A Zero-One Law for Secure Multi-Party Computation with Ternary Outputs

Gunnar Kreitz

KTH – Royal Institute of Technology
gkreitz@kth.se

Abstract. There are protocols to privately evaluate any function in the passive (honest-but-curious) setting assuming that the honest nodes are in majority. For some specific functions, protocols are known which remain secure even without an honest majority. The seminal work by Chor and Kushilevitz [7] gave a complete characterization of Boolean functions, showing that each Boolean function either requires an honest majority, or is such that it can be privately evaluated regardless of the number of colluding nodes.
The problem of discovering the threshold for secure evaluation of more general functions remains an open problem. Towards a resolution, we provide a complete characterization of the security threshold for functions with three different outputs. Surprisingly, the zero-one law for Boolean functions extends to $\mathbb{Z}_3$, meaning that each function with range $\mathbb{Z}_3$ either requires honest majority or tolerates up to n colluding nodes.

1 Introduction

Multi-party secure function evaluation (SFE) is a cornerstone of modern cryptography, and has been extensively studied since it was introduced by Yao [15]. In this work we consider the joint evaluation by n parties of a public n-ary function f in such a way that no collusion of parties learns anything more than what they do by knowing their own inputs and seeing the output. We consider the *symmetric* case where all participants receive the same output.

Several models of adversaries occur in the SFE literature. A first distinction is whether the adversary has limited computational power (*computational security*) or not (*information-theoretic security*). A second important distinction is whether the parties corrupted by the adversary must still follow the protocol (*passive*) or not (*active*). In the present work, we are concerned with information-theoretic security and all adversaries considered are passive. We assume that the parties communicate over a complete network with *private channels*, meaning that the adversary cannot see messages sent between two honest parties.

Another important limitation put upon the adversary is which parties she can corrupt. The most common adversary is allowed to corrupt up to a threshold $t \leq n$ participants for some t which is typically a function of n. We say that a function for which there is a protocol tolerating up to t corruptions is t-private. In this paper, we will only consider threshold adversaries. More general adversarial

models have also been studied, both in terms of a more general specification of the parties the adversary can corrupt by Hirt and Maurer [10] and considering a mix active and passive adversarial corruptions by Beerliová-Trubíniová *et al.* [2].

There exist protocols to securely evaluate any function $\lfloor (n-1)/2 \rfloor$-privately in our setting by Ben-Or, Goldwasser, and Wigderson [3], and Chaum, Crépeau, and Damgård [4]. For some functions, in particular Boolean disjunction, this has been proved to be an upper bound meaning that there are no protocols to evaluate them which remain secure against more than $\lfloor (n-1)/2 \rfloor$ colluding parties. For other functions, in particular summation over a finite Abelian group, there are n-private protocols. This raises the question of determining the privacy threshold of functions.

Chor and Kushilevitz [7] completely answered the question for Boolean functions. They proved a zero-one law showing that each Boolean function is either $\lfloor (n-1)/2 \rfloor$-private (and not $\lceil n/2 \rceil$-private) or n-private. Their work presents a proof that a function containing an OR-like substructure (an *embedded* OR) is $\lfloor (n-1)/2 \rfloor$-private and that all Boolean functions without such a substructure can be computed by a single Boolean summation.

Proving that a function f cannot be t-privately computed is often done by a partition argument, reducing to the two-party case. In these proofs, the parties are partitioned into two parts of size $\leq t$ and we think of f as a two-party function with each party supplying all inputs for one set of the partition. If the two-party function is not 1-private, then f is not t-private. Chor and Ishai [6] analyzed partition arguments and gave a generalization partitioning the parties into $k > 2$ sets which increases the power of the framework. However, in this paper, we will only need partitioning arguments with two sets.

Chor, Geréb-Graus, and Kushilevitz [5] showed that for every t, $\lceil n/2 \rceil \leq t \leq n-2$ there exists a function such that it is t-private but not $(t+1)$-private. We remark that the functions they construct in their proofs have very large ranges which grow exponentially with t.

The privacy of symmetric[1] functions with Boolean arguments has been studied by Chor and Shani [9]. For such functions, they prove a necessary condition on the preimages of outputs for the function to be $\lceil n/2 \rceil$-private. They also define a class called dense symmetric functions where this necessary condition is also sufficient for n-privacy. Thus, they also prove a zero-one law where for a class of functions, where each function in the class is either n-private or not $\lceil n/2 \rceil$-private.

For two-party computation, a complete characterization of the 1-private functions was made independently by Beaver [1] and Kushilevitz [14]. They both show that a function f is 1-private if and only if it is decomposable, and for decomposable functions, there is a straightforward 1-private protocol. One of our protocols, Protocol 3, can be viewed as a generalization of the protocol for decomposable functions to the multi-party case.

[1] Here, symmetric means the standard notion of a symmetric function, not the SFE-specific notion that all parties receive the same output.

Künzler, Müller-Quade, and Raub [13] give a combinatorial classification of functions computable in several different adversarial models, including the information-theoretic passive model which we work with in this paper. However, in this setting, they consider the broadcast model of communication which gives different results from private channels. For instance, summation is not n-private in the broadcast channel model.

1.1 Our Contribution

In this work, we extend the zero-one law of Boolean privacy to functions with three outputs. For notational convenience, we talk about functions with range $\mathbb{Z}_3$, but we would like to emphasize our results do not depend on any algebraic structure over the range of the function. More formally, we prove the following statement:

Theorem 1 (Main theorem). *For every n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$, f is either n-private, or it is $\lfloor (n-1)/2 \rfloor$-private and not $\lceil n/2 \rceil$-private.*

The core part of our proof is a structure lemma (Lemma 8) showing that every function f with range $\mathbb{Z}_3$ must have at least one of three properties (which we define more formally later):

- f has an embedded OR
- f is a permuted sum
- f is collapsible.

We provide protocols for n-privately evaluating those functions of the two latter types which do not contain an embedded OR.

Our definition of an embedded OR is a generalization of the one commonly found in the literature, but the presence of one implies that there is no protocol which can securely evaluate f and tolerate more than t colluding parties for some t (but potentially for a $t > \lceil n/2 \rceil$).

Finally, we prove (Theorem 22) that the existence of an embedded OR (in our generalized sense) also implies the existence of a "small" embedded OR, giving $t = \lceil n/2 \rceil$. By combining this result with our structure lemma and the result from [7] that a function with an embedded OR of size at most $\lceil n/2 \rceil$ cannot be $\lceil n/2 \rceil$-privately computed, our main theorem follows. We state the proof more formally in Section 6.

We remark that while our statements are true for $n = 2$, there are complete classifications [1,14] for the 2-party case which are simpler than ours (for $n = 2$, our protocols reduce to decomposition) and not limited to functions with range $\mathbb{Z}_3$. Our contribution lies in the case when $n \geq 3$.

The proof of our theorems are significantly more involved than the analogous proofs for Boolean functions. In several of our proofs we need to apply a fairly extensive case analysis.

