Inception Makes Non-malleable Codes Stronger

Divesh Aggarwal[1], Tomasz Kazana[2⋆], and Maciej Obremski[3]

[1] National University of Singapore
[2] University of Warsaw
[3] Aarhus University

Abstract. Non-malleable codes (NMCs), introduced by Dziembowski, Pietrzak and Wichs [DPW10], provide a useful message integrity guarantee in situations where traditional error-correction (and even error-detection) is impossible; for example, when the attacker can completely overwrite the encoded message. NMCs have emerged as a fundamental object at the intersection of coding theory and cryptography.

A large body of the recent work has focused on various constructions of non-malleable codes in the split-state model. Many variants of NMCs have been introduced in the literature i.e. strong NMCs, super strong NMCs and continuous NMCs. Perhaps the most useful notion among these is that of continuous non-malleable codes, that allows for continuous tampering by the adversary.

In this paper we give the first efficient, information-theoretic secure construction of continuous non-malleable codes in the split-state model. Enroute to our main result, we obtain constructions for almost all possible notions of non-malleable codes that have been considered in the split-state model, and for which such a construction is possible. Our result is obtained by a series of black-box reductions starting from the non-malleable codes from [ADL14].

One of the main technical ingredient of our result is a new concept that we call *inception coding*. We believe it may be of independent interest.

Also our construction is used as a building block for non-persistent (resettable) continuous non-malleable codes in constant split-state model in [ADN$^+$19].

⋆ Supported by NCN grant UMO-2014/13/D/ST6/03252. Partially done during a post-docinternship at NYU

1 Introduction

Non-malleable Codes. Non-malleable codes (NMCs), introduced by Dziembowski, Pietrzak and Wichs [DPW10], provide a useful message integrity guarantee in situations where traditional error-correction (and even error-detection) is impossible; for example, when the attacker can completely overwrite the encoded message. NMCs have emerged as a fundamental object at the intersection of coding theory and cryptography.

Informally, given a tampering family $\mathcal{F}$, an NMC $(\mathsf{Enc}, \mathsf{Dec})$ against $\mathcal{F}$ encodes a given message m into a codeword $c \leftarrow \mathsf{Enc}(m)$ in a way that, if the adversary modifies c to $c' = f(c)$ for some $f \in \mathcal{F}$, then the the message $m' = \mathsf{Dec}(c')$ is either the original message m, or a completely "unrelated value". As has been shown by the recent progress [DPW10,LL12,DKO13,ADL14, FMVW14,FMNV14,CG14a,CG14b,CZ14,Agg15,ADKO15b,ADKO15a,CGL15, AGM+15b, AGM+15a, AAnHKM+16, Li16] NMCs aim to handle a much larger class of tampering functions $\mathcal{F}$ than traditional error-correcting or error-detecting codes, at the expense of potentially allowing the attacker to replace a given message m by an unrelated message m'. NMCs are useful in situations where changing m to an unrelated m' is not useful for the attacker (for example, when m is the secret key for a signature scheme.)

(Super) Strong Non-malleable Codes. A stronger notion of non-malleability, called *strong non-malleable codes*, was also considered in [DPW10] in which, whenever the codeword c is modified to $c' = f(c) \neq c$, the decoded message $m' = \mathsf{Dec}(c')$ is independent of m. This is in contrast to the plain notion of non-malleability where some modification of the codeword c could still result in $m' = m$. Indeed, this is the case in some of the previous constructions of non-malleable codes like [ADL14, ADKO15a]. For the purpose of conveniently defining continuous non-malleable codes, an even stronger notion called *super-strong non-malleable codes* has been considered in the literature [FMNV14, JW15]. Informally speaking, in this notion, if $c' \neq c$ is a valid codeword, then c' must be independent of c.

An intermediate notion can also be considered where if $m' = \mathsf{Dec}(c') \notin \{m, \perp\}$, then c' must be independent of c. To be consistent with other notions of non-malleable codes, we call these *super non-malleable codes*.

Continuous Non-malleable Codes. It is clearly realistically possible that the attacker repeatedly tampers with the device and observes the outputs. As mentioned in [JW15], non-malleable codes can provide protection against these kind of attacks if the device is allowed to freshly re-encode its state after each invocation to make sure that the tampering is applied to a fresh codeword at each step. After each execution the entire content of the memory is erased. While such perfect erasures may be feasible in some settings, they are rather problematic in the presence of tampering. Due to this reason, Faust et al. [FMNV14] introduced an even stronger notion of non-malleable codes called continuous non-malleable

codes where security is achieved against continuous tampering of a single codeword *without* re-encoding. Jafargholi and Wichs [JW15] considered four variants of continuous non-malleable codes depending on

- Whether tampering is *persistent* in the sense that the tampering is always applied to the current version of the tampered codeword, and all previous versions of the codeword are lost. The alternative definition considers non-persistent tampering where the tampering always occurs on the original codeword.
- Whether tampering to an invalid codeword (i.e., when the decoder outputs $\perp$) causes a *"self-destruct"* and the experiment stops and the attacker cannot gain any additional information, or alternatively whether the attacker can always continue to tamper and gain information.

Split-State Model. Although any kind of non-malleable codes do not exist if the family of "tampering functions" $\mathcal{F}$ is completely unrestricted,[4] they are known to exist for many large classes of tampering families $\mathcal{F}$. One such natural family is the family of tampering functions in the so called *t-split-state* model. In this model, the codeword is "split" into $t > 1$ states $c = (c_1, \ldots, c_t)$; a tampering function f is viewed as a list of t functions $(f_1, \ldots, f_t)$ where each function f_i tampers with corresponding component c_i of the codeword independently: i.e., the tampered codeword is $c' = (f_1(c_1), \ldots, f_t(c_t))$.

This family is interesting since it seems naturally useful in applications, especially when t is low and the shares $y_1, \ldots, y_t$ are stored in different parts of memory, or by different parties. Not surprisingly, the setting of $t = 2$ appears the most useful (but also the most challenging from the technical point of view), so it received the most attention so far [DPW10, LL12, DKO13, ADL14, FMNV14, CG14a, CG14b, CZ14, CGL15, ADKO15b, ADKO15a, Li16] and is also the focus of our work.

While some of the above mentioned results achieve security against computationally bounded adversaries, we focus on security in the information-theoretic setting, i.e., security against unbounded adversaries. The known results in the information-theoretic setting can be summarized as follows. Firstly [DPW10] showed the existence of (strong) non-malleable codes, and this result was improved by [CG14a] who showed that the optimal rate of these codes is 1/2. Faust et al. [FMNV14] showed the impossibility of continuous non-malleable codes against non-persistent split-state tampering. Later [JW15] showed that continuous non-malleable codes exist in the split-state model if the tampering is persistent.

There have been a series of recent results culminating in constructions of efficient non-malleable codes in the split-state model [DKO13, ADL14, CZ14, CGL15, ADKO15a, Li16]. However, there is no known efficient construction in the continuous setting. Since the work of [FMNV14] rules out the possibility of

[4] In particular, $\mathcal{F}$ should not include "re-encoding functions" $f(c) = \mathsf{Enc}(f'(\mathsf{Dec}(c)))$ for any non-trivial function f', as $m' = \mathsf{Dec}(f(\mathsf{Enc}(m))) = f'(m)$ is obviously related to m.

such a construction for the case of non-persistent tampering, the best one can hope for is an efficient construction for the case of persistent tampering in the split-state model.

Our Results and Techniques. This brings us to the main result of the paper which is the following.

Theorem 1. *For any k, there exists an efficient (in k) information-theoretically secure persistent continuous $2^{-k^{\Omega(1)}}$-non-malleable code with self-destruct in the split-state model that encodes k-bit messages to $\mathsf{poly}(k)$-bit codewords.*

Enroute to Theorem 1, we obtain efficient constructions of almost all possible notions of non-malleable codes in the split-state model for which such a construction is possible.

While it might be argued that the most interesting case of continuous non-malleable codes is that of non-persistent tampering, it was shown to be impossible in the 2-split state model in [FMNV14]. In a recent work, it has been shown that our persistent continuous non-malleable codes can in fact be used to obtain an efficient construction of non-persistent continuous non-malleable codes in the constant split-state model [ADN+19].

The construction is obtained in a series of steps. We first show a reduction (Theorem 2 in Section 4) that any scheme in the split-state model that is a super-strong non-malleable code is also a persistent continuous non-malleable code with self-destruct in the split-state model. The key idea behind this reduction is the observation by Jafargholi and Wichs [JW15] that for the case of persistent continuous non-malleable codes with self-destruct, without loss of generality, we can assume that the experiment stops at the first instance (say at step I) when there is a non-trivial tampering. This is because if the tampered codeword decodes to $\perp$ then the experiment stops because of the self-destruct property, and if it does not decode to $\perp$, then the adversary learns the entire codeword and can simulate the remaining tampering experiment himself. Thus, the main ingredient of this reduction is showing that for any non-malleable code in the split-state model, the random variable I combined with first non-same tampering experiment output does not reveal the encoded message.

Our main technical reduction (Theorem 3 in Section 5) is one that shows that any coding scheme that is super non-malleable in the split-state model can be converted into a scheme that is super-strong non-malleable in the split-state model. To do that we develop a new technique we called *inception coding*. The key difference between a super non-malleable code and a super-strong non-malleable code is that in the former, the adversary is assumed to not gain any useful information if he tampers with and changes the codeword but the tampered codeword still decodes to the same message while in the latter, the adversary in this case gets to see the entire tampered codeword. Our inception coding essentially forces all these non-trivial tampered codewords (that originally decoded to the correct message) to decode to $\perp$. In our reduction, given a super non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$, we modify the encoding procedure to sacrifice a small suffix of the message (it will not carry any message

related information anymore) to replace it with validity checks for each of the states that detect whether these states have been tampered with. The message m is encoded as $\mathsf{Enc}(m, \mathsf{check}_x, \mathsf{check}_y) = (X, Y)$ subject to the condition that $\mathsf{Verify}(\mathsf{check}_x; X) = \mathsf{Verify}(\mathsf{check}_y; Y) = \mathrm{OK}$. This ensures that in the case when tampered codeword decodes correctly, the validity check can detect the tampering and output $\perp$. In order to use the super non-malleability of $(\mathsf{Enc}, \mathsf{Dec})$ to conclude super-strong non-malleability of the modified encoding scheme, we need to do rejection sampling to ensure that the codeword is valid with respect to the modified encoding algorithm. This blows up the error by a factor of about 2^{2t} where t is the length of each validity check, and so we require that $2^{2t} \ll 1/\varepsilon$, where ε is the error parameter for $(\mathsf{Enc}, \mathsf{Dec})$. We obtain a construction of the check function in Definition 8 using the well-studied Reed-Solomon error-correcting codes. In order to reduce the output length of this construction, we define a composition theorem on validity check functions, and show in Lemma 7 that using this composition theorem repeatedly, we can progressively make the length of the validity check shorter.

