. Then each shareholder only forwards the sum of all values received, hiding the individual subshares. Consequently, confidentiality is preserved. Accessibility is provided since the distributed subshares and the polynomials used for secret sharing are additively homomorphic. With respect to the algorithm Reset, each shareholder s_l of an authorized subset $A \in \Gamma$ use hierarchical secret sharing to distribute its share to a new (the same) set of shareholders. While security of this algorithm follows from the security of the used conjunctive or disjunctive secret sharing scheme, accessibility is provided by the homomorphic property of polynomials.

Verifiability is achieved with the help of homomorphic and computationally binding commitment schemes. They allow each shareholder $s_{i,j}$ to compute a commitment c^* to its share $\sigma_{i,j}$ using the commitments received, i.e. $c^* = \prod_{k=j}^{t-1} c_k^{\frac{k!}{(k-j)!} i^{k-j}} = g^{f^{(j)}(i)}$, where c_k is the commitment to coefficient a_k, for $k = 0, \ldots, t-1$. Thus, by verifying $c^* \equiv g^{\sigma_{i,j}}$ the correctness of its share can be checked.

Moreover, we argue that introducing dynamism and verifiability even increases the overall security of the secret sharing scheme when it is practically instantiated. If messages are distributed for a long period of time they are prone to *mobile adversaries* [13]. Given enough time a mobile adversary is able to collect enough shares to reconstruct the secret, e.g. by breaking into many servers storing shares or bribing a sufficient amount of former employees holding shares. Thus, to provide long-term security it is necessary to renew the shares from time to time and this is possible due to our Reset algorithm. In addition, the fact that our dynamic hierarchical secret sharing scheme is also verifiable ensures protection of shareholders from a malicious dealer and vice versa.

With respect to efficiency, the polynomial $f(x)$ is retrieved computing the value for each coefficient (see Section 4). However, in the secret sharing framework the only coefficient that matters is the free (last) coefficient for conjunctive (disjunctive) secret sharing. Therefore for message reconstruction only two determinants have to be computed. This leads to a complexity of $\mathcal{O}(t^3)$ for matrix A of dimension $t \times t$ in case the LU decomposition technique is used [1].

6 Conclusion and Future Work

In this work we introduced a framework for dynamic secret sharing and presented the first dynamic and verifiable secret sharing scheme based on Birkhoff interpolation. For future work, we would like to combine this technique with our solution to allow for distributed computations on secretly shared data.

References

1. Agarwal, M., Mehr, R.: Review of matrix decomposition techniques for signal processing applications. Int. Journal of Engineering Research and Applications 4(1), 90–93 (2014), www.ijera.com
2. Backes, M., Kate, A., Patra, A.: Computational verifiable secret sharing revisited. In: Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. pp. 590–609 (2011), http://dx.doi.org/10.1007/978-3-642-25385-0_32
3. Baron, J., Defrawy, K.E., Lampkins, J., Ostrovsky, R.: Communication-optimal proactive secret sharing for dynamic groups. In: Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers. pp. 23–41 (2015), http://dx.doi.org/10.1007/978-3-319-28166-7_2
4. Blundo, C., Cresti, A., Santis, A.D., Vaccaro, U.: Fully dynamic secret sharing schemes. In: Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. pp. 110–125 (1993), http://dx.doi.org/10.1007/3-540-48329-2_10
5. Brickell, E.F.: Some ideal secret sharing schemes. In: Advances in Cryptology EUROCRYPT'89. pp. 468–475. Springer (1990)
6. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In: 26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985. pp. 383–395 (1985), http://dx.doi.org/10.1109/SFCS.1985.64
7. Doganay, M.C., Pedersen, T.B., Saygin, Y., Savaş, E., Levi, A.: Distributed privacy preserving k-means clustering with additive secret sharing. In: Proceedings of the 2008 international workshop on Privacy and anonymity in information society. pp. 3–11. ACM (2008)
8. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: Foundations of Computer Science, 1987., 28th Annual Symposium on. pp. 427–438. IEEE (1987)
9. Fitzi, M., Garay, J.A., Gollakota, S., Rangan, C.P., Srinathan, K.: Round-optimal and efficient verifiable secret sharing. In: Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings. pp. 329–342 (2006), http://dx.doi.org/10.1007/11681878_17
10. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece. pp. 580–589 (2001), http://doi.acm.org/10.1145/380752.380853
11. Ghodosi, H., Pieprzyk, J., Safavi-Naini, R.: Secret sharing in multilevel and compartmented groups. In: Information Security and Privacy. pp. 367–378. Springer (1998)
12. Gupta, V., Gopinath, K.: $\mathrm{G}_{\mathrm{its}}{}^{2}$ VSR: : An information theoretical secure verifiable secret redistribution protocol for long-term archival storage. In: Security in Storage Workshop, 2007. SISW'07. Fourth International IEEE. pp. 22–33. IEEE (2007)
13. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive secret sharing or: How to cope with perpetual leakage. In: Advances in Cryptology CRYPTO'95, pp. 339–352. Springer (1995)

