# Perfect reductions

February 11, 2018

**Abstract**

We initiate the study of perfect (rather than merely statistical) reductions among cryptographic primitives. For simplicity, we focus on client-server functionalities. As opposed to the computational and statistical worlds, quite little is known here. While 1-out-of-2 bit-OT oblivious transfer (and several other functionalities) is known to be complete for client server functionalities since the seminal work of Killian. Furthermore [?] demonstrate a reduction reduction with improved efficiency that can be carried out in one round, making parallel calls to the OT (oracles), where the receiver plays the role of OT receiver in all instances [?].

We make first steps towards understanding perfect reductions, proving that a large class of client-server functions is perfectly reducible to 1-out-of-2 bit-OT. To the best of our knowledge. In particular, we show that for "most" finite functions of the form $f : X \times Y \to \{0,1\}$, where server domain size $|Y|$ is larger than client domain size $|X|$, a constant functions are perfectly reducible to 1-out-of-2 OT. This fraction grows roughly as $1 - exp(|X| - |Y|)$ Furthermore, the reduction is 1-round, and the receiver plays the role of the receiver in all OT calls. As far as we know, before now, very few functionalities such as $k$-out-of-$n$ OT were known to be computable in the 1-out-of-2 bit-OT hybrid model with perfect security.

Our work leaves open the question of whether all finite client-server functionalities are reducible to bit-OT. In general, are there any (OT or other) complete functionalities for client server SFE, even in more then 1 round. Next, once could study functionalities where both parties receive inputs, and randomized functionalities.

In addition to the obvious theoretical appeal of the question towards better understanding secure computation, perfect, as opposed to statistical reductions may be useful for designing MPC protocols with high concrete efficiency, achieved to eliminating the dependence on a security parameter.

## 1 Introduction

TODO.

## 2 Preliminaries

### 2.1 Notation

**Geometry.**

**Definition 2.1** (Affine dimension [?])**.** *For a set of vectors $V = \{v_1, \ldots, v_t\} \in \mathbb{R}^n$, we define their affine dimension $\mathcal{A}(V)$ as the dimension of the set $\{v_i - v_1\}_{i \geq 2}$.*

For a vector $v \in \mathbb{R}^m$, we let $|v|_1 = \sum_i |v_i|$ denote its $\ell_1$ norm, and $|v|_\infty = max_i|v_i|$ denote its $\ell_\infty$ norm. We say a function $f : X \times Y \to Z$ is full-dimensional if the affine dimension of the row set of its truth table $F$ is full (equals $(|Z| - 1)|X|$).

**Algebra.** For a matrix $A \in \mathbb{F}^{n \times n}$, where $\mathbb{F}$ is a field, let $|A|$ denote the determinant of $A$. $A^{i,j}$ denotes the $(i,j)$'th co factor of $A$, which is the $(n-1) \times (n-1)$ matrix obtained by removing the $i$'th row and $j$'th column of $A$. It is well known that:

**Fact 1.** *$A^{-1} = C$ where $|C_{i,j}| = |A_{i,j}|/|A|$ (Cramer's formula).*

For a pair of matrices $M_1 \in \mathbb{F}^{n_1 \times m}, M_2 \in \mathbb{F}^{n_2 \times m}$, we denote by $[M_1||M_2]$ the concatenation of $M_2$ below $M_1$.

## 2.2 Our model

We consider secure evaluation of client-server (non interactive, deterministic) functionalities $f : X \times Y \to Z$ for finite domains $X, Y, Z$, where the client outputs $f(x, y)$ and the server outputs $\perp$ (has no output). By default, we set $Z = \{0, 1\}$. We consider secure evaluation of such $f$ in the stand-alone setting, with perfect security, against a non adaptive malicious adversary corrupting a single party. The security notion is the standard simulation-based notion as in [?].

More specifically, we focus on perfect 1-round protocols $\Pi$ in the 1-out-of-2-bit-OT-hybrid model securely evaluating $f$, where the client plays the role of the receiver in all OT calls. Alternatively, such protocols can be viewed as 1-round reduction of evaluating $f$ to securely evaluating the functionality 1-out-of-2-bit-OT$^l$, receiving $l$ bits $x_1, \ldots, x_l$ from the client and $\mathbf{y}_1, \ldots, \mathbf{y}_l$ from the server , where $\mathbf{y}_i = (y_{i,0}, y_{i,1})$ is a pair of bits.

We denote protocols in our setting as tuples $\Pi = (\Pi_C, \Pi_R, \Pi_S)$ of randomized algorithms, where $\Pi_C(x) : X \to \{0, 1\}^l$ generates client's queries, $\Pi_R(x, c, v) : X \times \{0, 1\}^l \times \{0, 1\}^{2l}$ generates client's output based on $x, c$ and OT reply $v$, and $\Pi_S(y) : Y \to \{0, 1\}^{2l}$ is server's generator of OT inputs. $\Pi_C$ and $\Pi_R$ share randomness the same randomness $r$.

