

# Perfect reductions

February 13, 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  $f : X \times Y \rightarrow \{0, 1\}$ , 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 than 1 round. Next, one 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  $V = \{v_1, \dots, v_t\} \subseteq \mathbb{R}^n$ , 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. For a set of vectors  $V = \{v_1, \dots, v_t\} \subseteq \mathbb{R}^n$ , a linear combination  $\sum_i \alpha_i v_i$  where  $\sum_i \alpha_i = 1$  and  $\forall i \alpha_i \geq 0$  is a convex combination of  $V$ . The convex hull of  $V$ ,  $\text{CH}(V) = \{u = \sum_i \alpha_i v_i \mid u \text{ is a convex combination of } V\}$ .

**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  $f : X \times Y \rightarrow Z$  for finite domains  $X, Y, Z$ , where the client outputs  $f(x, y)$  and the server outputs  $\perp$  (has no output).<sup>1</sup>

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.

In our setting, a protocol for evaluating a client-server functionality  $f : X \times Y \rightarrow Z$  is defined as follows.

**Definition 2.2** (1-round protocols in OT-hybrid model). *A protocol for evaluating  $f : X \times Y \rightarrow Z$  are tuples  $\Pi = (\Pi_Q, \Pi_R, \Pi_D)$  of randomized algorithms, where  $\Pi_Q(x) : X \rightarrow \{0, 1\}^l$  generates client's query  $c$ .  $\Pi_R(x, c, v) : X \times \{0, 1\}^l \times \{0, 1\}^l \rightarrow Z$  generates client's output based on  $x, c$  and OT reply  $v$ .<sup>2</sup>  $\Pi_D(y) : Y \rightarrow \{0, 1\}^{2l}$  is server's generator of OT inputs. We refer to  $l$  as the communication complexity of  $\Pi$ .*

For  $s = (s_{1,0}, s_{1,1}, \dots, s_{l,0}, s_{l,1}) \in \{0, 1\}^{2l}$ , and  $c \in \{0, 1\}^l$ , we let  $s[c]$  denote  $(s_{1,c_1}, \dots, s_{l,c_l})$ .

A protocol  $\Pi = (\Pi_Q, \Pi_R, \Pi_D)$  as in Definition 2.2 with CC  $l$  operates in the 1-out-of-2-bit-OT <sup>$l$</sup> -hybrid model as described above.<sup>3</sup> That is, it is specified by a pair of randomized turing machines  $\Pi_C^?, \Pi_S^?$  with oracle access to an idealized functionality 1-out-of-2-bit-OT <sup>$l$</sup>  operating as follows

$\Pi_C^?(x; r) :$  Let  $c = \Pi_Q(x; r)$ . Send  $c$  as input to the 1-out-of-2-bit-OT <sup>$l$</sup>  oracle, and let  $v$  denote the oracle's output. Output  $\Pi_R(x, c, v; r)$ .

$\Pi_S^?(y; r) :$  Let  $s = \Pi_D(y; r)$ , and send it to the 1-out-of-2-bit-OT <sup>$l$</sup>  oracle (where  $(s_{i,0}, s_{i,1})$  go to the  $i$ 'th 1-out-of-2-bit-OT instance).

Both algorithms  $\Pi_C, \Pi_S$  run in parallel, in a single round. We will usually use the notation  $\Pi = (\Pi_Q, \Pi_R, \Pi_D)$  to denote protocols, while  $\Pi_C, \Pi_S$  are implicit.

Note that the above definition of a protocol is merely syntactic.

### 2.2.1 Restating security requirements geometrically.

We take a similar approach to that of [?] to representing protocols and their security requirements geometrically. More specifically, we represent the client's output distributions vector for a given server's strategy as a vector over  $\mathbb{R}^t$  for a suitable  $t$ . This vector bundles together client output distributions for all client's inputs, with a subset of coordinates corresponding to each client's input. A protocol  $\Pi$  defines a region  $P_S$  of achievable distributions corresponding to all possible (possibly malicious) server strategies.

Similarly, for a given client's strategy, we consider the client's vector of *views*, with a subset of coordinates corresponding to each server's input  $y$ . Similarly to  $P_S$ ,  $\Pi$  defines a region  $P_C$  of achievable distributions corresponding to all client's strategies.