14. Katz, J., Koo, C., Kumaresan, R.: Improving the round complexity of VSS in point-to-point networks. Inf. Comput. 207(8), 889–899 (2009), http://dx.doi.org/10.1016/j.ic.2009.03.007
15. Kothari, S.C.: Generalized linear threshold scheme. In: Advances in Cryptology. pp. 231–241. Springer (1985)
16. Nojoumian, M., Stinson, D.R., Grainger, M.: Unconditionally secure social secret sharing scheme. Information Security, IET 4(4), 202–211 (2010)
17. Pakniat, N., Eslami, Z., Nojoumian, M.: Ideal social secret sharing using Birkhoff interpolation method. IACR Cryptology ePrint Archive 2014, 515 (2014), http://eprint.iacr.org/2014/515
18. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Advances in Cryptology CRYPTO'91. pp. 129–140. Springer (1992)
19. Schultz, D.A., Liskov, B., Liskov, M.: MPSS: mobile proactive secret sharing. ACM Trans. Inf. Syst. Secur. 13(4), 34 (2010), http://doi.acm.org/10.1145/1880022.1880028
20. Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (Nov 1979), http://doi.acm.org/10.1145/359168.359176
21. Simmons, G.J.: How to (really) share a secret. In: Proceedings on Advances in cryptology. pp. 390–448. Springer-Verlag New York, Inc. (1990)
22. Tassa, T.: Hierarchical threshold secret sharing. Journal of Cryptology 20(2), 237–264 (2007)

Appendix

A Requirements for Birkhoff interpolation matrices Intepolation

In this section the necessary requirements and a sufficient condition for the interpolation matrix E are presented, such that the corresponding Birkhoff interpolation problem is well posed. For the corresponding proofs we refer to [22].

Lemma 1. *Let $A \subset S$ be an authorized subset of shareholders, i.e. $A \in \Gamma$, and E the corresponding interpolation matrix, where the entries $e_{i,j}$ of the matrix E satisfy the following condition:*

$$\sum_{j=0}^{k}\sum_{i=1}^{r} e_{i,j} \geq k+1, \quad 0 \leq k \leq d, \tag{1}$$

where d is the highest derivative order in the problem and $r := |A|$ is the number of interpolating points.

Before providing the sufficient condition (Theorem 3), the following definition is needed.

Definition 6 ([22]). *In the interpolation matrix E a* 1-sequence *is a maximal run of consecutive* 1*s in a row of the matrix E itself. Namely, it is a triplet of the form (i, j_0, j_1) where $1 \leq i \leq r$ and $0 \leq j_0 \leq j_1 \leq d$, such that $e_{i,j} = 1$ for all $j_0 \leq j \leq j_1$, while $e_{i,j_0-1} = e_{i,j_1+1} = 0$. A 1-sequence (i, j_0, j_1) is called* supported *if E has 1s both to the northwest and southwest of the leading entry in the sequence, i.e. there exist indexes nw and sw, where $i_{nw} < i < i_{sw}$ and $j_{nw}, j_{sw} < j_0$ such that $e_{i_{nw},j_{nw}} = e_{i_{sw},j_{sw}} = 1$.*

Theorem 3. *The interpolation Birkhoff problem for an authorized subset A and the corresponding interpolation matrix E has a unique solution, if the interpolation matrix E satisfies (1) and contains no supported 1-sequence of odd length.*

In case the Birkhoff interpolation problem is instantiated over a finite field $\mathbb{F}_q$ with $q > 0$ a prime number, then also the following condition has to hold.