### 2.2.1 Restating security requirements geometrically.

We take a similar approach to that of [?] to representing protocols and their security requirements geometrically.

**Geometric representation of client's output distributions.** Fix $Z = \{0, 1\}$, and a protocol $\Pi = (\Pi_C, \Pi_R, \Pi_S)$. For a given server's strategy $s^* \in \{0, 1\}^{2l}$ for its OT oracle input [1], we represent the distribution of the client's output at the end of a protocol execution intended to compute $f$ as a vector $o \in \mathbb{R}^{|X|}$ indexed by $x \in [|X|]$. Here $o_x = p$ denotes the probability of outputting 1 on input $x$. We refer to such a distribution corresponding to some server's strategy $s^*$ as a *geometric row distribution* for $\Pi$, or just *row distribution* for short. We shall also consider geometric row distributions for the ideal model, corresponding to server's input distribution $y \in Y$, referring to them as a row distribution for $f$. We ommit $f, \Pi$ whenever it is clear from the context.

Observe that the single number $o_x$ uniquely encodes a distribution over the client's output set $\{0, 1\}$.[2] We refer to such a distribution as a *geometric row distribution*. Similarly, we consider *geometric column distributions*: for a given client's strategy $c^* \in \{0, 1\}^l$ for its input to the OT oracle, we consider the corresponding *geometric column* distribution vector $o \in \{0, 1\}^{|Y|}$ indexed by $y \in [|Y|]$, where $o_y$ is the probability of the client outputting 1 for server input $y$.

Generalizing for larger $Z = \{0, 1, \ldots, k-1\}$, a (geometric) row distribution $o \in \mathbb{R}^{(k-1-1)|X|}$, has entries labeled by pairs $(x, i)$ where $x \in [|X|], i \in Z \setminus \{0\}$, and $o_{(x,i)}$ denotes the probability of outputting $i$ on input $x$. Thus, for every $x, i$ we have $\sum_j o_{(x,j)} \leq 1$, and $o_{(x,i)} \geq 0$.[3] As in the case of $|Z| = 2$, this vector fully represents the client's output distribution for each input $x$. A similar extension can be made for (geometric) row distributions. For a given $x \in X$, let $o_x$ denote the sub-vector $(o_{x,z})_{z \in [k-1]}$. Also, the notion naturally extends to secure computation of randomized client-server functionalities $f : X \times Y \to Z$. Here exactly the same notions of client output distributions apply.

**Truth tables.** In the truth table $F$ of $f$, we index rows by elements of $Y$ and columns by elements of $X$. For $Z = \{0, 1\}$, the truth table representation we consider is just the standard one: a table where entry $(y, x)$ equals $f(x, y)$. We use $Y_y$ to denote the row vector ($X_x$ to denote the column) in $F$ corresponding to $y$ ($x$). We observe that each row $Y_y$ in this case is a geometric row distribution in the ideal model, where the server inputs $y$. We can interpret $Y_{y,x}$ as $f(x, y) = p$, where $p$ is the probability of outputting 1 (either $p = 0$ or $p = 1$).

Let us generalize this form to larger $Z$. We represent the truth table in "unary", where for each $y, x$ we have $|Z| - 1$ columns $(x, z)_{z \leq |Z|-1}$, and we set $F((x, z), y) = 1$ if $f(x, y) = z$, and $F((x, z), y) = 0$ otherwise (if $f(x, y) = |Z|$, all entires $F((x, z), y))$ will be 0).[4] Again, each row $Y_y$ is a valid geometric row distribution in the ideal world (corresponding to an input of $y$).

[1] This notion naturally generalizes to mixed strategies, but we do not need this extent of generality here.

[2] The vector $o$ represents $|X|$ separate distributions. Nothing is implied about the correlation between client's output on different inputs.

[3] The decision to exclude 0 is merely auesthetic, intended to remain consistent with standard binary truth tables.

[4] This is instead of having a single entry for each $(x, y)$ with values in $Z$.