Our result answers in part a question raised by Chor and Ishai [6] by showing that partition reductions (with only two sets) are universal for proving non-privacy of functions mapping to $\mathbb{Z}_3$.

2 Notation and Preliminary Theorems

We use boldface letters to refer to vectors, like: $\boldsymbol{x}$, $\boldsymbol{y}$. We work with functions with range $\mathbb{Z}_3$, and use the three Greek letters α, β, and γ to denote the three different outputs of the function. We take as convention that the three represent distinct outputs (so $\alpha \neq \beta \neq \gamma$). Sometimes we need to discuss an output as being not α, which we denote by $\not\alpha$.

In the proceeding discussion, we often need to discuss the behavior of a subfunction when keeping some subset of its arguments fixed. To simplify this discussion, we introduce some notation. For disjoint $S_1, S_2, S_3 \subseteq [n]$ we define

$$\begin{aligned} f^{\boldsymbol{a}}_{\{S_1\}}(\boldsymbol{x}) &\stackrel{\text{def}}{=} f(\{x_i\}_{i\in S_1}, \{a_i\}_{i\in S_1^C}) \\ f^{\boldsymbol{a}}_{\{S_1,S_2\}}(\boldsymbol{x},\boldsymbol{y}) &\stackrel{\text{def}}{=} f(\{x_i\}_{i\in S_1}, \{y_i\}_{i\in S_2}, \{a_i\}_{i\in (S_1\cup S_2)^C}) \\ f^{\boldsymbol{a}}_{\{S_1,S_2,S_3\}}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) &\stackrel{\text{def}}{=} f(\{x_i\}_{i\in S_1}, \{y_i\}_{i\in S_2}, \{z_i\}_{i\in S_3}, \{a_i\}_{i\in (S_1\cup S_2\cup S_3)^C}) \,. \end{aligned}$$

We sometimes consider singleton sets S_1, S_2, S_3 and then denote them simply by their only element, with some abuse of notation. That is,

$$\begin{aligned} f^{\boldsymbol{a}}_{\{i\}}(x) &\stackrel{\text{def}}{=} f(a_1,\ldots,a_{i-1},x,a_{i+1},\ldots,a_n) \\ f^{\boldsymbol{a}}_{\{i,j\}}(x,y) &\stackrel{\text{def}}{=} f(a_1,\ldots,a_{i-1},x,a_{i+1},\ldots,a_{j-1},y,a_{j+1},\ldots,a_n) \,, \end{aligned}$$

and analogously for $f^{\boldsymbol{a}}_{\{i,j,k\}}(x,y,z)$ and $f^{\boldsymbol{a}}_{\{i,j,k,l\}}(x,y,z,w)$.

We need to describe details of functions' behaviors, and adopt a geometric viewpoint. In the proofs, we speak of inputs as being neighbors and of rows, diagonals, and rectangles and induced rectangles in the function table. By neighbors we mean points at Hamming distance 1. By a row, we mean the values taken by the function fixing all but one values, i.e. the values $f^{\boldsymbol{a}}_{\{i\}}(x)$ for all $x \in A_1$ with a fixed i and $\boldsymbol{a}$ which are clear from the context. By a rectangle, we mean the values $f^{\boldsymbol{e}}_{\{S_1,S_2\}}(\boldsymbol{a},\boldsymbol{c}), f^{\boldsymbol{e}}_{\{S_1,S_2\}}(\boldsymbol{a},\boldsymbol{d}), f^{\boldsymbol{e}}_{\{S_1,S_2\}}(\boldsymbol{b},\boldsymbol{c}), f^{\boldsymbol{e}}_{\{S_1,S_2\}}(\boldsymbol{b},\boldsymbol{d})$. Note that a rectangle by this definition is a high-dimensional structure. By induced rectangle, we mean a rectangle as before but where $|S_1| = |S_2| = 1$, thus looking like a rectangle in the function table. We only use the concept of a diagonal of a 2×2 induced rectangle. For fixed inputs $\boldsymbol{a}$ and dimensions i, j we say that $f^{\boldsymbol{a}}_{\{i,j\}}(x_1,y_1), f^{\boldsymbol{a}}_{\{i,j\}}(x_2,y_2)$ is a diagonal for $x_1 \neq x_2$ and $y_1 \neq y_2$.

Definition 1 (Redundant inputs). *For an n-argument function f, we say that inputs $x, y, x \neq y$ are* redundant for player k *if for all $\boldsymbol{a}$ it holds that $f^{\boldsymbol{a}}_{\{k\}}(x) = f^{\boldsymbol{a}}_{\{k\}}(y)$.*

Definition 2 (Normalized function). *An n-argument function f with no redundant inputs for any player is said to be* normalized.

We take as convention that all functions are normalized. This assumption is without loss of generality as a function can easily be normalized by for each set of

redundant inputs removing all but one. A protocol for evaluating the normalized function can be used to evaluate the original function as well by performing the same procedure.

To prove Theorem 1, we make use of a theorem by Chor and Kushilevitz [7] which states that there is no 1-private protocol for a 2-party computation of disjunction. Through standard simulation techniques, this gives impossibility results for multi-party protocols of functions containing an OR-like substructure. This is commonly referred to as an embedded OR, or a corner. We formally define an embedded OR and then restate their result. For a two-party function, the definition is straightforward:

Definition 3 (Embedded OR (2 parties)). *We say that a two-argument function f contains an* embedded OR *if there exists inputs x_1, x_2, y_1, y_2 ($x_1 \neq x_2, y_1 \neq y_2$) such that $f(x_1, y_1) = f(x_1, y_2) = f(x_2, y_1) \neq f(x_2, y_2)$.*

However, when considering the n-party case, the definition of an embedded OR becomes slightly more complex. In particular, we need our definition to capture the size of the collusion required to realize an embedded OR, as that size also limits the impossibility result that follows from the existence of such an embedded OR. To this end, we define an embedded OR as having a degree k. We remark that Kilian *et al.* [11] define an embedded OR as one of degree 1. Much of the previous literature has mostly been concerned with Boolean functions, and then, the existence of an embedded OR (of any degree) implies the existence of one of degree 1, as proved in [11]. However, for functions with larger ranges, the situation is more complex, as shown by our Theorem 22.

Definition 4 (Embedded OR (n parties, induced, generalized), corner-free). *We say that an n-argument function f contains an* embedded OR *of degree k if there exists disjoint subsets $S_1, S_2 \subset [n]$ where $|S_1|, |S_2| \leq k$, and values $\boldsymbol{a}$ such that the two-argument function $f'(\boldsymbol{x}, \boldsymbol{y}) = f^{\boldsymbol{a}}_{\{S_1,S_2\}}(\boldsymbol{x}, \boldsymbol{y})$ contains an embedded* OR*. We refer to an embedded* OR *of degree 1 as an* induced *embedded* OR*, and one of degree greater than 1 as a* generalized *embedded* OR*. A function without an embedded* OR *(of any degree) is said to be* corner-free*.*

With the definitions in place, we are ready to restate a result by Chor and Kushilevitz [7]. The result we need was not presented as a separate lemma in their paper, but instead follows as a corollary from two of their lemmas which we restate in simplified form.