Theorem 4. *The Birkhoff interpolation problem for an interpolation matrix E has a unique solution over the finite field $\mathbb{F}_q$, if Theorem 3 holds and in addition also the following inequality is satisfied:*

$$q > 2^{-d+2} \cdot (d-1)^{\frac{(d-1)}{2}} \cdot (d-1)! \cdot x_r^{\frac{(d-1)(d-2)}{2}}, \tag{2}$$

where d is the highest derivative order of the problem.

B Security Analysis

Conjunctive secret sharing has been introduced by Tassa in [22] and it has been proven ideal, perfect secure, and accessible. We argue that the algorithms Add and Reset we introduced enhance the protocol and do not affect the properties and the security of the original conjunctive secret sharing scheme. To prove that, we first provide a high level idea of the proof of perfect security and accessibility of Tassa's conjunctive secret sharing scheme. Then, we show that our dynamic hierarchical secret sharing scheme maintains perfect security and accessibility. Furthermore, it is possible to cope with malicious dealers and shareholders including a verification protocol to the algorithm Share, Add, Reset, and Reconstruct. If Pedersen commitments are used in the verification protocol unconditional hidingness is maintained while bindingness can only be achieved computationally. Feldmann commitments instead ensure unconditional bindingness, i.e. the correctness of the shares can be guaranteed, but at he expenses of providing only computational hidingness for the shares. Thus, the latter solution is not suitable if data is processed for which long-term or even everlasting confidentiality is required. Similarly, it can be proven that Add and Reset maintain also the same properties of disjunctive secret sharing. However, for readability in the following we focus on conjunctive secret sharing only.

Roughly speaking, reconstructing a distributed message is equal to finding a solution of the Birkhoff interpolation problem for a polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$. Thus, Tassa proved the security of his approach by showing that authorized sets of shareholders $A \in \mathit{\Gamma}$ lead to interpolation matrices E for which the Birkhoff interpolation problem is well posed. Thus, accessibility is provided. Furthermore, any unauthorized set of shareholders $U \notin \mathit{\Gamma}$ leads to an unsolvable system and perfect security is therefore proven.

The introduction of the protocols Add and Reset making the Birkhoff interpolation based secret sharing scheme dynamic does not affect these properties. First, we show that accessibility and perfect security is provided if all shareholders act honestly. This corresponds to the setup of Tassa's security proof. Second, we prove that our scheme even provides verifiability, i.e. can cope with malicious dealers and shareholders.

Theorem 5. *The dynamic secret sharing scheme composed of the protocols* Share*,* Add*,* Reset*, and* Reconstruct *described in Section 5.2 is accessible and perfectly secure according to Definition 2.*

Proof. The proof for the algorithms Share and Reconstruct follows from Tassa's security proof. The algorithms Add and Reset are discussed individually in the following.

Add If the shareholders follow the protocol correctly, then all shareholders, meaning the old set of shareholders together with the new shareholder, only hold shares of the polynomial $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$ or of one of its derivatives. This prevents unauthorized subsets from reconstructing the message, meaning that perfect security is achieved. However, the share $\sigma_{i',j'}$ for the new shareholder $s_{i',j'}$ is generated by old shareholders in distributed fashion. More precisely, each old shareholder uses its share to generate a piece of information from which the new shareholder $s_{i',j'}$ can compute its own share $\sigma_{i',j'}$. Therefore, what is left to show is that no information about the other shares is leaked during the generation of the share $\sigma_{i',j'}$. To compute the share of a new shareholder $s_{i',j'}$ each shareholder $s_l \in A$ of an authorized subset $A \in \Gamma$ computes $f_l^{j'}(i')$, where $f_l^{j'}(x)$ is the j'-th derivative of the polynomial $f_l(x)$. Note that this value leaks information about the share of s_l, since $f_l^{j'}(i') = \sigma_l \sum_{k=j'}^{t-1} \frac{k!}{(k-j')!} \frac{(-1)^{l-1+k} \det(A_{l-1,k}(E,X,\varphi))}{\det(A(E,X,\varphi))} i'^{k-j'}$ and the latter part $\sum_{k=j'}^{t-1} \frac{k!}{(k-j')!} \frac{(-1)^{l-1+k} \det(A_{l-1,k}(E,X,\varphi))}{\det(A(E,X,\varphi))} i'^{k-j'}$ can be computed from public information. Thus, it generates shares to this value using an additive secret sharing scheme [7], i.e. computes $f_l^{j'}(i') = \sum_{k,s_k \in A} \delta_{k,l}$, and sends $\delta_{k,l}$ to shareholder $s_k \in A$. Each shareholder s_l then adds all subshares received by the other shareholders, i.e. $\delta_l = \sum_{k,s_k \in A} \delta_{l,k}$, and forwards only the result δ_l to the new shareholder. Due to the use of the additive secret sharing scheme perfect security of all shares remains preserved.
Since $\sum_{l,s_l \in A} \delta_l = \sum_{l,s_l \in A} \sum_{k,s_k \in A} \delta_{k,l} = \sum_{k,s_k \in A} f_l^{j'}(i') = f^{j'}(i')$ also accessibility is provided. This ensures that the new shareholder holds together with the other shareholders a point of polynomial $f(x)$ or of one of its derivatives and the shares of authorized subsets including the new shareholders can reconstruct the message.

