# Continuously Non-Malleable Secret Sharing for General Access Structures

Gianluca Brian[1], Antonio Faonio[2], and Daniele Venturi[1]

[1] Department of Computer Science, Sapienza University of Rome, Rome, Italy[⋆]
[2] IMDEA Software Institute, Madrid, Spain[⋆⋆]

**Abstract.** We study leakage-resilient continuously non-malleable secret sharing, as recently introduced by Faonio and Venturi (CRYPTO 2019). In this setting, an attacker can continuously tamper and leak from a target secret sharing of some message, with the goal of producing a modified set of shares that reconstructs to a message related to the originally shared value. Our contributions are two fold.

– In the plain model, assuming one-to-one one-way functions, we show how to obtain noisy-leakage-resilient continuous non-malleability for arbitrary access structures, in case the attacker can continuously leak from and tamper with all of the shares independently.
– In the common reference string model, we show how to obtain a new flavor of security which we dub bounded-leakage-resilient continuous non-malleability under selective $k$-partitioning. In this model, the attacker is allowed to partition the target $n$ shares into any number of non-overlapping blocks of maximal size $k$, and then can continuously leak from and tamper with the shares within each block jointly. Our construction works for arbitrary access structures, and assuming (doubly enhanced) trapdoor permutations and collision-resistant hash functions, we achieve a concrete instantiation for $k \in O(\log n)$.

Prior to our work, there was no secret sharing scheme achieving continuous non-malleability against joint tampering, and the only known scheme for independent tampering was tailored to threshold access structures.

**Keywords:** Secret sharing · Non-malleability · Leakage resilience.

[⋆] The first and third authors were supported in part by the research projects "PRIvacy-preserving, Security, and MAchine-learning techniques for healthcare applications (PRISMA)" and "Protect yourself and your data when using social networks", both funded by Sapienza University of Rome, and in part by the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Computer Science Department of Sapienza University of Rome.

[⋆⋆] The second author is supported by the Spanish Government through the projects Datamantium (ref. RTC-2016-4930-7), SCUM (RTI2018-102043-B-I00), and ERC2018-092822, and by the Madrid Regional Government under project BLOQUES (ref. S2018/TCS-4339).

# 1 Introduction

A non-malleable secret sharing for an access structure $\mathcal{A}$ over $n$ parties allows to share a secret message $m$ into $n$ shares $s = (s_1, \ldots, s_n)$, in such a way that the following properties are guaranteed.

**Privacy:** No attacker given the shares belonging to an arbitrary unauthorized subset $\mathcal{U} \notin \mathcal{A}$ of the players can infer any information on $m$.

**Non-malleability:** No attacker tampering with all of the shares via some function $f \in \mathcal{F}$ within some family of allowed[1] modifications can generate a mauled secret sharing $\tilde{s} = f(s)$ that reconstructs to $\tilde{m} \neq m$ related to $m$.

Sometimes, non-malleability is considered together with leakage resilience. This means that the attacker can additionally leak partial information $g(s)$ from all of the shares (via functions $g \in \mathcal{G}$) before launching a tampering attack. Leakage resilience typically comes in one of two flavors: *bounded leakage* (i.e,. there is a fixed upper bound on the maximum amount of information retrieved from the shares) or *noisy leakage* (i.e., the length of the retrieved information is arbitrary as long as it does not decrease the entropy of the shares by too much).

In this work we focus on leakage-resilient *continuous* non-malleability with *adaptive concurrent reconstruction*, as recently introduced by Faonio and Venturi [19].[2] Here, the attacker can (leak from and) tamper poly-many times with a target secret sharing using functions $f^{(q)} \in \mathcal{F}$ as above, and for each tampering query $q$ it can also choose adaptively the reconstruction set $\mathcal{T}^{(q)} \in \mathcal{A}$ used to determine the reconstructed message. There are only two limitations: First, the attacker is computationally bounded; second, the experiment stops (we say it "self-destructs") after the first tampering query yielding an invalid set of shares. Both limitations are *inherent* for continuous non-malleability [20,6,19].

The only known scheme achieving such a strong flavor of non-malleability is the one by Faonio and Venturi, which tolerates the families $\mathcal{F}$ and $\mathcal{G}$ of *independent tampering/leakage*, i.e. for each query $q$ we have $f^{(q)} = (f_1^{(q)}, \ldots, f_n^{(q)}) \in \mathcal{F}$ where $f_i^{(q)}$ gets as input the $i$-th share (and similarly $g^{(q)} = (g_1^{(q)}, \ldots, g_n^{(q)}) \in \mathcal{G}$). The access structure $\mathcal{A}$ supported by their construction is the $\tau$-threshold access structure—i.e., any subset of at most $\tau$ players has no information about the message—with the caveat that reconstruction works with at least $\tau + 2$ shares, namely a *ramp* secret sharing, thus leaving a minimal gap between the reconstruction and privacy threshold. The following natural question arise:

*Problem 1.* Can we obtain leakage-resilient continuously non-malleable secret sharing against independent leakage/tampering, for general access structures?

Another open question is whether leakage-resilient continuous non-malleability is achievable for stronger tampering and leakage families $\mathcal{F}, \mathcal{G}$, e.g. in case the attacker can leak from and manipulate subsets of the shares jointly.

[1] It is easy to see that non-malleability is impossible for arbitrary (polynomial-time) tampering.

[2] From now on, we omit to explicitly mention the feature of adaptive concurrent reconstruction and simply talk about continuous non-malleability.

*Problem 2.* Can we obtain leakage-resilient continuously non-malleable secret sharing against joint leakage/tampering?

### 1.1 Our Contributions

We make significant progress towards solving the above problems. In particular, our first contribution is a positive answer to Problem 1:

**Theorem 1 (Informal).** *Assuming one-to-one one-way functions, for any access structure $\mathcal{A}$ over $n$ parties there exists a noisy-leakage-resilient continuously non-malleable secret sharing scheme realizing $\mathcal{A}$ against independent leakage and tampering, in the plain model.*

Our second contribution is a positive answer to Problem 2 assuming trusted setup, in the form of a common reference string (CRS). More in details, we put forward a new security notion for secret sharing dubbed continuous non-malleability under *selective $k$-partitioning*. This roughly means that the attacker, after seeing the CRS, must commit to a partition of the set $[n]$ into $\beta$ (non-overlapping) blocks $(\mathcal{B}_1, \ldots, \mathcal{B}_\beta)$ of size at most $k$; hence, the adversary can jointly, and continuously, tamper with and leak from each collection $s_{\mathcal{B}_i}$ of the shares.[3]

**Theorem 2 (Informal).** *Assuming (doubly-enhanced) trapdoor permutations and collision-resistant hash functions, for any access structure $\mathcal{A}$ over $n$ parties there exists a bounded-leakage-resilient continuously non-malleable secret sharing scheme realizing $\mathcal{A}$ against selective $O(\log n)$-joint leakage and tampering in the CRS model.*

Prior to our work, we had secret sharing schemes unconditionally achieving security either against joint leakage [29] or joint tampering [23,24], but nothing was known for both even in the much simpler case of one-time non-malleability.

### 1.2 Related Work

Non-malleable secret sharing was introduced by Goyal and Kumar [23]. For any $\tau \leq n$, they showed how to realize $\tau$-threshold access structures, against one-time tampering with either all of the shares independently, or jointly after partitioning the players into two non-overlapping blocks of size at most[4] $\tau - 1$. In a subsequent work [24], the same authors show how to extend the result for independent tampering to the case of arbitrary access structures; additionally, for the case of joint tampering, they provide a new scheme realizing the $n$-threshold access structure (i.e., an $n$-out-of-$n$ secret sharing) in a stronger model where

[3] The only restriction is that no block in the partition can contain an authorized set of players, otherwise trivial attacks are possible.

[4] An additional (artificial) requirement is that the size of the two blocks must be different in order for their technique to work.

the attacker can partition the players into two possibly overlapping blocks of size at most $n-1$. Srinivasan and Vasudevan [36] built the first non-malleable secret sharing schemes for general access structures against independent tampering, with non-zero rate[5] (in fact, even constant rate in case of threshold access structures). Chattopadhyay *et al.* [9] construct non-malleable secret sharing for threshold access structures, against affine tampering composed with joint split-state tampering. Lin *et al.* [31] consider non-malleability against affine tampering in an adaptive setting where the adversary gets to see an unauthorized subset of the shares before launching a single tampering attack.

Badrinarayanan and Srinivasan [6] generalize non-malleability to $p$-time tampering attacks, where $p$ is an a-priori upper bound on the number of tampering queries the adversary can ask. For each attempt, however, the reconstruction set $\mathcal{T}$ must be chosen in advance at the beginning of the experiment. In this model, they show how to realize arbitrary access structures against independent tampering with all of the shares. Aggarwal *et al.* [2] were the first to consider $p$-time non-malleability under *non-adaptive* concurrent reconstruction, i.e. the attacker now can specify a different reconstruction set $\mathcal{T}^{(q)}$ during the $q$-th tampering query, although the sequence of sets $\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(p)}$ must be chosen non-adaptively. Kumar, Meka, and Sahai [29] pioneered bounded-leakage-resilient one-time non-malleable secret sharing for general access structures, against independent leakage and tampering with all of the shares.

In the special case of 2-threshold access structures over $n = 2$ parties, the notion of (leakage-resilient) non-malleable secret sharing collapses to that of split-state (leakage-resilient) non-malleable codes [17,32,15,4,10,20,5,3,1,30,18,34,11].

**Organization.** All of our constructions rely on standard cryptographic primitives, which we recall in §2 (together with some basic notation). The new model of continuous tampering under selective partitioning is presented in §3.