Lemma 2 (Partition lemma, [7]). *Let $f : A_1 \times \ldots \times A_n \to R$ be $\lceil n/2 \rceil$-private. Then for every subset S_1 of size $\lceil n/2 \rceil$, the two-argument function $f'(\boldsymbol{x}, \boldsymbol{y}) = f_{\{S_1,S_1^C\}}(\boldsymbol{x}, \boldsymbol{y})$ is 1-private.*

Lemma 3 (Corners lemma, [7]). *A two-argument function is not 1-private if it contains an embedded* OR*.*

Corollary 4. *A function containing an embedded* OR *of degree at most $\lceil n/2 \rceil$ is not $\lceil n/2 \rceil$-private.*

We also make use of [7, Theorem 4] which states that a corner-free Boolean function can be expressed as a Boolean sum:

Theorem 5 ([7]). *For a corner-free Boolean function f there are functions f_i such that $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i)$ where the sum is computed modulo 2.*

We formally restate the theorem from [11] showing that a generalized embedded OR in a Boolean function implies an induced embedded OR. In our terminology:

Theorem 6 ([11]). *A Boolean function f containing an embedded* OR *contains an embedded* OR *of degree 1.*

We show functions and subfunctions which depend on up to 4 arguments. To be able to draw them, we show 2-dimensional projections separated by lines with vertical lines indicating a 3^{rd} dimension and horizontal lines indicating a 4^{th} dimension. We present sample function in Figure 1, showing a function which contains an embedded OR of degree 2 but does not contain an embedded OR of degree 1. The highlighted embedded OR occurs with the subsets $S_1 = \{P_1, P_3\}$ and $S_2 = \{P_2, P_4\}$ with inputs $(2, 1)$ and $(1, 2)$ for S_1 and $(1, 1)$ and $(2, 2)$ for S_2. As the function is drawn, the coalition S_1 in the embedded OR controls the horizontal position, and S_2 controls the vertical position.

0	**1**	**1**	2
1	0	2	1
2	0	0	1
0	**2**	**1**	0

Fig. 1. An example function containing an embedded OR of degree 2 (highlighted).

We use the following lemma which we believe is well-known. A proof is included in the full version of this paper [12].

Lemma 7. *If an n-argument function $f : A_1 \times \ldots \times A_n \to G$, where G is an Abelian group, has the property that for every pair of dimensions j, k and inputs $x_1, x_2, y_1, y_2, \boldsymbol{a}$ the following equality holds:*

$$f^{\boldsymbol{a}}_{\{j,k\}}(x_1, y_1) + f^{\boldsymbol{a}}_{\{j,k\}}(x_2, y_2) = f^{\boldsymbol{a}}_{\{j,k\}}(x_1, y_2) + f^{\boldsymbol{a}}_{\{j,k\}}(x_2, y_1), \tag{1}$$

then f can be rewritten as $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i)$.

3 A Structure Lemma

The main step towards proving Theorem 1 is the establishment of a structure lemma for functions with range $\mathbb{Z}_3$. Thus, we turn toward some global properties of functions (as opposed to the comparatively local property of the existence

of an embedded OR). The first such property captures the case when we can split the range of a function into two parts, and compute a Boolean sum to discover which part the output lies in. If we can then proceed with further such subdivisions until we arrive at a single possible output, this immediately gives a protocol to compute f. We prove that this further subdivision is always possible for corner-free f with range $\mathbb{Z}_3$ in Lemma 21. We remark that this is a further generalization of the multi-party decomposability defined in [13], which in turn was a generalization of 2-party decomposability defined in [14]. We show a collapsible function and the generalized decomposition of it in Figure 2.

Definition 5 (Collapsible). *We say that a function $f : A_1 \times \ldots A_n \to R$ is* collapsible *if there is a subset $R', \emptyset \subset R' \subset R$ such that the Boolean function*

$$f'(\boldsymbol{x}) = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) \in R' \\ 0 & \text{otherwise} \end{cases}$$

does not contain an embedded OR *and can thus be n-privately computed. We refer to f' as being collapsed.*

For a collapsible function f with range $\mathbb{Z}_3$ if f is collapsible we can choose R' with two elements α, β and say that f is collapsible by collapsing α and β.

0	1	2	2	2	0	1	1	0	0	0	1
1	0	2	2	2	1	1	1	0	0	0	1
2	2	0	1	0	2	0	0	1	1	1	0

(a) Collapsible f (b) f collapsed

Fig. 2. An example collapsible function and the collapsed function.

Summation in a finite Abelian group is a function which is known to be n-private [8]. In a summation, the effect of one party's input can be thought of as applying a permutation to the sum of the other parties' inputs. We generalize this by defining a permuted sum where we give one of the parties a special role and let her input select an arbitrary permutation to be applied to the sum of the other parties' inputs. All functions which are sums, i.e. can be rewritten as $\sum_{i=1}^{n} f_i(x_i)$, are also permuted sums. In our applications, the sum may be a Boolean sum or over $\mathbb{Z}_3$. We show two example functions which are permuted sums in Figure 3

Definition 6 (Permuted sum). *We say that a function is a* permuted sum *if it can be written as $\pi_{x_i}(\sum_{j \neq i} f_j(x_j))$ where π_x is a permutation. We refer to party i as the* permuter.

With these definitions, we are now ready to state and prove our structure lemma:

0 0 1 1 2 2		0 1 2 | 0 2 1 | 1 0 2
1 2 0 2 0 1		1 2 0 | 2 1 0 | 0 2 1
(a) f		2 0 1 | 1 0 2 | 2 1 0
		(b) g

Fig. 3. Two example permuted sums. In f, party 2 (selecting column) is the permuter selecting one of the 6 permutations. The function $g = \pi_{x_3}(x_1 + x_2)$ where π_1 is the identity permutation, $\pi_2 = (12)$ and $\pi_3 = (01)$.

Lemma 8 (Structure lemma). *For every normalized n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$, at least one of the following holds:*

- *f has an embedded* OR
- *f is a permuted sum*
- *f is collapsible*

We present protocols for n-privately evaluating permuted sums (Protocol 2) and collapsible functions (Protocol 3) which do not contain an embedded OR. In Theorem 22 we show that if f contains an embedded OR, it also contains a small embedded OR. This, together with Corollary 4 concludes the proof of our Theorem 1.

