# The Chaining Lemma and its application

Ivan Damgård[1], Sebastian Faust[2], Pratyay Mukherjee[1], and Daniele Venturi[3]

[1] *Department of Computer Science, Aarhus University*
[2] *Security and Cryptography Laboratory, EPFL*
[3] *Department of Computer Science, Sapienza University of Rome*

December 3, 2014

**Abstract**

We present a new information theoretic result which we call the Chaining Lemma. It considers a so-called "chain" of random variables, defined by a source distribution ($X^{(0)}$) with high min-entropy and a number (say, $t$ in total) of arbitrary functions ($T_1, \dots, T_t$) which are applied in succession to that source to generate the chain $X^{(0)} \xrightarrow{T_1} X^{(1)} \xrightarrow{T_2} X^{(2)} \cdots \xrightarrow{T_t} X^{(t)}$. Intuitively, the Chaining Lemma guarantees that, if the chain is not too long, then either (i) the entire chain is "highly random", in that every variable has high min-entropy; or (ii) it is possible to find a point $j$ ( $1 \leq j \leq t$) in the chain such that, conditioned on the end of the chain i.e. $X^{(j)} \xrightarrow{T_{j+1}} X^{(j+1)} \cdots \xrightarrow{T_t} X^{(t)}$, the preceding part viz. $X^{(0)} \xrightarrow{T_1} X^{(1)} \cdots \xrightarrow{T_j} X^{(j)}$ remains highly random. We believe this is an interesting information-theoretic result which is intuitive but nevertheless requires rigorous case-analysis to prove. We believe that it will find applications in cryptography. We give an example of this, namely we show an application of the lemma to protect essentially any cryptographic scheme against memory-tampering attacks. We allow several tampering request, the tampering functions can be arbitrary, however, they must be chosen from a bounded size set of functions that is fixed a priori.

## 1 Introduction

Assume that we have a uniform random distribution over some domain $\mathcal{X}$, represented by the random variable $X$. Let us now apply an arbitrary function $T$ to $X$ and denote the output random variable by $X' = T(X)$. Since $T$ is an arbitrary function, the variable $X'$ can also be arbitrarily distributed. Consider now the case, where $X'$ is "easy to predict", or more concretely where $X'$ has "low" min-entropy. Now, a natural question, in this case, is *how much information can $X'$ reveal about $X$?* or more formally, *how much min-entropy can $X$ have if we condition on $X'$?*

Intuitively, one might expect that since $X'$ has low entropy, it cannot tell us much about $X$, so $X$ should still be "close to random" and hence have high entropy. While this would be true for Shannon entropy, it turns out to be completely false for min-entropy. This may seem a bit counter-intuitive at first, but is actually easy to see from an example: Let $T$ be the function which maps half of the elements in $\mathcal{X}$ to one "heavy" point but is injective on all the other elements. For this $T$, $X'$ has very small min-entropy (namely 1) because the heavy point occurs with probability 1/2. But on the other hand, $X'$ reveals everything about $X$ half the time, and so the entropy of $X$ in fact decreases very significantly (on average) when $X'$ is given. So despite having very low min-entropy, $X' = T(X)$ does reveal a lot about $X$.

There is, however, a more refined statement that will be true for min-entropy: Let $E$ be the event that $X$ takes one of the values that are *not* mapped to the "heavy point" by $T$, while $\bar{E}$ is the event that $X$ is mapped to the heavy point. Now, conditioned on $E$, both $X_{|E}$ and $X'_{|E}$ have high min-entropy. On the other hand, conditioned on $\bar{E}$, $X_{|\bar{E}}$ will clearly have the same (high) min-entropy whether we are given $X'_{|\bar{E}}$ or not

This simple observation leads to the following conjecture: there always exists an event $E$ such that: (i) Conditioned on $E$, both $X$ and $X'$ have "high" min-entropy (ii) and conditioned on $\bar{E}$, $X'$ reveals "little" about $X$. In this paper, from a very high-level, we mainly focus into settling this conjecture, however, in a more general setting, which results in the main contribution of the paper, the information theoretic lemma which we call the Chaining Lemma.

We generalize the above setting and towards that, for notational convenience, let us rename the above symbols as follows: $X^{(0)} \equiv X$, $T_1 \equiv T$ and $X^{(1)} \equiv X'$. We consider $t$ functions $T_1, T_2, \ldots, T_t$ which are applied to the the variables sequentially starting from $X^{(0)}$. In particular, each $T_i$ is applied to $X^{(i-1)}$ to produce a new variable $X^{(i)} = T_i(X^{(i-1)})$ for $i \in \{1, \ldots, t\}$. We call the sequence of variables $(X^{(0)}, \ldots, X^{(t)})$ a "chain" which is completely defined by the "source" distribution $X^{(0)}$ and the sequence of $t$ functions $(T_1, \ldots, T_t)$. It can be presented more vividly as follows: $X^{(0)} \xrightarrow{T_1} X^{(1)} \xrightarrow{T_2} X^{(2)} \cdots \xrightarrow{T_t} X^{(t)}$.

We are now interested in the min-entropy of $X^{(1)}, \ldots, X^{(t)}$. Of course, each variable $X^{(i)}$ has min-entropy less than (or equal to) the preceding variable $X^{(i-1)}$ (as a deterministic function can not generate randomness). Assume now that we fix some threshold value $u$ and consider any value of min-entropy less than $u$ to be "low". Assume further that the source has min-entropy much larger than $u$. As a motivation, one may think of a setting where each $X^{(i)}$ is used as key in some cryptographic application, where, as long $X^{(i)}$ has high min entropy we are fine and the adversary will not learn something he should not. But if $X^{(i)}$ has low min-entropy, things might go wrong and the adversary might learn $X^{(i)}$.

Now, there are two possible scenarios for the above chain: (i) all the variables (hence the last variable $X^{(t)}$) in the chain have high min-entropy (ii) one or more variable (obviously including the last variable $X^{(t)}$) has low min-entropy.

In case (i), everything is fine. But in case (ii), things might go wrong at a certain point. We now want to ask if we can at least "save" some part of the chain, i.e., *can we find a point in the chain such that if we condition on all the variables after that point, all the preceding variables (obviously including the source $X^{(0)}$) would still have high min-entropy ?* This hope might be justified if $t$ is small enough compared to the entropy of $X^{(0)}$: since the entropy drops below $u$ after a small number of steps, there must be a point (say $j$) where the entropy falls "sharply", i.e., $X^{(j)}$ has much smaller min-entropy than $X^{(j-1)}$. However, as the above example shows, even if there is a large gap in min-entropy between two successive variables ($X^{(j)}$ and $X^{(j-1)}$ in this case), the succeeding one ($X^{(j)}$) might actually reveal a lot about the preceding one ($X^{(j-1)}$) on average. So it is not clear that we can use $j$ as the point we are looking for. However, one could hope that a generalised version of the above conjecture might be true, namely there might exist some event, further conditioning on which, all variables would have high min-entropy, and on the other hand, conditioning on the complement, $X^{(j-1)}$ (and hence the entire preceding chain) would have high min-entropy. Essentially that is what our Chaining Lemma says, which we present next although in an informal way. We give the formal statement and proof of the lemma in Section 3.

**Lemma 1.1.** *(The Chaining Lemma (informal)) Let $X^{(0)}$ be a uniform random variable over $\mathcal{X}$ and $(T_0, \ldots, T_t)$ be arbitrary functions mapping $\mathcal{X} \to \mathcal{X}$. They define a chain $X^{(0)} \xrightarrow{T_1} X^{(1)} \xrightarrow{T_2} X^{(2)} \cdots \xrightarrow{T_t} X^{(t)}$. If the chain is "sufficiently short", there exists an event $E$ such that (i) if $E$ happens, then all the variables $(X^{(0)}, \ldots, X^{(t)})$ (conditioned on $E$) would have "high" min-entropy; (ii) otherwise if $E$ does not happen there is an index $j$ such that conditioning on $X^{(j)}$ (and also on $\bar{E}$) all the previous variables namely $X^{(0)}, \ldots, X^{(j-1)}$ would have "high" min-entropy.*

**Application to tamper-resilient cryptography:** Although we think that the Chaining Lemma is interesting in its own right, in this paper we provide an application in cryptography, precisely in tamper-resilient cryptography. In tamper-resilient cryptography the main goal is to "theoretically" protect cryptographic schemes against the well-known fault attack which are found to be devastating as shown by [5, 12] and many more. In this model, the adversary is stronger than standard black-box setting because, in addition to standard (blackbox) access, it can change the secret state [9, 27, 22, 31, 8] or the cryptographic computation [29, 26, 17, 18, 24, 19] and observes the effect of such changes at the output. In this paper we restrict ourselves to the model where the adversary is not allowed to alter the computation, but only the secret state (i.e. only the memory of the device but not the circuitry).

To illustrate such memory tampering, consider a digital signature scheme $\mathsf{Sign}$ with public/secret key pair $(pk, sk)$. The tampering adversary obtains $pk$ and can replace $sk$ with $T(sk)$ for arbitrary tampering function $T$. Then, the adversary gets access to an oracle $\mathsf{Sign}(T(sk), \cdot)$, i.e., to a signing oracle running with the tampered key $T(sk)$. As usual the adversary wins the game by outputting a valid forgery with respect to the original public key $pk$.[1] In the most general setting, the adversary is allowed to ask an arbitrary polynomial number of tampering queries. However, a general impossibility result by Gennaro *et al.* [27] shows that the above flavor of tamper resistance is unachievable without further assumptions. There are several methods of countermeasures adapted in the literature, e.g. tamper-detection and self-destruct [22, 15, 1, 14, 13, 25, 16, 2, 3, 4, 30, 23], limiting the function to simple relations [9, 33, 7, 6, 28, 35, 36, 10, 11] etc.

However in contrast to those, very recently, Damgård *et al.* [20] proposed a different approach where, instead of limiting the type of allowed modifications, they assume an upper bound on the number of tampering queries that the adversary can ask, so that now the attacker can issue some a-priori fixed number $t$ of *arbitrary* tampering queries. As argued by [20], this limitation is more likely to capture realistic tampering attacks. They also show how to construct public key encryptions and identification schemes secure against bounded leakage and tampering (BLT)[2] attack. This modeling actually fits perfectly with the setting of the Chaining Lemma as in BLT also we consider a limited number of tampering functions $(T_1, \ldots, T_t)$ for some fixed bound $t$, applied on a uniform (or close to uniform) secret-state $X^{(0)}$. Now recall that, the Chaining Lemma 1.1 guarantees that, for "small enough" $t$, the source distribution stays unpredictable in essentially "any" case. Therefore, the source can be now used as a "highly unpredictable" secret-key for any arbitrary but limited number of tampering. As a basic application of the Chaining Lemma, we show that *any* cryptographic scheme can be made secure in the BLT model. To the best of our knowledge, this is the first such general result that holds for arbitrary tampering functions and multiple tampering queries. The price we pay for this is that the tampering functions must be chosen from a bounded size set that is fixed a priori.

Previous work by Faust et al. [25], shows how to protect generically against tampering using a new primitive called *non-malleable key-derivation.* This result also works for arbitrary tampering functions, does not require that a small set of functions is fixed in advance, but works only for one-time tampering.

We discuss the application briefly in Section 4 and in detail in Appendix A.

# 2 Preliminaries

We review the basic terminology used throughout the paper.

## 2.1 Notation

For $n \in \mathbb{N}$, we write $[n] := \{1, \ldots, n\}$. Given a set $\mathcal{S}$, we write $s \leftarrow \mathcal{S}$ to denote that element $s$ is sampled uniformly from $\mathcal{S}$. If $\mathsf{A}$ is an algorithm, $y \leftarrow \mathsf{A}(x)$ denotes an execution of $\mathsf{A}$ with input $x$ and output $y$; if $\mathsf{A}$ is randomized, then $y$ is a random variable.

We denote with $k$ the security parameter. A function $\delta(k)$ is called *negligible* in $k$ (or simply negligible) if it vanishes faster than the inverse of any polynomial in $k$. A machine $\mathsf{A}$ is called *probabilistic polynomial time* (PPT) if for any input $x \in \{0,1\}^*$ the computation of $\mathsf{A}(x)$ terminates in at most $poly(|x|)$ steps and $\mathsf{A}$ is probabilistic (i.e., it uses randomness as part of its logic). Random variables are usually denoted by capital letters. We sometimes abuse notation and denote a distribution and the corresponding random variable with the same capital letter, say $X$. We write $\mathsf{sup}(X)$ for the support of $X$. Given an event $E$, we let $X_{|E}$ be the conditional distribution of $X$ conditioned on $E$ happening. The statistical distance of two random variables $X$ and $Y$, defined over a common set $\mathcal{S}$ is $\Delta(X;Y) = \frac{1}{2}\sum_{s\in\mathcal{S}}|\Pr[X=s] - \Pr[Y=s]|$. Given a random variable $Z$, the statistical distance of $X$ and $Y$ conditioned on $Z$ is defined as $\Delta(X;Y|Z) = \Delta((X,Z);(Y,Z))$.