On the other hand, we consider distribution vectors achievable in the ideal model, and corresponding regions  $\tilde{P}_S, \tilde{P}_C$  of all achievable distributions corresponding to possible server strategies and client's strategies respectively. Jumping ahead, our representation of distributions will be such that the regions  $\tilde{P}_S, \tilde{P}_C$  will roughly correspond to convex combinations of row and column vectors of the function's truth table.

To achieve security against a malicious server and client respectively, we require that  $P_S \subseteq \tilde{P}_S, P_C \subseteq \tilde{P}_C$ . For honest correctness (that is, the protocol always outputs the correct value if both parties behave honestly), we make a separate requirement on distribution vectors corresponding to valid client and server strategies. While the first two requirements are readily expressed as a

<sup>1</sup> All the definitions below readily generalize to randomized functionalities  $f$ , but we focus on deterministic  $f$  for simplicity.

<sup>2</sup> 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.

<sup>3</sup> In particular, it first receives all inputs and only then returns all the outputs to the client. No rushing such as sending inputs to a certain OT instance after getting outputs from other OT instances is possible.

single LP (linear program) [], it is unclear how to incorporate the third requirement into a linear program.

**Geometric representation of client's output distributions.** Fix a protocol  $\Pi = (\Pi_Q, \Pi_R, \Pi_D)$  as in Definition 2.2 for evaluating a functionality  $f : X \times Y \rightarrow Z$ . For the sake of defining our output distributions we view the protocol as merely randomized mapping from  $X \times Y$  to client outputs  $Z$ , and do not require it to securely evaluate  $f$ , or even correctly evaluate it if everyone behaves honestly.

**Boolean functions.** Fix  $Z = \{0, 1\}$ . For a given server's strategy  $\Pi_D^*(x = y; r) = s^*$  for some fixed  $s^* \in \{0, 1\}^{2^l}$ , we consider the distributions of the client's output at the end of a protocol execution  $\Pi^* = (\Pi_Q, \Pi_R, \Pi_D^*)$  (that is, a protocol resulting from  $\Pi$  when the server runs  $\Pi_D^*$  instead of  $\Pi_D$ ). We denote this set of distributions by a vector  $o \in \mathbb{R}^{|X|}$  indexed by  $x \in [|X|]$ . Here  $o_x = p$  denotes the probability of the client outputting 1 on input  $x$ . That is,

$$o_x = \Pr_r[\Pi_Q(x; r) = c; \Pi_R(x, c, s^*[c]) = 1].$$

We refer to such a vector corresponding to some (possibly invalid) server's strategy  $s^*$  as a *geometric row distribution* for  $\Pi$ . We shall also consider geometric row distributions for the ideal model evaluating  $f$ , corresponding to server's input distributions  $y \in Y$ , referring to them as a row distribution for  $f$ . We omit  $\Pi, f$  whenever clear from the context.

Observe that the single number  $o_x$  uniquely encodes a distribution over the client's output set  $\{0, 1\}$  on input  $x$ .<sup>5</sup> Similarly, we consider *geometric column distributions* for  $\Pi$ : 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  $y$ . That is:

$$o_y = \Pr_r[\Pi_R(x, c^*, \Pi_D(y; r)[c^*]) = 1].$$

**General functions.** Generalizing for larger  $Z = \{0, 1, \dots, k-1\}$ , a (geometric) row distribution  $o \in \mathbb{R}^{(k-1) \cdot |X|}$ , 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  $o_{(x,i)} \geq 0$ .<sup>6</sup> 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  $(o_{x,z})_{z \in [k-1]}$ .

**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  $F_y$  to denote the row vector ( $X_x$  to denote the column) in  $F$  corresponding to  $y$  ( $x$ ). We observe that each row  $F_y$  in this case is a geometric row distribution in the ideal model, where the server inputs  $y$ . We can interpret  $F_{y,x}$  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  $(x, z)_{z \in [|Z|-1]}$ , and we set  $F((x, z), y) = 1$  if  $f(x, y) = z$ , and  $F((x, z), y) = 0$  otherwise (if  $f(x, y) = 0$ , all entries  $F((x, z), y)$  will be 0).<sup>7</sup> Again, each row  $F_y$  is a valid geometric row distribution in the ideal world (corresponding to a server input of  $y$ ).

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

Let us recall the standard definitions of security against malicious parties in the 1-out-of-2-bit-OT<sup>l</sup>-hybrid model.