Reset In this algorithm each shareholder $s_l \in A$ of an authorized subset $A \in \Gamma$ uses hierarchical secret sharing to distribute its share to a new set of shareholders. More precisely, it computes its partial Birkhoff interpolation coefficient

$$a_{l,0} := \sigma_l (-1)^{l-1} \frac{\det(A_{l-1,0}(E,X,\varphi))}{\det(A(E,X,\varphi))}$$

of coefficient a_0 and then chooses a polynomial $f'_l(x) = a'_{l,0} + a'_{l,1}x + a'_{l,2}x^2 + \cdots + a'_{l,t'-1}x^{t'-1}$, where $a'_{l,0} = a_{l,0}$, containing this value in the free coefficient. In this way, shares of shares are sent to the new shareholders, since only one point of this polynomial or of one of its derivatives is sent. Therefore, perfect security follows from the perfect security of conjunctive secret sharing. Furthermore, it computes the value to be sent to a new shareholder in accordance to the new access structure and the IDs assigned to each new shareholder. Thus, any unauthorized subset $U \notin \Gamma$ cannot reconstruct the message and perfect security is provided.
Accessibility of this protocol is provided due to the homomorphic property of polynomials. More precisely each new shareholder $s_{i,j}$ receives from each old shareholder s_l share $f'^j_l(i)$ of polynomial $f'_l(x) = a'_{l,0} + a'_{l,1}x + a'_{l,2}x^2 + \cdots + a'_{l,t'-1}x^{t'-1}$, where $a'_{l,0} = a_{l,0}$ is the partial Birkhoff interpolation coefficient of a_0. Since the new shareholder adds all shares received to compute its new share it follows that it holds a point of polynomial $f'(x) = \sum_{l,s_l \in A} f'_l(x) = \sum_{l,s_l \in A}(a'_{l,0} + a'_{l,1}x + \cdots + a'_{l,t'-1}x^{t'-1}) = \sum_{l,s_l \in A} a'_{l,0} + \sum_{l,s_l \in A} a'_{l,1} + \cdots + \sum_{l,s_l \in A} a'_{l,t'-1}x^{t'-1} = a_0 + \sum_{l,s_l \in A} a'_{l,1} + \cdots + \sum_{l,s_l \in A} a'_{l,t'-1}x^{t'-1}$ or of one of its derivatives. So the free coefficient of $f'(x)$ is still a_0, meaning that any authorized subset of the new access structure is still able to retrieve message $a_0 = m$.

Next we show that our verifiable and dynamic hierarchical secret sharing scheme indeed provides verifiability. For this we assume a majority of trustworthy shareholders within an authorized subset. This assumption can be weakened by letting all shareholders participate during the Add and Reset algorithm and choose an authorized subset among the majority. This majority can be identified during Add by checking who reports the same set of commitments to function $f(x)$ and during Reset by checking who reported the same commitments c_0 to the free coefficient of $f(x)$. Note that the presence of a majority of trustworthy shareholders is a common assumption of classical secret sharing schemes that allow to reset access structures, e.g. [12].

Theorem 6. *In the presence of a majority of trustworthy shareholders within an authorized subset the verifiable and dynamic secret sharing scheme composed of the protocols* Share*,* Add*,* Reset*, and* Reconstruct *described in Section 5.2 is a verifiable secret sharing scheme according to Definition 3.*

Proof. To prove that each authorized subset of shareholders $A \in \Gamma$ reconstruct the same message $a_0 = m$ each shareholder must hold a point of the to-be-found polynomial $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ or of one of its derivatives. Furthermore, each shareholder must hold the point assigned to its ID $(i, j) \in \mathcal{I} \times \mathcal{I}$, i.e. must receive share $\sigma_{i,j} = f^j(i)$, where $f^j(x)$ is the j-th derivative of the polynomial $f(x)$. In the following we show for each algorithm that generates shares, i.e. Share, Add, and Reset, that the shareholders receiving these shares are able to verify these conditions.

