New Techniques for Obfuscating Conjunctions

James Bartusek* Princeton
Tancrède Lepoint† SRI International
Fermi Ma‡ Princeton
Mark Zhandry§ Princeton

October 2, 2018

Abstract

A conjunction is a function $f(x_1, \ldots, x_n) = \bigwedge_{i \in S} l_i$ where $S \subseteq [n]$ and each l_i is x_i or $\neg x_i$. Bishop et al. (CRYPTO 2018) recently proposed obfuscating conjunctions by embedding them in the error positions of a noisy Reed-Solomon codeword and encoding the codeword in a group exponent. They prove distributional virtual black box (VBB) security in the generic group model for random conjunctions where $|S| \geq 0.226n$. While conjunction obfuscation was known from LWE [WZ17, GKW17], these constructions rely on substantial technical machinery.

In this work, we conduct an extensive study of *simple* conjunction obfuscation techniques.

- We abstract the Bishop et al. scheme to obtain an equivalent yet more efficient "dual" scheme that can handle conjunctions over exponential size alphabets. This scheme admits a straightforward proof of generic group security, which we combine with a novel combinatorial argument to obtain distributional VBB security for $|S|$ of *any size*.
- If we replace the Reed-Solomon code with a *random binary linear code*, we can prove security from standard LPN and avoid encoding in a group. This addresses an open problem posed by Bishop et al. to prove security of this simple approach in the standard model.
- We give a new construction that achieves *information theoretic* distributional VBB security and weak functionality preservation for $|S| \geq n - n^{\delta}$ and $\delta < 1$. Assuming discrete log and $\delta < 1/2$, we satisfy a stronger notion of functionality preservation for computationally bounded adversaries while still achieving information theoretic security.

*bartusek.james@gmail.com. This work was done while the author was an intern at SRI International.
†tancrede.lepoint@gmail.com.
‡fermima1@gmail.com. This work was done while the author was an intern at SRI International.
§mzhandry@princeton.edu.

Contents

1 Introduction

Program obfuscation [BGI+01] scrambles a program in order to hide its implementation details, while still preserving the program's functionality. Program obfuscation has recently received considerable attention, yielding new constructions [SW14, BZ14, CLTV15, BP15, DGL+16, MPS16] and demonstrating many applications throughout cryptography [MO14, BFM15, GPS16, CCC+16].

Much of the recent attention on obfuscation focuses on obfuscating *general* programs. Such obfuscation is naturally the most useful, but currently the only known constructions are extremely inefficient and rely on new uncertain complexity assumptions about cryptographic multilinear maps [GGH13, CLT13, GGH15]. Despite advances in terms of efficiency [AGIS14, AB15, Zim15] and security [AJ15, MSZ16, LV16, GMM+16, Lin16, AS17, BISW17, LT17, FRS17, MZ18, BGMZ18], obfuscating general programs remains far from usable.

For some specific functionalities, it is possible to avoid multilinear maps. A series of works have shown how to obfuscate point functions (i.e., boolean functions that output 1 on a single input) and hyperplanes [Can97, LPS04, Wee05, CD08, DKL09, GKPV10, CRV10, YZ16, BS16, KY18]. Brakerski, Vaikuntanathan, Wee, and Wichs [BVWW16] showed how to obfuscate conjunctions under a variant of the Learning with Errors (LWE) assumption. More recently it has been shown how to obfuscate a very general class of evasive functions including conjunctions under LWE [GKW17, WZ17].

1.1 This Work: Conjunction Obfuscation

In this work, we primarily consider obfuscating conjunctions. This class of programs has also been called pattern matching with wildcards [BKM+18], and in related contexts is known as bit-fixing [BW13].

A conjunction is any boolean function $f(x_1, \ldots, x_n) = \bigwedge_{i \in S} l_i$ for some $S \subseteq [n]$, where each literal l_i is either x_i or $\neg x_i$. This functionality can be viewed as pattern-matching for a pattern $\mathsf{pat} \in \{0, 1, *\}^n$, where the $*$ character denotes a wildcard. An input string $x \in \{0,1\}^n$ matches a pattern pat if and only if x matches pat at all non-wildcard positions. So for example $x = \mathtt{0100}$ matches $\mathsf{pat} = \mathtt{*10*}$ but not $\mathsf{pat} = \mathtt{1**0}$.

We are interested in obfuscating the boolean functions $f_{\mathsf{pat}} \colon \{0,1\}^n \to \{0,1\}$ which output 1 if and only if x matches pat. We additionally give obfuscation constructions for two generalizations of the pattern matching functionality: multi-bit conjunction functions $f_{\mathsf{pat},m}$ which output a secret message $m \in \{0,1\}^\ell$ on an accepting input, and conjunctions over arbitrary size alphabets (rather than just the binary alphabet).