Finally, to complete the proof, we show (in Theorem 5 in Section 6) that the coding scheme from [ADL14], which was shown to be a non-malleable code in the split-state model, is also super non-malleable. This proof was surprisingly involved, since we need to argue that for any two tampered codewords c'_1, c'_2 of two distinct messages, if they do not decode to $\perp$ or the original messages, respectively, then the two tampered codewords are indistinguishable. This required a careful re-analysis of the various cases in [ADL14], in particular those where their tampering experiment does not output same or $\perp$. Fortunately, this happens only when one of the two tampered parts $f(L)$ or $g(R)$ loses a lot of information about the two parts L and R of the original codeword, and since the construction of [ADL14] is based on the inner product function, which is a strong 2-source extractor, one can conclude that the tampered codeword $(f(L), g(R))$ is independent of the $\langle L, R\rangle$ and hence of the original message.

Background. The notion of non-malleability was introduced by Dolev, Dwork and Naor [DDN00], and has found many applications in cryptography. Traditionally, non-malleability is defined in the computational setting, but recently non-malleability has been successfully defined and applied in the information-theoretic setting (generally resulting in somewhat simpler and cleaner definitions than their computational counter-parts). For example, in addition to non-malleable codes studied in this work, the work of Dodis and Wichs [DW09] defined the notion of non-malleable extractors as a tool for building round-efficient privacy amplification protocols.

Finally, the study of non-malleable codes falls into a much larger cryptographic framework of providing counter-measures against various classes of tampering attacks. This work was pioneered by the early works of [ISW03, GLM+03, IPSW06], and has since led to many subsequent models. We do not list all such tampering models, but we refer to [KKS11, LL12] for an excellent discussion of various such models.

Other Related Work. In addition to the works mentioned above, non-malleable codes have been studied in various tampering models in several recent results. For tampering functions of size $2^{\mathsf{poly}(n)}$, rate-1 codes (with efficient encoding and decoding) exist, and can be obtained efficiently with overwhelming probability [FMVW14].

Cheraghchi and Guruswami [CG14b] gave a rate 1 non-malleable code against the class of bitwise-tampering functions, where each bit of the codewords is tampered independently. Recently, Agrawal et al. [AGM+15b, AGM+15a] improved this result by giving a explicit rate-1 code against a stronger class of tampering functions, which in addition to tampering with each bit of the codeword independently, can also permute the bits of the resulting codeword after tampering, was achieved in [AGM+15b, AGM+15a].

In the "split state" setting, an encoding scheme was proposed in [CKM11]. For the case of only two states, an explicit non-malleable code for encoding one-bit message was proposed by [DKO13]. This was improved by Aggarwal et al [ADL14] to a scheme that encodes larger messages but with rate $1/\mathsf{poly}(k)$ where k is the length of the message. This was further improved to obtain a constant-rate non-malleable code in [CZ14, ADKO15a].

Another related result by Aggarwal et al [ADKO15b] obtained efficient construction of non-malleable codes in a model where the adversary, in addition to performing split-state tampering, is also allowed some limited interaction between the two states.

Coretti et al. [CMTV15,CDTV16] have obtained constructions of information-theoretically secure continuous non-malleable codes in the bit-wise independent tampering model and have used this construct a non-malleable encryption scheme.

In the computational setting, there has been a sequence of works constructing non-malleable codes and its variants [LL12,FMNV14]. Chandran et al. [CGM+15] also rely on the computational setting in defining their new notion of *blockwise non-malleable codes*. Blockwise non-malleable codes are a generalization of the split-state model (and the recent lookahead model of [ADKO15a]) where the adversary tampers with one state at a time.

2 Preliminaries

For a set S, we let U_S denote the uniform distribution over S. For an integer $m \in \mathbb{N}$, we let U_m denote the uniform distribution over $\{0,1\}^m$, the bit-strings of length m. For a distribution or random variable X we write $x \leftarrow X$ to denote the operation of sampling a random x according to X. For a set S, we write $s \leftarrow S$ as shorthand for $s \leftarrow U_S$.

The Hamming distance between two strings $(a_1, \ldots, a_m)$, $(b_1, \ldots, b_m) \in \{0,1\}^m$ is the number of $i \in [m]$ such that $a_i \neq b_i$. We denote it as

$$\mathsf{Ham}((a_1, \ldots, a_m)\ ;\ (b_1, \ldots, b_m))\ .$$

Entropy and Statistical Distance. The *min-entropy* of a random variable X is defined as $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. We say that X is an (n, k)-*source* if $X \in \{0,1\}^n$ and $\mathbf{H}_\infty(X) \geq k$. For $X \in \{0,1\}^n$, we define the *entropy rate* of X to be $\mathbf{H}_\infty(X)/n$. We also define *average (aka conditional) min-entropy* of a random variable X conditioned on another random variable Z as

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(X|Z) &\stackrel{\text{def}}{=} -\log\left(\mathbb{E}_{z\leftarrow Z}\left[\max_x \Pr[X = x|Z = z]\right]\right)\\ &= -\log\left(\mathbb{E}_{z\leftarrow Z}\left[2^{-\mathbf{H}_\infty(X|Z=z)}\right]\right).\end{aligned}$$

where $\mathbb{E}_{z\leftarrow Z}$ denotes the expected value over $z \leftarrow Z$. We have the following lemma.

Lemma 1 ([DORS08]). *Let (X, W) be some joint distribution. Then,*

- *For any $s > 0$, $\Pr_{w\leftarrow W}[\mathbf{H}_\infty(X|W = w) \geq \widetilde{\mathbf{H}}_\infty(X|W) - s] \geq 1 - 2^{-s}$.*
- *If Z has at most 2^ℓ possible values, then $\widetilde{\mathbf{H}}_\infty(X|(W, Z)) \geq \widetilde{\mathbf{H}}_\infty(X|W) - \ell$.*

The *statistical distance* between two random variables W and Z distributed over some set S is

$$\Delta(W, Z) := \max_{T\subseteq S} |W(T) - Z(T)| = \frac{1}{2}\sum_{s\in S} |W(s) - Z(s)|.$$

Note that $\Delta(W, Z) = \max_D(\Pr[D(W) = 1] - \Pr[D(Z) = 1])$, where D is a probabilistic function. We say W is ε-close to Z, denoted $W \approx_\varepsilon Z$, if $\Delta(W, Z) \leq \varepsilon$. We write $\Delta(W, Z|Y)$ as shorthand for $\Delta((W, Y), (Z, Y))$, and note that $\Delta(W, Z|Y) = \mathbb{E}_{y\leftarrow Y}\Delta(W|Y = y, Z|Y = y)$.

Reed-Solomon Codes. In Section 5 we will use standard Reed-Solomon error-correcting codes. The following is a folklore result about Reed-Solomon codes. See, for example [RU08].

Lemma 2. *Let $n = 2^\ell$ for some positive integer ℓ, and let $q > 0$ be an integer. There exist a function $RS : \{0,1\}^n \to \{0,1\}^{n+q\log n}$[5] such that:*

- *Hamming distance between any two elements of the image of RS is at least $q + 1$,*
- *For any $x \in \{0,1\}^n$ there exist a unique sequence of bits $u \in \{0,1\}^{q\log n}$ such that $x\|u$ is an element of the image of RS;*
- *For every $u \in \{0,1\}^{q\log n}$ the set of all $x \in \{0,1\}^n$ such that $x\|u$ is an element of the image of RS is affine subspace of $\{0,1\}^n$.*

[5] The elements of the image of RS are called valid codewords for RS.

3 Various definitions of Non-Malleable Codes

Definition 1. *A coding scheme in the split-state model consists of two functions: a randomized encoding function* $\mathsf{Enc} : \{0,1\}^k \mapsto \{0,1\}^n \times \{0,1\}^n$*, and a deterministic decoding function* $\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^k \cup \{\perp\}$ *such that, for each* $m \in \mathcal{M}$*,* $\Pr(\mathsf{Dec}(\mathsf{Enc}(m)) = m) = 1$ *(over the randomness of the encoding algorithm). Additionally, we say that the coding scheme is almost uniform if for any* m*, any constant* $c > 1/2$ *and large enough* n*, and any* $\mathcal{L}, \mathcal{R} \subseteq \{0,1\}^n$*, such that* $|\mathcal{L}| \geq 2^{cn}$*, and* $|\mathcal{R}| \geq 2^{cn}$ *we have that*

$$\frac{|\mathcal{L}| \times |\mathcal{R}|}{2^{2n+1}} \leq \Pr(\mathsf{Enc}(m) \in \mathcal{L} \times \mathcal{R}) \leq \frac{|\mathcal{L}| \times |\mathcal{R}|}{2^{2n-1}} ,$$

where the probability is taken over the randomness of the encoding algorithm.

We now define non-malleable codes.

Definition 2. (Non-Malleable Code from [DPW10].) *Let* $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}, \mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ *be an encoding scheme. For* $f, g : \mathcal{X} \to \mathcal{X}$ *and for any* $m \in \mathcal{M}$ *define the experiment* $\mathsf{DPWTamp}_m^{f,g}$ *as:*

$$\mathsf{DPWTamp}_m^{f,g} = \left\{ \begin{array}{c} (X, Y) \leftarrow \mathsf{Enc}(m), \\ X' := f(X), Y' := g(Y) \\ m' := \mathsf{Dec}(X', Y') \\ \textit{output: } m' \end{array} \right\}$$

We say that an encoding scheme $(\mathsf{Enc}, \mathsf{Dec})$ *is* ε-DPW-non-malleable in split-state model *if for every functions* $f, g : \mathcal{X} \to \mathcal{X}$ *there exists distribution* $D^{f,g}$ *on* $\mathcal{M} \cup \{\mathsf{same}, \perp\}$ *such that for every* $m \in \mathcal{M}$ *we have*

$$\mathsf{DPWTamp}_m^{f,g} \approx_\varepsilon \left\{ \begin{array}{c} d \leftarrow D^{f,g} \\ \textit{if } d = \mathsf{same} \textit{ then output } m \\ \textit{otherwise output } d. \end{array} \right\}$$

We will consider the following alternative definition of non-malleable code, which will be a smoother transition to the subsequent definitions in this section. We show the equivalence of this definition to Definition 2 (originally formulated in [DPW10]) in Appendix A.