To prove the structure lemma, we perform a case-analysis based on a property of f we call a link:

Definition 7 (Link, link-free). *We say that an n-argument function has a* link *(*over output α*) in dimension k if there exists inputs $x, y, x \neq y$, and $\boldsymbol{a}$ such that $\alpha = f^{\boldsymbol{a}}_{\{k\}}(x) = f^{\boldsymbol{a}}_{\{k\}}(y)$. We say that f has* links in c dimensions *if there are precisely c distinct k such that f has a link in dimension k. We say that a function is* link-free *if it has no links.*

Lemma 9. *In a corner-free n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$, if there are links between inputs x and y in dimension k over two distinct outputs, then x and y are redundant for player k.*

Proof. Let f have links over α and β between inputs x and y in dimension k. That is, there exists values $\boldsymbol{a}$, $\boldsymbol{b}$ such that $f^{\boldsymbol{a}}_{\{k\}}(x) = f^{\boldsymbol{a}}_{\{k\}}(y) = \alpha$, and $f^{\boldsymbol{b}}_{\{k\}}(x) = f^{\boldsymbol{b}}_{\{k\}}(y) = \beta$. Suppose that for some $\boldsymbol{c}$ we have $f^{\boldsymbol{c}}_{\{k\}}(x) \neq f^{\boldsymbol{c}}_{\{k\}}(y)$. Then one of $f^{\boldsymbol{c}}_{\{k\}}(x)$ and $f^{\boldsymbol{c}}_{\{k\}}(y)$ equals α or β. If one of them is α then f has an embedded OR with $S_1 = \{k\}$, $S_2 = \{k\}^C$ using inputs (x, y) and $(\boldsymbol{a}, \boldsymbol{c})$. If one is β then f has an embedded OR with $S_1 = \{k\}$, $S_2 = \{k\}^C$ using inputs (x, y) and $(\boldsymbol{b}, \boldsymbol{c})$. □

Looking at the proof of Lemma 9 we begin to see the importance of the small range of the function to the analysis. It also highlights the added complexities compared to the Boolean case, as for a Boolean function any link implies that two inputs are redundant. From the lemma and its proof follow two corollaries about normalized functions with range $\mathbb{Z}_3$:

Corollary 10. *For a normalized n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with a link over α in dimension k for inputs x, y, for all $\boldsymbol{a}$, we have that $f^{\boldsymbol{a}}_{\{k\}}(x)$ uniquely determines $f^{\boldsymbol{a}}_{\{k\}}(y)$. More specifically, the possible combinations of values are $(\alpha, \alpha); (\beta, \gamma); (\gamma, \beta)$.*

Proof. Follows from the proof of Lemma 9. □

Corollary 11. *A normalized n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ cannot have links over α in dimension k for inputs x, y, and x, z.*

Proof. By Corollary 10 the value at x determines the value at both y and z and hence inputs y and z are redundant. □

Analogously to an embedded OR, we introduce notation for the various 2×2 substructures in a function. Apart from the embedded OR, two of them feature prominently in our proofs. Firstly, a 2×2 substructure with one output occurring on the diagonal, and the two other values occurring once each on the opposite diagonal is called Aff_3. Secondly, a 2×2 substructure where one output is on one diagonal, and another is on the other is referred to as an XOR. For the XOR, we also define the type of an XOR as the pair (without order) of outputs in the XOR. All the substructures which can occur (up to symmetries) are depicted in Figure 4. A 2×2 substructure where only one output occurs is called constant, and if we want to emphasize that it is the output α which occurs, we write (α)-constant.

$\alpha\ \alpha$	$\alpha\ \alpha$	$\alpha\ \alpha$	$\alpha\ \alpha$	$\alpha\ \beta$	$\alpha\ \beta$
$\alpha\ \alpha$	$\alpha\ \beta$	$\beta\ \beta$	$\beta\ \gamma$	$\beta\ \alpha$	$\gamma\ \alpha$
(a) Constant	(b) OR	(c) 2-link	(d) Link	(e) XOR	(f) Aff_3

Fig. 4. The six 2×2 substructures.

Definition 8 (Type of an XOR). *If an XOR consists of outputs α and β we say that it is an XOR of type (α, β), denoted (α, β)-XOR. The order of elements is not important, so for functions to $\mathbb{Z}_3$ there are three possible types of XOR: (α, β), (α, γ), (β, γ).*

Our name Aff_3 comes from the fact that it can be expressed as an affine function modulo 3, analogously to the fact that XOR can be expressed as a sum modulo 2. We do not need that it is affine, but we make use of the fact that a function where all subfunctions are of the form Aff_3 can be written as a sum on the form $\sum_{i=1}^{n} f_i(x_i)$ with summation in $\mathbb{Z}_3$.

Lemma 12. *An n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ such that all 2×2 subfunctions are of the form Aff_3 can be expressed as $\sum_{i=1}^{n} f_i(x_i)$ with summation in $\mathbb{Z}_3$.*

Proof. By Lemma 7 we need to verify that (1) holds for all 2×2 subfunctions, which are all of the form Aff_3. For all ways of assigning 0, 1 and 2 (distinctly) to α, β, γ we have that $2\alpha \equiv \beta + \gamma \pmod 3$. As $2 \equiv -1 \pmod 3$ this is equivalent to $\alpha + \beta + \gamma \equiv 0 \pmod 3$. □

In our proof of Lemma 8 we consider the substructures occurring in f. We begin by establishing three preliminary lemmas. The lemmas come into play primarily in cases when f contains links in few dimensions (none or one), and if f has an XOR spanned by dimensions i, j and f is link-free in those dimensions, then $|A_i| = |A_j| = 2$, giving some intuition for the condition on the size of the two inputs in the lemmas. We highlight the proof idea for each of the lemmas and give full proofs in the appendix.

Lemma 13. *Let f be an n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with i, j such that $|A_i| = |A_j| = 2$ such that for all $\boldsymbol{a}$, $f^{\boldsymbol{a}}_{\{i,j\}}$ is an XOR. If all three types of XOR's occur then there is a dimension k such that the input in dimension k determines the type of XOR.*

Proof (Idea). We show that if no such k exists then f contains an embedded OR. Full proof in Section A.1. □

Lemma 14. *Let f be an n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with i, j such that $|A_i| = |A_j| = 2$ and an output α such that for all $\boldsymbol{a}$ precisely one diagonal of $f^{\boldsymbol{a}}_{\{i,j\}}$ has two α's. Then f is collapsible.*

Proof (Idea). If f is not collapsible then the collapsed function contains an embedded OR. We show that this implies an embedded OR in f as well. Full proof in Section A.2. □

Lemma 15. *An n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with i, j such that $|A_i| = |A_j| = 2$ and such that for some $\boldsymbol{a}$, $f^{\boldsymbol{a}}_{\{i,j\}}$ is an Aff_3 and for some $\boldsymbol{b}$, $f^{\boldsymbol{b}}_{\{i,j\}}$ is an XOR is collapsible.*

Proof (Idea). We prove that f fulfills the conditions of Lemma 14. Full proof in Section A.3. □

Our proof of Lemma 8 proceeds in three separate lemmas, depending on whether the function f is link-free (Lemma 16), has links in one dimension (Lemma 17), or if it has links in two or more dimensions (Lemma 18). As the proofs are long and consist mainly of case analysis, we simply state the lemmas here. The proof of the first lemma is given in the appendix, and the two others in the full version of this paper [12].