In particular, we consider the notion of *distributional virtual black-box obfuscation* which guarantees that the obfuscation of a pattern drawn from some distribution can be simulated efficiently, given only oracle access to the truth table of the function defined by the pattern. We consider this notion of obfuscation in the *evasive* setting, where given oracle access to a pattern drawn from the distribution, a polynomial time algorithm cannot find an accepting input except with negligible probability. Thus our goal will be to produce obfuscations that are easily simulatable given no information about the underlying pattern. This naturally leads us to the goal of producing obfuscations that are computationally indistinguishable from random bits.

At CRYPTO 2018, Bishop, Kowalczyk, Malkin, Pastro, Raykova, and Shi [BKM+18] gave a simple and elegant obfuscation scheme for conjunctions, which they prove secure in the generic group model [Sho97]. Unfortunately, they did not prove security relative to any concrete (efficiently falsifiable [Nao03, GW11]) assumption on cryptographic groups. Before their work, obfuscation for

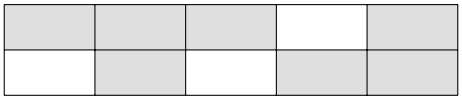
conjunctions was already known from LWE as a consequence of lockable obfuscation (also known as obfuscation for compute-and-compare programs) [WZ17, GKW17]. However, for the restricted setting of conjunctions, the Bishop et al. [BKM+18] construction is significantly simpler and more efficient.

In this work we show how to alter the Bishop et al. construction in various ways in order to:

- Simplify the generic group security analysis and improve the size, efficiency, and generality of the obfuscation.
- Obtain simple conjunction obfuscation under standard assumptions, or even no assumptions at all.

Review of the [BKM+18] Construction. We first recall the [BKM+18] scheme for obfuscating a pattern $\mathsf{pat} \in \{0, 1, *\}^n$. Begin by fixing a field $\mathbb{F}_q$ for a prime q exponential in n. Then sample uniformly random $s_1, \ldots, s_{n-1} \leftarrow \mathbb{F}_q$ and define the polynomial $s(t) := \sum_{k=1}^{n-1} s_k t^k \in \mathbb{F}_q[t]$. Note that $s(t)$ is a uniformly random degree $n-1$ polynomial conditioned on $s(0) = 0$.

Visualize a $2 \times n$ grid with columns indexed as $i = 1, \ldots, n$ and rows indexed as $j = 0, 1$. Correspond the ith character in pat with the ith grid column as follows: $\mathsf{pat}_i = \mathtt{0}$ points to the top cell, $\mathsf{pat}_i = \mathtt{1}$ to the bottom cell, and $\mathsf{pat}_i = \mathtt{*}$ to both cells. For example, when $n = 5$ the pattern $\mathsf{pat} = \mathtt{0*01*}$ naturally corresponds to the gray cells in the 2×5 grid below:



Next, we evaluate the polynomial $s(t)$ at $2i + j$ for each gray cell (i, j). For each white cell, we sample a uniformly random element $r_{2i+j} \leftarrow \mathbb{F}_q$. So for $\mathsf{pat} = \mathtt{0*01*}$, the resulting grid would be

$s(2)$	$s(4)$	$s(6)$	r_8	$s(10)$
r_3	$s(5)$	r_7	$s(9)$	$s(11)$

If we stop here and simply output these $2n$ field elements in the clear, we already have a functional (although insecure) construction. To evaluate whether or not an input $x \in \{0, 1\}^n$ matches the pattern, we read the n field elements corresponding to the bits of x. In our example, if $x = \mathtt{00011}$ (an accepting input for $\mathsf{pat} = \mathtt{0*01*}$), we read row $j = 0$ in columns $1, 2, 3$, and row $j = 1$ in columns 4 and 5:

$s(2)$	$s(4)$	$s(6)$		
			$s(9)$	$s(11)$

An input is accepting precisely when all $n = 5$ selected elements are evaluations of the random degree $n - 1 = 4$ polynomial $s(t)$. Thus, we can evaluate by simply performing a Lagrange interpolation and evaluating the polynomial at $s(0)$, i.e., we generate reconstruction coefficients $L_2, L_4, L_6, L_9, L_{11} \in \mathbb{F}_q$ that satisfy

$$L_2 p(2) + L_4 s(4) + L_6 s(6) + L_9 s(9) + L_{11} s(11) = s(0)$$

for any degree 4 polynomial $s(t)$. When we perform this reconstruction for an accepting input, we recover $s(0) = 0$. If the input is not accepting, say $x = \texttt{10010}$ in our example, then at least one of the n reconstruction points will be a random field element instead of the correct polynomial evaluation. With overwhelming probability, the reconstructed field element will not be 0, indicating x did not match pat.