Our main constructions appear in §4 (for joint tampering in the CRS model) and §5–§6 (for independent tampering in the plain model), respectively; there, we also explain how to instantiate these constructions with concrete building blocks, thus establishing Thm. 1 and Thm. 2. Finally, in §7, we conclude the paper with a list of open problems and interesting directions for further research.

## 2 Standard Definitions

*Basic notation.* For a string $x$, we denote its length by $|x|$; if $\mathcal{X}$ is a set, $|\mathcal{X}|$ represents the number of elements in $\mathcal{X}$. When $x$ is chosen randomly in $\mathcal{X}$, we write $x \leftarrow\!\!\$\; \mathcal{X}$. When $\mathsf{A}$ is a randomized algorithm, we write $y \leftarrow\!\!\$\; \mathsf{A}(x)$ to denote a run of $\mathsf{A}$ on input $x$ (and implicit random coins $r$) and output $y$; the value $y$ is a random variable, and $\mathsf{A}(x; r)$ denotes a run of $\mathsf{A}$ on input $x$ and randomness $r$. An algorithm $\mathsf{A}$ is *probabilistic polynomial-time* (PPT) if $\mathsf{A}$ is randomized and for

[5] The rate refers to the asymptotic ratio between the maximal length of a share and that of the message.

any input $x, r \in \{0,1\}^*$ the computation of $\mathsf{A}(x;r)$ terminates in a polynomial number of steps (in the size of the input).

*Negligible functions.* We denote with $\lambda \in \mathbb{N}$ the security parameter. A function $p$ is a polynomial, denoted $p(\lambda) \in \mathtt{poly}(\lambda)$, if $p(\lambda) \in O(\lambda^c)$ for some constant $c > 0$. A function $\nu : \mathbb{N} \to [0,1]$ is negligible in the security parameter (or simply negligible) if it vanishes faster than the inverse of any polynomial in $\lambda$, i.e. $\nu(\lambda) \in O(1/p(\lambda))$ for all positive polynomials $p(\lambda)$. We often write $\nu(\lambda) \in \mathtt{negl}(\lambda)$ to denote that $\nu(\lambda)$ is negligible.

Unless stated otherwise, throughout the paper, we implicitly assume that the security parameter is given as input (in unary) to all algorithms.

*Random variables.* For a random variable $\mathbf{X}$, we write $\mathbb{P}[\mathbf{X} = x]$ for the probability that $\mathbf{X}$ takes on a particular value $x \in \mathcal{X}$ (with $\mathcal{X}$ being the set where $\mathbf{X}$ is defined). The statistical distance between two random variables $\mathbf{X}$ and $\mathbf{X}'$ defined over the same set $\mathcal{X}$ is defined as $\mathbb{SD}(\mathbf{X}; \mathbf{X}') = \frac{1}{2}\sum_{x \in \mathcal{X}} |\mathbb{P}[\mathbf{X} = x] - \mathbb{P}[\mathbf{X}' = x]|$.

Given two ensembles $\mathbf{X} = \{\mathbf{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\mathbf{Y} = \{\mathbf{Y}_\lambda\}_{\lambda \in \mathbb{N}}$, we write $\mathbf{X} \equiv \mathbf{Y}$ to denote that they are identically distributed, $\mathbf{X} \approx_s \mathbf{Y}$ to denote that they are statistically close, i.e. $\mathbb{SD}(\mathbf{X}_\lambda; \mathbf{X}'_\lambda) \in \mathtt{negl}(\lambda)$, and $\mathbf{X} \approx_c \mathbf{Y}$ to denote that they are computationally indistinguishable, i.e., for all PPT distinguishers $\mathsf{D}$:

$$|\mathbb{P}[\mathsf{D}(\mathbf{X}_\lambda) = 1] - \mathbb{P}[\mathsf{D}(\mathbf{Y}_\lambda) = 1]| \in \mathtt{negl}(\lambda).$$

We extend the notion of computational indistinguishability to the case of interactive experiments (a.k.a. games) featuring an adversary $\mathsf{A}$. In particular, let $\mathbf{G}_{\mathsf{A}}(\lambda)$ be the random variable corresponding to the output of $\mathsf{A}$ at the end of the experiment, where wlog. we may assume $\mathsf{A}$ outputs a decision bit. Given two experiments $\mathbf{G}_{\mathsf{A}}(\lambda, 0)$ and $\mathbf{G}_{\mathsf{A}}(\lambda, 1)$, we write $\{\mathbf{G}_{\mathsf{A}}(\lambda, 0)\}_{\lambda \in \mathbb{N}} \approx_c \{\mathbf{G}_{\mathsf{A}}(\lambda, 1)\}_{\lambda \in \mathbb{N}}$ as a shorthand for

$$|\mathbb{P}[\mathbf{G}_{\mathsf{A}}(\lambda, 0) = 1] - \mathbb{P}[\mathbf{G}_{\mathsf{A}}(\lambda, 1) = 1]| \in \mathtt{negl}(\lambda).$$

The above naturally generalizes to statistical distance (in case of unbounded adversaries). We recall a useful lemma from [16,12].

**Lemma 1 ([12], Lemma 4).** *Let $\mathcal{O}_{\mathsf{leak}}(x, g)$ be an oracle that upon input a value $x$ and a function $g$ outputs $g(x)$, and let $\mathbf{X}$ and $\mathbf{Y}$ be two independently distributed random variables. For any adversary $\mathsf{A}$, and for any value $z$, the distributions $\left(\mathbf{X} | z = \mathsf{A}^{\mathcal{O}_{\mathsf{leak}}(\mathbf{X},\cdot), \mathcal{O}_{\mathsf{leak}}(\mathbf{Y},\cdot)}\right)$ and $\left(\mathbf{Y} | z = \mathsf{A}^{\mathcal{O}_{\mathsf{leak}}(\mathbf{X},\cdot), \mathcal{O}_{\mathsf{leak}}(\mathbf{Y},\cdot)}\right)$ are independently distributed.*

*Average min-entropy.* The min-entropy of a random variable $\mathbf{X}$ with domain $\mathcal{X}$ is $\mathbb{H}_\infty(\mathbf{X}) := -\log \max_{x \in \mathcal{X}} \mathbb{P}[\mathbf{X} = x]$, and intuitively it measures the best chance to predict $\mathbf{X}$ (by a computationally unbounded algorithm). For conditional distributions, unpredictability is measured by the conditional average min-entropy [14]: $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Y}) := -\log \mathbb{E}_y[2^{-\mathbb{H}_\infty(\mathbf{X}|\mathbf{Y}=y)}]$. The lemma below is sometimes known as the "chain rule" for conditional average min-entropy.

**Lemma 2** **([14], Lemma 2.2).** *Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be random variables. If $\mathbf{Y}$ has at most $2^\ell$ possible values, then $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) \geq \widetilde{\mathbb{H}}_\infty(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) - \ell \geq \widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Z}) - \ell$. In particular, $\widetilde{\mathbb{H}}_\infty(\mathbf{X}|\mathbf{Y}) \geq \widetilde{\mathbb{H}}_\infty(\mathbf{X}, \mathbf{Y}) - \ell \geq \widetilde{\mathbb{H}}_\infty(\mathbf{X}) - \ell$.*

### 2.1 Secret Sharing Schemes

An $n$-party secret sharing scheme $\Sigma$ in the common reference string (CRS) model consists of polynomial-time algorithms $(\mathsf{Init}, \mathsf{Share}, \mathsf{Rec})$ specified as follows: (i) The randomized initialization algorithm $\mathsf{Init}$ takes as input the security parameter $1^\lambda$, and outputs a CRS $\omega \in \{0,1\}^*$; (ii) The randomized sharing algorithm $\mathsf{Share}$ takes as input a CRS $\omega \in \{0,1\}^*$ and a message $m \in \mathcal{M}$, and outputs $n$ shares $s_1, \ldots, s_n$ where each $s_i \in \mathcal{S}_i$; (iii) The deterministic algorithm $\mathsf{Rec}$ takes as input a CRS $\omega \in \{0,1\}^*$ and a certain number of candidate shares, and outputs a value in $\mathcal{M} \cup \{\bot\}$. Given $s = (s_1, \ldots, s_n)$ and a subset $\mathcal{I} \subseteq [n]$, we often write $s_\mathcal{I}$ to denote the shares $(s_i)_{i \in \mathcal{I}}$.

The subset of parties allowed to reconstruct the secrets by pulling their shares together form the so-called access structure.

**Definition 1** **(Access structure).** *We say $\mathcal{A}$ is an access structure for $n$ parties if $\mathcal{A}$ is a monotone class of subsets of $[n]$, i.e., if $\mathcal{I}_1 \in \mathcal{A}$ and $\mathcal{I}_1 \subseteq \mathcal{I}_2$, then $\mathcal{I}_2 \in \mathcal{A}$. We call sets $\mathcal{I} \in \mathcal{A}$ authorized or qualified, and unauthorized or unqualified otherwise.*

Intuitively, a secure secret sharing scheme must be such that all qualified subsets of players can efficiently reconstruct the secret, whereas all unqualified subset have no information (possibly in a computational sense) about the secret.