Definition 3. (Non-Malleable Code.) *We say that an encoding scheme* $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}, \mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ *is* ε-non-malleable in split-state model *if for every functions* $f, g : \mathcal{X} \to \mathcal{X}$ *there exists family of distributions* $\{D_{x,y}^{f,g}\}_{x,y \in \mathcal{X}}$ *each on* $\{0,1\}$ *such that for every* $m_0, m_1 \in \mathcal{M}$

$$\mathsf{Tamp}_{m_0}^{f,g} \approx_\varepsilon \mathsf{Tamp}_{m_1}^{f,g}$$

where

$$\mathsf{Tamp}_m^{f,g} = \left\{ \begin{array}{c} (X, Y) \leftarrow \mathsf{Enc}(m), \\ \textit{output } \mathbf{same} \textit{ if } \mathsf{Dec}(X, Y) = \mathsf{Dec}(f(X), g(Y)) \wedge D_{X,Y}^{f,g} = 0 \\ \textit{else output: } \mathsf{Dec}(f(X), g(Y)) \end{array} \right\}$$

Some results in the literature like [FMNV14, JW15] have considered a notion of super-strong non-malleable codes. We introduce the following intermediate notion of super non-malleable codes.

Definition 4. (Super Non-Malleable Code.) *We say that an encoding scheme* $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}, \mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ *is* ε-super non-malleable in split-state model *if for every functions* $f, g : \mathcal{X} \to \mathcal{X}$ *there exists family of distributions* $\{D^{f,g}_{x,y}\}_{x,y \in \mathcal{X}}$ *each on* $\{0,1\}$ *such that for every* $m_0, m_1 \in \mathcal{M}$

$$\mathsf{SupTamp}^{f,g}_{m_0} \approx_\varepsilon \mathsf{SupTamp}^{f,g}_{m_1}$$

where $\mathsf{SupTamp}^{f,g}_{m} =$

$$\left\{ \begin{array}{c} (X, Y) \leftarrow \mathsf{Enc}(m), \\ \textit{output } \textbf{same } \textit{if } \mathsf{Dec}(X, Y) = \mathsf{Dec}(f(X), g(Y)) \wedge D^{f,g}_{X,Y} = 0 \\ \textit{else if } \mathsf{Dec}(f(X), g(Y)) = \perp \textit{ output } \perp \\ \textit{else output: } (f(X), g(Y)) \end{array} \right\}$$

Definition 5. (Super Strong Non-Malleable Code.) *We say that an encoding scheme* $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}, \mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ *is* ε-super strong non-malleable in split-state model *if for every functions* $f, g : \mathcal{X} \to \mathcal{X}$ *and for every* $m_0, m_1 \in \mathcal{M}$

$$\mathsf{SupStrTamp}^{f,g}_{m_0} \approx_\varepsilon \mathsf{SupStrTamp}^{f,g}_{m_1}$$

where

$$\mathsf{SupStrTamp}^{f,g}_{m} = \left\{ \begin{array}{c} (X, Y) \leftarrow \mathsf{Enc}(m), \\ \textit{output } \textbf{same } \textit{if } (X, Y) = (f(X), g(Y)) \\ \textit{else if } \mathsf{Dec}(f(X), g(Y)) = \perp \textit{ output } \perp \\ \textit{else output: } (f(X), g(Y)) \end{array} \right\}$$

Definition 6. (Continuous Non-Malleable Code.) *[JW15] define four types of continuous non-malleable codes based on two flags:* $\mathsf{sd} \in \{0,1\}$ (self-destruct) *and* $\mathsf{prs} \in \{0,1\}$ (persistent). *We say that an encoding scheme* $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}, \mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ *is* $(\mathsf{T}, \varepsilon)$-continuous $[\mathsf{sd}, \mathsf{prs}]$ non-malleable in split-state model *if for every Adversary* $\mathcal{A}$ *and for every* $m_0, m_1 \in \mathcal{M}$

$$\mathsf{ConTamper}_{\mathcal{A},\mathsf{T},m_0} \approx_\varepsilon \mathsf{ConTamper}_{\mathcal{A},\mathsf{T},m_1}$$

where $\mathsf{ConTamper}_{\mathcal{A},\mathsf{T},m} =$

$$
\left\{
\begin{array}{l}
(X, Y) \leftarrow \mathsf{Enc}(m), \\
f_0, g_0 \equiv \mathsf{id}, \\
\textbf{Repeat} \quad i = 1, 2, ..., \mathsf{T} \\
\quad \mathcal{A} \textit{ chooses functions } f'_i, g'_i \\
\quad \textbf{if } \mathsf{prs} = 1 \textbf{ then } f_i = f'_i \circ f_{i-1},\ g_i = g'_i \circ g_{i-1} \\
\qquad \textbf{else } f_i = f'_i,\ g_i = g'_i \\
\quad \textbf{if } (f_i(X), g_i(Y)) = (X, Y) \textbf{ then output } \textbf{same} \\
\qquad \textbf{else} \\
\qquad \textbf{if } \mathsf{Dec}(f_i(X), g_i(Y)) = \perp \textbf{ then output } \perp \textbf{ if } \mathsf{sd} = 1 \textbf{ then experiment stops} \\
\qquad\quad \textbf{else output } (f_i(X), g_i(Y)) \textbf{ if } \mathsf{prs} = 1 \textbf{ then experiment stops}
\end{array}
\right\}
$$

Remark 1. In the case of persistent tampering, the above definition by [JW15] assumes that the tampering experiment stops if there is a non-trivial tampering that does not decode to $\perp$ since in this case the adversary learns the entire tampered codeword, and can simulate the remaining tampering experiment himself.

Remark 2. [FMNV14] show that non-persistent continuous non-malleable codes are impossible to construct in 2-split state model.

Remark 3. In any model allowing bitwise tampering, in particular the $2-$split state model, it is not difficult to conclude that the *non-self-destruct* property is impossible to achieve even in the case of persistent tampering if the space of messages contains at least 3 elements. To see this, notice that one can tamper the codeword $c = (c_1, c_2, c_3, \ldots)$ to obtain $c'_1 = (0, c_2, \ldots)$. The adversary then obtains the output of the tampering experiment which is same if and only if $c_1 = 0$. Thus the adversary learns $c^\star_1 = c_1$ and continues the tampering experiment with $(c^\star_1, 0, c_3, \ldots)$ (note that this tampering is persistent). Thus, the adversary can continue learn the codeword one bit at a time, thereby learning the entire codeword in N steps where N is the length of the codeword. Such an argument has been used previously for proving impossibility results. See for instance the work of Gennaro et al. [GLM$^+$03].

4 From Super Strong NMCs to Continuous NMCs

In this section we will prove the following statement:

Theorem 2. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* ε*-super strong non-malleable code in the split-state model then* $(\mathsf{Enc}, \mathsf{Dec})$ *is a* $(T, (2T+1)\varepsilon)-$*continuous* $[1, 1]$ *non-malleable code in the split-state model.*

For proving Theorem 2, we will need the following lemmata. The following result states that any non-malleable code in the 2-split state model is a good 2-out-of-2 secret sharing scheme.

Lemma 3 ([ADKO15b, Lemma 6.1]). *Let* $\mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M}$*, and* $\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}$ *be an* $\varepsilon-$*non-malleable code in the split state model for some* $\varepsilon < \frac{1}{2}$*. For any pair of messages* $m_0, m_1 \in \mathcal{M}$*, let* $(X_1^0, X_2^0) \leftarrow \mathsf{Enc}(m_0)$*, and let* $(X_1^1, X_2^1) \leftarrow \mathsf{Enc}(m_1)$*. Then* $\Delta(X_1^0\,;\, X_1^1) \leq 2\varepsilon$*.*

The following result states that given a non-malleable code $(\mathsf{Enc}, \mathsf{Dec})$ in the split-state model, for any sets A, B, and any message m, the probability that $\mathsf{Enc}(m)$ falls in the set $A \times B$ is almost independent of the choice of the message m.

Lemma 4. *Let* $k \geq 3$*, and let* $\varepsilon < 1/20$*. Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n \times \{0,1\}^n$*,* $\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k$ *be an* $\varepsilon-$*non-malleable code in the split state model. For every sets* $A, B \subset \{0,1\}^n$ *and every messages* $m_0, m_1 \in \{0,1\}^k$

$$|\Pr(\mathsf{Enc}(m_0) \in A \times B) - \Pr(\mathsf{Enc}(m_1) \in A \times B)| \leq \varepsilon\ .$$

Proof. We claim that there exist $x, y, z, w \in \{0,1\}^n$ such that $m_0, m_1, \mathsf{Dec}(x, w)$, $\mathsf{Dec}(z, w)$, and $\mathsf{Dec}(z, y)$ are all different from $\mathsf{Dec}(x, y)$. Before proving this claim, we show why this implies the given result. Consider the tampering functions f, g such that $f(c) = x$ if $c \in A$, and $f(c) = z$, otherwise, and $g(c) = y$ if $c \in B$, and $g(c) = w$, otherwise. Thus, for $b = 0, 1$, $\mathsf{Tamp}_{m_b}^{f,g} = \mathsf{Dec}(x, y)$ if and only if $\mathsf{Enc}(m_b) \in A \times B$. The result then follows from the ε-non-malleability of $(\mathsf{Enc}, \mathsf{Dec})$.

Now, to prove the claim, we will use the probabilistic method. Let U be uniform in $\{0,1\}^k$, and let $X, Y \leftarrow \mathsf{Enc}(U)$. Furthermore, let $W, Z \in \{0,1\}^n$ be uniform and independent of X, Y, U. We claim that X, Y, Z, W satisfy the required property with non-zero probability.

It is easy to see that the probability that $\mathsf{Dec}(X, Y) = U$ is either of m_0 or m_1 is at most $2/2^k$. Also, by Lemma 3, we have that except with probability 2ε, X is independent of U. Also, W is independent of U. Thus, the probability that $\mathsf{Dec}(X, W) = U$ is at most $2\varepsilon + 1/2^k$. Similarly, the probability that $\mathsf{Dec}(Z, Y) = U$ is at most $2\varepsilon + 1/2^k$. Finally, W, Z are independent of U, and so the probability that $\mathsf{Dec}(Z, W) = U$ is at most $\frac{1}{2^k}$.

Thus, by union bound, the probability that X, Y, Z, W do not satisfy the condition of the claim is at most $\frac{5}{2^k} + 4\varepsilon \leq \frac{5}{8} + 4\varepsilon < 1$. □

Before proving Theorem 2, let us fix some notation. Let $\mathcal{A}^*$ be any adversary described in Definition 6. Let $(I)_m$ denote the index of a round when ***same*** is not output in the experiment $\mathsf{ConTamper}_{\mathcal{A}^*, \mathsf{T}, m}$ and (f_i, g_i) (for $i = 1, \ldots, T$) denote pairs of functions chosen by $\mathcal{A}^*$ (of course we can assume that they are always the same because the choice for the next round does not depend on (X, Y)).