Lemma 16. *Every n-argument link-free, corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ is collapsible or a permuted sum.*

Proof (Idea). Case analysis showing we can apply one of Lemma 13, Lemma 14 and Lemma 15. Full proof in Section A.4 □

Lemma 17. *Every n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with links in 1 dimension and without an embedded* OR *is collapsible or a permuted sum.*

Proof (Idea). Case analysis showing we can apply one of Lemma 13, Lemma 14 and Lemma 15. Proof in the full version of this paper [12]. □

Lemma 18. *Every n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with links in 2 or more dimensions and without an embedded* OR *is collapsible.*

Proof (Idea). We show that all links must be over the same output. This gives some implications for the substructures of f which we use to show f must be collapsible. Proof in the full version of this paper [12]. □

4 Protocols

With the structure lemma established, we can now turn to the question of n-private protocols for collapsible functions and permuted sums. From the definitions of the two classes, we have two natural and easy protocols. The main problem we need to address in this section is proving the existence of a protocol for collapsible functions. For a function which is collapsible by collapsing β and γ it is clear from the definition that we can n-privately evaluate if the output is α or if it is one of β and γ. The key issue is to prove that we can then proceed with a second step where we can n-privately evaluate whether the output is β or if it is γ.

The construction of this second step relies on the passive model of adversaries and the knowledge that the output of the function is not α. Thus, in our second step we compute a sum which may have different outputs at points where the original function had α's. Such a construction is inherently insecure with active adversaries, as they may switch inputs between the first step of the decomposition and the second and would then learn some information about the other parties' inputs.

In both of our protocols we use a subprotocol by Chor and Kushilevitz [8] for n-private summation over any finite Abelian group. For completeness, we include a description of their protocol as Protocol 1. When used in our protocol for a permuted sum, the summation is either Boolean or in $\mathbb{Z}_3$ depending on the function f (but not on the inputs).

Protocol 1 (Summation [8]). The protocol for summation where party P_i participates with input x_i proceeds as follows:

1. In round $1 \leq i \leq n-2$, party P_i sums all its received messages, $w_i = \sum_{j=1}^{i-1} z_{j,i}$. Then, it chooses random group elements $z_{i,i+1}, z_{i,i+2}, \ldots, z_{i,n-1}$. Finally, it computes $z_{i,n}$ such that $x_i + w_i = \sum_{j=i+1}^{n} z_{i,j}$ and sends $z_{i,j}$ to P_j $(j > i)$.
2. In round $n-1$, party P_{n-1} computes $z_{n-1,n} = x_{n-1} + \sum_{j=1}^{n-2} z_{j,n-1}$ and sends $z_{n-1,n}$ to P_n.

3. In round n, party P_n computes the sum s as $s = x_n + \sum_{j=1}^{n-1} z_{j,n}$.

All sums are computed over some fixed finite Abelian group.

Protocol 2 (Permuted sum). The protocol for evaluating a permuted sum f, where party P_i (without loss of generality we assume the permuter is party n) participates with input x_i proceeds as follows:

1. Use Protocol 1 to privately compute $s = \sum_{j=1}^{n-1} f_j(x_j)$ such that only the permuter learns s.
2. The permuter computes the output as $\pi_{x_n}(s)$ and sends it to the other parties.

The sum is computed modulo 2, or 3, depending on f.

Protocol 3 (Collapsible). The protocol for evaluating a function f collapsible with partition $R' = \{\gamma\}$, where party P_i participates with input x_i proceeds as follows:

1. Use Protocol 1 to compute $s = \sum_{i=1}^{n} f_i(x_i) \pmod 2$, with f_i such that $s = 1$ iff $f(\boldsymbol{x}) = \gamma$
2. If $s = 0$, compute $s' = \sum_{i=1}^{n} g_i(x_i) \pmod 4$, with g_i such that $f(\boldsymbol{x}) = \alpha$ implies $s' = 0$, and $f(\boldsymbol{x}) = \beta$ implies $s' = 2$.

The correctness of Protocol 2 follows immediately from the definition of a permuted sum. In Protocol 3, since f is collapsible, the functions f_i exist by the definition of a collapsible function. However, the existence of appropriate g_i is not as straightforward. We prove, constructively, in Lemma 21 that they always exist for corner-free collapsible functions with range $\mathbb{Z}_3$. We stress that the choice of g_i does not depend on the input $\boldsymbol{x}$, but only on the function f.

The privacy of both these protocols is straightforward, and we only sketch the arguments.

Theorem 19. *Protocol 2 is n-private.*

Proof. The subprotocol used for summation was proven to be n-private in [8]. Due to the structure of the function, we see that the permuter, P_n, learns the sum s from $f(\boldsymbol{x})$ and x_n, since $s = \pi_{x_n}^{-1}(f(\boldsymbol{x}))$. □

Theorem 20. *Protocol 3 is n-private.*

Proof. The subprotocol used for summation was proven to be n-private in [8]. When the output is γ then, by the privacy of the summation sub-protocol, the protocol is private. Furthermore, when the output is one of α, β, then the privacy of the composed protocol also follow directly from the privacy of the subprotocols. The first sum only reveals that the output is one of α, β, and then, the condition on g_i is sufficient to guarantee that the sum s' reveals nothing but whether the output is α or β, as with a passive adversary we are guaranteed that s' is either 0 or 2. □

While the privacy is straightforward, the proof that there are functions g_i as required by Protocol 3 is rather involved and we simply state the lemma here and give the proof in [12]. One may intuitively expect that such functions could simply be Boolean, but it turns out that for some f we do need the full range of $\mathbb{Z}_4$.

Lemma 21. *Protocol 3 can evaluate all corner-free, collapsible functions with range $\mathbb{Z}_3$.*

Proof (Idea). We construct a function g such that $f(\boldsymbol{x}) = \alpha \implies g(\boldsymbol{x}) = 0$ and $f(\boldsymbol{x}) = \beta \implies g(\boldsymbol{x}) = 2$. By case analysis on the induced rectangles in g, we show that g satisfies the conditions of Lemma 7 and hence there are g_i as required by Protocol 3. Proof in the full version of this paper [12]. □

5 An Embedded OR Implies a Small Embedded OR

Previously, we have often assumed that functions are free of embedded OR's of *any* degree (i.e., that they are corner-free). However, to be able to apply Corollary 4 we need to show that a sufficiently small embedded OR exists.

For Boolean functions f, if f has an embedded OR of any degree, then it also has an embedded OR of degree 1, as proved in [11], explaining the zero-one nature of Boolean privacy.

It turns out that for functions with range $\mathbb{Z}_3$, similarly to the Boolean case, the presence of a large embedded OR implies that the function also contains a small one. We state the theorem here and give the proof in [12].