**Definition 2** **(Secret sharing scheme).** *Let $n \in \mathbb{N}$, and $\mathcal{A}$ be an access structure for $n$ parties. We say that $\Sigma = (\mathsf{Init}, \mathsf{Share}, \mathsf{Rec})$ is a secret sharing scheme realizing access structure $\mathcal{A}$ in the CRS model, with message space $\mathcal{M}$ and share space $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, if it is an $n$-party secret sharing in the CRS model with the following properties.*

*(i)* **Correctness:** *For all $\lambda \in \mathbb{N}$, all $\omega \in \mathsf{Init}(1^\lambda)$, all messages $m \in \mathcal{M}$, and for all subsets $\mathcal{I} \in \mathcal{A}$, we have that $\mathsf{Rec}(\omega, (\mathsf{Share}(\omega, m))_\mathcal{I}) = m$, with overwhelming probability over the randomness of the sharing algorithm.*

*(ii)* **Privacy:** *For all PPT adversaries $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$, we have*

$$\{\mathbf{Privacy}_{\Sigma,\mathsf{A}}(\lambda, 0)\}_{\lambda \in \mathbb{N}} \approx_c \{\mathbf{Privacy}_{\Sigma,\mathsf{A}}(\lambda, 1)\}_{\lambda \in \mathbb{N}},$$

*where the experiment $\mathbf{Privacy}_{\Sigma,\mathsf{A}}(\lambda, b)$ is defined by*

$$\mathbf{Privacy}_{\Sigma,\mathsf{A}}(\lambda, b) := \left\{ \begin{array}{c} \omega \leftarrow\!\!\$\ \mathsf{Init}(1^\lambda); (m_0, m_1, \mathcal{U} \not\in \mathcal{A}, \alpha_1) \leftarrow\!\!\$\ \mathsf{A}_1(\omega) \\ s \leftarrow\!\!\$\ \mathsf{Share}(\omega, m_b); b' \leftarrow\!\!\$\ \mathsf{A}_2(\alpha_1, s_\mathcal{U}) \end{array} \right\}.$$

*If the above ensembles are statistically close (resp. identically distributed), we speak of* statistical *(resp.* perfect*) privacy.*

*Moreover, we say that $\Sigma$ is a secret sharing scheme realizing access structure $\mathcal{A}$ in the plain model, if for all $\lambda \in \mathbb{N}$ algorithm* $\mathsf{Init}$ *simply returns $\omega = 1^\lambda$.*

*Remark 1.* In the plain model, the above definition of privacy is equivalent to saying that for all pairs of messages $m_0, m_1 \in \mathcal{M}$, and for all unqualified subsets $\mathcal{U} \notin \mathcal{A}$, it holds that $\{(\mathsf{Share}(1^\lambda, m_0))_\mathcal{U}\}_{\lambda \in \mathbb{N}} \approx_c \{(\mathsf{Share}(1^\lambda, m_1))_\mathcal{U}\}_{\lambda \in \mathbb{N}}$.

### 2.2 Non-Interactive Commitments

A non-interactive commitment scheme $\Pi = (\mathsf{Gen}, \mathsf{Com})$ is a pair of polynomial-time algorithms specified as follows: (i) The randomized algorithm $\mathsf{Gen}$ takes as input $1^\lambda$ and outputs a public key $pk \in \mathcal{K}$; (ii) The randomized algorithm $\mathsf{Com}$ takes as input the public key $pk$ and a message $m \in \mathcal{M}$, and outputs a commitment $c = \mathsf{Com}(pk, m; r) \in \mathcal{C}$ using random coins $r \in \mathcal{R}$. The pair $(m, r)$ is called the opening. In the plain model, we omit the algorithm $\mathsf{Gen}$ and simply set $pk = 1^\lambda$.

Intuitively, a secure commitment satisfies two properties called binding and hiding. The first property says that it is hard to open a commitment in two different ways. The second property says that a commitment hides the underlying message. The formal definitions follow.

**Definition 3 (Binding).** *We say that a non-interactive commitment scheme $\Pi = (\mathsf{Gen}, \mathsf{Com})$ is* computationally binding *if the following probability is negligible for all PPT adversaries* $\mathsf{A}$*:*

$$\mathbb{P}\left[ m_0 \neq m_1 \wedge \mathsf{Com}(pk, m_0; r_0) = \mathsf{Com}(pk, m_1; r_1) : \begin{array}{c} pk \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda) \\ (m_0, r_0, m_1, r_1) \leftarrow\!\!\$\ \mathsf{A}(pk) \end{array} \right].$$

*In case the above definition holds for all unbounded adversaries, we say that $\Pi$ is* statistically binding*. Finally, in case the above probability is exactly* 0 *(i.e., each commitment can be opened to at most a single message), then we say that $\Pi$ is* perfectly binding.

**Definition 4 (Hiding).** *We say that a non-interactive commitment scheme $\Pi = (\mathsf{Gen}, \mathsf{Com})$ is* computationally hiding *if the following holds for all PPT adversaries* $\mathsf{A}$*:*

$$\left\{ \begin{array}{r} pk \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda); (m_0, m_1, \alpha_1) \leftarrow\!\!\$\ \mathsf{A}_1(pk) \\ c \leftarrow\!\!\$\ \mathsf{Com}(pk, m_0); b' \leftarrow\!\!\$\ \mathsf{A}_2(\alpha_1, c) \end{array} \right\}$$
$$\approx_c \left\{ \begin{array}{r} pk \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda); (m_0, m_1, \alpha_1) \leftarrow\!\!\$\ \mathsf{A}_1(pk) \\ c \leftarrow\!\!\$\ \mathsf{Com}(pk, m_1); b' \leftarrow\!\!\$\ \mathsf{A}_2(\alpha_1, c) \end{array} \right\}.$$

*In case the above ensembles are statistically close (resp. identically distributed), we speak of* statistical *(resp.* perfect*) hiding.*

Note that in the plain model the above definition of hiding is equivalent to saying that for all pairs of messages $m_0, m_1 \in \mathcal{M}$ the following holds:

$$\left\{c :\ c \leftarrow\!\!\$\ \mathsf{Com}(1^\lambda, m_0)\right\}_{\lambda \in \mathbb{N}} \approx_c \left\{c :\ c \leftarrow\!\!\$\ \mathsf{Com}(1^\lambda, m_1)\right\}_{\lambda \in \mathbb{N}}.$$

### 2.3 Non-Interactive Zero Knowledge

Let $\mathsf{R}$ be a relation, corresponding to an NP language $\mathcal{L}$. A non-interactive zero-knowledge (NIZK) proof system for $\mathsf{R}$ is a tuple of efficient algorithms $\Pi = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Ver})$ specified as follows. (i) The randomized algorithm $\mathsf{CRSGen}$ takes as input the security parameter and outputs a common reference string $\omega$; (ii) The randomized algorithm $\mathsf{Prove}(\omega, \phi, (x, w))$, given $(x, w) \in \mathsf{R}$ and a label $\phi \in \{0,1\}^*$, outputs a proof $\pi$; (iii) The deterministic algorithm $\mathsf{Ver}(\omega, \phi, (x, \pi))$, given an instance $x$, a proof $\pi$, and a label $\phi \in \{0,1\}^*$, outputs either 0 (for "reject") or 1 (for "accept"). We say that a NIZK for relation $\mathsf{R}$ is *correct* if for every $\lambda \in \mathbb{N}$, all $\omega$ as output by $\mathsf{Init}(1^\lambda)$, any label $\phi \in \{0,1\}^*$, and any $(x, w) \in \mathsf{R}$, we have that $\mathsf{Ver}(\omega, \phi, (x, \mathsf{Prove}(\omega, \phi, (x, w)))) = 1$.

We define two properties of a NIZK proof system. The first property says that honest proofs do not reveal anything beyond the fact that $x \in \mathcal{L}$.

**Definition 5 (Adaptive multi-theorem zero-knowledge).** *A NIZK with labels $\Pi$ for a relation $\mathsf{R}$ satisfies adaptive multi-theorem zero-knowledge if there exists a PPT simulator $\mathsf{S} := (\mathsf{S}_0, \mathsf{S}_1)$ such that the following holds:*

*(i)* *$\mathsf{S}_0$ outputs $\omega$, a simulation trapdoor $\zeta$ and an extraction trapdoor $\xi$.*
*(ii)* *For all PPT distinguishers $\mathsf{D}$, we have that*

$$\Big|\mathbb{P}\left[\mathsf{D}^{\mathsf{Prove}(\omega,\cdot,(\cdot,\cdot))}(\omega) = 1 : \ \omega \leftarrow\!\!\$\ \mathsf{Init}(1^\lambda)\right] \\ - \mathbb{P}\left[\mathsf{D}^{\mathcal{O}_{\mathsf{sim}}(\zeta,\cdot,\cdot,\cdot)}(\omega) = 1 : \ (\omega, \zeta) \leftarrow\!\!\$\ \mathsf{S}_0(1^\lambda)\right]\Big|$$

*is negligible in $\lambda$, where the oracle $\mathcal{O}_{\mathsf{sim}}(\zeta,\cdot,\cdot,\cdot)$ takes as input a tuple $(\phi, x, w)$ and returns $\mathsf{S}_1(\zeta, \phi, x)$ iff $\mathsf{R}(x, w) = 1$ (and otherwise it returns $\perp$).*

Groth [26] introduced the concept of simulation-extractable NIZK, which informally states that knowledge soundness should hold even if the adversary can see simulated proofs for possibly false statements of its choice. For our purpose, it will suffice to consider the weaker notion of true simulation extractability, as defined by Dodis *et al.* [13].