## 2.2 Information Theory Basics

The min-entropy of a random variable $X$ over a set $\mathcal{X}$ is defined as $\mathbf{H}_\infty(X) := -\log\max_x \Pr[X=x]$, and measures how $X$ can be predicted by the best (unbounded) predictor. The conditional average min-entropy [21]

[1] Notice that $T$ may be the identity function, in which case we get the standard security notion of digital signature scheme as a special case.

[2] Although in our actual model we consider both leakage and tampering, we do not talk about leakage until Appendix A where we formally present the model in complete detail as the main result here is about tampering and the extension to leakage is straightforward.

of $X$ given a random variable $Z$ (over a set $\mathcal{Z}$) possibly dependent on $X$, is defined as

$$\widetilde{\mathbf{H}}_\infty(X|Z) := -\log \mathbb{E}_{z\leftarrow Z}[2^{-\mathbf{H}_\infty(X|Z=z)}] = \sum_{z\in\mathcal{Z}} \Pr[Z=z]\cdot 2^{-\mathbf{H}_\infty(X|Z=z)}.$$

We say that a distribution $X$ over a set $\mathcal{X}$ of size $|\mathcal{X}| = 2^n$ is $(\alpha, n)$-good if $\mathbf{H}_\infty(X) \geq \alpha$ and $\Pr[X=x] \geq 2^{-n}$ for all $x \in \mathsf{sup}(X)$.

We will rely on the following basic properties (see [21, Lemma 2.2]).

**Lemma 2.1.** *For all random variables $X, Z$ and $\Lambda$ over sets $\mathcal{X}$, $\mathcal{Z}$ and $\{0,1\}^\lambda$ such that $\widetilde{\mathbf{H}}_\infty(X|Z) \geq \alpha$, we have that*

$$\widetilde{\mathbf{H}}_\infty(X|Z,\Lambda) \geq \widetilde{\mathbf{H}}_\infty(X|Z) - \lambda \geq \alpha - \lambda.$$

The above lemma can be easily extended to the case of random variables $\Lambda$ with bounded support, i.e., $\widetilde{\mathbf{H}}_\infty(X|Z,\Lambda) \geq \widetilde{\mathbf{H}}_\infty(X|Z) - \log|\mathsf{sup}(\Lambda)|$.

**Lemma 2.2.** *For any $\epsilon > 0$, $\mathbf{H}_\infty(X|Z=z)$ is at least $\widetilde{\mathbf{H}}_\infty(X|Z) - \log(1/\epsilon)$ with probability at least $1-\epsilon$ over the choice of $z$.*

# 3 The Chaining Lemma

Before presenting the main proof of the Chaining Lemma, we state and prove two sub-lemmas. We do not provide any intuitions at this point regarding the whole proof of the Chaining Lemma due to involvement of rigorous case-analysis. Instead, we take a modular approach presenting intuitions step-by-step for each of the sub-lemmas and finally providing an intuition of the Chaining Lemma after the proof of these sub-lemmas.

The first lemma states that if the support of a distribution is sufficiently large then there always exists an event $E$ such that conditioned on $E$ the conditional distribution has high min-entropy.

**Lemma 3.1.** *For $n \in \mathbb{N}_{>1}$ let $c$ be some parameter such that $\sqrt{n} < c < n$. Let $\mathcal{X}$ be a set of size $2^n = |\mathcal{X}|$ and $X$ be a distribution over $\mathcal{X}$ with $|\mathsf{sup}(X)| > 2^c$ such that for all $x \in \mathsf{sup}(X)$ we have $\Pr[X=x] \geq \frac{1}{2^n}$. There exists an event $E$ such that:*

1. *$\mathbf{H}_\infty(X_{|E}) > c - 2\sqrt{n}$, and*
2. *$|\mathsf{sup}(X_{|\overline{E}})| < |\mathsf{sup}(X)|$.*

*Proof.* Intuitively, the lemma is proven by showing that if a distribution has sufficiently large support, then over a large subset of the support the distribution must be "almost" flat. We will describe below what it means for a distribution to be "almost flat". We then define an event $E$ that occurs when $X$ takes some value in the almost flat area. Clearly, $X$ conditioned on $E$ must be "almost" uniformly distributed, and if furthermore the support of $X$ conditioned on $E$ is still sufficiently large, we get that $\mathbf{H}_\infty(X_{|E})$ must be large. We proceed with the formal proof.

We introduce a parameter $b$ which is a positive integer such that $c > n/b$. Later we fix $b$ to its optimal value. For ease of description we assume that $n$ is a multiple of $b$. We start by defining what it means for an area to be flat. For some probability distribution $X$ we define $k \in [2^{n/b} - 1]$ sets as follows:

1. For $k \in [2^{n/b} - 1]$ we have: $I_k := \left\{ x \in \mathsf{sup}(X) : \frac{k^b}{2^n} \leq \Pr[X=x] < \frac{(k+1)^b}{2^n} \right\}$, and
2. $I_{2^{n/b}} = \{x \in \mathsf{sup}(X) : \Pr[X=x] = 1\}$.

These sets characterize the (potential) flat areas in the distribution $X$ as the probability of all values in some set $I_k$ lie in a certain range that is bounded from below and above. Clearly, the sets $I_k$ are pairwise disjoint and cover the whole space between $1/2^n$ and 1. Therefore, each $x \in \mathsf{sup}(X)$ with some probability $\Pr[X=x]$ must fall into some unique set $I_k$.

We denote by $I_m$ the set that contains the most elements among all sets $I_k$, and define the event $E$ as the event that occurs when $x \in \mathsf{sup}(X)$ falls into $I_m$, i.e., $X$ takes a value that falls in the largest set $I_m$. We now lower bound the probability that $E$ occurs.

$$\begin{aligned} \Pr[E] &\geq |I_m| \frac{m^b}{2^n} && (1) \\ &\geq 2^{c-n/b} \frac{m^b}{2^n} && (2) \end{aligned}$$

Inequality (1) holds as for all $x \in I_m$ we have $\Pr[X = x] \geq \frac{m^b}{2^n}$. (2) follows from the fact that $I_m$ must have size at least $2^{c-n/b}$, as there are $2^{n/b}$ sets and there are at least $2^c$ elements in the support of $X$.

As $\mathbf{H}_\infty(X_{|E}) = \max_x \Pr[X = x|E]$, we can give a lower bound for the min entropy of $X_{|E}$ by upper bounding $\Pr[X = x|E]$. More precisely,

$$\begin{aligned} \Pr[X = x|E] &= \frac{\Pr[X = x \wedge E]}{\Pr[E]} \\ &< \frac{(m+1)^b/2^n}{2^{(c-n/b)} m^b/2^n} && (3) \\ &= \left(1 + \frac{1}{m}\right)^b 2^{-c+n/b} \\ &\leq 2^{b-c+n/b} && (4) \end{aligned}$$

Inequality (3) uses (2) and the fact that $\Pr[X = x \wedge E] < \frac{(m+1)^b}{2^n}$ by definition of $I_m$. (4) follows from $m \geq 1$. This implies that $\mathbf{H}_\infty(X_{|E}) > c - n/b - b$ as required in the lemma.

For the second requirement, it is easy to see from the definition of $E$ that the support of the conditional probability distribution $X_{|\overline{E}}$ decreases by at least $2^{(c-n/b)}$ points (as these points belong to $E$). Clearly, $|\mathsf{sup}(X_{|\overline{E}})| \leq |\mathsf{sup}(X)| - 2^{c-n/b} < |\mathsf{sup}(X)|$ as stated in the lemma.

Now, we observe that, the loss in min-entropy, given by $(b + n/b)$ is minimum when $b = \sqrt{n}$. Since $b$ is a free parameter, we fix $b := \sqrt{n}$ (note that, since $c > \sqrt{n}$, the constraint $c > n/b$ holds) to get $\mathbf{H}_\infty(X_{|E}) > n - 2\sqrt{n}$ as stated in the lemma.

□

In the following lemma we consider an arbitrary distribution $X$ with sufficiently high min-entropy and some arbitrary function $T$. We show that if the support of $Y = T(X)$ is sufficiently large, then there exists an event $E$ such that one of the following happens:

(i) The min-entropy of $Y$ conditioned on the event $E$ is high, i.e., $Y$ conditioned on $E$ has an almost flat area with large support;

(ii) If $\overline{E}$ happens, then the average min-entropy of $X$ given $Y$ is high. Intuitively, this means that $Y$ conditioned on $\overline{E}$ has small support as then it does not "reveal" too much about $X$.

We formalize this statement in the lemma below.

**Lemma 3.2.** *For $n \in \mathbb{N}_{>1}$ let $c, \alpha$ be some parameters such that $\sqrt{n} < c < \alpha \leq n$. Let $\mathcal{X}$ be some set of size $2^n = |\mathcal{X}|$ and $X$ be an $(\alpha, n)$-good distribution over $\mathcal{X}$. For any function $T : \mathcal{X} \to \mathcal{X}$, let $Y = T(X)$ be such that $|\mathsf{sup}(Y)| > 2^c$. There exists an event $E$ such that the following holds:*

*(i)* $\mathbf{H}_\infty(Y_{|E}) > c - 2\sqrt{n}$.

*(ii)* $\widetilde{\mathbf{H}}_\infty(X_{|\overline{E}} | Y_{|\overline{E}}) \geq \alpha - c - \log \frac{1}{1-\Pr[E]}$.

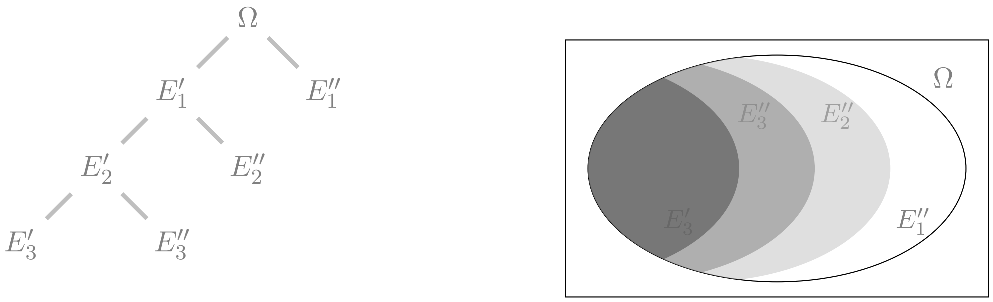

Figure 1: Events covering the probability space in the proof of Lemma 3.2 and Lemma 3.3.

*Proof.* Intuitively, in the proof below we apply Lemma 3.1 iteratively to the distribution $Y$ to find flat areas in $Y$. We “cut off” these flat areas until we have a distribution (derived from $Y$) which has sufficiently small support. Clearly such restricted $Y$ cannot reveal too much information about $X$. To formalize this approach, we construct iteratively an event $E$ by combining the events $E_i$ obtained by applying Lemma 3.1 to $Y$. If $E$ happens then $Y$ takes values that lie in a large flat area. On the other hand $\overline{E}$ characterizes only a relatively small support, and hence giving such $Y$ does not reveal much information (on average) about $X$. The formal proof with an explicit calculation of the parameters follows. We will define the event $E$ depending on events $\{E_i, E'_i, E''_i\}_{i\in\{0,\ldots,m-1\}}$ (for some integer $m$) which we will specify later. These events partition the probability space as follows (cf. Figure 1):

$$E'_i := \bigwedge_{j=0}^{i} \overline{E}_j = \overline{E}_i \wedge E'_{i-1} \qquad\qquad E''_i := E_i \wedge \left( \bigwedge_{j=0}^{i-1} \overline{E}_j \right) = E_i \wedge E'_{i-1}. \tag{5}$$

We will rely on some properties of the above partition. In particular, note that for all $i \in \{0, \ldots, m-1\}$ we have

$$E'_i \vee E''_i = E'_{i-1} \qquad\qquad E'_i \wedge E''_i = \emptyset. \tag{6}$$

We start by constructing the events $\{E_i, E'_i, E''_i\}$ and conditional probability distributions $Y^{(i)}$ that are derived from $Y$ by applying Lemma 3.1. Lemma 3.1 requires the following two conditions:

1. $|\mathsf{sup}(Y^{(i)})| > 2^c$, and
2. for all $y \in \mathsf{sup}(Y^{(i)})$ we have $\Pr[Y^{(i)} = y] \geq 2^{-n}$.

Clearly these two conditions are satisfied by $Y^{(0)} = Y$, since $Y^{(0)}$ is computed from $X$ by applying a function $T$ and for all $x \in \mathsf{sup}(X)$ the statement assumes $\Pr[X = x] \geq 2^{-n}$. Hence, Lemma 3.1 gives us an event $E_0$. We set and we define $Y^{(1)} = Y^{(0)}_{|\overline{E}_0}$. For all $i \geq 1$ we proceed to construct events $E_i$ and conditional distributions $Y^{(i+1)} = Y^{(i)}_{|\overline{E}_i}$ as long as the requirements from above in 1 and 2 are satisfied. Notice that by applying Lemma 3.1 to distribution $Y^{(i)}$ we get for each event $E_i$:

1. $\mathbf{H}_\infty(Y^{(i)}_{|E_i}) > c - 2\sqrt{n}$, and
2. $|\mathsf{sup}(Y^{(i+1)})| < |\mathsf{sup}(Y^{(i)})|$.