Share During this algorithm the dealer commits to each coefficient a_k of $f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ by computing a commitment $c_k := g^{a_k} \mod p$, for $k = 0, \ldots, t-1$. It broadcasts the commitments and sends each share $\sigma_{i,j}$ to shareholder $s_{i,j} \in L_h$, for $i = 1, \ldots, n_h$ and $h = 0, \ldots, \ell$. If shareholder $s_{i,j}$ accepts $\sigma_{i,j}$ then the following equation holds

$$g^{\sigma_{i,j}} \equiv \prod_{k=j}^{t-1} c_k^{\frac{k!}{(k-j)!} i^{k-j}} = g^{f^j(i)}.$$

From this it follows directly that incorrect shares can be detected and rejected.

Add During this algorithm the shareholders $s_l \in A$ of an authorized subset $A \in \Gamma$ compute share $\sigma_{i',j'}$ for a new shareholder $s_{i',j'} \in S$ in distributed fashion. Furthermore, each shareholder broadcasts the commitments to the coefficients $c_k := g^{a_k} \mod p$, for $k = 0, \ldots, t-1$ received from the dealer. Under the assumption that at least a majority of these shareholders is honest the new shareholder has access to a correct set of commitments and can verify whether

$$g^{\sigma_{i',j'}} \equiv \prod_{k=j'}^{t-1} c_k^{\frac{k!}{(k-j')!} i'^{k-j'}} = g^{f^{j'}(i')}.$$

From this it follows directly that incorrect shares can be detected and rejected.

Reset During this algorithm the shareholders $s_l \in A$ of an authorized subset $A \in \Gamma$ compute shares for a set of new shareholders $S' = \{s'_1, \ldots, s'_{n'}\}$, each accompanied with a unique ID $(i', j') \in \mathcal{I} \times \mathcal{I}$, and an access structure $\Gamma' \subset \mathcal{P}(S')$. Like for the other algorithms it has to be checked that share $\sigma_{i',j'}$ for the shareholder $s'_{i',j'} \in S'$ with ID $(i', j') \in \mathcal{I} \times \mathcal{I}$ are computed as $f'^{j'}(i')$. However, this algorithm has an additional requirement for correctness. The free coefficient of the to-be-found polynomial must be equal to the message m distributed by the dealer. To verify the first condition each shareholder $s_{i',j'}$ of the new access structure checks

$$g^{\sigma_{l,i',j'}} \equiv \prod_{k=j'}^{t'-1} c'^{\frac{k!}{(k-j')!} i'^{k-j'}}_{l,k} = g^{f_l'^{j'}(i')}, \text{ for } s_l \in A,$$

for each share $\sigma_{l,i',j'}$ received from shareholder l of the old set of shareholders. Finally, it checks that the sum of all shares is a point of a polynomial with free coefficient $a_0 = m$. This can be verified by multiplying all commitments to the individual free coefficients, i.e.

$$c_0 \equiv \prod_{l, s_l \in A} c'_{l,0} = \prod_{l, s_l \in A} g^{a_{l,0}} = g^{a_0} = g^m.$$

Under the assumption that a majority of the old shareholders sent the correct commitments incorrect shares can be detected.

Note that our scheme is also ideal. This clearly comes from the fact that each shareholder $s_i \in R$ receives a share $\sigma_{i,j} \in \mathbb{F}_q$ that is a field element of the same field as the message $m \in \mathbb{F}_q$.

C Example of Tassa's Hierarchical Secret Sharing

In the following, an example explaining how Tassa's hierarchical secret sharing scheme [22] works is provided. More precisely, we show a numerical instantiation of the algorithms Share and Reconstruct described in Definition 5 for conjunctive secret sharing. Note that we shall perform all computations assuming a finite field $\mathbb{F}_q$ for a very large prime q. Thus, we do not perform the modulo operation assuming the values computed are always smaller than q.