**Definition 6 (True simulation extractability).** *Let $\Pi$ be a NIZK proof systems for a relation $\mathsf{R}$, that satisfies adaptive multi-theorem zero-knowledge w.r.t. a simulator $\mathsf{S} := (\mathsf{S}_0, \mathsf{S}_1)$. We say that $\Pi$ is* true simulation extractable *if there exists a PPT algorithm $\mathsf{K}$ such that every PPT adversary $\mathsf{A}$ has a negligible probability of winning in the following game:*

- *The challenger runs $(\omega, \zeta, \xi) \leftarrow\!\!\$\ \mathsf{S}_0(1^\lambda)$, and gives $\omega$ to $\mathsf{A}$.*
- *Adversary $\mathsf{A}$ can ask polynomially many queries of the form $(\phi, x, w)$, upon which the challenger returns $\mathsf{S}_1(\zeta, \phi, x)$ if $(x, w) \in \mathsf{R}$ and $\perp$ otherwise.*
- *Adversary $\mathsf{A}$ outputs a tuple $(\phi^*, x^*, \pi^*)$.*
- *The challenger runs $w \leftarrow\!\!\$\ \mathsf{K}(\xi, \phi^*, (x^*, \pi^*))$.*

*We say that $\mathsf{A}$ wins iff: (a) $(\phi^*, x^*)$ was not queried in the second step; (b) $\mathsf{Ver}(\omega, \phi^*, (x^*, \pi^*)) = 1$; (c) $(x^*, w) \notin \mathsf{R}$.*

## 3 Continuous Tampering under Selective Partitioning

In this section we define a new notion of non-malleability against joint memory tampering and leakage for secret sharing. Our definition generalizes the one in [19] which was tailored to threshold access structures and to independent leakage/tampering from the shares.

Very roughly, in our model the attacker is allowed to partition the set of share holders into $\beta$ (non-overlapping) blocks with size at most $k$, covering the entire set $[n]$. This is formalized through the notion of a $k$-partition.

**Definition 7 ($k$-partition).** *Let $n, k, \beta \in \mathbb{N}$. We call $\mathcal{B} = (\mathcal{B}_1, \dots, \mathcal{B}_\beta)$ a $k$-partition of $[n]$ when: (i) $\bigcup_{i=1}^{\beta} \mathcal{B}_i = [n]$; (ii) $\forall i_1, i_2 \in [\beta]$, with $i_1 \neq i_2$, we have $\mathcal{B}_{i_1} \cap \mathcal{B}_{i_2} = \emptyset$; (iii) $\forall i = 1, \dots, \beta : |\mathcal{B}_i| \leq k$.*

### 3.1 The Definition

To define non-malleability, we consider an attacker $\mathsf{A}$ playing the following game. At the beginning of the experiment, $\mathsf{A}$ chooses two messages $m_0, m_1$ possibly depending on the CRS $\omega$ of the underlying secret sharing scheme, and a $k$-partition $(\mathcal{B}_1, \dots, \mathcal{B}_\beta)$ of the set $[n]$. Hence, the adversary interacts with a target secret sharing $s = (s_1, \dots, s_n)$ of either $m_0$ or $m_1$, via the following queries:

- **Leakage queries.** For each $j \in [\beta]$, the attacker can leak jointly from the shares $s_{\mathcal{B}_j}$. This can be done repeatedly and in an adaptive fashion, the only limitation being that the overall amount of leakage on each block is at most $\ell \in \mathbb{N}$ bits.
- **Tampering queries.** For each $j \in [\beta]$, the attacker can tamper jointly the shares $s_{\mathcal{B}_j}$. Each such query yields mauled shares $(\tilde{s}_1, \dots, \tilde{s}_n)$, for which the adversary is allowed to see the corresponding reconstructed message w.r.t. an arbitrary reconstruction set $\mathcal{T} \in \mathcal{A}$ that is also chosen adversarially. This can be done for at most $p \in \mathbb{N}$ times, and in an adaptive fashion.

The above naturally yields a notion of joint bounded-leakage and tampering admissible adversary, as defined below. Note that, in order to rule out trivial attacks, we must require that the partition $\mathcal{B}$ chosen by the attacker be such that no block of the partition is an authorized set for the underlying access structure.

**Definition 8 (Joint bounded-leakage and tampering admissible adversaries).** *Let $n, k, \ell, p \in \mathbb{N}$, and fix an arbitrary message space $\mathcal{M}$, sharing domain $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ and access structure $\mathcal{A}$ for $n$ parties. We say that a (possibly unbounded) adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ is $k$-joint $\ell$-bounded-leakage and $p$-tampering admissible ($(k, \ell, p)$-BLTA for short) if it satisfies the following conditions:*

*(i) $\mathsf{A}_1$ outputs two messages $m_0, m_1 \in \mathcal{M}$ and a $k$-partition $\mathcal{B} = (\mathcal{B}_1, \dots, \mathcal{B}_\beta)$ of $[n]$ such that $\forall j \in [\beta]$ we have $\mathcal{B}_j \notin \mathcal{A}$.*

```
$\mathbf{JSTamper}_{\Sigma,\mathsf{A}}(\lambda, b)$:
$\omega \leftarrow\!\!\$\ \mathsf{Init}(1^\lambda)$
$(\mathcal{B} = (\mathcal{B}_1, \ldots, \mathcal{B}_\beta), m_0, m_1, \alpha_1) \leftarrow\!\!\$\ \mathsf{A}_1(\omega)$
$s := (s_1, \ldots, s_n) \leftarrow\!\!\$\ \mathsf{Share}(\omega, m_b)$
$\mathtt{stop} \leftarrow \mathtt{false}$
$(\alpha_2, i^* \in [\beta]) \leftarrow\!\!\$\ \mathsf{A}_2^{\mathcal{O}_{\mathsf{nmss}}(s,\cdot,\cdot),\mathcal{O}_{\mathsf{leak}}(s,\cdot)}(\alpha_1)$
Return $\mathsf{A}_3(\alpha_2, s_{\mathcal{B}_{i^*}})$
Oracle $\mathcal{O}_{\mathsf{leak}}(s, (g_1, \ldots, g_\beta))$:
Return $g_1(s_{\mathcal{B}_1}), \ldots, g_\beta(s_{\mathcal{B}_\beta})$

Oracle $\mathcal{O}_{\mathsf{nmss}}(s, \mathcal{T}, (f_1, \ldots, f_\beta))$:
If $\mathtt{stop} = \mathtt{true}$
    Return $\perp$
Else
    $\forall i \in [\beta] : \tilde{s}_{\mathcal{B}_i} := f_i(s_{\mathcal{B}_i})$
    $\tilde{s} = (\tilde{s}_1, \ldots, \tilde{s}_n)$
    $\tilde{m} = \mathsf{Rec}(\omega, \tilde{s}_\mathcal{T})$
    If $\tilde{m} \in \{m_0, m_1\}$
        Return ♥
    If $\tilde{m} = \perp$
        Return $\perp$
        $\mathtt{stop} \leftarrow \mathtt{true}$
    Else return $\tilde{m}$
```

Fig. 1: Experiment defining leakage-resilient (continuously) non-malleable secret sharing under adaptive concurrent reconstruction. The instructions boxed in red are considered only for continuous non-malleability, in which case the oracle $\mathcal{O}_{\mathsf{nmss}}$ is implicitly parameterized by the flag `stop`.

*(ii)* $\mathsf{A}_2$ *outputs a sequence of poly-many leakage queries, chosen adaptively,* $(g_1^{(q)}, \ldots, g_\beta^{(q)})_{q \in \mathtt{poly}(\lambda)}$ *such that* $\forall j \in [\beta]$ *it holds that* $\sum_q |g_j^{(q)}(\cdot)| \leq \ell$*, where* $g_j^{(q)} : \bigtimes_{i \in \mathcal{B}_j} \mathcal{S}_i \to \{0,1\}^*$.

*(iii)* $\mathsf{A}_2$ *outputs a sequence of* $p$ *tampering queries, chosen adaptively,* $(\mathcal{T}^{(q)}, (f_1^{(q)}, \ldots, f_\beta^{(q)}))_{q \in [p]}$ *such that* $\mathcal{T}^{(q)} \in \mathcal{A}$*, and* $\forall j \in [\beta]$ *it holds that* $f_j^{(q)} : \bigtimes_{i \in \mathcal{B}_j} \mathcal{S}_i \to \bigtimes_{i \in \mathcal{B}_j} \mathcal{S}_i$.

Very roughly, leakage-resilient non-malleability states that no admissible adversary as defined above can distinguish whether it is interacting with a secret sharing of $m_0$ or of $m_1$. In the definition below, the attacker is further allowed to obtain in full the shares belonging to one of the partitions, at the end of the experiment. This is reminiscent of augmented (leakage-resilient) non-malleability, as considered in [20,1,25,11].

**Definition 9 (Leakage-resilient non-malleability under selective partitioning).** *Let* $n, k, \ell, p \in \mathbb{N}$ *be parameters, and* $\mathcal{A}$ *be an access structure for* $n$ *parties. We say that* $\Sigma = (\mathsf{Init}, \mathsf{Share}, \mathsf{Rec})$ *is an* augmented *$\ell$-bounded leakage-resilient $p$-time non-malleable secret sharing scheme realizing* $\mathcal{A}$ *against selective $k$-joint leakage and tampering in the CRS model (resp., in the plain model)—augmented $(k, \ell, p)$-BLR-CNMSS for short—if it is an $n$-party secret sharing scheme realizing* $\mathcal{A}$ *in the CRS model (resp., in the plain model) as per Def. 2, and additionally for all $(k, \ell, p)$-BLTA adversaries* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *we have:*

$$\left\{\mathbf{JSTamper}_{\Sigma,\mathsf{A}}(\lambda, 0)\right\}_{\lambda \in \mathbb{N}} \approx_s \left\{\mathbf{JSTamper}_{\Sigma,\mathsf{A}}(\lambda, 1)\right\}_{\lambda \in \mathbb{N}},$$

*where, for* $b \in \{0,1\}$*, experiment* $\mathbf{JSTamper}_{\Sigma,\mathsf{A}}(\lambda, b)$ *is depicted in Fig. 1.*

In case the above definition holds for all $p(\lambda) \in \mathtt{poly}(\lambda)$, but w.r.t. all PPT adversaries $\mathsf{A}$ (i.e., $\approx_s$ is replaced with $\approx_c$ in the above equation), we call $\Sigma$ (augmented, bounded leakage-resilient) continuously non-malleable. As shown by [19], already for the simpler case of independent tampering, it is impossible to achieve this notion without assuming self-destruct (i.e., the oracle $\mathcal{O}_{\mathsf{nmss}}$ must stop answering tampering queries after the first such query yielding an invalid reconstructed message).