Clearly, there are only finitely many (say $m$) events before we stop the iteration as the size of the support is strictly decreasing. At the stopping point we have $|\mathsf{sup}(Y^{(m-1)})| > 2^c$ and $|\mathsf{sup}(Y^{(m)})| \leq 2^c$. We define $E = \bigvee_{i=0}^{m-1} E_i = \bigvee_{i=0}^{m-1} E''_i$ and $\overline{E} = \bigwedge_{i=0}^{m-1} \overline{E}_i = E'_{m-1}$ and show in the claims below that they satisfy the conditions of the lemma.

**Claim 1.** $\mathbf{H}_\infty(Y_{|E}) > c - 2\sqrt{n}$.

*Proof.* Recall that for each $0 \leq i \leq m-1$ we have

$$\begin{aligned} Y^{(i)}_{|E_i} &= Y_{|E_i \wedge \overline{E}_{i-1} \ldots \wedge \overline{E}_0} && (7) \\ &= Y_{|E''_i} && (8) \end{aligned}$$

Eq. (7) follows from the definition of the conditional probability distribution $Y^{(i)}_{|E_i}$. Eq. (8) from the definition of the constructed events. From Eq. (8) and Lemma 3.1 we have for each $0 \leq i \leq m-1$ that $\mathbf{H}_\infty(Y_{|E''_i}) > c - 2\sqrt{n}$. As for each $0 \leq i \leq m-1$ we have $|\mathsf{sup}(Y_{|E})| \geq |\mathsf{sup}(Y_{|E''_i})|$ we get that $\mathbf{H}_\infty(Y_{|E}) > c - 2\sqrt{n}$. This concludes the proof of this claim. □

**Claim 2.** $\widetilde{\mathbf{H}}_\infty(X_{|\overline{E}}|Y_{|\overline{E}}) \geq \alpha - c - \log \frac{1}{1-\Pr[E]}$.

*Proof.* We first lower bound $\mathbf{H}_\infty(X_{|\overline{E}})$.

$$\begin{aligned} \mathbf{H}_\infty(X_{|\overline{E}}) &= -\log\left(\max_x \frac{\Pr[X = x \wedge \overline{E}]}{\Pr[\overline{E}]}\right) && (9) \\ &\geq -\log\left(\frac{1}{\Pr[\overline{E}]} \max_x \Pr[X = x]\right) && (10) \\ &\geq \mathbf{H}_\infty(X) - \log\frac{1}{\Pr[\overline{E}]} \geq \alpha - \log\frac{1}{1-\Pr[E]}. && (11) \end{aligned}$$

Eq. (9) follows from the definition of min-entropy and the definition of conditional probability. Eq. (10) follows from the basic fact that for two event $\Pr[E \wedge E'] \leq \Pr[E]$. Finally, we get Eq. (11) from our assumption that $\mathbf{H}_\infty(X) \geq \alpha$. To conclude the claim we compute:

$$\begin{aligned} \widetilde{\mathbf{H}}_\infty(X_{|\overline{E}}|Y_{|\overline{E}}) &\geq \mathbf{H}_\infty(X_{|\overline{E}}, Y_{|\overline{E}}) - \log|\mathsf{sup}(Y_{|\overline{E}})| && (12) \\ &= \mathbf{H}_\infty(X_{|\overline{E}}) - \log|\mathsf{sup}(Y_{|\overline{E}})| && (13) \\ &\geq \alpha - \log\frac{1}{1-\Pr[E]} - c = \alpha - c - \log\frac{1}{1-\Pr[E]}. && (14) \end{aligned}$$

Eq. (12) follows from Lemma 2.1 and (13) from the fact that $Y_{|\overline{E}}$ is computed as a function from $X_{|\overline{E}}$. Inequality (14) follows from (11) and the fact that the size of $\mathsf{sup}(Y_{|\overline{E}})$ is at most $c$. The latter follows from the definition of the event $\overline{E} = E'_{m-1}$ which in turn implies that $|\mathsf{sup}(Y_{|\overline{E}})| = |\mathsf{sup}(Y_{|E'_{m-1}})| = |\mathsf{sup}(Y^{(m-1)}_{|\overline{E}_{m-1}})| = |\mathsf{sup}(Y^{(m)})| \leq 2^c$, which concludes the proof. □

The above two claims finish the proof. □

We now turn to the proof of the chaining lemma.

**Lemma 3.3 (The Chaining Lemma).** *For $n \in \mathbb{N}_{>1}$ let $\alpha, \beta, t, \epsilon$ be some parameters where $t \in \mathbb{N}$, $0 < \alpha \leq n$, $\beta > 0$, $\epsilon \in (0, 1]$ and $t \leq \frac{\alpha-\beta}{\beta+2\sqrt{n}}$. Let $\mathcal{X}$ be some set of size $|\mathcal{X}| = 2^n$ and let $X^{(0)}$ be a $(\alpha, n)$-good distribution over $\mathcal{X}$. For $i \in [t]$ let $T_i : \mathcal{X} \to \mathcal{X}$ be arbitrary functions and $X^{(i)} = T_i(X^{(i-1)})$. There exists an event $E$ such that:*

*(i) If $\Pr[E] > 0$, for all $i \in [t]$, $\mathbf{H}_\infty(X^{(i)}_{|E}) \geq \beta$.*

*(ii) If $\Pr[\overline{E}] \geq \epsilon$ there exists an index $j \in [t]$ such that*

$$\widetilde{\mathbf{H}}_\infty(X^{(j-1)}_{|\overline{E}}|X^{(j)}_{|\overline{E}}) \geq \beta - \log\frac{t}{\epsilon}.$$

*Proof of Lemma 3.3.* Consider the chain of random variables $X^{(0)} \xrightarrow{T_1} X^{(1)} \xrightarrow{T_2} \ldots \xrightarrow{T_t} X^{(t)}$. Given a pair of random variables in the chain, we refer to $X^{(i-1)}$ as the "source distribution" and to $X^{(i)}$ as the "target distribution". The main idea is to consider different cases depending on the characteristics of the target distribution. In case the min-entropy of $X^{(i)}$ is high enough to start with, we get immediately property (i) of the statement and we can immediately move to the next pair of random variables in the chain. In case the min-entropy of $X^{(i)}$ is small, we further consider two different sub-cases depending on some bound on the support of the variable. If the support of $X^{(i)}$ happens to be "small", intuitively we can condition on the target distribution since this cannot reveal much about the source; roughly this implies property (ii) of the statement. On the other hand, if the support happens to be not small enough, we are not in a position which allows us to condition on $X^{(i)}$.

In the latter case, we will invoke Lemma 3.2. Roughly this guarantees that there exists some event such that, conditioned on this event happening, the target lies in a large "flat" area and the conditional distribution has high min-entropy; this yields property (i) of the statement. If instead the event does not happen, then conditioning on the event not happening we get a "restricted" distribution with small enough support which leads again to property (ii) of the second statement.

Whenever we are in those cases where (possibly conditioning on some event) the target distribution has high min-entropy, we move forward in the chain by considering $X^{(i)}$ as the source and $X^{(i+1)}$ as the target. However, when we reach a situation where we can "reveal" the target distribution we do not proceed further, since the remaining values can be computed as a deterministic function of the revealed distribution and, as such, do not constrain the min-entropy further. We now proceed with the formal proof.

Similar to Lemma 3.2, we will define the event $E$ depending on events $\{E_i, E'_i, E''_i\}_{i\in[t]}$ which we will specify later. These events partition the probability space as follows (cf. Figure 1):

$$E'_i := \bigwedge_{j=1}^{i} E_j = E_i \wedge E'_{i-1} \qquad\qquad E''_i := \overline{E}_i \wedge \left(\bigwedge_{j=1}^{i-1} E_j\right) = \overline{E}_i \wedge E'_{i-1}. \tag{15}$$

We will rely on some properties of the above partition. In particular, note that for all $i \in [t]$ we have

$$E'_i \vee E''_i = E'_{i-1} \qquad\qquad E'_i \wedge E''_i = \emptyset. \tag{16}$$

For all $i \in [t+1]$, define the following parameters:

$$s_i = (t-i+1)(\beta + 2\sqrt{n}) \tag{17}$$
$$\alpha_{i-1} = \beta + s_i. \tag{18}$$

Note that using the bound on $t$ from the statement of the lemma, we get $\alpha \geq \alpha_0$; moreover, it is easy to verify that $\alpha_{i-1} > s_i > \sqrt{n}$ for all $i \in [t]$.

In the next claim we construct the events $\{E_i, E'_i, E''_i\}_{i\in[t]}$.

**Claim 3.** *For all $i = 0, \ldots, t-1$, there exist events $E'_{i+1}$ and $E''_{i+1}$ (as given in Eq. (16)) such that the following hold:*

*(*) If $\Pr\left[E'_{i+1}\right] > 0$, $\mathbf{H}_\infty(X^{(i+1)}_{|E'_{i+1}}) \geq \alpha_{i+1}$.*

*(**) If $\Pr\left[E''_{i+1}\right] \geq \epsilon'$, $\widetilde{\mathbf{H}}_\infty(X^{(i)}_{|E''_{i+1}} | X^{(i+1)}_{|E''_{i+1}}) \geq \beta - \log\frac{1}{\epsilon'}$. where $0 < \epsilon' \leq 1$.*

*Proof.* We prove the claim by induction.

**Base Case:** In this case we let $E_0$ denote the whole probability space and thus $\Pr[E_0] = 1$. note that $\mathbf{H}_\infty(X^{(0)}_{|E_0}) = \mathbf{H}_\infty(X^{(0)}) = \alpha \geq \alpha_0$. The rest of the proof for the base case is almost the same to that of the inductive step except the use of above statement instead of induction hypothesis. Therefore we only prove the induction step in detail here. The proof-details for the base case would be a straightforward extension of the induction step with some notational changes.

<u>**Induction Step:**</u> The following holds by *induction hypothesis*:

(*) If $\Pr[E'_i] > 0$, then $\mathbf{H}_\infty(X^{(i)}_{|E'_i}) \geq \alpha_i$.

(**) If $\Pr[E''_i] \geq \epsilon'$ then, $\widetilde{\mathbf{H}}_\infty(X^{(i-1)}_{|E''_i}|X^{(i)}_{|E''_i}) \geq \beta - \log\frac{1}{\epsilon'}$ where $0 < \epsilon' \leq 1$.

By construction of events, $E'_i$ is partitioned into two sub-events $E'_{i+1}$ and $E''_{i+1}$ (cf. Eq. 16). From the statement of the claim, we observe that: since we are assuming $\Pr\left[E'_{i+1}\right] > 0$ in (*) and $\Pr\left[E''_{i+1}\right] \geq \epsilon' > 0$ in (**), in both cases we have $\Pr[E'_i] > 0$. Then (*) from the induction hypothesis holds: $\mathbf{H}_\infty(X^{(i)}_{|E'_i}) \geq \alpha_i$ which we use to prove the inductive step. We will define the events $E'_{i+1}$ and $E''_{i+1}$ differently depending on several (complete) cases. For each of these cases we will show that property (*) and (**) hold.

**The case where $\mathbf{H}_\infty(X^{(i+1)}_{|E'_i}) \geq \alpha_{i+1}$.** In this case we define $E'_{i+1}$ to be $E'_i$, which implies $E''_{i+1} = \emptyset$ by Eq. (16). Moreover property (*) holds since, if $\Pr\left[E'_{i+1}\right] > 0$, then $\Pr[E'_i] > 0$ and $\mathbf{H}_\infty(X^{(i+1)}_{|E'_{i+1}}) = \mathbf{H}_\infty(X^{(i+1)}_{|E'_i}) \geq \alpha_{i+1}$; as for property (**) there is nothing to prove, since $\Pr\left[E''_{i+1}\right] = 0$ in this case.

**The case where $\mathbf{H}_\infty(X^{(i+1)}_{|E'_i}) < \alpha_{i+1}$.** Here we consider two sub-cases, depending on the support size of $X^{(i+1)}$.