Theorem 22. *Every n-argument function $f : A_1 \times \ldots A_n \to \mathbb{Z}_3$ that has an embedded* OR *of any degree has an embedded* OR *of degree at most* 3*. Furthermore, every* 4*-argument function $f : A_1 \times A_2 \times A_3 \times A_4 \to \mathbb{Z}_3$ that has an embedded* OR*, also has one of degree at most* 2*.*

Proof (Idea). The basic idea is similar to that used in the proof of Theorem 6. However, while the boolean case is fairly straightforward, our proof results in a fairly extensive case analysis. Proof in the full version of this paper [12]. □

6 Proof of the Main Theorem

We now conclude by re-stating our main theorem and presenting the proof.

Theorem 1 (Main theorem). *For every n-argument function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$, f is either n-private, or it is $\lfloor (n-1)/2 \rfloor$-private and not $\lceil n/2 \rceil$-private.*

Proof. If f is corner-free, then by Lemma 8 it is a permuted sum, collapsible, or both. Thus, it can be n-privately computed by Protocol 2 or Protocol 3.

If f is not corner-free, then by Theorem 22 it contains an embedded OR of degree at most $\lceil n/2 \rceil$. Thus, by Corollary 4, f is not $\lceil n/2 \rceil$-private. □

Acknowledgements

I would like to thank Johan Håstad for his many helpful insights, Dominik Raub for introducing me to this problem and for the interesting discussions, and Per Austrin for Protocol 3.

References

1. Donald Beaver. Perfect privacy for two-party protocols. In J. Feigenbaum and M. Merritt, editors, *Proceedings of DIMACS Workshop on Distributed Computing and Cryptology*, volume 2, pages 65–77. American Mathematical Society, 1989.
2. Zuzana Beerliová-Trubíniová, Matthias Fitzi, Martin Hirt, Ueli M. Maurer, and Vassilis Zikas. MPC vs. SFE: Perfect security in a unified corruption model. In Ran Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 231–250. Springer, 2008.
3. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10. ACM, 1988.
4. David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, pages 11–19. ACM, 1988.
5. Benny Chor, Mihály Geréb-Graus, and Eyal Kushilevitz. On the structure of the privacy hierarchy. *J. Cryptology*, 7(1):53–60, 1994.
6. Benny Chor and Yuval Ishai. On privacy and partition arguments. *Inf. Comput.*, 167(1):2–9, 2001.
7. Benny Chor and Eyal Kushilevitz. A zero-one law for boolean privacy. *SIAM J. Discrete Math.*, 4(1):36–47, 1991.
8. Benny Chor and Eyal Kushilevitz. A communication-privacy tradeoff for modular addition. *Inf. Process. Lett.*, 45(4):205–210, 1993.
9. Benny Chor and Netta Shani. The privacy of dense symmetric functions. *Computational Complexity*, 5(1):43–59, 1995.
10. Martin Hirt and Ueli M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *J. Cryptology*, 13(1):31–60, 2000.
11. Joe Kilian, Eyal Kushilevitz, Silvio Micali, and Rafail Ostrovsky. Reducibility and completeness in private computations. *SIAM J. Comput.*, 29(4):1189–1208, 2000.
12. Gunnar Kreitz. A zero-one law for secure multi-party computation with ternary outputs (full version), 2011. `http://www.csc.kth.se/~gkreitz/mpc_zero_one_three`.
13. Robin Künzler, Jörn Müller-Quade, and Dominik Raub. Secure computability of functions in the IT setting with dishonest majority and applications to long-term security. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 238–255. Springer, 2009.
14. Eyal Kushilevitz. Privacy and communication complexity. *SIAM J. Discrete Math.*, 5(2):273–284, 1992.
15. Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164. IEEE, 1982.

A Proofs

A.1 Proof of Lemma 13

Lemma 13. *Let f be an n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with i, j such that $|A_i| = |A_j| = 2$ such that for all $\boldsymbol{a}$, $f^{\boldsymbol{a}}_{\{i,j\}}$ is an* XOR*. If all three types of* XOR*'s occur then there is a dimension k such that the input in dimension k determines the type of* XOR*.*

Proof. We assume that there is an (α, β)-XOR and an (α, γ)-XOR at Hamming distance 1. We denote the dimension by which they differ by k and relabel the inputs in dimension k such that $f^{\boldsymbol{a}}_{\{i,j,k\}}(\cdot, \cdot, 1)$ is an (α, β)-XOR and $f^{\boldsymbol{a}}_{\{i,j,k\}}(\cdot, \cdot, 2)$ is an (α, γ)-XOR.

We proceed to show that there is no $\boldsymbol{b}$ such that $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 1)$ or $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 2)$ is a (β, γ)-XOR. Assume to the contrary that there is a $\boldsymbol{b}$ such that $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 1)$ is a (β, γ)-OR (the case if it occurs at input 2 in dimension k is analogous). We illustrate this case in Figure 5 where we for simplicity show $\boldsymbol{b}$ as differing from $\boldsymbol{a}$ in only one dimension, which is not something we assume in the proof.

$\alpha\ \beta$	$\alpha\ \gamma$
$\beta\ \alpha$	$\gamma\ \alpha$
$\gamma\ \beta$	
$\beta\ \gamma$	

Fig. 5. Illustration of a contradiction in the proof of Lemma 13.

What values can the function take at $f^{\boldsymbol{b}}_{\{i,j,k\}}(1, 1, 2)$? We claim that any output at that position would violate the assumption that f is corner-free. In Figure 5 we can see why this is true for a simple function (writing α, β or γ anywhere in the missing 2×2 field can be verified to result in an embedded OR). For brevity, we only discuss the case when $f^{\boldsymbol{b}}_{\{i,j,k\}}(1, 1, 2) = \alpha$ here (the other cases are almost identical and the core idea is captured by Figure 5). Proofs of all three cases are given in the full version of this paper [12].

Assume $f^{\boldsymbol{b}}_{\{i,j,k\}}(1, 1, 2) = \alpha$. Then we can find y (equal to 1 or 2) such that $f^{\boldsymbol{a}}_{\{i,j,k\}}(1, y, 2) = \alpha$ as $f^{\boldsymbol{a}}_{\{i,j,k\}}(\cdot, \cdot, 2)$ is an (α, γ)-XOR. We can also find x such that $f^{\boldsymbol{a}}_{\{i,j,k\}}(x, y, 1) = \alpha$ as $f^{\boldsymbol{a}}_{\{i,j,k\}}(\cdot, \cdot, 1)$ is an (α, β)-XOR. However, as $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 1)$ is a (β, γ)-XOR we are guaranteed that $f^{\boldsymbol{b}}_{\{i,j,k\}}(x, 1, 1) \neq \alpha$. Thus, f contains an embedded OR with $S_1 = \{i, k\}$ and $S_2 = S_1^C$ using $(1, 2); (x, 1)$ on S_1 and $y; 1$ or j and $\boldsymbol{a}; \boldsymbol{b}$ on the rest of S_2.