It is also well-known that computational security is inherent for obtaining continuously non-malleable secret sharing realizing threshold access structures [6]. Unless stated otherwise, when we refer to non-malleable secret sharing in this paper we implicitly assume security holds in the computational setting (both for privacy and non-malleability).

**On Augmented Non-Malleability.** When dropping the adversary $\mathsf{A}_3$ from the above definition, we obtain the standard (non-augmented) notion of (leakage-resilient, continuous) non-malleability. The theorem below, however, says that augmented security is essentially for free whenever non-malleability is considered together with leakage resilience. Intuitively, this is because in the reduction we can simply simulate all leakage queries, and then ask a final leakage query which reveals the output guess of an hypothetical distinguisher attacking augmented non-malleability.[6] A similar proof strategy was used in [29, Lemma 7]. The formal proof appears in the full version [8].

**Theorem 3.** *Let $\Sigma$ be a $(k, \ell+1, p)$-BLR-CNMSS realizing access structure $\mathcal{A}$ for $n$ parties in the CRS model (resp. plain model). Then, $\Sigma$ is an* augmented *$(k, \ell, p)$-BLR-CNMSS realizing $\mathcal{A}$ in the CRS model (resp. plain model).*

## 3.2 Related Notions

We finally argue that known definitions from the literature can be cast by either restricting, or slightly tweaking, Def. 9.

*Independent leakage and tampering.* The definition below restricts the adversary to leak/tamper from/with each of the shares individually; this is sometimes known as local or independent leakage/tampering. The condition on leakage admissibility, though, is more general, in that the attacker can leak an arbitrary amount of information as long as the total leakage reduces the uncertainty on each share (conditioned on the other shares) by at most $\ell$ bits.

**Definition 10 (Independent noisy-leakage and tampering admissible adversaries).** *Let $n, \ell, p \in \mathbb{N}$, and fix an arbitrary message space $\mathcal{M}$, sharing domain $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ and access structure $\mathcal{A}$ for $n$ parties. We say that a (possibly unbounded) adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ is independent $\ell$-noisy-leakage and tampering admissible ($(n, \ell, p)$-NLTA for short) if it satisfies the following conditions:*

[6] While we state the theorem for the case of bounded leakage, an identical statement holds in the noisy-leakage setting.

*(i)* $\mathsf{A}_1$ *outputs two messages* $m_0, m_1 \in \mathcal{M}$ *and the partition* $\mathcal{B} = (\{1\}, \ldots, \{n\})$.

*(ii)* $\mathsf{A}_2$ *outputs a sequence of poly-many leakage queries (chosen adaptively)* $(g_1^{(q)}, \ldots, g_n^{(q)})_{q \in \texttt{poly}(\lambda)}$ *such that* $\forall i \in [n]$ *we have* $g_i^{(q)} : \mathcal{S}_i \to \{0,1\}^*$, *and* $\forall m \in \mathcal{M}$ *it holds that:*

$$\widetilde{\mathbb{H}}_\infty \left( \mathbf{S}_i | (\mathbf{S}_j)_{j \neq i}, g_i^{(1)}(\mathbf{S}_i), \cdots, g_i^{(p)}(\mathbf{S}_i) \right) \geq \widetilde{\mathbb{H}}_\infty(\mathbf{S}_i | (\mathbf{S}_j)_{j \neq i}) - \ell,$$

*where* $(\mathbf{S}_1, \ldots, \mathbf{S}_n)$ *is the random variable corresponding to* $\mathsf{Share}(\mathsf{Init}(1^\lambda), m)$.

*(iii)* $\mathsf{A}_2$ *outputs a sequence of tampering queries (chosen adaptively)* $(\mathcal{T}^{(q)}, (f_1^{(q)}, \ldots, f_n^{(q)}))_{q \in [p]}$ *such that* $\mathcal{T}^{(q)} \in \mathcal{A}$, *and* $\forall i \in [n]$ *it holds that* $f_i^{(q)} : \mathcal{S}_i \to \mathcal{S}_i$.

When restricting Def. 9 to all PPT $(n, \ell, \texttt{poly}(\lambda))$-NLTA adversaries, we obtain the notion of (augmented) $\ell$-noisy leakage-resilient continuously non-malleable secret sharing against individual leakage and tampering (with adaptive concurrent reconstructions) [19]. Finally, if we consider $n = 2$ and the threshold access structure with reconstruction parameter $\varrho = 2$ (i.e., both shares are required in order to reconstruct the message), we immediately obtain noisy leakage-resilient continuously non-malleable codes in the split-state model [20,34]. In what follows, we write $\mathbf{Tamper}(\lambda, b)$ to denote the random variable in the security experiment of Def. 9 with an $(n, \ell, p)$-NLTA adversary.

*Leakage-resilient secret sharing.* Further, when no tampering is allowed (i.e., $p = 0$), we obtain the notion of leakage-resilient secret sharing [12,29,36,2,33] as a special case. In particular, we write $\mathbf{JSLeak}(\lambda, b)$ to denote the random variable in the security experiment of Def. 9 with a $(k, \ell, 0)$-BLTA adversary, and $\mathbf{Leak}(\lambda, b)$ to denote the random variable in the security experiment of Def. 9 with an $(n, \ell, 0)$-NLTA adversary.

Recall that, by Theorem 3, the augmented variant is without loss of generality as long as leakage resilience holds for $\ell \geq 2$.

## 4 Construction in the CRS Model

### 4.1 Description of the Scheme

We show how to obtain leakage-resilient continuously non-malleable secret sharing for arbitrary access structures in the CRS model, with security against selective joint leakage and tampering. Our construction combines a commitment scheme $(\mathsf{Gen}, \mathsf{Com})$ (cf. §2.2), a non-interactive proof system $(\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Ver})$ for proving knowledge of a committed value (cf. §2.3), and an auxiliary $n$-party secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Rec})$, as depicted in Fig. 2.

The main idea behind the scheme is as follows. The CRS includes the CRS $\omega$ for the proof system and the public key $pk$ for the commitment scheme. Given a message $m \in \mathcal{M}$, the sharing procedure first shares $m$ using $\mathsf{Share}$, obtaining shares $(s_1, \ldots, s_n)$. Then, it commits to the $i$-th share $s_i$ along with the position $i$ using randomness $r_i$, and finally generates $n-1$ proofs $(\pi_j^i)_{j \neq i}$ for the statement

Let $\Sigma = (\mathsf{Share}, \mathsf{Rec})$ be an auxiliary secret sharing scheme realizing access structure $\mathcal{A}$, with message space $\mathcal{M}$ and share space $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. Let $(\mathsf{Gen}, \mathsf{Com})$ be a commitment scheme with domain $\{0,1\}^*$, and $(\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Ver})$ be a non-interactive argument system for the language $\mathcal{L}_{\mathsf{com}}^{pk} = \{c \in \{0,1\}^\gamma : \exists i \in [n], s \in \mathcal{S}_i, r \in \mathcal{R}$ s.t. $\mathsf{Com}(pk, i||s; r) = c\}$ that supports labels in $\{0,1\}^\gamma$. Define the following secret sharing scheme $\Sigma^* = (\mathsf{Init}^*, \mathsf{Share}^*, \mathsf{Rec}^*)$ in the CRS model.

**Initialization algorithm $\mathsf{Init}^*$:** Sample $\omega \leftarrow\!\!\$\, \mathsf{CRSGen}(1^\lambda)$ and $pk \leftarrow\!\!\$\, \mathsf{Gen}(1^\lambda)$, and return $\omega^* = (\omega, pk)$.

**Sharing algorithm $\mathsf{Share}^*$:** Upon input $\omega^* = (\omega, pk)$ and a value $m \in \mathcal{M}$, compute $(s_1, \ldots, s_n) \leftarrow\!\!\$\, \mathsf{Share}(m)$. For each $i \in [n]$, generate $r_i \leftarrow\!\!\$\, \mathcal{R}$ and define $c_i = \mathsf{Com}(pk, i||s_i; r_i)$. For each $i, j \in [n]$ such that $i \neq j$, define $\pi_i^j \leftarrow\!\!\$\, \mathsf{Prove}(\omega, c_j, (c_i, i||s_i, r_i))$. Return the shares $s^* = (s_1^*, \ldots, s_n^*)$, where for each $i \in [n]$ we set $s_i^* = (s_i, r_i, (c_j)_{j\neq i}, (\pi_j^i)_{j\neq i})$.

**Reconstruction algorithm $\mathsf{Rec}^*$:** Upon input $\omega^* = (\omega, pk)$ and shares $(s_i^*)_{i\in\mathcal{I}}$ parse $s_i^* = (s_i, r_i, (c_j^i)_{j\neq i}, (\pi_j^i)_{j\neq i})$ for each $i \in \mathcal{I}$. Hence, proceed as follows:

(a) If $\exists i_1, i_2 \in \mathcal{I}$ and $j \in [n]$ such that $c_j^{i_1} \neq c_j^{i_2}$, output $\bot$; else let the input shares be $s_i^* = (s_i, r_i, (c_j)_{j\neq i}, (\pi_j^i)_{j\neq i})$ for each $i \in \mathcal{I}$.

(b) If $\exists i \in \mathcal{I}$ such that $\mathsf{Com}(pk, i||s_i; r_i) \neq c_i$, output $\bot$.

(c) If $\exists i, j \in \mathcal{I}$ such that $i \neq j$ and $\mathsf{Ver}(\omega, c_j, (c_i, \pi_i^j)) = 0$, output $\bot$.