1. $|\mathsf{sup}(X^{(i+1)}_{|E'_i})| \leq 2^{s_{i+1}}$. We define $E''_{i+1} = E'_i$, which implies $E'_{i+1} = \emptyset$ by Eq. (16). As for property (*) there is nothing to prove, since $\Pr\left[E'_{i+1}\right] = 0$. To prove property (**) we observe the following:
   If $\Pr\left[E''_{i+1}\right] \geq \epsilon' > 0$, then $\Pr[E'_i] > 0$. Hence,

$$\begin{aligned}
\widetilde{\mathbf{H}}_\infty(X^{(i)}_{|E''_{i+1}}|X^{(i+1)}_{|E''_{i+1}}) &= \widetilde{\mathbf{H}}_\infty(X^{(i)}_{|E'_i}|X^{(i+1)}_{|E'_i}) \qquad (19)\\
&\geq \mathbf{H}_\infty(X^{(i)}_{|E'_i}, X^{(i+1)}_{|E'_i}) - \log(|\mathsf{sup}(X^{(i+1)}_{|E'_i})|) \qquad (20)\\
&\geq \alpha_i - s_{i+1} \qquad (21)\\
&= \beta + s_{i+1} - s_{i+1} = \beta.
\end{aligned}$$

   Eq. (19) follows as $E''_{i+1} = E'_i$. Eq. (20) follows from Lemma 2.1. Eq. (21) follows from two facts: (i) $X^{(i+1)}$ is a deterministic function of $X^{(i)}$, which means $\mathbf{H}_\infty(X^{(i+1)}_{|E'_i}, X^{(i+1)}_{|E'_i}) = \mathbf{H}_\infty(X^{(i)}_{|E'_i}) \geq \alpha_i$ (plugging-in the value from induction hypothesis), and (ii) $|\mathsf{sup}(X^{(i+1)}_{|E'_i})| \leq 2^{s_{i+1}}$.

2. $|\mathsf{sup}(X^{(i+1)}_{|E'_i})| > 2^{s_{i+1}}$. Here using induction hypothesis: $\mathbf{H}_\infty(X^{(i)}_{|E'_i}) \geq \alpha_i$, we invoke Lemma 3.2 on the distribution $X^{(i+1)}_{|E'_i}$ (recall that $\alpha_i > s_{i+1} > \sqrt{n}$), to obtain the event $E_{i+1}$ such that:

$$\mathbf{H}_\infty(X^{(i+1)}_{|E'_i \wedge E_{i+1}}) > s_{i+1} - 2\sqrt{n} \tag{22}$$

$$\widetilde{\mathbf{H}}_\infty(X^{(i)}_{|E'_i \wedge \overline{E}_{i+1}}|X^{(i+1)}_{|E'_i \wedge \overline{E}_{i+1}}) > \alpha_i - s_{i+1} - \log\frac{1}{1 - \Pr[E_{i+1}]}. \tag{23}$$

   Note that by our definitions of the events $E'_i, E''_i$ (cf. Eq. (15)), we have $E'_i \wedge E_{i+1} = E'_{i+1}$ and $E'_i \wedge \overline{E}_{i+1} = E''_{i+1}$.
   Now, to prove (*) we have: if $\Pr\left[E'_{i+1}\right] > 0$, then $\Pr[E'_i] > 0$ and $\Pr[E_{i+1}] > 0$. Plugging the values of $\alpha_i$ and $s_{i+1}$ from Eq. (18) and (17) into Eq. (22), we get

$$\begin{aligned}
\mathbf{H}_\infty(X^{(i+1)}_{|E'_{i+1}}) &> s_{i+1} - 2\sqrt{n}\\
&= (t - i)(\beta + 2\sqrt{n}) - 2\sqrt{n}\\
&= \beta + (t - i - 1)(\beta + 2\sqrt{n})\\
&= \beta + s_{i+2} = \alpha_{i+1},
\end{aligned}$$

Similarly, to prove (**) we have: if $\Pr\left[E''_{i+1}\right] \geq \epsilon'$, then $\Pr\left[E'_i\right] \geq \epsilon' > 0$ and $\Pr\left[\overline{E}_{i+1}\right] \geq \epsilon'$. Using Eq. (23), we obtain:

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(X^{(i)}_{|E''_{i+1}} | X^{(i+1)}_{|E''_{i+1}}) &> \alpha_i - s_{i+1} - \log \frac{1}{\Pr\left[\overline{E}_{i+1}\right]} \\ &= \beta - \log \frac{1}{\Pr\left[\overline{E}_{i+1}\right]} \\ &\geq \beta - \log \frac{1}{\epsilon'},\end{aligned}$$

This concludes the proof of the claim. □

We define the event $E$ to be $E = E'_t = \bigwedge_{i=1}^t E_i = \bigwedge_{i=1}^t E'_i$. It is easy to verify that this implies $\overline{E} = \bigvee_{i=1}^t E''_i$. We distinguish two cases:

- If $\Pr[E] > 0$, by definition of $E$ we get that $\Pr[E'_i] > 0$ for all $i \in [t]$. In particular, $\Pr[E'_t] > 0$. Hence, $\mathbf{H}_\infty(X^{(t)}_{|E}) = \mathbf{H}_\infty(X^{(t)}_{|E'_t}) \geq \alpha_t = \beta$, where the last inequality follows from (*) of Claim 3 putting $i = t-1$. Also, we observe that for all $i \in [t]$, $\mathbf{H}_\infty(X^{(i-1)}_{|E}) \geq \mathbf{H}_\infty(X^{(i)}_{|E})$. This proves property (i) of the lemma.

- If $\Pr\left[\overline{E}\right] \geq \epsilon$, then we get

$$\Pr\left[\bigvee_{i=1}^t E''_i\right] \geq \epsilon. \tag{24}$$

$$\sum_{i=1}^t \Pr\left[E''_i\right] \geq \epsilon. \tag{25}$$

Eq. 24 follows from the definition of $E$ and Eq. 25 follows applying union bound. Clearly, from Eq. 25, there must exists some $j$ such that $\Pr\left[E''_j\right] \geq \epsilon/t$.

Hence, putting $i = j-1$ and $\epsilon' = \epsilon/t$ in (**) of Claim 3, we get:

$$\widetilde{\mathbf{H}}_\infty(X^{(j-1)}_{|E''_j} | X^{(j)}_{|E''_j}) \geq \beta - \log \frac{t}{\epsilon}.$$

From the definition of $E$, $E''_j$ implies $\overline{E}$ and hence property (ii) of the lemma follows.

□

# 4 Application to Tamper-resilient Cryptography (overview)

We show that *any* cryptographic primitive where the secret key can be chosen as a uniformly random string can be made secure in the BLT model by a simple and efficient transformation. Our results therefore cover pseudorandom functions (PRF), block ciphers and many encryption and signature schemes. However, the result holds in a restricted model of tampering: the adversary first selects an arbitrary set of tampering functions of bounded size and as he interacts with the scheme. He must choose every tampering function from the set that was specified initially. We call this the *semi-adaptive* BLT model. Our result holds only when the set of functions is "small"[3].

The basic intuition behind the construction using the Chaining Lemma is easy to explain. We use a random string $X_0$ as secret key, and a universal hash function $h$ as public (and tamper proof) parameter. The

[3] In particular, the adversary can choose a "short enough" sequence of tampering functions from a set containing polynomially many such sequences.

construction then computes $K_0 = h(X_0)$ and uses $K_0$ as secret key for the original primitive. The intuitive reason why one might hope this would work is as follows: each tampering query changes the key, so we get a chain of keys: $X^{(0)} \xrightarrow{T_1} X^{(1)} \xrightarrow{T_2} X^{(2)} \dots \xrightarrow{T_t} X^{(t)}$. Recall that the chaining lemma guarantees that for such a chain, there exists an event $E$ such that (i) when $E$ takes place then all $X_i$ have high min-entropy, then we can show that with a suitable choice of $h$, all the hash values $K_0 = h(X_0), K_1 = h(X_1), \dots,$ are statistically close to uniformly and independently chosen keys; (ii) otherwise when $E$ does not happen, for some index $j \in \{1, \dots, t\}$ we should be able to reveal the value of $X_j$ to the adversary as the $X_i$'s with $i < j$ should still have high entropy and hence hash to independent values. On the other hand the $X_i$'s with $i \geq j$ are deterministic function of $X_j$ and hence the tampering queries corresponding to any subsequent key can be simulated by the adversary himself.

Due to its generality the above result suffers from two limitations. First, as already mentioned above (without making non-standard assumptions) the tampering has to satisfy a somewhat limited form of adaptivity. Second, the number of tampering queries one can tolerate is upper bounded by the length $n$ of the secret key. While this is true in general for schemes without key update, for our general result the limitation is rather strong. More concretely, with appropriately chosen parameters our transformation yields schemes that can tolerate up to $O(\sqrt[3]{n})$ tampering queries.

We discuss the application in full detail in Section A

**Comaprison with Faust *et al.* [25].** Very recently, Faust *et al.* [25] introduced the concept of non-malleable key derivation which is similar in spirit with this particular application of the Chaining Lemma. Intuitively a function $h$ is a non-malleable key derivation function if $h(X)$ is close to uniform even given the output of $h$ applied to a related input $T(X)$, as long as $T(X) \neq X$. They show that a random $t$-wise independent hash function already meets this property, and moreover that such a function can be used to protect arbitrary cryptographic schemes (with a uniform key) against "one-time" tampering attacks (i.e., the adversary is allowed a single tampering query) albeit against a much bigger class of functions[4]

However, we stress that the novelty of our result is in discovering the Chaining Lemma rather than this application, which can be instead thought of as a new technique, fundamentally different from that of [25] to achieve tamper-resilience in BLT model. We believe that, the Chaining Lemma itself is a new discovery from the perspective of information theory and the new techniques might find more application in cryptography in future.

## References

[1] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:81, 2013. To appear in STOC 2014.

[2] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations. *IACR Cryptology ePrint Archive*, 2014:316, 2014.

[3] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations and perturbations. *IACR Cryptology ePrint Archive*, 2014:841, 2014.

[4] Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit optimal-rate non-malleable codes against bit-wise tampering and permutations. *IACR Cryptology ePrint Archive*, 2014:842, 2014.

[5] Ross Anderson and Markus Kuhn. Tamper resistance: a cautionary note. In *WOEC'96: Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce*, pages 1–1, Berkeley, CA, USA, 1996. USENIX Association.

[4] However, if they extend their model to multiple (but bounded) time tampering, it might be possible that they could achieve the same parameters in the semi-adaptive model but against much larger tampering family, essentially subsuming our result.

[6] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In *ICS*, pages 45–60, 2011.

[7] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO*, pages 666–684, 2010.

[8] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT*, pages 486–503, 2011.

[9] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In *EUROCRYPT*, pages 491–506, 2003.

[10] Mihir Bellare, Kenneth G. Paterson, and Susan Thomson. RKA security beyond the linear barrier: IBE, encryption and signatures. In *ASIACRYPT*, pages 331–348, 2012.

[11] Rishiraj Bhattacharyya and Arnab Roy. Secure message authentication against related key attack. In *FSE*, 2013.

[12] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of eliminating errors in cryptographic computations. *J. Cryptology*, 14(2):101–119, 2001.

[13] Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *ITCS*, pages 155–168, 2014.

[14] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *TCC*, pages 440–464, 2014.

[15] Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. Bitr: Built-in tamper resilience. In *ASIACRYPT*, pages 740–758, 2011.

[16] Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. *IACR Cryptology ePrint Archive*, 2014:324, 2014.

[17] Dana Dachman-Soled and Yael Tauman Kalai. Securing circuits against constant-rate tampering. In *CRYPTO*, pages 533–551, 2012.

[18] Dana Dachman-Soled and Yael Tauman Kalai. Securing circuits and protocols against 1/poly(k) tampering rate. In *TCC*, pages 540–565, 2014.

[19] Dana Dachman-Soled, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Locally decodable and updatable non-malleable codes and their applications. Cryptology ePrint Archive, Report 2014/663, 2014. `http://eprint.iacr.org/`.

[20] Ivan Damgård, Sebastian Faust, Pratyay Mukherjee, and Daniele Venturi. Bounded tamper resilience: How to go beyond the algebraic barrier. In *ASIACRYPT (2)*, pages 140–160, 2013.

[21] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[22] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.

[23] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *TCC*, pages 465–488, 2014.

[24] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. A tamper and leakage resilient random access machine. *IACR Cryptology ePrint Archive*, 2014:338, 2014.

[25] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *EUROCRYPT*, pages 111–128, 2014.

[26] Sebastian Faust, Krzysztof Pietrzak, and Daniele Venturi. Tamper-proof circuits: How to trade leakage for tamper-resilience. In *ICALP (1)*, pages 391–402, 2011.

[27] Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. In *TCC*, pages 258–277, 2004.

[28] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In *TCC*, pages 182–200, 2011.

[29] Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In *EUROCRYPT*, pages 308–327, 2006.

[30] Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. Cryptology ePrint Archive, Report 2014/956, 2014. `http://eprint.iacr.org/`.

[31] Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *CRYPTO*, pages 373–390, 2011.

[32] Eike Kiltz, Krzysztof Pietrzak, Martijn Stam, and Moti Yung. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT*, pages 590–609, 2009.

[33] Stefan Lucks. Ciphers secure against related-key attacks. In *FSE*, pages 359–370, 2004.

[34] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[35] Krzysztof Pietrzak. Subspace lwe. In *TCC*, pages 548–563, 2012.

[36] Hoeteck Wee. Public key encryption against related key attacks. In *Public Key Cryptography*, pages 262–279, 2012.