**Definition of security** We require stand-alone security against a single malicious party. This requirement can be restated as three separate requirements. The equivalence to the original

**Definition 2.2.** *We say that a reduction $\Pi$ for evaluating $f : X \times Y \to Z$ as above is perfectly secure against a single malicious party if it satisfies:*

1. *Client correctness: For every server's input $s^* \in 0,1^{2l}$, the corresponding row distribution $o^*$ at the end of the protocol execution corresponds to a convex combination of the rows of $f$'s truth table. That is:*
$$o^* = \sum_i \alpha_i Y_i \text{ where } \alpha_i \geq 0, \sum_i \alpha_i = 1$$
2. *Server privacy: View the protocol $\Pi$ as a modified protocol $\Pi'$ performing an evaluation of a randomized functionality $f' : X \times Y \to \{0,1\}^l$, where the client's output is its (partial) view $s[c]$ during the protocol execution.*

   *We say that a column distribution vector $m_x$ is consistent for $x$, if for all $y, y'$ such that $f(x,y) = f(x,y')$ and $i \in [|Z|-1]$, we have $m_x[(y,i)] = m_x[(y',i)]$. We require that for each client strategy $c^* \in \{0,1\}^t$, the resulting column distribution (of $s[c]$, for evaluating $f'$) is a convex combination*
$$\sum_{x \in X} \alpha_x m_x$$
   *where each $m_x$ is consistent for $x$.*
3. *Honest correctness: Let $C_x = \text{support}(\Pi_C), S_x = \text{support}(\Pi_S)$. Then for all $c \in C_x, s \in S_y$, $\Pi_R(x, c, s[c]) = f(x,y)$.*[5]

We will also need a relaxed definition of client correctness, where the client may also output an error symbol $\perp$. We further relax it by allowing for a simulation error $\epsilon$.[6]

**Definition 2.3.** *We say a protocol $\Pi$ for computing $f$ is secure with $\epsilon$-relaxed client correctness, if it satisfies Definition 2.2, but client correctness is relaxed as follows. We allow the client to output a special error symbol $\perp$, in particular, we reinterpret the function $f$ as having an output domain $Z' = Z \cup \{\perp\}$. We say a row distribution $o$ is admissible, if it is a convex combination*
$$o^* = \sum_i \alpha_i o_i$$
*where each $o_i$ is either a row $Y_y$ in the truth table $F$, or is the all-$\perp$ vector $o_\perp$.*

*Then, for every deterministic server strategy $s^* \in \{0,1\}^{2l}$, the resulting row distribution $o^*$ satisfies either: (1) There exists an admissible row distribution $o$ where $\alpha_\perp \geq 1 - \epsilon$ such that $|o_x - o^*_x|_1 \leq \epsilon$.*[7] *Or (2) There exists an admissible row distribution $o$, so that $o^* = o$.*

# 3 A perfect reduction for any full-rank matrix

A protocol satisfying Definition 2.3 for all $f$ has been put forward in [**?**][8]

**Theorem 3.1.** *Let $f : X \times Y \to Z$ denote any (finite) function. Then, there exists $\epsilon_0$, such that for all $\epsilon \leq \epsilon_0$, there exists a protocol $\Pi$ evaluating $f$ with $\epsilon$-relaxed client correctness. The complexity of the protocol is $\ell(\epsilon, |X|, |Z|) = polylog(\epsilon^{-1}) \log |Z| + polylog(\epsilon^{-1}, \log |X|)$ parallel 1-out-of-2-bit-OT calls.*

To obtain concrete OT complexity of our resulting perfect protocol, we should calculate the concrete dependence on $\epsilon$.

We show how to transform a protocol into a perfectly secure one when $f$ is full-dimensional.

**Corollary 3.2.** *Let $f : X \times Y \to Z$ denote a full-dimensional function. Let $g = |X|(|Z| - 1)$. Then there exists a protocol $\Pi$ evaluating $f$ (with perfect security) and OT complexity $l = \ell(max(\epsilon_0, 1/10(g-1)g!), |X|, |Z|)$, where $\ell, \epsilon_0$ is as in Theorem 3.1.*

[5] For some, but not all functions $f$, $x$ is not required as an input to $R$, as $C_x \cup C_{x'} = \phi$ for all $x \neq x'$. It is not hard to prove that a sufficient condition on $f$ for having $C_x \cup C_{x'} = \phi$ in all secure protocols for $f$ is the existence of a $2 \times 2$ rectangle $\{y, y'\} \times \{x, x'\}$ in which 3 of the entries are identical, and the other entry differs from these three.

[6] This is a re-interpretation of the security of [**?**]'s protocol using our language.

[7] This simply means that for every input $x$, every non-$\perp$ value is output with probability at most $f$

[8] Consider their statistical construction for $NC^0$ functions. As we are not concerned with efficiency, but rather consider finite functions, the construction works for all functions, as far as we are concerned.

**Proof sketch.** The idea is simple: start with protocol $\Pi$ with $\epsilon$-relaxed client security for $f$ guaranteed by Theorem 3.1. We pick a sufficiently small $\epsilon > 0$, to be set later. Fix some vector $v$ in the convex hall of $F$'s (not $F'$'s) rows $Y_y$, which is "far enough from the edges" of that polygon. By "far enough" we mean that adding up to $\pm\epsilon$ in every coordinate results in a point which is still inside the polygon. Our protocol $\Pi'$ proceeds as follows. Whenever $\Pi$ outputs $\perp$ as an output on input $x$, output a distribution consistent with $v_x$. Otherwise, output the value output by $\Pi$. Indeed, if $\Pi$ satisfies condition (2), the resulting row distribution is a convex combination $\alpha_i o_i$, where $o_i$ is either $v$, or a row vector in $f$'s truth table $F$. As $v = \beta_i Y_i$ is a convex combination This is therefor a convex combination of the $Y_i$'s, as required.