(d) Else, output $\mathsf{Rec}((s_i)_{i\in\mathcal{I}})$.

Fig. 2: Leakage-resilient continuously non-malleable secret sharing for arbitrary access structures against selective joint leakage and tampering, in the CRS model.

$c_i$ using each time the value $c_j = \mathsf{Com}(pk, j||s_j; r_j)$ as label. The final share of player $i$ consists of $s_i$, along with the randomness $r_i$ used to obtain $c_i$ and all the values $(c_j)_{j\neq i}$ and $(\pi_j^i)_{j\neq i}$. The reconstruction procedure, given a set of shares $s_{\mathcal{I}}^*$, first checks that for each $i \in \mathcal{I}$ the commits $(c_j)_{j\neq i}$ contained in each share are all equal, and moreover each $c_i$ is indeed obtained by committing $i||s_i$ with the randomness $r_i$; further, it checks that all the proofs verify correctly w.r.t. the corresponding statement and label. If any of the above checks fails, the algorithm returns $\bot$ and otherwise it outputs the same as $\mathsf{Rec}(s_{\mathcal{I}})$.

Intuitively, our scheme can be seen as a generalization of the original construction of continuously non-malleable codes in the split-state model from [20]. In particular, when $n = 2$, the two constructions are identical except for two differences: (i) We commit to each share, whereas [20] uses a collision-resistant hash function; (ii) We include the position of each share in the commitment. Roughly speaking, the first modification is necessary in order to prove privacy (as hash functions do not necessarily hide their inputs). The second modification is needed in order to avoid that an attacker can permute the shares within one of the partitions, which was not possible in the setting of independent tampering. We establish the following result, whose proof appears in the full version [8].

**Theorem 4.** *Let $n, k \in \mathbb{N}$, and $\mathcal{A}$ be any access structure for $n$ parties. Assume that:*

*(i)* *$\Sigma$ is an $n$-party augmented $\ell$-bounded leakage-resilient secret sharing scheme realizing access structure $\mathcal{A}$ against selective $k$-joint leakage in the plain model;*

*(ii)* $(\mathsf{Gen}, \mathsf{Com})$ *is a statistically hiding and computationally binding commitment scheme with commitment length $\gamma = O(\lambda)$;*

*(iii)* $(\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Ver})$ *is a true-simulation extractable non-interactive zero-knowledge argument system for the language $\mathcal{L}^{pk}_{\mathsf{com}} = \{c \in \{0,1\}^{\gamma} : \exists i \in [n], s \in \mathcal{S}_i, r \in \mathcal{R}$ s.t. $\mathsf{Com}(pk, i||s; r) = c\}$.*

*Then, the secret sharing scheme $\Sigma^*$ described in Fig. 2 is an $n$-party augmented $\ell^*$-bounded leakage-resilient continuously non-malleable secret sharing scheme realizing access structure $\mathcal{A}$ against selective $k$-joint leakage and tampering in the CRS model, as long as $\ell = 2\ell^* + n\gamma + O(\lambda \log \lambda)$.*

### 4.2 Concrete Instantiation

Finally, we show how to instantiate Thm. 4 from generic assumptions, thus yielding the statement of Thm. 2 as a corollary. It is well known that true-simulation extractable NIZKs can be obtained from (doubly-enhanced) trapdoor permutations [21,35,13], whereas statistically hiding non-interactive commitments—with commitment size $O(\lambda)$ and $2^{-\Omega(\lambda)}$-statistical hiding—can be instantiated from collision-resistant hash functions [27].

As for the underlying leakage-resilient secret sharing, we can use the recent construction from [29] which achieves information-theoretic security in the stronger setting where the attacker can adaptively leak from subsets of shares of size at most $O(\log n)$, in a joint manner. The latter clearly implies leakage resilience against selective $O(\log n)$-joint leakage.

## 5 Construction in the Plain Model

### 5.1 Description of the Scheme

We show how to obtain leakage-resilient continuously non-malleable secret sharing for arbitrary access structures in the plain model, with security against independent leakage and tampering attacks. Our construction combines a non-interactive commitment scheme $\mathsf{Com}$ with an auxiliary $n$-party secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Rec})$, as depicted in Fig. 3. The basic idea is to compute a commitment $c$ to the message $m$ being shared, using random coins $r$; hence, we secret share the string $m||r$ using the underlying sharing function $\mathsf{Share}$, yielding shares $(s_1, \ldots, s_n)$. Hence, the final share of the $i$-th player is $s_i^* = (c, s_i)$.

We establish the following result. Note that when $n = 2$, we get as a special case the construction of split-state continuously non-malleable codes in the plain model that was originally proposed in [34], and later simplified in [19] by relying on noisy leakage. Our proof can be seen as a generalization of the proof strategy in [19] to the case $n > 2$. We refer the reader to the full version [8] for the details.

Let $\mathsf{Com}$ be a non-interactive commitment scheme with message space $\mathcal{M}$, randomness space $\mathcal{R}$, and commitment space $\mathcal{C}$. Let $\Sigma = (\mathsf{Share}, \mathsf{Rec})$ be an auxiliary secret sharing scheme realizing access structure $\mathcal{A}$, with message space $\mathcal{M} \times \mathcal{R}$ and share space $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. Define the following secret sharing scheme $\Sigma^* = (\mathsf{Share}^*, \mathsf{Rec}^*)$, with message space $\mathcal{M}$ and share space $\mathcal{S}^* = \mathcal{S}_1^* \times \cdots \times \mathcal{S}_n^*$ where for each $i \in [n]$ we have $\mathcal{S}_i^* = \mathcal{C} \times \mathcal{S}_i$.

**Sharing algorithm $\mathsf{Share}^*$:** Upon input a value $m \in \mathcal{M}$, sample random coins $r \leftarrow_{\$} \mathcal{R}$ and compute $c = \mathsf{Com}(m; r)$ and $(s_1, \ldots, s_n) \leftarrow_{\$} \mathsf{Share}(m||r)$. Return the shares $s^* = (s_1^*, \ldots, s_n^*)$, where for each $i \in [n]$ we set $s_i^* = (c, s_i)$.

**Reconstruction algorithm $\mathsf{Rec}^*$:** Upon input shares $(s_i^*)_{i \in \mathcal{I}}$ parse $s_i^* = (s_i, c_i)$ for each $i \in \mathcal{I}$. Hence, proceed as follows:

(a) If $\exists i_1, i_2 \in \mathcal{I}$ for which $c_{i_1} \neq c_{i_2}$, return $\perp$; else, let the input shares be $s_i^* = (s_i, c)$.

(b) Run $m||r = \mathsf{Rec}((s_i)_{i \in \mathcal{I}})$; if the outcome equals $\perp$ return $\perp$.

(c) If $c = \mathsf{Com}(m; r)$ return $m$, else return $\perp$.

Fig. 3: Leakage-resilient continuously non-malleable secret sharing for arbitrary access structures against independent leakage and tampering in the plain model.

**Theorem 5.** *Let $n \in \mathbb{N}$, and let $\mathcal{A}$ be an arbitrary access structure for $n$ parties without singletons. Assume that:*

*(i)* $\mathsf{Com}$ *is a perfectly binding and computationally hiding non-interactive commitment;*

*(ii)* $\Sigma$ *is an $n$-party $\ell$-noisy leakage-resilient one-time non-malleable secret sharing scheme realizing access structure $\mathcal{A}$ against independent leakage and tampering in the plain model, with information-theoretic security and with message space $\mathcal{M}$ such that $|\mathcal{M}| \in \omega(\log(\lambda))$.*

*Then, the secret sharing scheme $\Sigma^*$ described in Fig. 3 is an $n$-party $\ell^*$-noisy leakage-resilient continuously non-malleable secret sharing scheme realizing access structure $\mathcal{A}$ against independent leakage and tampering with computational security in the plain model, as long as $\ell = \ell^* + 1 + \gamma + O(\log \lambda)$ where $\gamma = \log |\mathcal{C}|$ is the size of a commitment.*

## 6 Statistical One-Time Non-Malleability with Noisy Leakage

Since non-interactive, perfectly binding, commitments can be obtained in the plain model assuming one-to-one one-way functions [22], all that remains in order to derive Thm. 1 as a corollary of Thm. 5 is an unconditional construction of noisy-leakage resilient one-time non-malleable secret sharing for arbitrary access structures against independent leakage and tampering. The only known scheme achieving all these properties unconditionally is the one in [29], but unfortunately that scheme only tolerates bounded leakage, and it is unclear how to generalize the proof to the setting of noisy leakage.[7] Hence, we take a different approach and we instead show how to generalize a recent transformation from [7], which is tailored to the case $n = 2$.

### 6.1 Asymmetric Noisy-Leakage-Resilient Secret Sharing

Our construction exploits so-called leakage-resilient encryption, as recently introduced by Ball, Guo, and Wichs [7]. To keep the exposition more uniform, we cast their definition in terms of a special 2-out-of-2 leakage-resilient secret sharing satisfying three additional properties: (i) One of the shares is uniformly random, and can be sampled independently from the message; (ii) The shares are almost uncorrelated, namely the distribution of one share in isolation and conditioned on the other share have very similar min-entropy; (iii) The size of the shares are asymmetric, namely one share is substantially larger than the other share. Given a 2-out-of-2 secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Rec})$, abusing notation, for any fixed $s_1 \in \mathcal{S}_1$ and $m \in \mathcal{M}$, we write $s_2 \leftarrow\!\!\$\, \mathsf{Share}(m, s_1)$ for the sharing algorithm that computes share $s_2$ subject to $(s_1, s_2)$ being a valid sharingof $m$.