# A Application of Chaining Lemma to Tamper-Resilient Cryptography (in detail)

## A.1 Abstract Security Games with Tampering

We start by defining a general structure of abstract security games for cryptographic schemes $\mathcal{CS}$. Most standard security notions such as IND-CPA or pseudorandomness of PRFs follow the structure given below, and we will later give concrete instantiations of our abstract definition. We then show how to extend such games to the BLT setting. Consider some cryptographic scheme $\mathcal{CS}$ with associated key generation algorithm $\mathsf{KeyGen}$. $\mathsf{KeyGen}$ outputs a secret key $X$, and in some cases some public parameters $pp$. We will grant the adversary access to an oracle $\mathcal{O}(pp, X, \cdot)$. The definition of the oracle depends on the actual security definition. For instance, it can be a signing oracle or an oracle that gives access to the outputs of a PRF with key $K$. To simplify notation, we will assume that such oracles can be multi-functional. That is, if we want to model CCA security of a symmetric encryption scheme, $\mathcal{O}$ offers interfaces for both encryption and decryption queries.

**Definition A.1** (Abstract security game)**.** Let $k \in \mathbb{N}$ be the security parameter and let $\mathcal{CS}$ be a cryptographic scheme with key generation algorithm $\mathsf{KeyGen}$. Let $\mathcal{O}(X, pp, \cdot)$, $\mathsf{C}(X, pp, \cdot)$ be some oracles, where $\mathsf{C}(X, pp, \cdot)$ is the challenge oracle that outputs at the end a bit $b$. We assume in the following that both oracles share state. An abstract security game $\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k)$ consists of 3 phases given below.

1. *Setup:* Run the key generation $(X, pp) \leftarrow \mathsf{KeyGen}(1^k)$. The public parameters $pp$ are given to $\mathsf{A}$.

2. *Query phase:* The adversary gets access to the oracle $\mathcal{O}(X, pp, \cdot)$ and is allowed to query them in any order.

3. *Challenge phase:* The adversary looses access to all oracles and interacts with the challenge oracle $\mathsf{C}(X, pp, \cdot)$ that at the end outputs a bit $b$. $b$ is returned by the game, where $b = 1$ indicates that $\mathsf{A}$ won the game.

We define the security of a cryptographic scheme according to the above definition depending on whether it is an unpredictability or indistinguishability type game.

- For *unpredictability* security games we say that a scheme is $\delta(k)$-secure if for any PPT adversary $\mathsf{A}$ the advantage of $\mathsf{A}$ is
$$\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(1^k) = 1] \leq \delta(k).$$

- For *indistinguishability* games we say that a scheme is $\delta(k)$-secure if for any PPT adversary $\mathsf{A}$ the advantage of $\mathsf{A}$ is
$$\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(1^k) = 1] - 1/2 \leq \delta(k).$$

For indistinguishability games, we assume wlog. that the challenger $\mathsf{C}$ internally keeps a bit $b$ and $\mathsf{A}$ submits as its last message to $\mathsf{C}$ a bit $b'$. If $b = b'$ then the challenger returns 1; otherwise 0. In the following, we will usually omit the parameter $\delta(k)$ and just say that a scheme is secure if $\delta(k)$ is negligible in $k$.

We now extend the above definition to the BLT setting. We will give the extension specifically for the case of a semi-adaptive choice of tampering functions. Notice that BLT security can also be defined for a fully adaptive choice, but our general construction from this section does not achieve this stronger notion. We emphasize, however, that for some specific schemes fully adaptive BLT security can be achieved for free (as shown in [20]). We start by providing some intuitive description of *semi-adaptive* BLT security.

In contrast to Definition A.1, in the BLT setting the adversary $\mathsf{A}$ can learn $\lambda$ bits about the key in the setup phase. More importantly, after the query phase he may decide on a particular choice of tampering functions $T \in \mathcal{T}$ and gets access to the tampered oracles. Finally, in the challenge phase, he looses access to all oracles (including the tampered oracles) and plays the standard security game against the challenger $\mathsf{C}(X, pp, \cdot)$. As in Definition A.1 the challenge oracle outputs a bit indicating whether the adversary won or lost the BLT security game.

**Definition A.2** (Abstract BLT security game)**.** Let $\lambda, t, v$ be functions in the security parameter $k \in \mathbb{N}$ and let $\mathcal{CS}$ be a cryptographic scheme with key generation algorithm $\mathsf{KeyGen}$. Let $\mathcal{O}(X, pp, \cdot)$, $\mathsf{C}(X, pp, \cdot)$ be some oracles, where $\mathsf{C}(X, pp, \cdot)$ is the challenge oracle that outputs at the end either 0 or 1. We assume in the following that all these oracles share state. An abstract BLT-security game $\mathsf{BLT}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k, t, \lambda, v)$ consists of 4 phases given below.

1. *Setup:* Run $(X, pp) \leftarrow \mathsf{KeyGen}(1^k)$ and obtain from $\mathsf{A}$ a description of a leakage function $L : \mathcal{X} \to \{0,1\}^\lambda$ and a set of tolerated tampering functions $\mathcal{T} = \{(T_1, \ldots, T_t) | T_i : \mathcal{X} \to \mathcal{X}\}$ with $|\mathcal{T}| = v$. Give $pp, L(X)$ to the adversary.

2. *Query phase:* The adversary gets access to the oracle $\mathcal{O}(X, pp, \cdot)$ and is allowed to query them in any order.

3. *Tampering phase:* The adversary decides on a choice of tampering functions $T = (T_1, \ldots, T_t) \in \mathcal{T}$ and gets access to the tampered oracles. That is, he can query (in any order) the tampered oracles $\mathcal{O}(T_1(X_0), pp, \cdot), \ldots \mathcal{O}(T_t(X_{t-1}), pp, \cdot)$, where $X_0 = X$ and $X_i = T_i(X_{i-1})$.

4. *Challenge phase:* The adversary looses access to all oracles and interacts with the challenge oracle $\mathsf{C}(pp, X, \cdot)$ that eventually outputs a bit $b$. $b$ is returned by the game and indicates if $\mathsf{A}$ won the game.

We define BLT security for unpredictability or indistinguishability type games analogous to standard security where the adversary now runs in the BLT game as given by Definition A.2.

**Definition A.3** (Semi-adaptive BLT security of $\mathcal{CS}$)**.** We say that a cryptographic scheme $\mathcal{CS}$ is $(\lambda, t, v)$*-BLT-secure* in the semi-adaptive BLT model if for all PPT adversaries $\mathsf{A}$ we have

$$\Pr[\mathsf{BLT}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k, t, \lambda, v) = 1] \leq negl(k).$$

Here, $|\mathcal{T}| = v$ and each element of $\mathcal{T}$ is a tuple $(T_1, \ldots, T_t)$ of tampering functions.

Some remarks are in order to explain the relation between $\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k)$ and the BLT game $\mathsf{BLT}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k, t, \lambda, v)$. Consider for instance the standard security notion of existential unforgeability of digital signature scheme. Our abstract security game from Definition A.1 clearly covers this security notion when $\mathcal{O}(pp, K, \cdot)$ is the signing oracle and the challenge oracle returns 1 if the adversary submits a valid forgery to $\mathsf{C}$ during the challenge phase. Another example is the indistinguishability based security notion of PRFs. Here, $\mathcal{O}$ is the oracle that returns outputs of the PRF on inputs of the adversary's choice, while the challenge oracle returns either the output of the PRF or the output of a random function. Notice that in this case the challenge oracle returns $\perp$ for inputs that have been already queried to oracle $\mathcal{O}$. As in the standard definition for PRFs the adversary wins the game when he can correctly tell apart random or real queries.

The BLT security game extends these abstract security games by giving the adversary additional access to tampered oracles. That is, for all an oracle $\mathcal{O}$ that the adversary can access in the standard security game, he now gets access to a "tampered copy" of this oracle. More precisely, for any oracle $\mathcal{O}$ from the standard security game and for any set of tampering functions $(T_1, \ldots, T_t)$ the adversary gets now additionally access to the oracles $\mathcal{O}(pp, T_i(X_{i-1}), \cdot)$.

## A.2 Semi-Adaptive BLT Security for General Primitives

In this section, we show that "any" primitive (e.g., any PRF or signature scheme) can be made BLT secure by combining it with a universal hash function. We put forward a notion of semi-adaptive BLT security for general primitives in Section A.1. In Section A.2.1, we describe our transformation based on universal hashing and state our main theorem (Theorem A.1). Section A.2.2 contains a high-level overview of the proof of Theorem A.1; a formal proof appears in Section A.2.4 and A.2.3. Finally, in Section A.3 we discuss a few extensions of our main theorem.

**Notation for this section.** In this section $n = poly(k)$ denotes the length of the key unless explicitly mentioned otherwise, where $k$ is the security parameter. Given some event $E$ we write $\widetilde{\mathbf{H}}_\infty(X|Y_1, Y_2, \ldots, E)$ to denote that every random variable is conditioned on the event $E$.

### A.2.1 A General Transformation

We now describe a general transformation to leverage security of a cryptographic scheme $\mathcal{CS}$ (as per Definition A.1) to semi-adaptive BLT security (as per Definition A.3). The transformation is based on a family $\mathcal{H} = \{h_S : \mathcal{X} \rightarrow \mathcal{Y}\}$ of $(2t+1)$-wise independent hash functions. Recall that $\mathcal{H}$ is called $t$-wise independent if for any sequence of distinct elements $X^{(1)}, \ldots, X^{(t)} \in \mathcal{X}$ the random variables $h_S(X^{(1)}), \ldots, h_S(X^{(t)})$ are uniform, where $h_S \leftarrow \mathcal{H}$.[5]

**The transformation.** Consider a cryptographic scheme $\mathcal{CS}$ with a key generation algorithm $\mathsf{KeyGen}$ outputting a secret key $X$ and public parameters $pp$. In the following we consider schemes that have a BLT admissible key generation algorithm. That is, (1) the secret key $X$ is sampled uniformly at random from the key space $\mathcal{X}$, and (2) given the secret key $X$, there exists an efficient algorithm $\mathsf{KeyGen}'(X)$ that takes as input $X$ and outputs corresponding public parameters $pp$. Furthermore, let $\mathsf{CS}$ be some cryptographic algorithm

[5] A concrete construction is given by the following function $h_S : \mathbb{Z}_p \rightarrow \mathbb{Z}_p$: Sample $S$ by choosing $t$ random elements $s_0, s_1, \ldots, s_{t-1} \leftarrow \mathbb{Z}_p^t$ and define $h_S(X) = s_0 + s_1 \cdot X + \ldots + s_{t-1} \cdot X^{t-1} \bmod p$.

that uses the secret key $X$, the public parameters $pp$ and some input $M$ and outputs $Z \leftarrow \mathsf{CS}(pp, K, M)$. For instance, $\mathcal{CS}$ may be a block cipher and $\mathsf{CS}$ the associated encryption algorithm with $M$ being the message and $Z$ the corresponding ciphertext. We transform $\mathcal{CS}$ into $\mathcal{CS}'$ as follows. Let $\mathcal{H} = \{h_S : \mathcal{X} \to \mathcal{Y}\}_{S \in \mathcal{S}}$ be a family of $2(t+1)$-wise independent hash functions. At setup we sample a key for the hash function $S$ and $X \leftarrow \mathcal{X}$ uniformly at random and compute $K = h_S(X)$. We then run $\mathsf{KeyGen}'(K)$ to sample the corresponding public parameters $pp$, where $\mathsf{KeyGen}$ is the underlying key generation of $\mathcal{CS}$. Let $X$ be the secret key of $\mathcal{CS}'$ and $pp' = (pp, S)$ the corresponding public parameters. To compute the cryptographic algorithm $\mathsf{CS}'$ on some input $M$, we run $Z \leftarrow \mathsf{CS}(pp, h_S(X), M)$, i.e., we map the key $X$ for $\mathcal{CS}'$ to the key $K$ for the underlying cryptoscheme $\mathcal{CS}$ by applying the hash function.

The theorem below states that the above transformation is BLT-secure in the semi-adaptive BLT model whenever $\mathcal{CS}$ is secure in the standard sense (cf. Definition A.1 and its key generation algorithm is BLT admissible.

**Theorem A.1.** *If $\mathcal{CS}$ is secure and $\mathcal{H} = \{h_S : \mathcal{X} \to \mathcal{Y}\}_{S \in \mathcal{S}}$ is a family of $2(t+1)$-wise independent hash functions with $|\mathcal{X}| = 2^n$ and $|\mathcal{Y}| = 2^\ell$, then we have that $\mathcal{CS}'$ is $(\lambda, t, \delta, v)$-secure in the semi-adaptive BLT model, where*

$$\lambda = O(\sqrt[3]{n}) \qquad t = O(\sqrt[3]{n}) \qquad \delta \le negl(k) \qquad v = O(n^d) \qquad \ell = O(\sqrt[4]{n}),$$

*for some constant $d > 0$.*

Concretely, we can think of $\mathcal{CS}$ being a PRF (or a signature scheme) with security in the standard sense, i.e., the adversary has negligible advantage when playing against the underlying challenger. The Theorem A.1 says that, for sufficiently large $n$, the transformed PRF $\mathcal{CS}'$ achieves semi-adaptive BLT security against adversaries tampering $O(\sqrt[3]{n})$ times and leaking $O(\sqrt[3]{n})$ bits from the original key. Notice that the hash function compresses the $n$-bit input to $O(\sqrt[4]{n})$ bits and the set of admissible (sequences of) tampering functions has size $O(n^d)$ for some constant $d > 0$. Notice that if the underlying primitive is super-polynomial secure than we can increase the size of admissible tampering functions. In the extreme case when the underlying primitive has exponential security, the size of $\mathcal{T}$ may be sub-exponentially large.