Proof (of Theorem 2).

We will show that

$$\Delta([(I)_{m_0}, f_{I_{m_0}}(X_0), g_{I_{m_0}}(Y_0)]\ ;\ [I_{m_1}, f_{I_{m_1}}(X_1), g_{I_{m_1}}(Y_1)]) \leq (2T+1)\varepsilon. \quad (4.1)$$

The desired result will follow from the observation that $\mathsf{ConTamper}_{\mathcal{A}^*, \mathsf{T}, m_b}$ for $b = 0, 1$ depends only on $(I)_{m_b}$, $f_{(I)_{m_b}}(X_b)$, and $g_{(I)_{m_b}}(Y_b)$

In order to simplify the proof, we make use of the following fact about statistical distance: The statistical distance between two random variables Z_0 and Z_1 is at most δ if and only if for any computationally unbounded algorithm that is given as input a sample distributed as Z_b, for a uniformly random bit b, the probability that the algorithm can guess the bit b is at most $1/2 + \delta/2$.

Thus, we wish to bound the probability of guessing the bit b, given $I, f_I(X), g_I(Y)$, where I, X, Y are shorthand for I_{m_b}, X_b, Y_b.

We can partition the codeword space $\{0,1\}^n \times \{0,1\}^n$ into $(2T+1)$ sets: $(A_1^i \times B_1^i), (A_2^i \times B_2^i)$ for $1 \leq i \leq T$, and the set $C \times D$, where

$$A_1^i = \{X \subset \{0,1\}^n | f_j(X) = X, \text{ for all } j < i \text{ and } f_i(X) \neq X\},$$
$$B_1^i = \{Y \subset \{0,1\}^n | g_j(Y) = Y, \text{ for all } j < i\},$$

$$A_2^i = \{X \subset \{0,1\}^n | f_j(X) = X, \text{ for all } j \leq i\},$$
$$B_2^i = \{Y \subset \{0,1\}^n | g_j(Y) = Y, \text{ for all } j < i \text{ and } g_i(Y) \neq Y\},$$

$$C = \{X \subset \{0,1\}^n | f_j(X) = X, \text{ for all } j \leq T\},$$
$$D = \{Y \subset \{0,1\}^n | g_j(Y) = Y, \text{ for all } j \leq T\}.$$

Note that if $(X,Y) \in A_j^i \times B_j^i$ for $j = 1, 2$, and $i \in [T]$, then $I = i$, and if $(X,Y) \in C \times D$, then $I = T + 1$. Also $f_I(X), g_I(Y)$ are empty strings if $I = T + 1$.

We call these partitions $P_1, ..., P_{2T+1}$.

Now suppose there is an adversary $\mathcal{A}$ that guesses the bit b with probability greater than $1/2 + (2T+1)\varepsilon/2$ given $I, f_I(X), g_I(Y)$. Let us say that $\mathcal{A}$ wins if $\mathcal{A}$ guesses the bit b correctly. Then

$$\begin{aligned}
&1/2 + (2T+1)\varepsilon/2 < \Pr[\mathcal{A} \text{ wins}] \\
&= \sum_{r=1}^{2T+1} \Pr[\mathcal{A} \text{ wins} \mid (X,Y) \in P_r] \cdot \Pr[(X,Y) \in P_r] \\
&= 1/2 + \sum_{r=1}^{2T+1} (\Pr[\mathcal{A} \text{ wins} \mid (X,Y) \in P_r] - 1/2) \cdot \Pr[(X,Y) \in P_r] \,.
\end{aligned}$$

Thus, there exists some r such that:

$$(\Pr[\mathcal{A} \text{ wins} \mid (X,Y) \in P_r] - 1/2) \cdot \Pr[(X,Y) \in P_r] > \varepsilon/2. \tag{4.2}$$

We now show that this contradicts the fact that $(\mathsf{Enc}, \mathsf{Dec})$ is $\varepsilon-$ super strong non-malleable in the split state model.

Case 1: $P_r = A_1^i \times B_1^i$ for some $i \in [T]$

Define the tampering function (f, g) as:

$$f(x) := \begin{cases} f_i(x) \text{ if } x \in A_1^i \\ u, \text{ otherwise.} \end{cases}$$

where u is some element not in $f_i(A_1^i)$.

$$g(y) := \begin{cases} g_i(y) \text{ if } y \in B_1^i \\ v, \text{ otherwise.} \end{cases}$$

where v is some element not in $g_i(B_1^i)$.

Then define an adversary $\mathcal{A}^*$ that given the tampering experiment of a random message m_b, outputs a fresh uniform random bit if it sees any of $(u, y), (x, v), \mathsf{same}$, or $\perp$, and calls $\mathcal{A}$ with input i, and the output of the tampering experiment otherwise. The success probability of $\mathcal{A}^*$ in guessing bit b is

$\Pr[\mathcal{A} \text{ wins} \mid (X, Y) \in P_r] \cdot Pr[(X, Y) \in P_r] + \frac{1}{2} \cdot (1 - Pr[(X, Y) \in P_r])$, which is greater than $\frac{1}{2} \cdot \Pr[(X, Y) \in P_r] + \frac{\varepsilon}{2} + \frac{1}{2} \cdot (1 - \Pr[(X, Y) \in P_r]) = 1/2 + \varepsilon/2$ using equation 4.2.

This contradicts the assumption that $(\mathsf{Enc}, \mathsf{Dec})$ is $\varepsilon-$ super strong non-malleable in the split state model.

Case 2: $P_r = A_2^i \times B_2^i$ for some $i \in [T]$

This case is similar to *Case 1.*

Case 3: $P_r = C \times D$.

In this case, the only information that $\mathcal{A}$ has is that $I = T + 1$, which is equivalent to saying that $(X, Y) \in C \times D$. Then let p_b be $\Pr((X_b, Y_b) \in C \times D)$ for $b = 0, 1$. By Lemma 4, we have that $|p_0 - p_1| \leq \varepsilon$. Without loss of generality, let $p_0 = p_1 + \varepsilon'$ for some $\varepsilon' \in [0, \varepsilon]$. Then given $(X_b, Y_b) \in C \times D$, the adversary has higher chance of winning if the adversary outputs 0.

Thus, $\Pr[\mathcal{A} \text{ wins} \mid (X_b, Y_b) \in C \times D] = \Pr[b = 0 | (X_b, Y_b) \in C \times D]$.
So, rewriting equation 4.2 assuming $P_r = C \times D$, we get that $\Pr[(X_b, Y_b) \in C \times D \wedge b = 0] - \frac{1}{2} \cdot \Pr[(X_b, Y_b) \in C \times D] > \varepsilon/2$. This implies, $\frac{1}{2} \cdot (p_1 + \varepsilon') - \frac{1}{2} \cdot \frac{1}{2} \cdot (p_1 + p_1 + \varepsilon') > \varepsilon/2$, which is equivalent to $\varepsilon' > 2\varepsilon$, which is a contradiction. □

Remark 4. The above reduction is in the split-state model. It may be interesting to note that the only place that we use a particular property of this model is equation 4.1, which can be generalized to saying that the random variable I combined with the output of tampering experiment should not reveal the message. It is also obvious that if this statement does not hold for some model then the reduction will not hold. That means that the above mentioned statement is in some sense a necessary and sufficient property of a tampering model in which the main reduction of this section is true.

5 Super Strong NMCs from Super NMCs via Inception Coding

In this section, we will show that any super non-malleable code in the split-state model can be converted into a super-strong non-malleable code in the split-state model. The main technique used here and called by us 'inception' is described in 5.2 (i.e. Definition 9). However before we start the actual definition and construction let us define some auxiliary objects in Section 5.1

5.1 Check Functions

In order to detect possible tampering with a string x, we introduce the following variant of *Universal Hashing Family.*

Definition 7. *A function $C : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^t$ is called an* ε-check *if for any $x, y \in \{0,1\}^n$ such that $x \neq y$,*

$$\Pr_{R \leftarrow \{0,1\}^s}(C(R,x) = C(R,y)) \leq \varepsilon$$

Remark 5. Every ε-check is also $(\varepsilon \cdot 2^t - 1)$-universal hashing family. Due to unnecessarily complicated normalization of parameters in standard UHF definition it is simply more convenient for us to use the *check* notion all through the paper.

In this section we give a construction of an efficient check function that has a short output length, short seed and has preimages with affine structure. Consider the following function.

Definition 8. *Let $q, t, n > 0$ be integers. Let $\mathsf{Check}_1 : \{0,1\}^n \to \{0,1\}^{q \log n}$ be such that for all $x \in \{0,1\}^n$, $x||\mathsf{Check}_1(x)$ is a valid Reed-Solomon code.*[6] *Let $\mathsf{Check}_2 : \{0,1\}^{t \log n} \times \{0,1\}^n \to \{0,1\}^t$ be a simple sampler function defined as follows. Let $r = r_1 \| r_2 \| \cdots \| r_t$ be such that each r_j is a $\log n$-bit string. Then $\mathsf{Check}_2(r, x) := x_{r_1} \dots x_{r_t}$, where x_{r_j} is the bit of x at position r_j, when written in binary form. Then we define the function $C_0 : \{0,1\}^{t \log(n)} \times \{0,1\}^n \to \{0,1\}^{q \log n + t}$ as $C_0(r, x) := \mathsf{Check}_1(x) \| \mathsf{Check}_2(r, x)$.*

Lemma 5. *The function C_0 defined above is a $e^{-\frac{qt}{n}}$-check.*

Proof. We want to bound the probability that for any two distinct $x, y \in \{0,1\}^n$ and $R = R_1 \| \dots \| R_t$ chosen uniformly at random from $\{0,1\}^{t \log n}$, $C_0(R, x) = C_0(R, y)$.

By Lemma 2, we have that the Hamming distance between $x \| \mathsf{Check}_1(x)$ and $y \| \mathsf{Check}_1(y)$ is at least $q+1$. Thus, if $\mathsf{Ham}(x; y) < q$ then $\mathsf{Check}_1(x) \neq \mathsf{Check}_1(y)$. So, for $C_0(R, x) = C_0(R, y)$ we must have that $\mathsf{Ham}(x; y) \geq q$. Additionally, we have that $\mathsf{Check}_2(R, x) = \mathsf{Check}_2(R, y)$ which implies $x_{R_j} = y_{R_j}$ for all $j \in [t]$.

[6] Correctness of this definition follows from Lemma 2.

This holds if none of $R_1, \ldots, R_t$ belong to the set of positions on which x and y are not different which occurs with probability at most

$$\left(1 - \frac{q}{n}\right)^t \leq e^{-\frac{qt}{n}} .$$

□

For our application, we require a check with the output having length upper bounded by n^α for a small constant $\alpha > 0$. Now, let us describe a composition lemma for check functions that will help us to reach the expected parameters.