**Definition 11 (Asymmetric secret sharing).** *Let $\Sigma = (\mathsf{Share}, \mathsf{Rec})$ be a 2-out-of-2 secret sharing scheme. We call $\Sigma$ $(\alpha, \sigma_1, \sigma_2)$*-asymmetric*, if it satisfies the following properties:*

*(i) For any $s_1 \in \mathcal{S}_1$, and any $m \in \mathcal{M}$, it holds that $\mathsf{Rec}(s_1, \mathsf{Share}(m, s_1)) = m$;*

*(ii) For any message $m \in \mathcal{M}$, and for all $i \in \{1, 2\}$, it holds that $\widetilde{\mathbb{H}}_\infty(\mathbf{S}_i | \mathbf{S}_{3-i}) \geq \log |\mathcal{S}_i| - \alpha$, where $\mathbf{S}_1, \mathbf{S}_2$ are the random variables corresponding to sampling $s_1 \leftarrow\!\!\$\, \mathcal{S}_1$ and $s_2 \leftarrow\!\!\$\, \mathsf{Share}(m, s_1)$;*

*(iii) It holds that $\log |\mathcal{S}_1| = \sigma_1$ and $\log |\mathcal{S}_2| = \sigma_2$.*

As for security we consider the same security experiment of a leakage-resilient secret sharing, however, we consider a more general class of admissible adversaries:

**Definition 12 (Independent noisy-leakage admissibility for asymmetric secret sharing).** *Let $\Sigma = (\mathsf{Share}, \mathsf{Rec})$ be a 2-out-of-2 secret sharing scheme. We say that an unbounded adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ is independent $(\ell_1, \ell_2)$-asymmetric noisy-leakage admissible ($(\ell_1, \ell_2)$-NLA for short) if it satisfies Def. 10 without property (iii), and using the following variant of property (ii):*

*(ii) $\mathsf{A}_2$ outputs a sequence of leakage queries (chosen adaptively) $(g^{(q)})_{q \in [p]}$, with $p(\lambda) \in \mathtt{poly}(\lambda)$, such that for all $i \in \{1, 2\}$, and for all $m \in \mathcal{M}$:*

$$\widetilde{\mathbb{H}}_\infty\left(\mathbf{S}_i | \mathbf{S}_{3-i}, g_i^{(1)}(\mathbf{S}_i), \cdots, g_i^{(p)}(\mathbf{S}_i)\right) \geq \widetilde{\mathbb{H}}_\infty(\mathbf{S}_i | \mathbf{S}_{3-i}) - \ell_i,$$

*where $\mathbf{S}_1$ is uniformly random over $\mathcal{S}_1$ and $\mathbf{S}_2$ is the random variable corresponding to $\mathsf{Share}(m, \mathbf{S}_1)$.*

[7] This is because [29] relies on lower bounds in communication complexity.

Let $\Sigma' = (\mathsf{Share}', \mathsf{Rec}')$ be a secret sharing scheme realizing access structure $\mathcal{A}$, with message space $\mathcal{M}$ and share space $\mathcal{S}' = \mathcal{S}'_1 \times \cdots \times \mathcal{S}'_n$ where $\mathcal{S}'_i \subseteq \mathcal{M}''$. Let $\Sigma'' = (\mathsf{Share}'', \mathsf{Rec}'')$ be a 2-out-of-2 *asymmetric* secret sharing scheme with message space $\mathcal{M}''$ and share space $\mathcal{S}'' = \mathcal{S}''_1 \times \mathcal{S}''_2$. Define the following secret sharing scheme $\Sigma = (\mathsf{Share}, \mathsf{Rec})$, with message space $\mathcal{M}$ and share space $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$, where for each $i \in [n]$ we have $\mathcal{S}_i \subseteq (\mathcal{S}''_1)^{n-1} \times (\mathcal{S}''_2)^{n-1}$.

**Sharing algorithm** $\mathsf{Share}$**:** Upon input a value $m \in \mathcal{M}$, compute $(s'_1, \ldots, s'_n) \leftarrow\!\!\$\; \mathsf{Share}'(m)$. For each $i \in [n]$ and $j \in [n] \setminus \{i\}$, sample a random share $s''_{i,j,1} \leftarrow\!\!\$\; \mathcal{S}''_1$ and compute $s''_{i,j,2} \leftarrow\!\!\$\; \mathsf{Share}''(s'_i, s''_{i,j,1})$. Return the shares $s = (s_1, \ldots, s_n)$, where for each $i \in [n]$ we set $s_i = ((s''_{j,i,1})_{j \neq i}, (s''_{i,j,2})_{j \neq i})$.

**Reconstruction algorithm** $\mathsf{Rec}$**:** Upon input shares $(s_i)_{i \in \mathcal{I}}$ with $\mathcal{I} \in \mathcal{A}$, parse $s_i = ((s''_{j,i,1})_{j \neq i}, (s''_{i,j,2})_{j \neq i})$. Hence, proceed as follows:
(a) Compute $s'_i = \mathsf{Rec}''(s''_{i,\mathtt{nxt}(i),1}, s''_{i,\mathtt{nxt}(i),2})$ for $i \in \mathcal{I}$;
(b) Return $\mathsf{Rec}'((s'_i)_{i \in \mathcal{I}})$.

Fig. 4: Noisy-leakage-resilient one-time statistically non-malleable secret sharing for arbitrary access structures against independent leakage and tampering in the plain model.

Finally, we say that a 2-out-of-2 secret sharing is augmented $(\ell_1, \ell_2)$-noisy-leakage resilient if it is secure as per Def. 9, against the class of all unbounded adversaries that are $(\ell_1, \ell_2)$-NLA. The theorem below says that there is an unconditional construction of such a leakage-resilient secret sharing that is also asymmetric as per Def. 11. The proof appears in the full version [8].

**Theorem 6.** *For any $\alpha \in \mathbb{N}$, and for any large enough $\ell_1, \ell_2 \in \mathtt{poly}(\lambda, \alpha)$, there exists $\sigma_1, \sigma_2 \in \mathtt{poly}(\lambda, \alpha)$ and an $(\alpha, \sigma_1, \sigma_2)$-asymmetric secret sharing scheme $\Sigma$ with message space $\{0,1\}^\alpha$ that is augmented $(\ell_1, \ell_2)$-noisy leakage resilient.*

**Construction.** Before presenting our scheme, we establish some notation. Given a reconstruction set $\mathcal{I} = \{i_1, \ldots, i_k\}$, we always assume that $i_j \leq i_{j+1}$ for $j \in [k]$. further, we define the function $\mathtt{nxt}_\mathcal{I} : \mathcal{I} \to \mathcal{I}$ as:

$$\mathtt{nxt}_\mathcal{I}(i_j) := \begin{cases} i_{j+1} & j < k \\ i_1 & \text{otherwise} \end{cases}$$

and the function $\mathtt{prv}_\mathcal{I}$ to be the inverse of $\mathtt{nxt}_\mathcal{I}$. Whenever it is clear from the context we omit the reconstruction set $\mathcal{I}$ and simply write $\mathtt{nxt}$ and $\mathtt{prv}$.

Intuitively, our construction (cf. Fig. 4) relies on a one-time non-malleable (but not leakage resilient) secret sharing $\Sigma'$, and on an asymmetric leakage-resilient secret sharing $\Sigma'$. The sharing of a message $m$ is obtained by first sharing $m$ under $\Sigma'$, obtaining $n$ shares $(s'_1, \ldots, s'_n)$, and then sharing each $s_i$ independently $n-1$ times under $\Sigma''$, obtaining pairs of shares $(s''_{i,j,1}, s''_{i,j,2})_{j \neq i}$; the final share of party $i$ is then set to be the collection of right shares corresponding to $i$ and all the left shares corresponding to the parties $j \neq i$. We can now state the main theorem of this section.

**Theorem 7.** *Let $n \in \mathbb{N}$, and let $\mathcal{A}$ be an arbitrary access structure for $n$ parties without singletons. Assume that:*

*(i) $\Sigma'$ is an $n$-party one-time non-malleable secret sharing scheme realizing access structure $\mathcal{A}$ against independent tampering in the plain model, with information-theoretic security;*
*(ii) $\Sigma''$ is an $(\alpha, \sigma_1, \sigma_2)$-asymmetric augmented $(\ell_1, \ell_2)$-noisy leakage-resilient secret sharing scheme.*

*Then, the secret sharing scheme $\Sigma$ described in Fig. 4 is an $n$-party $\ell$-noisy leakage-resilient one-time non-malleable secret sharing scheme realizing access structure $\mathcal{A}$ against independent leakage and tampering with statistical security in the plain model, as long as $\ell_1 = \ell + (2n-3)\alpha$ and $\ell_2 = \ell + (2n-3)\alpha + \sigma_1$.*

The proof to the above theorem appears in the full version [8], here we discuss the main intuition. Privacy of $\Sigma$ follows in a fairly straightforward manner from privacy of $\Sigma'$. In fact, recall that the shares $s''_{i,j,1}$, with $i, j \in [n]$ and $i \neq j$, are sampled uniformly at random and independently of $s'$. Thus, in the reduction we can sample these values locally and then define the shares $(s_u)_{u \in \mathcal{U}}$ as a function of the shares $(s'_u)_{u \in \mathcal{U}}$. As for the proof of leakage-resilient one-time non-malleability, the idea is to reduce to the one-time non-malleability of $\Sigma'$ and simulate the leakage by sampling dummy values for the shares $s''_{i,j,1}, s''_{i,j,2}$.