We emphasize that we can obtain stronger leakage resilience as we inherit the security properties from the underlying cryptoscheme $\mathcal{CS}$. Hence, if $\mathcal{CS}$ is secure against adaptive leakage attacks from the key $K$, then also $\mathcal{CS}'$ is secure against adaptive leakage attacks from the key $h_S(X)$ used by the actual cryptographic scheme.

### A.2.2 Outline of the Proof

We explain the intuition and the main ideas behind the proof of Theorem A.1. The proof is by reduction: Given an adversary $\mathsf{A}$ with non-negligible advantage in the semi-adaptive BLT game for $\mathcal{CS}'$ (cf. Definition A.3), we build an adversary $\mathsf{B}$ against standard security of $\mathcal{CS}$ (cf. Definition A.1). The main difficulty is that $\mathsf{B}$ has only access to the standard oracle $\mathcal{O}(pp, K)$, so it is not a priori clear how $\mathsf{B}$ can answer $\mathsf{A}$'s tampering queries and simulate the tampered oracles.

The idea is to let $\mathsf{B}$ sample the initial key $X^{(0)}$ independently of the target key $K$ (which is anyway not known to $\mathsf{B}$) and compute the keys $X^{(1)}, \ldots, X^{(t)}$ as specified by the tampering functions $T_i$ in order to simulate the tampered view of $\mathsf{A}$, i.e., the oracles $\mathcal{O}(pp', T_i(X^{(i-1)}))$. To give a first hint why this may indeed be a good strategy, consider the simple case where all tampered keys have high min-entropy (say higher than some threshold $\beta$). In this case, we can rely on a property of $2(t+1)$-wise independent hashing, namely for a uniformly sampled hash function $h_S$ the tuple $(h_S(X^{(0)}), h_S(X^{(1)}), \ldots, h_S(X^{(t)}))$ is statistically close to uniform and thus $\mathsf{B}$'s simulation of $\mathsf{A}$'s view is indistinguishable from the real view. The proof is a straightforward extension of [32, Lemma 3.2]. and is deferred to Appendix B.

**Lemma A.1.** *Let $(X_1, X_2, \ldots, X_t) \in \mathcal{X}^t$ be $t$ (possibly dependent) random variables such that $\mathbf{H}_\infty(X_i) \ge \beta$ and $(X_1, \ldots, X_t)$ are pairwise different. Let $\mathcal{H} = \{h_S : \mathcal{X} \to \mathcal{Y}\}$ be a family of $2t$-wise independent hash functions, with $|\mathcal{Y}| = 2^\ell$. Then for random $h_S \leftarrow \mathcal{H}$ we have that*

$$\Delta((h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)); (h_S, \underbrace{U_{\mathcal{Y}}, \ldots, U_{\mathcal{Y}}}_{t\ times})) \le \frac{t}{2} \cdot 2^{(t \cdot \ell - \beta)/2}.$$

Of course, in our general tampering model nothing guarantees that all keys have high min-entropy, and hence we cannot immediately apply Lemma A.1. At this point, a careful reader may object that at the end this does not matter too much: if the compression of the hash function is high enough (as it is the case for our choice of $\ell \approx \sqrt[4]{n}$ in Theorem A.1) the hashed keys are short anyway, and thus the entropy of $X^{(0)}$ given the hash of the tampered keys remains high. At this point it looks tempting to apply the leftover hash lemma, and argue that $h_S(X^{(0)})$ is statistically close to uniform even given the hashed tampered keys. The leftover hash lemma, however, requires that the key $S$ can be sampled uniformly and independently from the distribution of $X^{(0)}$. Unfortunately, the conditional distribution of $X^{(0)}$ (given the tampered hashed keys) may now depend on $S$, and we cannot apply the leftover hash lemma directly.

At this point we use the Chaining Lemma. We restate the Chaining Lemma 3.3 below for reader's convenience.

**Lemma 3.3.** *For $n \in \mathbb{N}_{>1}$ let $\alpha, \beta, t, \epsilon$ be some parameters where $t \in \mathbb{N}$, $0 < \alpha \leq n$, $\beta > 0$, $\epsilon \in (0,1]$ and $t \leq \frac{\alpha-\beta}{\beta+2\sqrt{n}}$. Let $\mathcal{X}$ be some set of size $|\mathcal{X}| = 2^n$ and let $X^{(0)}$ be a $(\alpha, n)$-good distribution over $\mathcal{X}$. For $i \in [t]$ let $T_i : \mathcal{X} \to \mathcal{X}$ be arbitrary functions and $X^{(i)} = T_i(X^{(i-1)})$. There exists an event $E$ such that:*

*(i) If $\Pr[E] > 0$, for all $i \in [t]$, $\mathbf{H}_\infty(X^{(i)}_{|E}) \geq \beta$.*

*(ii) If $\Pr[\overline{E}] \geq \epsilon$ there exists an index $j \in [t]$ such that*

$$\widetilde{\mathbf{H}}_\infty(X^{(j-1)}_{|\overline{E}} | X^{(j)}_{|\overline{E}}) \geq \beta - \log\frac{t}{\epsilon}.$$

Instead of the real experiment we can now turn to a mental experiment where at some point in the chain we reveal an entire source $X^{(i)}$. By the Chaining Lemma 3.3 we are guaranteed that $X^{(0)}, X^{(1)}, \ldots, X^{(i-1)}$ individually all have high min-entropy even given $X^{(i)}$, which allows us to apply Lemma A.1 and conclude that $h_S(X^{(0)}), \ldots, h_S(X^{(i-1)})$ are jointly close to uniform. Notice that in the mental experiment clearly the remaining sources $X^{(0)}, X^{(1)}, \ldots, X^{(i-1)}$ remain independent from $S$ even given $X^{(i)}$. At this point we are almost done except for two technical difficulties: (1) the Lemma A.1 requires that all $X^{(j)}$ (for $j < i$) are pairwise distinct, and (2) the adversary picks its tampering choice *adaptively* from a fixed set $\mathcal{T}$ after seeing the key for the hash function $S$ and after interacting with the original challenger (the so-called semi-adaptive model). We solve the first by changing the above mental experiment and eliminate all sources that appear multiple times in the source chain. We then show that given a short advice we can re-sample the complete $X^{(0)}, \ldots, X^{(t)}$ from the reduced chain. To complete the proof, we address the semi-adaptivity mentioned in (2) by a counting argument as the size of the set of potential tampering queries $\mathcal{T}$ is not too big (polynomial in the security parameter).

We conclude the above outline by defining two experiments that describe how the keys $X^{(0)}, X^{(1)}, \ldots, X^{(t)}$ are sampled in the real game and in the simulation. For $t, \lambda \in \mathbb{N}$ and any set of functions $T_1, \ldots, T_t : \mathcal{X} \to \mathcal{X}$, $L : \mathcal{X} \to \{0,1\}^\lambda$ consider the two experiments as given in Figure 2. In the lemma below we show that for a distribution $X$ with a sufficient amount of min-entropy and certain set of carefully chosen parameters the distance between $\mathsf{Real}(X, \mathcal{T}, L)$ and $\mathsf{Sim}(X, \mathcal{T}, L)$ is statistically close. We will in the following omit to explicitly mention the inputs to the experiments.

**Lemma A.2.** *Denote with $k \in \mathbb{N}$ the security parameter and let $n, t, q, v, \lambda, \epsilon, \ell, \alpha$ be functions in $k$ such that $\lambda, t < \alpha \leq n$ and $\epsilon \in (0, 1/2)$. Let $\mathcal{H} = \{h_S : \mathcal{X} \to \mathcal{Y}\}$ be a family of $q$-wise independent hash functions and $\mathcal{T} = \{T_i : \mathcal{X} \to \mathcal{X}\}$, such that $|\mathcal{T}| = 2^v$. Let $|\mathcal{X}| = 2^n$, $|\mathcal{Y}| = 2^\ell$, and $X$ be an $(\alpha, n)$-good distribution over $\mathcal{X}$. For all sequences of functions $(T_1, \ldots, T_t) \in \mathcal{T}$, and for all $L : \mathcal{X} \to \{0,1\}^\lambda$ as specified in Figure 2:*

$$\Delta(\mathsf{Real}; \mathsf{Sim}) \leq 2^{t(v+\ell)+2} \left(\frac{q}{4t\epsilon^2 2^{c-t\ell}}\right)^{\frac{q}{2t}} + 6\epsilon,$$

*where $c := \beta - 2\log t/\epsilon - \lambda - 2t\log(t)$ and $\beta := \frac{\alpha - 2t\sqrt{n}}{t+1}$.*

Some comments are in order to explain the mechanics of the parameters defining the statistical distance between $\mathsf{Real}$ and $\mathsf{Sim}$ in Lemma A.2. To obtain a negligible quantity (as we will need for the proof of Theorem A.1 in Section A.2.3), the value $c$ must be chosen to be sufficiently larger than the value $(t+1) \cdot \ell$; this

**Experiment** $\mathsf{Real}$ **vs.** $\mathsf{Sim}$

1. Experiment $\mathsf{Real}(X, \mathcal{T}, L)$*:* Let $X^{(0)}$ be a random variable with distribution $X$ and $h_S \leftarrow \mathcal{H}$ a uniformly sampled hash function. For a sequence of functions $T_1, \dots, T_t \in \mathcal{T}$ let $X^{(i)} = T_i(X^{(i-1)})$ and output:

$$\mathsf{Real} := (D_0, \dots, D_{t+2}) = \Big( h_S(X^{(0)}), \dots, h_S(X^{(t)}), L(X^{(0)}), S \Big) .$$

2. Experiment $\mathsf{Sim}(X, \mathcal{T}, L)$*:* Let $X^{(0)}$ be a random variable with distribution $X$ and $h_S \leftarrow \mathcal{H}$ a uniformly sampled hash function. For a sequence of functions $T_1, \dots, T_t \in \mathcal{T}$ let $X^{(i)} = T_i(X^{(i-1)})$, and proceed as follows:

```
Sample $D_0 \leftarrow U_{\mathcal{Y}}$.
For $i \in [t]$ compute:
    If $X^{(i)} \neq X^{(0)}$ then $D_i = h_S(X^{(i)})$
    Else $D_i = D_0$
Output $\mathsf{Sim} = (D_0, \dots, D_t, L(X^{(0)}), S)$.
```

Figure 2: Experiment $\mathsf{Real}$ denotes the real tampering experiment and $\mathsf{Sim}$ our simulation.

shows a clear trade-off between the value of $t$ and the value of $\ell$. We instantiate Lemma A.2 with concrete values in the following corollary. It shows a setting where we try to maximize the number of tampering queries we can tolerate by using a very high compression factor.

**Corollary 1.** *For sufficiently large $n$, if we set $\ell = O(\sqrt[4]{n})$, $\lambda = O(\sqrt[3]{n})$, $\alpha = n - O(\sqrt[3]{n})$ and $\epsilon = \exp(-\Theta(\sqrt[3]{n}))$ in Lemma A.2 we get $t = O(\sqrt[3]{n})$ for which the distance $\Delta(\mathsf{Real}; \mathsf{Sim}) \leq \exp(-\Omega(\sqrt[3]{n}))$ in $n$.*

### A.2.3 Proof of Theorem A.1

We now turn to the proof of Theorem A.1.

*Proof of Theorem A.1.* Suppose there exists an adversary $\mathsf{A}$ and a polynomial $p(.)$ such that $\mathsf{A}$ breaks the $(\lambda, t, v)$ semi-adaptive BLT security of $\mathcal{CS}'$ with advantage at least $1/p(k)$ for infinitely many $k$. Then, we construct an adversary $\mathsf{B}$ that breaks the security of $\mathcal{CS}$ according to the challenge oracle $\mathsf{C}(pp, K)$ with advantage at least $1/p'(k)$ for some polynomial $p'(.)$. To this end, adversary $\mathsf{B}$ needs to simulate the environment specified by BLT game according to Definition A.2 to $\mathsf{A}$ given only access to its target oracle $\mathcal{O}(pp, K)$ and $\mathsf{C}(pp, K)$. At a high-level this simulation is carried out as follows: $\mathsf{B}$ uses its target oracles to simulate the interaction of $\mathsf{A}$ in the query and challenge phase. In the tampering phase, it will either use access to $\mathcal{O}(pp, K)$ (if the adversary did not tamper with the key) or simulates the tampered view with the keys sampled uniformly at random. The simulation closely follows the structure of the BLT game as specified in Definition A.2 and is given below:

1. *setup phase:* In the first step $\mathsf{B}$ receives a leakage function $L : \mathcal{X} \rightarrow \{0,1\}^{\lambda}$ and a set of tolerated tampering functions $\mathcal{T}$ from $\mathsf{A}$. It also receives the public parameters $pp$ from its own target game. $\mathsf{B}$ chooses uniformly at random an index $j^* \in [v]$. Recall that $v = |\mathcal{T}| = O(n^d)$ for some constant $d$.