Lemma 6. *If $C_0 : \{0,1\}^{s_1} \times \{0,1\}^n \mapsto \{0,1\}^{t_1}$ is an ε_1-check and $C : \{0,1\}^{s_2} \times \{0,1\}^{t_1} \mapsto \{0,1\}^{t_2}$ is an ε_2-check then $C_1 : \{0,1\}^{s_1+s_2} \times \{0,1\}^n \mapsto \{0,1\}^{t_2}$ given by*

$$C_1(r_1 \| r_2, x) := C(r_2, C_0(r_1, x))$$

is an $(\varepsilon_1 + \varepsilon_2)$-check.

Proof. Let $R_1 \| R_2 \leftarrow U_{s_1+s_2}$, and let $E_1 = E_1(R_1, x)$ be the event that $C_0(R_1, x) = C_0(R_1, y)$ and $E_2 = E_2(R_1, R_2, x)$ be the event that $C(R_2, C_0(R_1, x)) = C(R_2, C_0(R_1, y))$. Then

$$\begin{aligned} \Pr(E_2) &\leq \Pr(E_1) + \Pr(E_2 \mid \overline{E_1}) \\ &\leq \varepsilon_1 + \varepsilon_2 . \end{aligned}$$

□

We now apply Lemma 6 repeatedly to the construction of Lemma 5 to obtain a check with small length of both the output and the seed.

Lemma 7. *For any constant $\delta \in (0, 1/2)$ and for a large enough integer n, there exists an efficient $2^{-n^{\delta^2/5}}$-check $\mathsf{Check}^\star : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^t$ with $s \leq n^\delta$ and $t \leq n^\delta$.*

Proof. Let $\delta' = \delta/5$. We start with the construction from Lemma 5, and we set $t = n^{3\delta'}$, and $q = n^{1-2\delta'}$. Furthermore, we assume that output length $n_1 = q \log n + t \leq n^{1-\delta'}$, and $s_1 = t \log n \leq n^{4\delta'}$, which hold for a large enough n. The error is $e^{-n^{\delta'}}$.

We then define a check function for the output of length n_1, with seed length s_2 being at most $n_1^{4\delta'} \leq n^{(1-\delta')\cdot 4\delta'}$, output length n_2 being at most $n_1^{1-\delta'} \leq n^{(1-\delta')^2}$, and error is at most $e^{-n_1^{\delta'}}$.

We continue this procedure for ℓ steps until $n_\ell \leq n^\delta$. Thus $n_{\ell-1} > n^\delta$. The number of steps ℓ is upper bounded by $\log(1-\delta')/\log\delta$. Thus, using Lemma 6, the error is upper bounded by

$$\frac{\log(1-\delta')}{\log\delta} \cdot e^{-n^{5\delta'^2}} \leq 2^{-n^{5\delta'^2}}$$

and the total seed length is

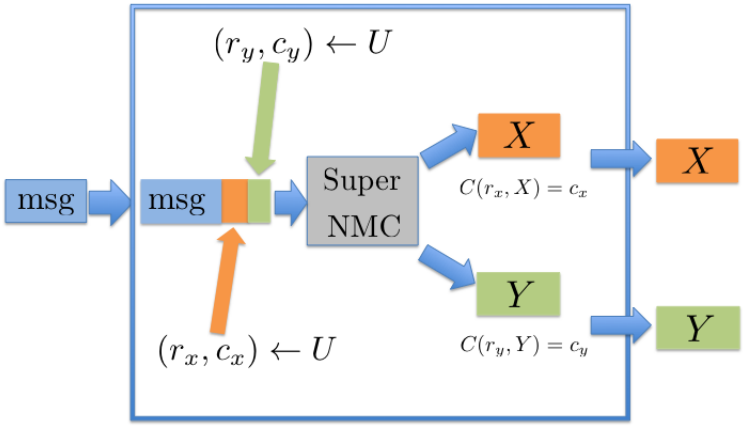
$$s_1 + \cdots + s_\ell \leq n^{4\delta'} \cdot \frac{\log(1-\delta')}{\log\delta} \leq n^\delta ,$$

where we again used that n is large enough. □

5.2 Inception Coding

In this section, we show that any super non-malleable code in the split-state model can be converted into a super-strong non-malleable code in the split-state model. Notice that for some message m with $(X, Y) \leftarrow \mathsf{Enc}(m)$, the only possible scenario in which the output of the tampering experiment in the super-strong non-malleability definition and that in the super non-malleability definition are different is when $\mathsf{Dec}(X, Y) = \mathsf{Dec}(f(X), g(Y))$ even in the case of a non-trivial tampering, i.e., $(X, Y) \neq (f(X), g(Y))$. Our idea is to use some of the least significant bits of the message to store a seed and an output of a "Check" such that if the decoder outputs the correct message in case of a non-trivial tampering, then the "Check" can detect this and force the output to be $\perp$. This technique of installing a validity check for a codeword within the message is what we call inception coding and is defined below.

Figure 1. Inception coding using super non-malleable code.



Definition 9. *Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n \times \{0,1\}^n$, $\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k \cap \{\perp\}$ *be a coding scheme. Let* $C : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^t$ *be some function.*[7] *The* Inception version *of* $(\mathsf{Enc}, \mathsf{Dec}, C)$ *is a coding scheme denoted as* $\mathcal{I}\mathsf{Enc} : \{0,1\}^{k-2s-2t} \to \{0,1\}^n \times \{0,1\}^n$, $\mathcal{I}\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{k-2s-2t} \cup \{\perp\}$ *and is defined as follows. The encoding algorithm* $\mathcal{I}\mathsf{Enc}$*, for a given message* $m \in \{0,1\}^{k-2s-2t}$*, does the following.*

- *Choose uniformly at random* r_x, r_y *from* $\{0,1\}^s$*, and* c_x, c_y *from* $\{0,1\}^t$*.*
- *Sample* (X, Y) *as the output of the encoding algorithm* Enc *on input* $(m \| r_x \| c_x \| r_y \| c_y)$ *conditioned on the fact that* $C(r_x, X) = c_x$ *and* $C(r_y, Y) = c_y$*.*

[7] We will use this definition with C being a check function.

– *Output* (X, Y).

The decoding algorithm $\mathcal{I}\mathsf{Dec}$*, on input* $x, y \in \{0,1\}^n$*, does the following.*

– *Obtain* $\mathsf{Dec}(x, y) \in \{0,1\}^k$*, and interpret the output as* $(m \| r_x \| c_x \| r_y \| c_y)$*, where* $m \in \{0,1\}^{k-2s-2t}$*,* $r_x, r_y \in \{0,1\}^s$*, and* $c_x, c_y \in \{0,1\}^t$*.*
– *If* $C(r_x, x) = c_x$ *and* $C(r_y, y) = c_y$ *then output* m*, else output* $\perp$*.*

We now state our main result.

Theorem 3. *Let* $\varepsilon_1, \varepsilon_2 > 0$*.* $C : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^t$ *be an* ε_1*-check. Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n \times \{0,1\}^n$*,* $\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k \cap \{\perp\}$ *be a uniform* ε_2*-super non-malleable code in the split-state model such that for any* m, r_x, c_x, r_y, c_y*, there is an efficient algorithm to sample* $(X, Y) \leftarrow \mathsf{Enc}(m)$ *conditioned on* $C(r_x, X) = c_x$ *and* $C(r_y, Y) = c_y$*. Then* $(\mathcal{I}\mathsf{Enc}, \mathcal{I}\mathsf{Dec})$ *is an efficient* ε'*-super strong non-malleable code in the split-state model with* $\varepsilon' = \frac{16\varepsilon_2}{2^{-2t}} + 2\varepsilon_1 + 3\varepsilon_2$*.*

Proof. Let $f : \{0,1\}^n \mapsto \{0,1\}^n$, $g : \{0,1\}^n \mapsto \{0,1\}^n$ be arbitrary functions and $m, m' \in \{0,1\}^{k-2s-2t}$ be arbitrary messages. We will bound the statistical distance between $\mathsf{SupStrTamp}_m^{f,g}$ and $\mathsf{SupStrTamp}_{m'}^{f,g}$ for the encoding scheme $(\mathcal{I}\mathsf{Enc}, \mathcal{I}\mathsf{Dec})$. For this purpose, we intend to use the fact that $(\mathsf{Enc}, \mathsf{Dec})$ is super non-malleable. However, the main issue with using this is that the codeword obtained by using Enc might not be a valid encoding for $\mathcal{I}\mathsf{Enc}$. The main idea to make sure that the encoding is valid is to (artificially) do rejection sampling. We modify the tampering functions f, g to f', g' such that the tampered codeword becomes irrelevant if the code is not a valid codeword with respect to $\mathcal{I}\mathsf{Enc}$. This is the reason that the error is blown-up by a factor 2^{2t}.

Let the space of all $x \in \{0,1\}^n$ such that $C(r, x) = c$ be $A_{r,c}$, i.e.,

$$A_{r,c} := \{x \in \{0,1\}^n \,|\, C(r, x) = c\} \,.$$

We choose fresh uniformly random and independent strings r_x, r_y from $\{0,1\}^s$, and c_x, c_y from $\{0,1\}^t$. Consider the following functions:

$$f'(x) := \begin{cases} f(x) \text{ if } x \in A_{r_x,c_x} \\ 0^n, \text{ otherwise.} \end{cases}$$

$$g'(y) := \begin{cases} g(y) \text{ if } y \in A_{r_y,c_y} \\ 0^n \text{ otherwise.} \end{cases}$$

Let $(X, Y) \leftarrow \mathsf{Enc}(m, r_x, c_x, r_y, c_y)$ and let $(X', Y') \leftarrow \mathsf{Enc}(m', r_x, c_x, r_y, c_y)$. We shorthand $\mathsf{SupTamp}^{f',g'}_{(m,r_x,c_x,r_y,c_y)}$ by T and $\mathsf{SupTamp}^{f',g'}_{(m',r_x,c_x,r_y,c_y)}$ by T'. The range of T and T' is $\mathcal{R} = \{0,1\}^n \times \{0,1\}^n \cup \{\perp, \mathsf{same}\}$. Also, let $\mathcal{A} = A_{r_x,c_x} \times A_{r_y,c_y}$, and let $\Pr((X, Y) \in \mathcal{A}) = p$ and $\Pr((X', Y') \in \mathcal{A}) = p'$. By Lemma 4, we have that $|p - p'| \leq \varepsilon_2$, and by the fact that $(\mathsf{Enc}, \mathsf{Dec})$ is almost uniform, we have that $p \geq 2^{-2t-1}$.