The main challenge is to make sure that the answer to tampering query $f = (f_1, \ldots, f_n)$ is consistent with the simulated leakage. To this end, in the reduction we define the tampering function $f' = (f'_1, \ldots, f'_n)$, acting on the shares $s' = (s'_1, \ldots, s'_n)$, as follows. Each function $f'_i$, upon input $s'_i$ and given the values $(s''_{i,j,1})_{j \neq i}$, samples $(\hat{s}_{i,j,2})_{j \neq i}$ in such a way that for any $j$ the reconstruction $\mathsf{Rec}''(s_{i,j,1}, \hat{s}_{i,j,2})$ yields a share $s'_i$ that is consistent with the simulated leakage using the dummy values. Noisy-leakage resilience of $\Sigma''$ guarantees that the function $f'_i$ samples from a valid distribution (namely, a non-empty one). Note that the function $f'_i$ might not be efficiently computable; however, as we are reducing to statistical non-malleability, this is not a problem.

An additional difficulty is that the functions $(f'_t)_{t \in \mathcal{T}}$ need to communicate in order to produce their outputs. In fact, for any $t \in \mathcal{T}$, the function $f'_t$ returns a tampered share for $\Sigma'$ that depends on the mauled share $\tilde{s}_{\mathtt{prv}(t),t,1}$ (generated by $f_{\mathtt{prv}(t)}$). To overcome this problem, we let the reduction perform an additional leakage query on the dummy values before tampering. Thanks to this extra leakage, the reduction learns the values $\tilde{s}_{\mathtt{prv}(t),t,1}$ for all $t \in \mathcal{T}$, which can be hard-coded in the description of $(f'_t)_{t \in \mathcal{T}}$. Here is where we rely on the asymmetric property of $\Sigma''$, which allows us to leak $\sigma_1$ bits from the second share.

At this point, a reader familiar with [7] might notice that the two proofs proceed very similarly. However, our proof requires extra care when bounding the amount of leakage performed by the reduction. The key ideas are that: (i) Each of the shares under $\Sigma'$ is shared using $n-1$ independent invocations of $\Sigma''$; and (ii) our reconstruction procedure depends only on one of those (chosen as function of the reconstruction set). Property (i) allows to reduce independent leakage on $n$ shares under $\Sigma$ to independent leakage on 2 shares under $\Sigma''$ by sampling locally the missing $n-2$ shares when reducing to noisy-leakage resilience of $\Sigma''$. Property (ii) allows to bound the amount of information the reduction needs to simulate the tampering query to a single short leakage from each of the shares (i.e., the value $\tilde{s}_{\mathtt{prv}(t),t,1}$ for $t \in \mathcal{T}$).

## 7 Conclusions and Open Problems

We have shown new constructions of leakage-resilient continuously non-malleable secret sharing schemes, for general access structures. Our first scheme is in the plain model, and guarantees security against independent noisy leakage and tampering with all of the shares. Our second scheme is in the CRS model, and guarantees security against joint bounded leakage and tampering using a fixed partition of the $n$ shares into non-overlapping blocks of size $O(\log n)$.

The two major questions left over by our work are whether continuous non-malleability against joint tampering is achievable in the plain model, or against adaptive (rather than selective) joint tampering with the shares. Interestingly, our proof strategy breaks down in the case of adaptive tampering, and this holds true even assuming that the inner leakage-resilient secret sharing is secure in the presence of adaptive joint leakage. Intuitively, the reason is that in the reduction we must run different copies of the adversary inside the leakage oracle; in particular, we use each block of the shares in order to simulate the answer to all tampering queries asked by each copy of the attacker, and this is clearly possible only if the adversary does not change the partition within each query.

It would also be interesting to achieve continuous non-malleability under joint selective partitioning for better values of the parameter $k$ (namely, the attacker can tamper jointly with blocks of size super-logarithmic in $n$). Note that this would follow immediately by our result if we plug in our construction a leakage-resilient secret sharing scheme tolerating joint leakage from subsets of shares with size $\omega(\log n)$. Unfortunately, the only known secret sharing scheme achieving joint-leakage resilience is the one in [29], and as the authors explain improving the parameters in their construction would lead to progress on longstanding open problems in complexity theory. We leave it open to establish whether this holds true even in the case of selective partitioning (recall that the scheme of [29] achieves adaptive leakage resilience), or whether the current state of affairs can be improved in the computational setting (with or without trusted setup).

A further open question is to improve the rate of our constructions. Note that by applying the rate compiler of [19], we do get *rate-one* continuously non-malleable secret sharing for general access structures, against independent tampering in the plain model. However, this is well-known to be sub-optimal in the computational setting, where the optimal share size would be $O(\mu/n)$, with $\mu$ being the size of the message [28]. Note that it is unclear whether the same rate compiler works also for our construction against joint tampering under selective partitioning. This is because the analysis in [19] crucially relies on the resilience of the initial rate-zero non-malleable secret sharing against noisy leakage, whereas our construction only achieves security in the bounded-leakage model.

## References

1. Aggarwal, D., Agrawal, S., Gupta, D., Maji, H.K., Pandey, O., Prabhakaran, M.: Optimal computational split-state non-malleable codes. In: TCC. pp. 393–417 (2016)
2. Aggarwal, D., Damgård, I., Nielsen, J.B., Obremski, M., Purwanto, E., Ribeiro, J., Simkin, M.: Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In: CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 510–539 (Aug 2019)
3. Aggarwal, D., Dodis, Y., Kazana, T., Obremski, M.: Non-malleable reductions and applications. In: STOC. pp. 459–468 (2015)
4. Aggarwal, D., Dodis, Y., Lovett, S.: Non-malleable codes from additive combinatorics. In: STOC. pp. 774–783 (2014)
5. Aggarwal, D., Dziembowski, S., Kazana, T., Obremski, M.: Leakage-resilient non-malleable codes. In: TCC. pp. 398–426 (2015)
6. Badrinarayanan, S., Srinivasan, A.: Revisiting non-malleable secret sharing. In: EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 593–622 (May 2019)
7. Ball, M., Guo, S., Wichs, D.: Non-malleable codes for decision trees. In: CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 413–434 (Aug 2019)
8. Brian, G., Faonio, A., Venturi, D.: Continuously non-malleable secret sharing for general access structures (2019), https://eprint.iacr.org/2019/602
9. Chattopadhyay, E., Li, X.: Non-malleable codes, extractors and secret sharing for interleaved tampering and composition of tampering. Cryptology ePrint Archive, Report 2018/1069 (2018), https://eprint.iacr.org/2018/1069
10. Cheraghchi, M., Guruswami, V.: Non-malleable coding against bit-wise and split-state tampering. In: TCC. pp. 440–464 (2014)
11. Coretti, S., Faonio, A., Venturi, D.: Rate-optimizing compilers for continuously non-malleable codes. In: ACNS 19. LNCS, vol. 11464, pp. 3–23. Springer, Heidelberg (Jun 2019)
12. Davì, F., Dziembowski, S., Venturi, D.: Leakage-resilient storage. In: SCN. pp. 121–137 (2010)
13. Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient public-key cryptography in the presence of key leakage. In: ASIACRYPT. pp. 613–631 (2010)
14. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
15. Dziembowski, S., Kazana, T., Obremski, M.: Non-malleable codes from two-source extractors. In: CRYPTO. pp. 239–257 (2013)
16. Dziembowski, S., Pietrzak, K.: Intrusion-resilient secret sharing. In: FOCS. pp. 227–237 (2007)
17. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: Innovations in Computer Science. pp. 434–452 (2010)
18. Faonio, A., Nielsen, J.B., Simkin, M., Venturi, D.: Continuously non-malleable codes with split-state refresh. In: ACNS. pp. 1–19 (2018)
19. Faonio, A., Venturi, D.: Non-malleable secret sharing in the computational setting: Adaptive tampering, noisy-leakage resilience, and improved rate. In: CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 448–479 (Aug 2019)
20. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. In: TCC. pp. 465–488 (2014)

21. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In: FOCS. pp. 308–317 (1990)
22. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: STOC. pp. 218–229 (1987)
23. Goyal, V., Kumar, A.: Non-malleable secret sharing. In: STOC. pp. 685–698 (2018)
24. Goyal, V., Kumar, A.: Non-malleable secret sharing for general access structures. In: CRYPTO. pp. 501–530 (2018)
25. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: STOC. pp. 1128–1141 (2016)
26. Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: ASIACRYPT. pp. 444–459 (2006)
27. Halevi, S., Micali, S.: Practical and provably-secure commitment schemes from collision-free hashing. In: CRYPTO. pp. 201–215 (1996)
28. Krawczyk, H.: Secret sharing made short. In: CRYPTO. pp. 136–146 (1993)
29. Kumar, A., Meka, R., Sahai, A.: Leakage-resilient secret sharing. Cryptology ePrint Archive, Report 2018/1138 (2018), https://ia.cr/2018/1138
30. Li, X.: Improved non-malleable extractors, non-malleable codes and independent source extractors. In: STOC. pp. 1144–1156 (2017)
31. Lin, F., Cheraghchi, M., Guruswami, V., Safavi-Naini, R., Wang, H.: Non-malleable secret sharing against affine tampering. CoRR **abs/1902.06195** (2019), http://arxiv.org/abs/1902.06195
32. Liu, F., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: CRYPTO. pp. 517–532 (2012)
33. Nielsen, J.B., Simkin, M.: Lower bounds for leakage-resilient secret sharing. Cryptology ePrint Archive, Report 2019/181 (2019), https://eprint.iacr.org/2019/181
34. Ostrovsky, R., Persiano, G., Venturi, D., Visconti, I.: Continuously non-malleable codes in the split-state model from minimal assumptions. In: CRYPTO. pp. 608–639 (2018)
35. Santis, A.D., Crescenzo, G.D., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: CRYPTO. pp. 566–598 (2001)
36. Srinivasan, A., Vasudevan, P.N.: Leakage resilient secret sharing and applications. In: CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 480–509. Springer, Heidelberg (Aug 2019)