2. $\mathsf{B}$ samples a random key $S$ for the hash function and uniformly at random an initial key $X^{(0)}$ from $\mathcal{X}$. It forwards $L(X^{(0)})$ and $pp' = (pp, S)$ as the public parameters to $\mathsf{A}$.

3. $\mathsf{B}$ uses its underlying target oracle $\mathcal{O}(pp, K)$ for the cryptoscheme $\mathcal{CS}$ to simulate $\mathsf{A}$'s interaction with the original (un-tampered) key in the query phase.

4. $\mathsf{B}$ receives a tuple of tampering functions $V_j = (T_1, \dots, T_t) \in \mathcal{T}$ from $\mathsf{A}$. If $j \neq j^*$ then we proceed as follows:

   (a) If $\mathsf{B}$ runs an unpredictability game, then it aborts.

   (b) If $\mathsf{B}$ runs an indistinguishability game, then it samples a random bit $b$ and submits $b$ as its last message to its challenge oracle $\mathsf{C}$.

5. $\mathsf{B}$ computes $X^{(i)} = T_i(X^{(i-1)})$ and simulates interactions with the oracles as follows:

(a) For all $i \geq 1$ with $X^{(i)} \neq X^{(0)}$, it uses $X^{(i)}$ to simulate $\mathsf{A}$'s interaction with the oracles $\mathcal{O}(pp', h_S(X^{(i)}))$ in the tampering phase. As $X^{(i)}$ is known to $\mathsf{B}$ this can be done efficiently.

(b) For all $i \geq 0$ with $X^{(i)} = X^{(0)}$, it uses its target oracle to simulate $\mathsf{A}$'s view. Notice that this includes the case when $\mathsf{A}$ interacts with the scheme running on the original key $X^{(0)}$.

We argue that when $\mathsf{A}$ wins the tampering game with advantage at least $1/p(k)$, then $\mathsf{B}$ wins the underlying game against challenger $\mathsf{C}$ with advantage $1/p'(k)$. To this end, we first show that conditioned on $j = j^*$ the view of the adversary in the simulation and the adversary's view in the real experiment are statistically close. If $j = j^*$ the simulation above is identically distributed to the simulation given in $\mathsf{Sim}$ from Figure 2. This follows from the following observations:

1. In the simulation $\mathsf{A}$ is committed to the tampering option $j^*$ *before* he starts to interact with the challenge oracles as otherwise $\mathsf{B}$ will abort the simulation. Notice that this commitment is in particular before seeing the hash key $S$, and the view with the original key. Hence, $\mathsf{B}$'s simulation corresponds to the non-adaptive case as given in $\mathsf{Sim}$.

2. $\mathsf{B}$ uses its own challenge oracle running with a uniform key to simulate $\mathsf{A}$'s un-tampered view. This is exactly as in the simulation $\mathsf{Sim}$ from Figure 2, where we replace the first output of the hash function with a uniformly and independently sampled value (independently of $S$ and the first input to the hash function).

3. $\mathsf{B}$ simulates the tampering queries by using an initial input $X^{(0)}$ for the hash function that is chosen independently from the un-tampered view. That is exactly what happens in $\mathsf{Sim}$.

The above concludes that the simulation of $\mathsf{B}$ and the simulation given in $\mathsf{Sim}$ are identical if $j = j^*$. By Lemma A.2 and Corollary 1 we get for the choice of parameters given in the theorem's statement (notice that this choice corresponds to the parameters of Corollary 1) that for $j = j^*$

$$\Delta(\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,\lambda,v); \overline{\mathsf{BLT}}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,\lambda,v)) \leq \exp(-\Omega(\sqrt[3]{(n)}), \tag{26}$$

where $\overline{\mathsf{BLT}}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,\lambda,v)$ is the game where $\mathsf{A}$ runs in the experiment as defined by $\mathsf{B}$. To complete the proof we need to lower bound the advantage of $\mathsf{B}$ when running against challenger $\mathsf{C}$. We discuss how to handle unpredictability and indistinguishability games separately.

1. *Unpredictability security notion:* For unpredictability games $\mathsf{B}$ aborts in Step 4 if $j \neq j^*$. Hence, we get:

$$\begin{aligned}
\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k) = 1] =& \Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k) = 1 | j = j^*] \Pr[j = j^*] \\
&+ \Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k) = 1 | j \neq j^*] \Pr[j \neq j^*] \\
\geq& \Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k) = 1 | j = j^*] \Pr[j = j^*] \\
\geq& \Big(\Pr[\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,\lambda,v) = 1] - \exp(-\Omega(\sqrt[3]{n})\Big) \frac{1}{v} \qquad (27) \\
>& \frac{1}{p'(k)}. \qquad (28)
\end{aligned}$$

(27) follows from Eq. (26) and the fact that conditioned on $j = j^*$ $\mathsf{B}$ wins game $\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}$ when $\mathsf{A}$ wins $\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,\lambda,v)$. Finally, (28) holds because $v = O(n^d)$ (for some constant $d$) and by assumption $\Pr[\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,\lambda,v) = 1] \geq 1/p(k)$. Clearly, (28) yields a contradiction.

2. *Indistinguishability security notion:* For indistinguishability games $\mathsf{B}$ aborts and sends a random bit $b$ to the challenger. As above we need to lower bound the advantage of $\mathsf{B}$ when playing against the challenge

oracle $\mathsf{C}$.

$$\begin{aligned}
\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k)=1]-\frac{1}{2} =&\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k)=1|j=j^*]\Pr[j=j^*]\\
&+\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k)=1|j\neq j^*]\Pr[j\neq j^*]-\frac{1}{2}\\
\geq&\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k)=1|j=j^*]\Pr[j=j^*]+\frac{\Pr[j\neq j^*]}{2}-\frac{1}{2} && (29)\\
\geq&\frac{\left(\Pr[\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,v,\lambda)=1]-\exp(-\Omega(\sqrt[3]{n}))\right)}{v}+\frac{v-1}{2v}-\frac{1}{2} && (30)\\
>&\frac{1}{p'(k)}. && (31)
\end{aligned}$$

(29) holds because $\Pr[\mathsf{Game}^{\mathcal{CS}}_{\mathsf{C},\mathcal{O},\mathsf{B}}(1^k)=1|j\neq j^*]=1/2$. (30) follows from Eq. (26) and the fact that conditioned on $j=j^*$ $\mathsf{B}$ wins its game when $\mathsf{A}$ wins its BLT game. Finally, (31) holds because $v=O(n^d)$ (for some constant $d$) and $\Pr[\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,v,\lambda)=1]\geq 1/2+1/p(k)$ for some polynomial $p(.)$.

The above yields a contradiction as for both game types the adversary $\mathsf{B}$ has a non-negligible advantage against the underlying challenger $\mathsf{C}$. Hence, we get

$$\Pr[\mathsf{BLT}^{\mathcal{CS}'}_{\mathsf{C},\mathcal{O},\mathsf{A}}(k,t,v,\lambda)=1]\leq \mathit{negl}(k)$$

as claimed in the theorem. This concludes the proof. □

### A.2.4 Proof of Lemma A.2

We start by describing a distribution $D^1$ that together with a short advice $Z$ allows to sample $\mathsf{Real}$. Distribution $D^1$ is defined exactly as $\mathsf{Real}$ except that it only contains distinct values (in particular notice that $D^1$ can contain less values than $\mathsf{Real}$). More precisely, $D^1$ is sampled as follows:

1. Let $X^{(0)}$ be distributed according to $X$ and compute $X^{(i)}=T_i(X^{(i-1)})$ (this is exactly as in $\mathsf{Real}$).
2. For all $i\in[t]$ output $h_S(X^{(i)})$ if for all $j<i$ we have $X^{(i)}\neq X^{(j)}$. Denote these outputs by $D^1$. We also output $(L(X^{(0)}),S)$.

The advice $Z$ (depending on $X^{(0)}$ and the functions $T_1,\ldots,T_t$) describes where the values from $D^1$ appear in $\mathsf{Real}$. An easy way to describe such an advice $Z$ requires $t^2/2$ bits. A more thorough analysis shows that one can encode the information necessary to map from $D^1$ to $\mathsf{Real}$ by $2t\log(t)$ bits.[6] In the following we denote the mapping algorithm that maps $(D^1,Z)$ to $\mathsf{Real}$ as $\mathsf{Samp}(D^1,Z)$. Clearly, $\mathsf{Samp}(D^1,Z)$ and $\mathsf{Real}$ are identically distributed and all values in $D^1$ are distinct. For ease of notation we will reuse the parameter $t$ to denote the number of elements in $D^1$.

**Claim 4.** *Let $\beta=\frac{\alpha-2t\sqrt{n}}{t+1}$. There exists an $i\in[t]$ and an event Good such that $\Pr[\mathit{Good}]\geq 1-\epsilon$ and*

$$\widetilde{\mathbf{H}}_\infty(X^{(i)}|X^{(i+1)},L(X^{(0)}),S,Z,\mathit{Good})\geq\beta-\log t/\epsilon-\lambda-2t\log(t). \tag{32}$$

*In the above $X^{(t+1)}$ denotes a random variable that is chosen uniformly and independently from $\mathcal{X}$.*

*Proof.* Recall that by putting *Good* in the condition of (32) we denote that all random variables are conditioned on the fact that *Good* happens. We prove this statement by relying on Lemma 3.3 which shows that each $X^{(i)}$ has average min-entropy at least $\beta-\log t/\epsilon$. Lemma 3.3 puts a constraint on $\beta$, i.e., $\beta\leq\frac{\alpha-2t\sqrt{n}}{t+1}$. Clearly, our choice of $\beta$ satisfies the above constraint. As $X^{(0)}$ is $(\alpha,n)$-good, we can now apply Lemma 3.3:

1. If $\Pr[E]>0$, for all $i\in[t]$: $\mathbf{H}_\infty(X^{(i)}_{|E})\geq\beta$,

[6] This can be done by first describing for each element in $D^1$ how often it appears in $\mathsf{Real}$ and then by defining a mapping that maps each element to its position in $\mathsf{Real}$. Each of these steps require at most $t\log(t)$ bits.

2. If $\Pr[\overline{E}] \geq \epsilon$ then there exists $i \in [t]$ such that : $\widetilde{\mathbf{H}}_\infty(X^{(i-1)}_{|\overline{E}} | X^{(i)}_{|\overline{E}}) \geq \beta - \log \frac{t}{\epsilon}$.

Consider now the setting when $\Pr[E] > 0$. Hence we know by Step (1) from above that for all $i \in [t]$: $\mathbf{H}_\infty(X^{(i)}_{|E}) \geq \beta$, and in particular $\mathbf{H}_\infty(X^{(t)}_{|E}) \geq \beta$. As $X^{(t+1)}$ is uniformly and independently chosen from all other variables, we get in this case that

$$\widetilde{\mathbf{H}}_\infty(X^{(t)}_{|E} | X^{(t+1)}_{|E}) \geq \beta \geq \beta - \log t/\epsilon. \tag{33}$$

Again if $\Pr[\overline{E}] \geq \epsilon$ then by Step (2) from above there exists an $i \in [t]$ such that

$$\widetilde{\mathbf{H}}_\infty(X^{(i-1)}_{|\overline{E}} | X^{(i)}_{|\overline{E}}) \geq \beta - \log t/\epsilon. \tag{34}$$

We define *Good* as follows: $\mathit{Good} = E$ if $\Pr[\overline{E}] < \epsilon$ and $\mathit{Good} = \Omega$ if $\Pr[\overline{E}] \geq \epsilon$ where $\Omega$ denotes the whole probability space. We can bound the probability of the event *Good* considering two cases:

- When $\Pr\left[\overline{E}\right] \geq \epsilon$, then $\Pr\left[\mathit{Good}\right] = 1$.
- When $\Pr\left[\overline{E}\right] < \epsilon$ then, $\Pr\left[\mathit{Good}\right] = \Pr\left[E\right] > 1 - \epsilon$.

So clearly $\Pr\left[\mathit{Good}\right] > 1 - \epsilon$.