Also, if $(X, Y) \notin \mathcal{A}$, then $(f'(X), g'(Y))$ depends on at most one of X, Y, and if $(X', Y') \notin \mathcal{A}$, then $(f'(X'), g'(Y'))$ depends on at most one of X', Y'. Hence the respective tampering experiments T and T' depend on at most one of the shares and by Lemma 3, we have that in this case T and T' are statistically close, i.e.,:

$$\frac{1}{2} \cdot \sum_{z \in \mathcal{R}} |\Pr(T = z \ \wedge \ (X, Y) \notin \mathcal{A}) - \Pr(T' = z \ \wedge \ (X', Y') \notin \mathcal{A})| \leq 2\varepsilon_2 \ . \quad (5.1)$$

Also, by the super non-malleability assumption, we have that $\Delta(T; T') \leq \varepsilon_2$. Thus, using Equation 5.1, and the triangle inequality, we have that

$$\begin{aligned}
6\varepsilon_2 &\geq \sum_{z \in \mathcal{R}} \Big| \Pr(T = z \ \wedge \ (X, Y) \in \mathcal{A}) - \Pr(T' = z \ \wedge \ (X', Y') \in \mathcal{A})| \\
&= \sum_{z \in \mathcal{R}} \Big| \Pr(T = z \mid (X, Y) \in \mathcal{A}) \cdot p - \Pr(T' = z | (X', Y') \in \mathcal{A}) \cdot p' \Big| \\
&\geq p \cdot \sum_{z \in \mathcal{R}} |\Pr(T = z | (X, Y) \in \mathcal{A}) - \Pr(T' = z | (X', Y') \in \mathcal{A})| - |p - p'| \\
&\geq (2^{-2t-1}) \cdot \sum_{z \in \mathcal{R}} |\Pr(T = z | (X, Y) \in \mathcal{A}) - \Pr(T' = z | (X', Y') \in \mathcal{A})| - 2\varepsilon_2 \ .
\end{aligned}$$

This implies that

$$\sum_{z \in \mathcal{R}} |\Pr(T = z | (X, Y) \in \mathcal{A}) - \Pr(T' = z | (X', Y') \in \mathcal{A})| \leq \frac{8\varepsilon_2}{2^{-2t-1}} \ .$$

Let $\widetilde{T}$ be the tampering experiment T conditioned on the event $(X, Y) \in \mathcal{A}$. Similarly define $\widetilde{T}'$.

We now compare the experiments $\widetilde{T}$ and $\mathsf{SupStrTamp}_m^{f,g}$. For the purpose of this comparison, we assume that the random coins needed to generate r_x, c_x, r_y, c_y, and $(X, Y) \leftarrow \mathsf{Enc}(m)$ conditioned on $(X, Y) \in \mathcal{A}$ are the same. Then, we have that if $\widetilde{T} \neq \mathsf{same}$, then $\mathsf{SupStrTamp}_m^{f,g}$ is equal to $\widetilde{T}$ except with probability at most ε_2. To see this, notice that if both $\widetilde{T}$ and $\mathsf{SupStrTamp}_m^{f,g}$ are not same, then they are equal. The event that $\widetilde{T} \neq \mathsf{same}$ and $\mathsf{SupStrTamp}_m^{f,g} = \mathsf{same}$ happens if $f(X) = X, g(Y) = Y$ but $D_{X,Y}^{f,g} = 1$. This cannot happen with probability more than ε_2, since this would mean that $T = (X, Y)$ which would immediately reveal the message thereby contradicting the non-malleability of $(\mathsf{Enc}, \mathsf{Dec})$.

Also, we claim that if $\widetilde{T} = \mathsf{same}$, then $\mathsf{SupStrTamp}_m^{f,g} \in \{\mathsf{same}, \perp\}$, except with probability at most ε_1. This follows from the fact that if $\widetilde{T} = \mathsf{same}$, and $\mathsf{SupStrTamp}_m^{f,g} \notin \{\mathsf{same}, \perp\}$, then this implies that at least one of $f(X) \neq X$, or $g(Y) \neq Y$ but $C(r_x, f(X)) = c_x$, and $C(r_y, g(Y)) = c_y$ which happens with probability at most ε_1.

Thus, we can bound the statistical distance between $\mathsf{SupStrTamp}_m^{f,g}$ and $\mathsf{SupStrTamp}_{m'}^{f,g}$ by

$$\frac{8\varepsilon_2}{2^{-2t-1}} + 2\varepsilon_1 + 2\varepsilon_2 + |\Pr(\mathsf{SupStrTamp}_m^{f,g} = \mathsf{same}) - \Pr(\mathsf{SupStrTamp}_{m'}^{f,g} = \mathsf{same})| \ .$$

Finally, using Lemma 4, we can conclude that

$$|\Pr(\mathsf{SupStrTamp}_m^{f,g} = \mathsf{same}) - \Pr(\mathsf{SupStrTamp}_{m'}^{f,g} = \mathsf{same})| \leq \varepsilon_2$$

by setting $A = \{x \in \{0,1\}^n : f(x) = x\}$, and $B = \{y \in \{0,1\}^n : g(y) = y\}$. □

6 Instantiating a Super Non-malleable Code

In [ADL14], Aggarwal et al. gave a construction of non-malleable codes in the split-state model. Here, we argue that the construction of [ADL14] is also super-non-malleable.

In [AB16], Aggarwal et al. improved the analysis of the [ADL14] construction, when talking about parameters we will be recalling parameters from [AB16].

Note that for any message m with $\mathsf{Enc}(m) = (X, Y)$, and any functions f, g, the output of the tampering experiment in Definition 3 is the same as that in Definition 4 if $\mathsf{Dec}(f(X), g(Y)) = m$ or $\mathsf{Dec}(f(X), g(Y)) = \perp$. This leads to the following simple observation.

Observation 6.1 *Let $\varepsilon, \varepsilon' > 0$. Let $(\mathsf{Enc} : \mathcal{M} \to \mathcal{X} \times \mathcal{X}, \mathsf{Dec} : \mathcal{X} \times \mathcal{X} \to \mathcal{M} \cup \{\perp\})$ be an ε-non-malleable code in the split-state model. Given $f, g : \mathcal{X} \mapsto \mathcal{X}$, assume there exists a partitioning $(\mathcal{S}_1, \cdots, \mathcal{S}_{s+t}, \mathcal{S}^\star)$ of $\mathcal{X} \times \mathcal{X}$ such that the following hold:*

1. *For all $m \in \mathcal{M}$, $1 \leq i \leq s$, $\Pr_{(X,Y) \leftarrow \mathsf{Enc}(m)}(\mathsf{Dec}(f(X), g(Y)) \in \{m, \perp\} | (X, Y) \in \mathcal{S}_i) \geq 1 - \varepsilon'$.*
2. *For all $m_1, m_2 \in \mathcal{M}$, $s+1 \leq i \leq s+t$, let (X_1, Y_1), (X_2, Y_2) be the encoding of m_1, m_2 respectively, conditioned on the fact that $(X_1, Y_1), (X_2, Y_2) \in \mathcal{S}_i$. Then $\Delta((f(X_1), g(Y_1)), (f(X_2), g(Y_2)) \leq \varepsilon'$.*
3. *For any $m \in \mathcal{M}$, $\Pr(\mathsf{Enc}(m) \in \mathcal{S}^\star) \leq \varepsilon'$.*

Then, the scheme $(\mathsf{Enc}, \mathsf{Dec})$ is $(\varepsilon + O(\varepsilon'))$-super-non-malleable.

In the above observation, we set $D^{f,g}_{(X,Y)}$ to be 1 if $(X, Y) \in \mathcal{S}_1, \ldots, \mathcal{S}_s$, and 0, otherwise.

Before describing the encoding scheme from [ADL14], we will need the following definition of an affine-evasive function.

Definition 10. *Let $\mathbb{F} = \mathbb{F}_p$ be a finite field. A surjective function $h : \mathbb{F} \mapsto \mathcal{M} \cup \{\perp\}$ is called (γ, δ)-*affine-evasive *if or any $a, b \in \mathbb{F}$ such that $a \neq 0$, and $(a, b) \neq (1, 0)$, and for any $m \in \mathcal{M}$,*

1. *$\Pr_{U \leftarrow \mathbb{F}}(h(aU + b) \neq \perp) \leq \gamma$*
2. *$\Pr_{U \leftarrow \mathbb{F}}(h(aU + b) \neq \perp \mid h(U) = m) \leq \delta$*
3. *A uniformly random X such that $h(X) = m$ is efficiently samplable.*

Aggarwal [Agg15] showed the following.

Lemma 8. *There exists an efficiently computable $(p^{-3/4}, \Theta(|\mathcal{M}| \log p \cdot p^{-1/4}))$-affine-evasive function $h : \mathbb{F} \mapsto \mathcal{M} \cup \{\perp\}$.*

We now describe the coding scheme from [ADL14] combined with the affine-evasive function promised by Lemma 8. Let $\mathcal{M} = \{1, \ldots, K\}$ and $\mathcal{X} = \mathbb{F}^N$, where $\mathbb{F}$ is a finite field of prime order p such that $p \geq (K/\varepsilon)^{16}$, and N chosen as $C \log^4 p$, where C is some universal constant.

Then for any $m \in \mathcal{M}$, $\mathsf{Enc}(m) = \mathsf{Enc}_1 \circ \mathsf{Enc}_2(m)$, where for any $m \in \mathcal{M}$, $\mathsf{Enc}_2(m)$ is X where X is uniformly random such that $h(X) = m$, where h is affine-evasive function defined earlier, and for any $x \in \mathbb{F}$, $\mathsf{Enc}_1(x) = (L, R)$, where $L, R \in \mathbb{F}^N$ are uniform such that $\langle L, R \rangle = x$.

The decoding algorithm is as follows. For $\ell, r \in \mathbb{F}^N \times \mathbb{F}^N$, $\mathsf{Dec}(\ell, r) = \mathsf{Dec}_2 \circ \mathsf{Dec}_1(\ell, r)$, where for any $\ell, r \in \mathbb{F}^N$, $\mathsf{Dec}_1(\ell, r) = \langle \ell, r \rangle$, and for any $x \in \mathbb{F}$, $\mathsf{Dec}_2(x) = h(x)$.

The following is implicit in [ADL14].