We now conclude that there is no $\boldsymbol{b}$ such that $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 1)$ or $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 2)$ is a (β, γ)-XOR. As f has all types of XOR's, there must still be a (β, γ)-XOR in the function. Thus, we see that $|A_k| \geq 3$, and we can assume there is a $\boldsymbol{b}$ such that $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot, \cdot, 3)$ is a (β, γ)-XOR.

We claim that $f^{\boldsymbol{a}}_{\{i,j,k\}}(\cdot,\cdot,3)$ must be a (β,γ)-XOR. To see this we observe that if it was another type of XOR, then by the same proof that showed that there is no (β,γ)-XOR for $k=1,2$ we could have shown that there was not (β,γ)-XOR for $k=3$, but we know that $f^{\boldsymbol{b}}_{\{i,j,k\}}(\cdot,\cdot,3)$ is a (β,γ)-XOR. We now see that for a given x_k, all XOR's must be of the same type as that at $f^{\boldsymbol{a}}_{\{i,j,x_k\}}(\cdot,\cdot,k)$ which concludes our proof. □

A.2 Proof of Lemma 14

Lemma 14. *Let f be an n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with i,j such that $|A_i| = |A_j| = 2$ and an output α such that for all $\boldsymbol{a}$ precisely one diagonal of $f^{\boldsymbol{a}}_{\{i,j\}}$ has two α's. Then f is collapsible.*

Proof. We claim that f is collapsible by collapsing β and γ. To prove this we show that the collapsed function

$$g(\boldsymbol{x}) = \begin{cases} 1 & \text{if } f(\boldsymbol{x}) \in \{\beta,\gamma\} \\ 0 & \text{if } f(\boldsymbol{x}) = \alpha \end{cases}$$

does not contain an embedded OR of degree 1. Then by Theorem 6 we have that g is corner-free and Theorem 5 implies that the collapsed function can be written as a Boolean sum.

We begin by observing that as each 2×2 plane spanned by dimensions i,j contains exactly one diagonal with α's, each such 2×2 plane contains two α's, as if it had three α's it would be an embedded OR and if it had four α's both diagonals would have two α's. We further make the observation that a pair of neighboring outputs in dimension i (and analogously in j) are such that exactly one of them is α. More formally, for all $\boldsymbol{c}$ if $f^{\boldsymbol{c}}_{\{i\}}(1) = \alpha$ then $f^{\boldsymbol{c}}_{\{i\}}(2) \neq \alpha$ and if $f^{\boldsymbol{c}}_{\{i\}}(1) \neq \alpha$ then $f^{\boldsymbol{c}}_{\{i\}}(2) = \alpha$.

As g is Boolean, by Theorem 6 we know that if g has an embedded OR (of any degree), it also has an embedded OR of degree 1. We assume by contradiction that there is an embedded OR of degree 1 in g. We reorder inputs and dimensions such that the embedded OR is spanned by dimensions $1,2$ using inputs $(1,2);(1,2)$, with other inputs as $\boldsymbol{a}$. We say that $g^{\boldsymbol{a}}_{\{1,2\}}$ is an embedded OR with slight abuse of notation (as $|A_1|$ or $|A_2|$ could be greater than 2). We see that the embedded OR cannot have three 1's as g takes the value 0 where f takes the value α, so an embedded OR with three 0's corresponds to an embedded OR with three α in f, which is corner-free. Thus, the embedded OR must have three 1's.

From our observation we know that each 2×2 plane in g spanned by i,j has two 0's, so there cannot be an OR in g with three 1's spanned by dimensions i,j. Thus, at least one of i and j must be different from both 1 and 2. We assume $i \neq 1,2$ and reorder inputs such that the embedded OR occurs when $x_i = 1$. Let $\boldsymbol{b}$ be $\boldsymbol{a}$ with the value at x_i removed.

We now consider what values occur at $f^{\boldsymbol{b}}_{\{1,2,i\}}(\cdot,\cdot,2)$. We know that of the four outputs of $f^{\boldsymbol{b}}_{\{1,2,i\}}(\cdot,\cdot,1)$ one is α and three are different from α. But by our observation, this implies that of the four outputs of $f^{\boldsymbol{b}}_{\{1,2,i\}}(\cdot,\cdot,2)$ three are α and

one is different from α. This concludes our proof as it shows that an embedded OR in g implies an embedded OR in f which we assumed to be corner-free. □

A.3 Proof of Lemma 15

Lemma 15. *An n-argument corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ with i, j such that $|A_i| = |A_j| = 2$ and such that for some $\boldsymbol{a}$, $f^{\boldsymbol{a}}_{\{i,j\}}$ is an Aff_3 and for some $\boldsymbol{b}$, $f^{\boldsymbol{b}}_{\{i,j\}}$ is an XOR is collapsible.*

Proof. Let the output that appears twice in $f^{\boldsymbol{a}}_{\{i,j\}}$ be α, and reorder inputs such that $f^{\boldsymbol{a}}_{\{i,j\}}(1,1) = f^{\boldsymbol{a}}_{\{i,j\}}(2,2) = \alpha$.

We now claim that $f^{\boldsymbol{b}}_{\{i,j\}}(1,2) = f^{\boldsymbol{b}}_{\{i,j\}}(2,1) = \alpha$. As $f^{\boldsymbol{b}}_{\{i,j\}}$ is an XOR we know that $f^{\boldsymbol{b}}_{\{i,j\}}(1,2) = f^{\boldsymbol{b}}_{\{i,j\}}(2,1)$. If $f^{\boldsymbol{b}}_{\{i,j\}}(1,2) = f^{\boldsymbol{b}}_{\{i,j\}}(2,1) \in \{\beta, \gamma\}$ then there is an embedded OR with $S_1 = \{i, j\}$ and $S_2 = S_1^C$ as using inputs $(1,2); (2,1)$ on S_1 and $\boldsymbol{a}; \boldsymbol{b}$ on S_2. We assume the other diagonal of the XOR consists of β's, i.e. $f^{\boldsymbol{b}}_{\{i,j\}}(1,1) = f^{\boldsymbol{b}}_{\{i,j\}}(2,2) = \beta$, and that $f^{\boldsymbol{a}}_{\{i,j\}}(1,2) = \beta$, $f^{\boldsymbol{a}}_{\{i,j\}}(2,1) = \gamma$. This is without loss of generality as we can relabel outputs and switch the roles of parties 1 and 2.

$\boldsymbol{\alpha}\ \beta$	$\beta\ \boldsymbol{\alpha}$
$\boldsymbol{\gamma}\ \alpha$	$\alpha\ \boldsymbol{\beta}$
(a) $f^{\boldsymbol{a}}_{\{i,j\}}$	(b) $f^{\boldsymbol{b}}_{\{i,j\}}$

Fig. 6. An XOR and Aff_3 in f. The outputs involved in proving that f has no links in dimension i are highlighted.