Otherwise, the resulting row distribution vector $o_x$ in $\Pi$ is of the form $(1-\epsilon)v + e$, where $|e|_\infty \leq \epsilon$. If $e_{x,\perp} = \alpha_{x,\perp}$, we add $\alpha_{x,\perp} v_x$ to $o'_x$, the row distribution vector in $\Pi'$. Thus, $o'_x$ is a (syntactically) valid distribution for $f$

$$(1-\epsilon)v + e'$$

where $|e'_x|_1 \leq \epsilon$ for all $x$ (by a simple calculation, that stems from the fact that $F$ has 0/1 entries, and $v$ has entries in $[0,1]$). In particular, $|e'|_\infty \leq \epsilon$ as well, leaving us inside the convex hall of the $Y_i$'s for sufficiently small $\epsilon$ (since $f$ is full rank).

It remains to prove that there is a way to pick $v, \epsilon$ so that the resulting $o'_x$ satisfies the client correctness requirement. Since $f$ is full-rank, let us pick $g = |X|(|Z|-1)+1$ rows of $F$, $V = \{Y_1, \ldots, Y_g\}$, such that the dimension of $\{\Delta_{i-1} = Y_i - Y_1\}_{i \geq 2}$ is $g-1$. We will pick $v$ as a convex combination

$$\sum_{i \leq g} \alpha_i Y_i$$

of the $Y_i$'s. As $F$ has 0-1 entries we have

**Observation 1.** *$\Delta_j$ is has entires in the set $\{0, 1, -1\}$.*

Let us pick

$$v = 3/4 Y_1 + \sum_{2 \leq i \leq g} 1/4(g-1) Y_i$$

Now, we adapt $\epsilon$ to this choice based on the following observation.

**Claim 3.3.** *Let $u \in \mathbb{R}^{g-1}$ be a vector with $|u|_\infty \leq \epsilon$. Then, in the representation $u = \sum_{i \leq g-1} \alpha_i \Delta_i$ (it exists and is unique as the $\Delta_i$'s form a basis of $\mathbb{R}^{g-1}$)), it must be the case that $|\alpha_i| \leq \epsilon \cdot g!$.*

To prove the claim we apply Fact 1 to $M = (\Delta_1 || \Delta_2 || \ldots \Delta_{g-1})$, and use Observation 1 to bound each $|M^{-1}_{i,j}|$ as $|C_{i,j}| \leq (g-1)!/1 = (g-1)!$. Now, the solution to $Mx = u$ is $M^{-1}u$, thus we get $|x|_\infty \leq (g-1)(g-1)!\epsilon \leq g!\epsilon$, as required (here $g-1$ is the length of $u$).

We can rewrite a convex combination of the $Y_i$'s as

$$\sum_{i \leq g} \alpha_i Y_i = Y_1 + \sum_{i \leq g-1} \alpha_{i+1} \Delta_i \tag{1}$$

where the $\alpha_i$'s on the right hand side are non-negative integers summing to *at most* 1 (and move back and forth between the two representations). In particular, we have

$$v = Y_1 + 1/4(g-1) \sum_{i \leq g-1} \Delta_i$$

Recall also that the resulting row distribution is $o' = (1-\epsilon)v + e'$, where $|e'|_\infty \leq \epsilon$. Thus, we have $e' = \sum_{i \leq g-1} \alpha_i \Delta_i$, where $|\alpha_i| \leq g!\epsilon$.

Thus, we have

$$o' = (1-\epsilon)v + e' = Y_1 + (1-\epsilon)1/g \sum_{i \leq g-1} \Delta_i + (e' - \epsilon Y_1)$$

Let us write $w = e' - \epsilon Y_1$. Clearly, $|w|_\infty \leq 2\epsilon$. Here $|\beta_i| \leq 2\epsilon$ for each $i$. Thus, from Claim 3.3 we have

$$o' = Y_1 + \sum_{i \leq g-1} ((1-\epsilon)1/4(g-1) + \beta_i)\Delta_i$$

where $|\beta_i| \leq 2g!\epsilon$. Thus (since $g \geq 2$), picking $\epsilon = 1/10(g-1)(g!)$, we obtain $o'$ of the form presented in Equation 1, which falls in the required region (the coefficients of the $\Delta_i$'s are all non-negative and sum to at most 1)

# 4 Applications to concrete security