Theorem 4. *Let $f, g : \mathbb{F}^N \mapsto \mathbb{F}^N$ be arbitrary functions. Let $s = \lfloor N/20 \rfloor$, and let $t = \lfloor \frac{s^{1/4}}{c \log p} \rfloor$, for some universal constant c. Then, there exists a set $\mathcal{S} \subset \mathbb{F}^N \times \mathbb{F}^N$ of size at most p^{2N-s} such that $\mathbb{F}^N \times \mathbb{F}^N \setminus \mathcal{S}$ can be partitioned into sets of the form*

1. *$\mathcal{L} \times \mathcal{R}$ such that $(\langle L', R' \rangle, \langle f(L'), g(R') \rangle)$ is p^{-t}-close to uniform for L', R' uniform in $\mathcal{L}, \mathcal{R}$ respectively.*
2. *$\mathcal{L} \times \mathcal{R}$, such that $|\mathcal{L} \times \mathcal{R}| \geq p^{2N-7s}$, and there exists $A \in \mathbb{F}^{N \times N}$, $a \neq 0 \in \mathbb{F}, b \in \mathbb{F}^n$ such that $f(\ell) = A\ell$ for all $\ell \in \mathcal{L}$, and $A^T g(r) = ar + b$ for all $r \in \mathcal{R}$.*
3. *$\mathcal{L} \times \mathcal{R}$, such that $|\mathcal{L} \times \mathcal{R}| \geq p^{2N-7s}$, and there exists $y \in \mathbb{F}^N$, such that $g(r) = y$ for all $y \in \mathcal{R}$.*

To argue that the construction given above is also super-non-malleable, we will need the following:

Lemma 9. *Let L and R be independent random variables over $\mathbb{F}^N$. If*

$$\mathbf{H}_\infty(L) + \mathbf{H}_\infty(R) \geq (N+1) \log p + 2 \log \left(\frac{1}{\varepsilon} \right) ,$$

then

$$\Delta((L, \langle L, R \rangle) \; ; \; (L, U_{\mathbb{F}})) \leq \varepsilon \text{ and } \Delta((R, \langle L, R \rangle) \; ; \; (R, U_{\mathbb{F}})) \leq \varepsilon.$$

Lemma 10. *Let $X_1, Y_1 \in \mathcal{A}$, and $X_2, Y_2 \in \mathcal{B}$ be random variables such that $\Delta((X_1, X_2) \; ; \; (Y_1, Y_2)) \leq \varepsilon$. Then, for any non-empty set $\mathcal{A}_1 \subseteq \mathcal{A}$, we have*

$$\Delta(X_2 \mid X_1 \in \mathcal{A}_1 \; ; \; Y_2 \mid Y_1 \in \mathcal{A}_1) \leq \frac{2\varepsilon}{\Pr(X_1 \in \mathcal{A}_1)} .$$

Theorem 5. *The scheme $(\mathsf{Enc}, \mathsf{Dec})$ is almost uniform, $O(\varepsilon)$-super-non-malleable code in the split-state model.*

Proof. We first show that the scheme is a super non-malleable code in the split-state model. We will argue that each partition promised by Theorem 4 is one of $\mathcal{S}_1, \ldots, \mathcal{S}_{s+t}, \mathcal{S}^\star$ as in Observation 6.1 with $\varepsilon' = \varepsilon$. Clearly, for any $m \in \mathcal{M}$, $\Pr(\mathsf{Enc}(m) \in \mathcal{S}) \leq p^{-s+1} \leq \varepsilon$, and hence we can set $\mathcal{S}^\star = \mathcal{S}$. So, we consider the partitioning of $\mathbb{F}^n \times \mathbb{F}^n \setminus \mathcal{S}$.

1. $\mathcal{L} \times \mathcal{R}$ such that $(\langle L', R' \rangle, \langle f(L'), g(R') \rangle)$ is p^{-t}-close to uniform for L', R' uniform in $\mathcal{L}, \mathcal{R}$ respectively. In this case, for any message m, if $(L, R) \leftarrow \mathsf{Enc}(m)$, then $\mathsf{Dec}(f(L), g(R))$ conditioned on $(L, R) \in \mathcal{L} \times \mathcal{R}$ is $h(\langle f(L'), g(R') \rangle)$ conditioned on $h(\langle L', R' \rangle) = m$. By Lemma 10, we have that this is $2 \cdot p^{-t+1}$-close to uniform, and hence, by Lemma 8, we have that $h(\langle f(L'), g(R') \rangle) = \perp$ with probability at least $1 - p^{-3/4} - p^{-t+1} \geq 1 - \varepsilon$.
2. $\mathcal{L} \times \mathcal{R}$, such that $|\mathcal{L} \times \mathcal{R}| \geq p^{2N-7s}$, and there exists $A \in \mathbb{F}^{N \times N}$, $a \in \mathbb{F}, b \in \mathbb{F}^N$ such that $f(\ell) = A\ell$ for all $\ell \in \mathcal{L}$, and $A^T g(r) = ar + b$ for all $r \in \mathcal{R}$. In this case, using the same argument as in the previous item, we have that $\mathsf{Dec}(f(L), g(R))$ conditioned on $(L, R) \in \mathcal{L} \times \mathcal{R}$ is $\perp$ with probability at least $1 - p^{-1/4} \log p - p^{-t+1} \geq 1 - \varepsilon$.
3. $\mathcal{L} \times \mathcal{R}$, such that $|\mathcal{L} \times \mathcal{R}| \geq p^{2N-7s}$, and there exists $y \in \mathbb{F}^N$, such that $g(r) = y$ for all $y \in \mathcal{R}$. Let L', R' uniform in $\mathcal{L}, \mathcal{R}$, respectively. Then, using Lemma 9, we have that $\langle L', R' \rangle$ is $p^{-(N-7s-1)/2}$-close to uniform given $f(L')$, and $g(R') = y$, and so, using Lemma 10, this partition satisfies item 2 from Observation 6.1.

The result then follows from Observation 6.1.

We now show that the scheme is uniform. Let $\mathcal{X}_0, \mathcal{Y}_0 \subset \mathbb{F}^N$ such that $|\mathcal{X}_0| = p^{c_1 N}$, and $|\mathcal{Y}_0| = p^{c_2 N}$ for some $c_1, c_2 \in (1/2, 1)$, and let $\mathcal{X}_1 = \mathbb{F}^N \setminus \mathcal{X}_0$, and $\mathcal{Y}_1 = \mathbb{F}^N \setminus \mathcal{Y}_0$. Let X_0, X_1, Y_0, Y_1 be uniform in $\mathcal{X}_0, \mathcal{X}_1, \mathcal{Y}_0, \mathcal{Y}_1$, respectively. Then by Lemma 9, there exists $c > 0$, such that for $i, j \in \{0, 1\}$,

$$\Delta(\langle X_i, Y_j \rangle \; ; \; U_{\mathbb{F}}) \leq p^{-cN} \; .$$

Thus, for any $a \in \mathbb{F}_p$, the number of $x \in \mathcal{X}_i, y \in \mathcal{Y}_j$ such that $\langle x, y \rangle = a$ is

$$|\mathcal{X}_i| \cdot |\mathcal{Y}_j| \cdot (\frac{1}{p} \pm p^{-cN}) \; .$$

Thus the fraction of $(x, y) \in \mathcal{X}_0 \times \mathcal{Y}_0$ such that $\langle x, y \rangle = a$ is in the interval

$$\left(\frac{|\mathcal{X}_i| \cdot |\mathcal{Y}_j|}{p^{2N}} \cdot \frac{1 - p^{-cN+1}}{1 + p^{-cN+1}} \; , \; \frac{|\mathcal{X}_i| \cdot |\mathcal{Y}_j|}{p^{2N}} \cdot \frac{1 + p^{-cN+1}}{1 - p^{-cN+1}} \right) \; ,$$

which implies the result. □

7 Final proof of the main result

Theorem 5 proves that non-malleable code from [ADL14] is super non-malleable. The only additional requirement that needs to be fulfilled in order to be able

to use this code to obtain super strong non-malleable codes using Theorem 3 is that there is an efficient algorithm to sample $(X, Y) \leftarrow \mathsf{Enc}(m)$ conditioned on $C(r_x, X) = c_x$ and $C(r_y, Y) = c_y$ for some given r_x, r_y, c_x, c_y, m. Note that here, $X, Y \in \mathbb{F}^N$, which is thought of as being embedded in to $\{0,1\}^n$ for $n = N\lceil \log p \rceil$. A way to sample this will be to sample $a \leftarrow \mathsf{Enc}_2(m) \in \mathbb{F}_p$, and then try to sample X, Y such that $\langle X, Y \rangle = a$ (where X, Y are interpreted as elements of $\mathbb{F}^N$) and $C(r_x, X) = c_x$ and $C(r_y, Y) = c_y$ (where X, Y are interpreted as elements of $\{0,1\}^n$).

Since we don't know how to sample this efficiently, we resolve this issue by introducing an alternate definition of inception coding, which we call partial inception coding, that installs only a check for X into the message.

Definition 11. *Let* $\mathsf{Enc} : \{0,1\}^k \to \{0,1\}^n \times \{0,1\}^n$, $\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k \cap \{\perp\}$ *be a coding scheme. Let* $C : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^t$ *be some function.*[8] *The* Partial Inception version *of* $(\mathsf{Enc}, \mathsf{Dec}, C)$ *is a coding scheme denoted as* $\mathcal{I}\mathsf{Enc} : \{0,1\}^{k-s-t} \to \{0,1\}^n \times \{0,1\}^n$, $\mathcal{I}\mathsf{Dec} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{k-s-t} \cup \{\perp\}$ *and is defined as follows. The encoding algorithm* $\mathcal{I}\mathsf{Enc}$*, for a given message* $m \in \{0,1\}^{k-s-t}$*, does the following.*

- *Choose uniformly at random* r_x *from* $\{0,1\}^s$*, and* c_x *from* $\{0,1\}^t$*.*
- *Sample* (X, Y) *as the output of the encoding algorithm* Enc *on input* $(m \| r_x \| c_x)$ *conditioned on the fact that* $C(r_x, X) = c_x$*.*
- *Output* (X, Y)*.*

The decoding algorithm $\mathcal{I}\mathsf{Dec}$*, on input* $x, y \in \{0,1\}^n$*, does the following.*

- *Obtain* $\mathsf{Dec}(x, y) \in \{0,1\}^k$*, and interpret the output as* $(m \| r_x \| c_x)$*, where* $m \in \{0,1\}^{k-s-t}$*,* $r_x \in \{0,1\}^s$*, and* $c_x \in \{0,1\}^t$*.*
- *If* $C(r_x, x) = c_x$ *then output* m*, else output* $\perp$*.*

Then, it is easy to sample from the desired distribution. One can efficiently sample X conditioned on $C(X, r_X) = c_X$ since for any $r \in \{0,1\}^s$ and any $c \in \{0,1\}^t$ the set of all x such that $C(r, x) = c$ is an affine subspace of $\{0,1\}^n$. This follows immediately from Lemma 2 and Definition 8. Then, Y can be sampled easily conditioned on the constraint that $\langle X, Y \rangle = a$.