**Definition 2.3.** Let  $\Pi$  denote a protocol for evaluating a function  $f$  as in Definition 2.2 in the 1-out-of-2-bit-OT<sup>l</sup>-hybrid model. Then it evaluates  $f$  with malicious security against a non adaptive malicious adversary if it satisfies:

<sup>4</sup>This notion naturally generalizes to randomized strategies, but we do not need this extent of generality here.

<sup>5</sup>The vector  $o$  represents  $|X|$  separate distributions, one for each client's input  $x$ . Nothing is implied about the correlation between client's outputs on different inputs for a given server's strategy  $s^*$ .

<sup>6</sup>The decision to exclude 0 is merely aesthetic, intended to remain consistent with standard binary truth tables.

<sup>7</sup>This is instead of having a single entry for each  $(x, y)$  with values in  $Z$ .

*Security against malicious servers:* For all algorithms  $\Pi_D^*(y; r)$ , there exists a randomized simulator algorithm  $\text{Sim}_D^*(y; r) : Y \rightarrow Y$  such that for all  $x \in X, y \in Y, r_s$ , the following equality of distributions holds.

$$\Pi_R(x, \Pi_Q(x; r_c), \Pi_D(y; r_s)) = f(x, \text{Sim}_D^*(y; r_s))$$

8

**Definition 2.4.** We say that a reduction  $\Pi$  for evaluating  $f : X \times Y \rightarrow Z$  as above is perfectly secure against a single malicious party if it satisfies:

1. *Client correctness:* For every server's strategy  $s^* \in \{0, 1\}^{2^l}$ , the corresponding row distribution  $o^*$  of  $f'$  is in  $\text{CH}(\{F_y\}_{y \in Y})$ , where the  $F_y$ 's are the rows of the truth table  $F$  of  $f$ .
2. *Server privacy:* Define a modified protocol  $\Pi'$  where the client's output is replaced by its (partial) view  $s[c]$  recieved from the OT oracle. That is,  $\Pi' = (\Pi_Q, \Pi'_R, \Pi_D)$ , where  $\Pi'_R(x, c, v) = v$ . This protocol is a mapping from  $X \times Y$  to  $Z' = \{0, 1\}^l$ .

We say that a column distribution for  $\Pi'$  vector  $m_x$  is consistent with  $f, x$ , if for all  $y, y'$  such that  $f(x, y) = f(x, y')$  and  $i \in [2^l - 1]$ , we have  $m_x[(y, i)] = m_x[(y', i)]$ . We require that for each client strategy  $\Pi_Q^*(x; r) = c^*$  for  $c^* \in \{0, 1\}^l$ , the resulting column distribution in  $\Pi'$  is a convex combination

$$\sum_{x \in X} \alpha_x m_x$$

where each  $m_x$  is consistent for  $f, x$ .

3. *Honest correctness:* Let  $C_x = \text{support}(\Pi_Q(x))$ ,  $S_y = \text{support}(\Pi_D(y))$ . Then for all  $c \in C_x, s \in S_y$ ,  $\Pi_R(x, c, s[c]) = f(x, y)$ .

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  $\epsilon$ .<sup>9</sup>

**Definition 2.5.** We say a protocol  $\Pi$  for computing  $f$  is secure with  $\epsilon$ -relaxed client correctness, if it satisfies Definition 2.4, 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  $F_y$  in the truth table  $F$ , or is the all- $\perp$  vector  $o_\perp$ .

Then, for every deterministic server strategy  $s^* \in \{0, 1\}^{2^l}$ , the resulting row distribution  $o^*$  satisfies either: (1) There exists an admissible row distribution  $o$  where  $\alpha_\perp \geq 1 - \epsilon$  such that  $|o_x - o_x^*|_1 \leq \epsilon$  for all  $x \in X$ .<sup>10</sup> Or (2) There exists an admissible row distribution  $o$ , so that  $o^* = o$ .

For client-server functionalities and protocols  $\Pi$  as in Definition 2.2 as we consider, our Definition ?? of perfect security is equivalent to standard perfect security against malicious adversaries, and the equivalence is straight forward to prove.

**Lemma 2.1.** Let  $\Pi = (\Pi_Q, \Pi_R, \Pi_D)$  be a protocol for evaluating a client-server functionality  $f : X \times Y \rightarrow Z$ .  $\Pi$  is perfectly secure by Definition 2.4 iff. it is perfectly secure against a single malicious party in the 1-out-of-2-bit-OT<sup>l</sup>-hybrid model according to standard simulation based definition [1] (against a non-adaptive malicious adversary).