We conclude that there must exist an $i \in [t]$ such that

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(X^{(i)} | X^{(i+1)}, L(X^{(0)}), S, Z, \mathit{Good}) &\geq \widetilde{\mathbf{H}}_\infty(X^{(i)} | X^{(i+1)}, S, \mathit{Good}) - \lambda - 2t\log(t) &(35)\\ &= \widetilde{\mathbf{H}}_\infty(X^{(i)} | X^{(i+1)}, \mathit{Good}) - \lambda - 2t\log(t) &(36)\\ &\geq \beta - \log t/\epsilon - \lambda - 2t\log(t). &(37)\end{aligned}$$

Eq. (35) follows from the chain rule for conditional average min entropy (cf. Lemma 2.1). Eq. (36) holds because $S$ is chosen uniformly and independently from all other variables. Finally, as either $E$ or $\overline{E}$ must happen and we condition on *Good*, we get from Eq. (33) and Eq. (34) that Eq. (37) holds. This concludes the proof of the claim. □

By using the union bound and Lemma 2.1, we can now restate Claim 4 in terms of min-entropy and condition all random variables on event *Good* happening. Thus we get that there exists an $i \in [t]$ such that with probability at least $1 - 2\epsilon$ the following holds:

$$\mathbf{H}_\infty(X^{(i)} | (X^{(i+1)}, L(X^{(0)}), S, Z) = r) \geq \beta - 2\log t/\epsilon - \lambda - 2t\log(t).$$

Recall that $X^{(i)}$ can be computed as a (deterministic) function from $X^{(j)}$ where $j < i$. Hence, the above holds for all $X^{(j)}$ where $j \leq i$, i.e.,

$$\mathbf{H}_\infty(X^{(j)} | (X^{(i+1)}, L(X^{(0)}), S, Z) = r) \geq \beta - 2\log t/\epsilon - \lambda - 2t\log(t) =: c.$$

As with probability at least $1 - 2\epsilon$ all $X^{(j)}$ individually have min-entropy $c$ and by assumption all $X^{(j)}$ are distinct, we can apply Lemma A.1:

$$\Delta((D^1, Z); (\underbrace{\overbrace{U_\mathcal{Y}, \ldots, U_\mathcal{Y}}^{i+1 \text{ times}}, X^{(i+1)}, L(X^{(0)}), S}_{D^2}, Z)) \leq 2^{t(v+\ell)+1} \left(\frac{q}{4t\epsilon^2 2^{c-t\ell}}\right)^{\frac{q}{2t}} + 3\epsilon =: \epsilon'.$$

As $\mathsf{Samp}$ is a deterministic algorithm, the above implies:

$$\Delta(\mathsf{Samp}(D^1, Z); \mathsf{Samp}(D^2, Z)) \leq \epsilon'.$$

Notice that in $\mathsf{Samp}(D^2, Z)$ the first $i + 1$ values are now sampled uniformly and independently from $U_\mathcal{Y}$. Consider now a distribution $D^3$ where only the first element is replaced by $U_\mathcal{Y}$ and the following $i$ elements are computed correctly as the output of the hash function $h_S$. By a standard argument, we get

$$\Delta(\mathsf{Samp}(D^1, Z); \mathsf{Samp}(D^3, Z)) \leq 2\epsilon'.$$

To conclude the proof notice that $\mathsf{Real}$ and $\mathsf{Sim}$ are identically distributed except for the effect that the first element has on the two distributions.[7] Hence, $\mathsf{Samp}(D^3, Z)$ and $\mathsf{Sim}$ are identically distributed. As moreover $\mathsf{Samp}(D^1, Z)$ and $\mathsf{Real}$ are identically distributed this concludes the proof. □

[7] In both cases we start with an element sampled from $X$ and apply the functions $T_1, \ldots, T_t$ to it.

### A.3 Extensions

**Beyond semi-adaptivity.** Notice that since $\mathcal{T}$ is a set of tuples of functions, Definition A.3 clearly implies non-adaptive security where the adversary commits to a single chain of tampering functions $(T_1, \ldots, T_t)$. We further notice that we can obtain a stronger form of semi-adaptivity by paying a higher price in the security loss. In this model, after committing to a set of functions $\mathcal{T} = \{T_i : \mathcal{X} \to \mathcal{X}\}$ (in Step 1 of Definition A.2), the adversary can adaptively choose individual functions from $\mathcal{T}$ (in Step 3 of Definition A.2). The loss in security however increases by a factor $v^t$ (instead of just $v$ as in Theorem A.1). Finally, observe that we can replace the $2(t+1)$-wise independent hash function with any cryptographic hash function and model it in the security proof of Theorem A.1 as a random oracle. As long as the tampering function cannot query the random oracle, the tampering choice may now be fully adaptively. Notice also that the random oracle allows us to improve some of the parameters from the theorem—in particular, the compression rate $\ell$.

It is an interesting question if also for general primitives stronger adaptivity security notions in the BLT model can be obtained. The following simple example shows, however, that this question may be hard—at least in its most general form.

**Example A.1.** Consider a PRF $\psi(K, M)$ that is a function of a $d$-key $K$ and input $M$ and is secure in the standard sense (without tampering or leakage). We also assume that the function can be broken if one learns a constant fraction of the key bits. We turn this into a new scheme with a public parameter $x_1, \ldots, x_d$ chosen from a large finite field of characteristic 2. The secret key is now a random polynomial $f$ of degree at most $d-1$. To evaluate the function on input $M$, we first compute $K = (lsb(f(x_1)), \ldots, lsb(f(x_d)))$ where $lsb$ denotes the least significant bit, and output $\psi(K, M)$. If there is no tampering, this is still secure, since $K$ is random, even given the $x_i$'s.

However, a fully adaptive tampering function that has full information on the $x_i$'s can interpolate a polynomial that takes any set of desired values in the $x_i$'s. It can therefore tamper freely with individual bits of $K$, and use a generic attack to learn $t$ bits of $K$ using $t$ tampering queries and break the function.

On the other hand, a non-adaptive tampering function is not allowed to depend on the public parameters. Assume it replaces the polynomial by $f' \neq f$. Then if $f - f'$ is constant, either $K$ is not changed or all bits of $K$ are flipped. We can reasonably assume that $\psi$ is secure against such a related-key attack. If $f - f'$ is not constant, then $(f - f')(x_i)$ is close to uniform for all $i$ because the degree of $f - f'$ is at most $d$ and this is much smaller than the size of the field. Although the values $(f - f')(x_i)$ are not independent, it is certainly not possible to change only one or a small number of key bits. So assuming $\psi$ has some form of related-key security, non-adaptive tampering cannot break the function.

**Avoid hashing by assuming RKA security.** We discuss a simple extension of our result from Theorem A.1 which allows to lift the statement to a fully adaptive setting, in case one is willing to assume the underlying cryptographic scheme has an additional security property (essentially a form of related-key attack security [8]). The scheme $\mathcal{CS}$ should remain secure even against an adversary which is allowed to see outputs $Z'$ produced with keys related to $X$ but that still retain high enough min-entropy. In this case, we can avoid entirely the transformation based on hashing and apply directly this assumption in the proof of Lemma A.2.

One natural question to ask is whether one can hope to prove that all primitives are secure in the non-adaptive BLT model, without necessarily using our transformation. The question to this answer is negative. Consider for instance the Naor-Reingold construction of a PRF [34]. For a group $\mathbb{G}$ of prime order $p$ with generator $g$, let $\mathrm{NR} : (\mathbb{Z}_p^*)^{n+1} \times \{0,1\}^n \to \mathbb{G}$ be defined as $\mathrm{NR}(\mathbf{x}, m) = g^{x_0 \cdot \prod_{i=1}^n x_i^{m_i}}$. The following is a simple non-adaptive attack on NR. Before the public parameters are sampled, commit to tampering function $T$, such that $T(x_0, \ldots, x_n) = (x_0, x_2, x_1, x_3 \ldots, x_n)$ (i.e., $T$ just swap $x_1$ and $x_2$). Query the function on input $m' = (1, 0, \ldots, 0)$; this yields the value $y' = g^{x_0 \cdot x_2}$. Now, run the challenge phase using input $m'' = (0, 1, 0, \ldots, 0)$. This is clearly distinguishable from random, as $y'' = y'$ for NR.

## B Proof of Lemma A.1

We restate the lemma for the reader's convenience.

**Lemma A.1.** *Let $(X_1, X_2, \ldots, X_t) \in \mathcal{X}^t$ be $t$ (possibly dependent) random variables such that $\mathbf{H}_\infty(X_i) \geq \beta$ and $(X_1, \ldots, X_t)$ are pairwise different. Let $\mathcal{H} = \{h_S : \mathcal{X} \to \mathcal{Y}\}$ be a family of $2t$-wise independent hash functions, with $|\mathcal{Y}| = 2^\ell$. Then for random $h_S \leftarrow \mathcal{H}$ we have that*

$$\Delta((h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)); (h_S, \underbrace{U_\mathcal{Y}, \ldots, U_\mathcal{Y}}_{t\ \textit{times}})) \leq \frac{t}{2} \cdot 2^{(t\cdot\ell-\beta)/2}.$$

*Proof.* Denote with $d = \log|\mathcal{H}|$ the size of the set $\mathcal{H}$. For random variables $Z$, $Z'$ such that $Z'$ is an independent copy of $Z$ we write $Col(Z) = \Pr[Z = Z']$ for the collision probability of $Z$. In particular,

$$\begin{aligned}
&Col((h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)))\\
&= \Pr\left[(h_S, h_S(X_1), h_s(X_2), \ldots, h_S(X_t)) = (h'_S, h'_S(X'_1), h'_s(X'_2), \ldots, h'_s(X'_t))\right]\\
&= \Pr\left[h_S = h'_S\right] \cdot \Pr\left[(h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)) = (h'_S, h'_S(X'_1), h'_S(X'_2), \ldots, h'_S(X'_t)) \mid h_S = h'_S\right]\\
&= 2^{-d} \cdot \Pr\left[(h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)) = (h_S, h_S(X'_1), h_S(X'_2), \ldots, h_S(X'_t))\right] \qquad (38)
\end{aligned}$$

where the probabilities above are over the choices of $h_S, (X_1, X_2, \ldots, X_t)$ and $h'_S, (X'_1, X'_2, \ldots, X'_t)$.

We define an event $E$ such that conditioning on $E$ happening we can apply the assumption that $h_S$ is $2t$-wise independent, and thus bound the probability in Eq. (38) by $2^{-t\cdot\ell}$. The event $E$ becomes true when $X_1, X_2, \ldots, X_t, X'_1, X'_2, \ldots, X'_t$ are pairwise different. Notice that there are $\binom{2t}{2}$ such pairs, however by assumption $(X_1, \ldots, X_t)$ are pairwise different; this leaves us with $\binom{2t}{2} - 2\binom{t}{2} = t^2$ pairs. Hence, by the union bound

$$\Pr\left[\overline{E}\right] = \Pr\left[X_1 = X_2 \vee X_2 = X_3 \vee \ldots \vee X'_{t-1} = X'_t\right] \leq t^2 \cdot 2^{-\beta},$$

where the inequality comes from the assumption that all random variables have individually min-entropy at least $\beta$ and by applying the union bound.

Plugging the last expression in Eq. (38) and using the fact that $h_S$ is $2t$-wise independent yields

$$\begin{aligned}
&Col((h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)))\\
&\leq 2^{-d} \cdot \left(\Pr\left[(h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)) = (h_S, h_S(X'_1), h_S(X'_2), \ldots, h_S(X'_t)) \mid E\right] + \Pr\left[\overline{E}\right]\right)\\
&\leq 2^{-d} \cdot (2^{-t\cdot\ell} + t^2 \cdot 2^{-\beta}).
\end{aligned}$$

Let $Z$ be a random variable with support $\mathcal{Z}$ and $U$ be uniform over $\mathcal{Z}$. Then $\|Z - U\|_2^2 = Col(Z) - |\mathcal{Z}|^{-1}$. In particular,

$$\begin{aligned}
\|(h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t))\|_2^2 &= Col((h_S, h_S(X_1), h_s(X_2), \ldots, h_S(X_t))) - 2^{-d-t\cdot\ell}\\
&\leq 2^{-d} \cdot \left(2^{-t\cdot\ell} + t^2 \cdot 2^{-\beta}\right) - 2^{-d-t\cdot\ell} = t^2 \cdot 2^{-d-\beta}.
\end{aligned}$$

Finally, using that $\|Z\|_1 \leq \sqrt{|\mathcal{Z}|} \cdot \|Z\|_2$ for any random variable $Z$ with support $\mathcal{Z}$, we obtain

$$\begin{aligned}
&\Delta((h_S, h_S(X_1), h_s(X_2), \ldots, h_S(X_t)); (h_S, \underbrace{U_\mathcal{Y}, \ldots, U_\mathcal{Y}}_{t\ \text{times}}))\\
&\qquad= \frac{1}{2}\|(h_S, h_S(X_1), h_S(X_2), \ldots, h_S(X_t)) - (h_S, \underbrace{U_\mathcal{Y}, \ldots, U_\mathcal{Y}}_{t\ \text{times}})\|_1\\
&\qquad\leq \frac{1}{2}\sqrt{2^{d+t\cdot\ell}} \cdot \sqrt{t^2 \cdot 2^{-d-\beta}} = \frac{t}{2} \cdot 2^{(t\cdot\ell-\beta)/2}.
\end{aligned}$$

□