Share Let us assume a hierarchy composed of three levels L_0, L_1, L_2 (where L_0 is the highest level and L_2 is the lowest level) and thresholds $t_1 = 1, t_2 = 2, t_3 = 3$. Furthermore, let us assume the set S is composed of $n = 6$ shareholders. More precisely, one shareholder $s_{1,0}$ is assigned to level L_0, two shareholders $s_{1,1}, s_{2,1}$ are assigned to level L_1, and three shareholders $s_{1,2}, s_{2,2}$, and $s_{3,2}$ are assigned to level L_2. Finally, let us assume that a dealer wants to secretly share the message $m := 2$. Denoted $t := t_3$, the dealer selects a polynomial $f(x) = a_0 + a_1 x + a_2 x^2$ of degree $t - 1$ setting $a_0 := 2$ and choosing the remaining two coefficients a_1, a_2 uniformly at random., e.g. $a_1 = 3, a_2 = 1$, and $f(x) = 2+3x+x^2$. The shares are computed as points over $f(x)$ or one of its derivatives $f'(x) = 3+2x$ or $f''(x) = 2$. With respect to level L_0 shareholder $s_{1,0}$ gets share $\sigma_{1,0} = f(1) = 6$. With respect to level L_1 shareholder $s_{1,1}$ gets share $\sigma_{1,1} = f'(1) = 5$ and shareholder $s_{2,1}$ gets share $\sigma_{2,1} = f'(2) = 7$. With respect to level L_2 shareholder $s_{1,2}$ gets share $\sigma_{1,2} = f''(1) = 2$, shareholder $s_{2,2}$ gets share $\sigma_{2,2} = f''(2) = 2$, and $s_{3,2}$ gets share $\sigma_{3,2} = f''(3) = 2$.

Reconstruct For conjunctive secret sharing, the thresholds $0 < t_0 < t_1 < t_2$ have to be considered as a chain. More precisely, the access structure defined is such that the message can be retrieved if at least $t_2 = 3$ shareholders in total collaborate, at least $t_1 = 2$ of them belong to level L_1 or L_0, and at least $t_0 = 1$ of them belong to level L_0. Without loss of generality, let us assume that the shareholders collaborating are $s_{1,0}, s_{2,1}$, and $s_{3,2}$. The access structure is satisfied because the corresponding interpolation matrix

$$E = \begin{pmatrix} 1\;0\;0 \\ 0\;1\;0 \\ 0\;0\;1 \end{pmatrix}$$

leads to a Birkhoff interpolation problem with unique solution (see Appendix A). The message $m = 2$ can be retrieved as follows:

1. the set containing the coordinates of E in lexicographic order is $I(E) = \{(1,0),(2,1),(3,2)\}$ and the column containing the shares in lexicographic order is $(6,7,2)^t$;
2. the vector of the functions involved is $\varphi = \{1, x, x^2\}$;

3. the matrices involved in the Birkhoff's reconstruction formula are:

$$A(E,X,\varphi) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 4 \\ 0 & 0 & 2 \end{pmatrix}, \qquad A(E,X,\varphi_0) = \begin{pmatrix} 6 & 1 & 1 \\ 7 & 1 & 4 \\ 2 & 0 & 2 \end{pmatrix},$$

$$A(E,X,\varphi_1) = \begin{pmatrix} 1 & 6 & 1 \\ 0 & 7 & 4 \\ 0 & 2 & 2 \end{pmatrix}, \qquad A(E,X,\varphi_2) = \begin{pmatrix} 1 & 1 & 6 \\ 0 & 1 & 7 \\ 0 & 0 & 2 \end{pmatrix};$$

4. the determinants are $\det(A(E,X,\varphi)) = 2, \det(A(E,X,\varphi_0)) = 4$, $\det(A(E,X,\varphi_1)) = 6$ and $\det(A(E,X,\varphi_2)) = 2$, respectively;
5. applying Birkhoff's reconstruction formula the coefficients a_0, a_1, a_2 of polynomial $f(x)$ are computed as:

$$a_0 = \frac{\det(A(E,X,\varphi_0))}{\det(A(E,X,\varphi))} = \frac{4}{2} = 2, a_1 = \frac{\det(A(E,X,\varphi_1))}{\det(A(E,X,\varphi))} = \frac{6}{2} = 3,$$

$$a_2 = \frac{\det(A(E,X,\varphi_2))}{\det(A(E,X,\varphi))} = \frac{2}{2} = 1;$$

6. the polynomial reconstructed is exactly $f(x) = 2 + 3x + x^2$ and the secret is retrieved as $f(0) = a_0 = 2$.