<sup>8</sup>Wlog. we may assume  $\Pi_D^*$  is deterministic. Also note that we do not require that the algorithms (as we generally consider finite functionalities). Due to the simple structure of  $\Pi$  the above requirement can be further simplified to requiring that for all  $\cdot$  there exists a distribution  $Y^*$  over  $Y$  such that

$$\Pi_R(x, \Pi_Q(x; r_c), s^*) = f(x, Y^*)$$

<sup>9</sup>This is a re-interpretation of the security of [?]’s protocol using our language.

<sup>10</sup>This simply means that for every input  $x$ , a non- $\perp$  value is output with probability at most  $\epsilon$

### 3 A perfect reduction for any full-rank matrix

Our starting point is a protocol as in Definition 2.2 (1-round protocol in the (1-out-of-2-bit-OT)<sup>l</sup>-hybrid model for some  $l$ ) by [?] [Section 3]. We denote this protocol by  $\Pi_{\text{IKOPSW}}$ .

**Theorem 3.1.** *Let  $f : X \times Y \rightarrow Z$  denote any (finite) function. Then for all  $\epsilon \geq 0$ , there exists a protocol  $\Pi$  evaluating  $f$  with  $\epsilon$ -relaxed client correctness. Let  $h_i$  denote the smallest formula for evaluating the  $i$ 'th bit of  $f(x, y)$ , and let  $h = \max_{i \leq \log(|Z|)} h_i$ . Then,  $\Pi$  has communication complexity of  $l = \log(|Z|) \tilde{O}(\log(|X|)(\log(\epsilon^{-1}) + h)^2) + \text{poly}(\epsilon^{-1})$ .*

We transform it into a perfectly secure protocol in two steps. First, we transform it into a perfectly server-private (yet not perfectly client-correct) protocol  $\Pi'$ . Then we transform  $\Pi'$  into  $\Pi''$  preserving perfect server privacy and adding perfect client correctness.

#### 3.1 Making $\Pi_{\text{IKOPSW}}$ perfectly server-private

$\Pi_{\text{IKOPSW}}$

In this section we describe the first step of the construction.

Definition 2.5 for all  $f$  has been put forward in [?]<sup>11</sup>

**Definition 3.1.** *We say a function  $f : X \times Y \rightarrow Z$  is full-dimensional if the affine dimension of the row set of its truth table  $F$  is the maximal possible - equals  $(|Z| - 1)|X|$ .*

**Theorem 3.2.** [ ] Let  $f : X \times Y \rightarrow Z$  denote any (finite) function. Then for all  $\epsilon \geq 0$ , there exists a protocol  $\Pi$  evaluating  $f$  with  $\epsilon$ -relaxed client correctness. Let  $h_i$  denote the smallest formula for evaluating the  $i$ 'th bit of  $f(x, y)$ , and let  $h = \max_{i \leq \log(|Z|)} h_i$ . Then,  $\Pi$  has communication complexity of  $l = \log(|Z|) \tilde{O}(\log(|X|)(\log(\epsilon^{-1}) + h)^2)$ .

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

The main technical contribution of this paper is a way to transform a protocol into a perfectly secure one when  $f$  is full-dimensional.

**Corollary 3.3.** *Let  $f : X \times Y \rightarrow Z$  denote a full-dimensional function. Let  $g = |X|(|Z| - 1)$ . Then there exists a protocol 1-round protocol evaluating  $f$  with perfect security against a malicious adversary in the 1-out-of-2-bit-OT<sup>l</sup>-hybrid model for  $l = \ell(\max(\epsilon_0, 1/10(g - 1)g!), |X|, |Z|)$ , where  $\ell, \epsilon_0$  is as in Theorem 3.2.*

**Proof sketch.** The idea is simple: start with protocol  $\Pi$  with  $\epsilon$ -relaxed client security for  $f$  guaranteed by Theorem 3.2. We pick a sufficiently small  $\epsilon > 0$ , to be set later. Fix some vector  $v$  in the convex hull of  $F$ 's (not  $F'$ 's) rows  $F_y$ , 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 hull 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  $F$ ,  $V =$

<sup>11</sup>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.

$\{Y_1, \dots, Y_g\}$ , 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 3.4.** *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  $M = (\Delta_1 || \Delta_2 || \dots \Delta_{g-1})$ , and use Observation 1 to bound each  $|M_{i,j}^{-1}|$  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 3.4 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