We claim that the function f cannot have any links in dimensions i or j. To see this for dimension i, we see that $f^{\boldsymbol{a}}_{\{i,j\}}(1,1) = \alpha$ and $f^{\boldsymbol{a}}_{\{i,j\}}(2,1) = \gamma$ but also $f^{\boldsymbol{b}}_{\{i,j\}}(1,2) = \alpha$ and $f^{\boldsymbol{b}}_{\{i,j\}}(2,2) = \beta$. Thus, the value of $f^{\boldsymbol{c}}_{\{i\}}(2)$ is not a function of the value of $f^{\boldsymbol{c}}_{\{i\}}(1)$ for all $\boldsymbol{c}$ and the contrapositive form of Corollary 10 gives that f cannot have a link between inputs 1 and 2 in dimension i. Similarly for dimension j, we have that $f^{\boldsymbol{a}}_{\{i,j\}}(2,2) = \alpha$ and $f^{\boldsymbol{a}}_{\{i,j\}}(2,1) = \gamma$, but also $f^{\boldsymbol{b}}_{\{i,j\}}(1,2) = \alpha$ and $f^{\boldsymbol{b}}_{\{i,j\}}(1,1) = \beta$. This demonstrates that $f^{\boldsymbol{c}}_{\{j\}}(1)$ is not a function of $f^{\boldsymbol{c}}_{\{j\}}(2)$ for all $\boldsymbol{c}$, and by the contrapositive form of Corollary 10, there is no link between inputs 2 and 1 in dimension j.

We proceed by proving that for all $\boldsymbol{c}$ precisely one of the two diagonals of $f^{\boldsymbol{c}}_{\{i,j\}}$ contains two α's. What are the possible values for $(f^{\boldsymbol{c}}_{\{i,j\}}(1,2), f^{\boldsymbol{c}}_{\{i,j\}}(2,1))$? We proved (when $\boldsymbol{c} = \boldsymbol{b}$ but we made no use of any properties of $\boldsymbol{b}$) that they cannot be (β, β) or (γ, γ). Furthermore, as $f^{\boldsymbol{b}}_{\{i,j\}}(1,2) = f^{\boldsymbol{b}}_{\{i,j\}}(2,1) = \alpha$ it cannot be that precisely one of the values is α, as then f would have an embedded OR with $S_1 = \{i, j\}$ and $S_2 = S_1^C$ using inputs $(1,2); (2,1)$ on S_1 and $\boldsymbol{b}; \boldsymbol{c}$ on S_2. Thus the only remaining possibilities are $(\alpha, \alpha); (\beta, \gamma); (\gamma, \beta)$. As f has no links in

dimension i or j we see that in the first case neither $f^{\boldsymbol{c}}_{\{i,j\}}(1,1)$ nor $f^{\boldsymbol{c}}_{\{i,j\}}(2,2)$ can equal α. In the two latter cases we have again by the link-freeness in dimensions i and j that $f^{\boldsymbol{c}}_{\{i,j\}}(1,1) = f^{\boldsymbol{c}}_{\{i,j\}}(2,2) = \alpha$. By Lemma 14 we have that f is collapsible as claimed. □

A.4 Proof of Lemma 16

Lemma 16. *Every n-argument link-free, corner-free function $f : A_1 \times \ldots \times A_n \to \mathbb{Z}_3$ is collapsible or a permuted sum.*

Proof. For a link-free and corner-free function, only two types of induced rectangles are possible: XOR and Aff_3. If f contains an XOR spanned by dimensions i and j, then $|A_i| = |A_j| = 2$ since f is link-free.

We proceed with a case analysis. If f does not contain an XOR, then we select the first case. Otherwise, we pick an arbitrary XOR occurring in f and fix the dimensions i and j spanning it, and denote by $\boldsymbol{a}$ a set of inputs such that $f^{\boldsymbol{a}}_{\{i,j\}}$ is an (α,β)-XOR (if f has an XOR, we can relabel outputs such that there is an (α,β)-XOR). When we have fixed dimensions i, j we select among the four last cases of our proof based *only* on the 2×2-planes spanned by dimensions i, j.

Case 1: Only Aff_3 (all dimensions). If all induced rectangles of f are of the form Aff_3, then f satisfies the condition of Lemma 12 and is a (permuted) sum.

Case 2: Both XOR and Aff_3 (spanned by i, j). By Lemma 15, f is collapsible.

Case 3: Only XOR, one type of XOR (spanned by i, j). If only one type of XOR's occur, then f is a Boolean corner-free function and by Theorem 5 we have that f is a sum, and thus also a permuted sum.

Case 4: Only XOR, two types of XOR (spanned by i, j). We assume that f contains XOR's of types (α,β) and (α,γ). Then α occurs on exactly one diagonal of all 2×2 planes spanned by dimensions i, j and by Lemma 14 f is collapsible by collapsing β and γ.

Case 5: Only XOR, three types of XOR (spanned by i, j). By Lemma 13 we see that there must be a dimension k such that the input in dimension k determines the type of the XOR. Reorder inputs such that for input 1 in dimension k the 2×2-planes spanned by i and j are (α,β)-XOR's. We let $\boldsymbol{a} = (1)$ and $S_1 = \{k\}^C$ and see that $f^{\boldsymbol{a}}_{\{S_1\}}$ is a Boolean corner-free function. Thus, Theorem 5 implies that $f^{\boldsymbol{a}}_{\{S_1\}}(x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n) = \sum_{i\neq k} f_i(x_i)$ with the sum computed modulo 2.

We claim that f is a permuted sum with P_k as the permuter and the sum computed modulo 2. To see this, we prove that for all $x_k \in A_k$ and for all inputs $\boldsymbol{b}$ we have $f^{\boldsymbol{b}}_{\{k\}}(x_k) = \pi_{x_k}\{f^{\boldsymbol{b}}_{\{k\}}(1)\}$. As f is link-free, we have that $f^{\boldsymbol{b}}_{\{k\}}(x_k) \neq f^{\boldsymbol{b}}_{\{k\}}(1)$. If x_k is such that the 2×2-planes spanned by dimensions 1 and 2 when the input in dimension k is x_k are (α,β)-XOR then this means that $f^{\boldsymbol{b}}_{\{k\}}(1) = \alpha \implies f^{\boldsymbol{b}}_{\{k\}}(x_k) = \beta$ and $f^{\boldsymbol{b}}_{\{k\}}(1) = \beta \implies f^{\boldsymbol{b}}_{\{k\}}(x_k) = \alpha$. Similarly if the XOR's are (α,γ)-XOR's $f^{\boldsymbol{b}}_{\{k\}}(1) = \alpha \implies f^{\boldsymbol{b}}_{\{k\}}(x_k) = \gamma$, and as the 2×2-planes are XOR's we have $f^{\boldsymbol{b}}_{\{k\}}(1) \neq \alpha \implies f^{\boldsymbol{b}}_{\{k\}}(x_k) = \alpha$. The case for x_k with (β,γ)-XOR's is analogous, concluding the proof. □