However, this introduces an additional requirement on the non-malleable code that the adversary cannot decode to the same message by changing just one part of the codeword, i.e., for any function $g : \{0,1\}^n \mapsto \{0,1\}^n$, and any message m with $(X, Y) \leftarrow \mathsf{Enc}(m)$, the probability that $g(Y) \neq Y$ and $\mathsf{Dec}(X, g(Y)) = m$ is small. This condition, fortunately, is immediate from the proof of Theorem 5, where item (2) with A being the identity matrix corresponds to this case, and unless g is also the identity function, we conclude that $\mathsf{Dec}(X, g(Y)) = m$ with probability at most ε.

Remark 6. The main reason that we did not define partial inception coding to start with is because we did not want to restrict Theorem 3 in the sense that

[8] We will use this definition with C being a check function.

it only works if we instantiate it with a non-malleable code that has the special property that the probability that $g(Y) \neq Y$ and $\mathsf{Dec}(X, g(Y)) = m$ is small. This, we believe is just a minor technicality since we are having difficulty in sampling X, Y conditioned on $C(r_X, X) = c_X$, $C(r_Y, Y) = c_Y$ and $\langle X, Y \rangle = a$. Perhaps using a clever sampling algorithm like the one used by Chattopadhyay and Zuckerman [CZ14], such a sampling is possible. Even if this is not the case, we want Theorem 3 to be general enough so that it can be instantiated with other super non-malleable codes.

Thus, using a result analogous to Theorem 3 for the case of Partial Inception coding introduced in Definition 11 and instantiating it with $(\mathsf{Enc}, \mathsf{Dec})$ from [ADL14] gives us the following result.

Theorem 6. *There exists an efficient $2^{-k^{\Omega(1)}}$-super-strong non-malleable code in the split-state model from k-bit messages to k^5-bit codewords.*

Combining Theorem 6 with Theorem 2 gives us the main result of the paper, i.e., a construction of a persistent continuous non-malleable code in the split-state model.

Theorem 7. *There exists an efficient $(T, (T+1) \cdot 2^{-k^{\Omega(1)}})-$continuous $[1, 1]$ non-malleable code in the split-state model from k-bit messages to k^5-bit codewords.*

References

AAnHKM+16. Divesh Aggarwal, Shashank Agrawal, Divya Gupta nad Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Optimal computational split state non-malleable codes. *To appear in TCC 16-A*, 2016.

AB16. Divesh Aggarwal and Jop Briët. Revisiting the sanders-bogolyubov-ruzsa theorem in fpn and its application to non-malleable codes. *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1322–1326, 2016.

ADKO15a. Divesh Aggarwal, Yevgeniy Dodis, Tomasz Kazana, and Maciej Obremski. Leakage-resilient non-malleable codes. In *The 47th ACM Symposium on Theory of Computing (STOC)*, 2015.

ADKO15b. Divesh Aggarwal, Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Leakage-resilient non-malleable codes. In *Theory of Cryptography*, volume 9014 of *Lecture Notes in Computer Science*, pages 398–426. Springer Berlin Heidelberg, 2015.

ADL14. Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *STOC*. ACM, 2014.

ADN+19. Divesh Aggarwal, Nico Döttling, Jesper Buus Nielsen, Maciej Obremski, and Erick Purwanto. Continuous non-malleable codes in the 8-split-state model. *Eurocrypt 2019*, 2019.

Agg15. Divesh Aggarwal. Affine-evasive sets modulo a prime. *Information Processing Letters*, 115(2):382–385, 2015.

AGM+15a. Shashank Agrawal, Divya Gupta, Hemanta K. Maji, Omkant Pandey, and Manoj Prabhakaran. Explicit non-malleable codes resistant to permutations. *Advances in Cryptology - CRYPTO*, 2015.

AGM+15b. Shashank Agrawal, Divya Gupta, HemantaK. Maji, Omkant Pandey, and Manoj Prabhakaran. A rate-optimizing compiler for non-malleable codes against bit-wise tampering and permutations. In *Theory of Cryptography*, volume 9014 of *Lecture Notes in Computer Science*, pages 375–397. Springer Berlin Heidelberg, 2015.

CDTV16. Sandro Coretti, Yevgeniy Dodis, Björn Tackmann, and Daniele Venturi. Non-malleable encryption: Simpler, shorter, stronger. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 306–335, 2016.

CG14a. Mahdi Cheraghchi and Venkatesan Guruswami. Capacity of non-malleable codes. In *ITCS*, 2014.

CG14b. Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *TCC*, 2014.

CGL15. Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Non-malleable extractors and codes, with their many tampered extensions. *CoRR*, abs/1505.00107, 2015.

CGM+15. Nishanth Chandran, Vipul Goyal, Pratyay Mukherjee, Omkant Pandey, and Jalaj Upadhyay. Block-wise non-malleable codes. *IACR Cryptology ePrint Archive*, 2015:129, 2015.

CKM11. Seung Geol Choi, Aggelos Kiayias, and Tal Malkin. Bitr: built-in tamper resilience. In *Advances in Cryptology–ASIACRYPT 2011*, pages 740–758. Springer, 2011.

CMTV15. Sandro Coretti, Ueli Maurer, Björn Tackmann, and Daniele Venturi. From single-bit to multi-bit public-key encryption via non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 532–560, 2015.

CZ14. Eshan Chattopadhyay and David Zuckerman. Non-malleable codes in the constant split-state model. *FOCS*, 2014.

DDN00. D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM*, 30:391–437, 2000.

DKO13. Stefan Dziembowski, Tomasz Kazana, and Maciej Obremski. Non-malleable codes from two-source extractors. In *Advances in Cryptology-CRYPTO 2013*. Springer, 2013.

DORS08. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

DPW10. Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *ICS*, pages 434–452. Tsinghua University Press, 2010.

DW09. Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 601–610, Bethesda, MD, USA, 2009. ACM.

FMNV14. S. Faust, P. Mukherjee, J. Nielsen, and D. Venturi. Continuous non-malleable codes. In *Theory of Cryptography Conference - TCC*. Springer, 2014.

FMVW14. S. Faust, P. Mukherjee, D. Venturi, and D. Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *Eurocrypt*. Springer, 2014.

GLM+03. Rosario Gennaro, Anna Lysyanskaya, Tal Malkin, Silvio Micali, and Tal Rabin. Algorithmic Tamper-Proof (ATP) security: Theoretical foundations for security against hardware tampering. In Moni Naor, editor, *First Theory of Cryptography Conference — TCC 2004*, volume 2951 of *LNCS*, pages 258–277. Springer-Verlag, February 19–21 2003.

IPSW06. Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and David Wagner. Private circuits II: Keeping secrets in tamperable circuits. In Serge Vaudenay, editor, *Advances in Cryptology—EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 308–327. Springer-Verlag, 2006.

ISW03. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology—CRYPTO 2003*, volume 2729 of *LNCS*. Springer-Verlag, 2003.

JW15. Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. In *Theory of Cryptography*, volume 9014 of *Lecture Notes in Computer Science*, pages 451–480. Springer Berlin Heidelberg, 2015.

KKS11. Yael Tauman Kalai, Bhavana Kanukurthi, and Amit Sahai. Cryptography with tamperable and leaky memory. In *Advances in Cryptology–CRYPTO 2011*, pages 373–390. Springer, 2011.

Li16. Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. *arXiv*, 2016.

LL12. Feng-Hao Liu and Anna Lysyanskaya. Tamper and leakage resilience in the split-state model. In *Advances in Cryptology–CRYPTO 2012*, pages 517–532. Springer, 2012.

RU08. Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. Cambridge University Press, New York, NY, USA, 2008.

A Equivalence of Our Non-malleable Codes Definition (Def. 3) with that of [DPW10]

Theorem 8. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* $\varepsilon-$*non-malleable code then it is also an* $\varepsilon-$*non-malleable code according to the definition from [DPW10].*

Proof. Let us define transform $T_m : \mathcal{M} \cup \{\bot, \mathsf{same}\} \to \mathcal{M} \cup \{\bot\}$ as follows: for any $m' \in \mathcal{M}$ let $T_m(m') = m'$, $T_m(\bot) = \bot$, $T_m(\mathsf{same}) = m$. Notice that $T_m(\mathsf{Tamp}_m^{f,g}) = \mathsf{DPWTamp}_m^{f,g}$. Fix any message m_0, and take $D^{f,g} = \mathsf{Tamp}_{m_0}^{f,g}$. We know that $\mathsf{Tamp}_m^{f,g} \approx_\varepsilon \mathsf{Tamp}_{m_0}^{f,g}$ for any functions f, g and any message m. Thus

$$T_m(\mathsf{Tamp}_m^{f,g}) \approx_\varepsilon T_m(\mathsf{Tamp}_{m_0}^{f,g}),$$
$$\mathsf{DPWTamp}_m^{f,g} \approx_\varepsilon T_m(D^{f,g}).$$

□

Theorem 9. *If* $(\mathsf{Enc}, \mathsf{Dec})$ *is an* $\varepsilon-$*-non-malleable code according to the definition from [DPW10], then it is* $4\varepsilon-$*non-malleable code.*

Proof. Using the notation from Theorem 8, we know that, irrespective of the choice of $D^{f,g}_{x,y}$ distributions, the following is true:

$$T_m(\mathsf{Tamp}^{f,g}_m) = \mathsf{DPWTamp}^{f,g}_m.$$

Now let $D^{f,g}_{x,y}$ as follows:

$$\Pr(D^{f,g}_{x,y} = 0) = \min\left\{\frac{\Pr(D^{f,g} = \mathsf{same})}{\Pr(\mathsf{DPWTamp}^{f,g}_{\mathsf{Dec}(x,y)} = \mathsf{Dec}(x,y))}, 1\right\}$$

if $\Pr(\mathsf{DPWTamp}^{f,g}_{\mathsf{Dec}(x,y)} = \mathsf{Dec}(x,y)) \neq 0$. Otherwise let $\Pr(D^{f,g}_{x,y} = 0) = 0$.

Notice that now

$$|\Pr(\mathsf{Tamp}^{f,g}_m = \mathsf{same}) - \Pr(D^{f,g} = \mathsf{same})| < \varepsilon\ .$$

By DPW-non-malleable codes definition we get

$$T_m(\mathsf{Tamp}^{f,g}_m) \approx_\varepsilon T_m(D^{f,g})$$

thus

$$\mathsf{Tamp}^{f,g}_m \approx_{2\varepsilon} D^{f,g},$$

and thus that for any m_0, m_1 we get

$$\mathsf{Tamp}^{f,g}_{m_0} \approx_{4\varepsilon} \mathsf{Tamp}^{f,g}_{m_1}.$$

□