Bishop et al. [BKM+18] observe that these $2n$ field elements are essentially a noisy Reed-Solomon codeword with the white grid cells representing error positions. If the number of error positions is small enough, an attacker can run any Reed-Solomon error correction algorithm to recover $s(t)$ and learn pat. The key observation of [BKM+18] is that honest functionality only requires linear operations on the field elements, whereas all known error-correction algorithms for Reed-Solomon codes are non-linear. Thus, the final step of their construction is to place the $2n$ field elements in the exponent of a group $\mathbb{G} = \langle g \rangle$ of order q. Evaluation is done by performing the reconstruction in the exponent and accepting if and only if Lagrange reconstruction yields g^0. For security, they prove the following:

Theorem 1 ([BKM+18])**.** *Let $\mathcal{U}_{n,w}$ be the uniform distribution over all patterns in $\{0,1,*\}^n$ with exactly w wildcards. For any $w < 0.774n$, this construction attains distributional virtual black box security in the generic group model.*

The proof in [BKM+18] is fairly technical and spans over 10 pages. Bishop et al. do not address whether the scheme actually becomes insecure for $0.774n < w < n - \omega(\log n)$, or if the bound is a limitation of their analysis.[1]

1.2 Technical Overview

We provide several new interpretations of the [BKM+18] scheme. Through these interpretations, we are able to obtain improved security, efficiency, and generality, as well as novel constructions secure under standard cryptographic assumptions.

1.2.1 Interpretation 1: The Primal

Our first observation is that the $2n$ field elements generated by the [BKM+18] construction can be rewritten as a product of a Vandermonde-style matrix (henceforth the "Vandermonde matrix") and a random vector, plus a certain "error vector." So if the pattern is $\mathsf{pat} = \texttt{11*0}$, instead of writing the elements in grid form as

r_2	r_4	$s(6)$	$s(8)$
$s(3)$	$s(5)$	$s(7)$	r_9

[1] If $w = n - O(\log n)$, the distributional virtual black box security notion is vacuous since an attacker can guess an accepting input and recover pat entirely.

we can stack them in a column as

$$\begin{pmatrix} r_2 \\ s(3) \\ r_4 \\ s(5) \\ s(6) \\ s(7) \\ s(8) \\ r_9 \end{pmatrix} = \begin{pmatrix} 2^1 & 2^2 & 2^3 \\ 3^1 & 3^2 & 3^3 \\ 4^1 & 4^2 & 4^3 \\ 5^1 & 5^2 & 5^3 \\ 6^1 & 6^2 & 6^3 \\ 7^1 & 7^2 & 7^3 \\ 8^1 & 8^2 & 8^3 \\ 9^1 & 9^2 & 9^3 \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} + \begin{pmatrix} r'_2 \\ 0 \\ r'_4 \\ 0 \\ 0 \\ 0 \\ 0 \\ r'_9 \end{pmatrix}.$$

If $s_1, s_2, s_3, r'_2, r'_4, r'_9$ are chosen uniformly at random from $\mathbb{F}_q$, the resulting column vector has the same distribution as the $2n$ group exponents in the [BKM+18] construction. The product of the Vandermonde matrix and the random vector $s^\top \coloneqq \begin{pmatrix} s_1 & s_2 & s_3 \end{pmatrix}^\top$ is simply $\begin{pmatrix} s(2) & s(3) & \cdots & s(9) \end{pmatrix}^\top$ for the polynomial $s(t) \coloneqq s_3 t^3 + s_2 t^2 + s_1 t$. Since r'_2, r'_4, r'_9 are uniformly random, adding the "error vector" with these entries is equivalent to replacing the corresponding polynomial evaluations with random field elements as in the [BKM+18] scheme. For sake of clarity, we use $A \cdot s + e$ to denote this view of the scheme, and we write the $2n$ group elements that result from encoding in the exponent as $g^{A \cdot s + e} \in \mathbb{G}^{2n}$.

So far, nothing has changed — [BKM+18] obfuscation is precisely $g^{A \cdot s + e}$. But if we revisit the evaluation procedure in the $A \cdot s + e$ format, a possible improvement to the construction becomes apparent. Recall that evaluation is simply polynomial interpolation: on input $x \in \{0,1\}^n$, the evaluator generates a vector $v \in \mathbb{F}_q^{2n}$ where $v_{2i+x_i-1} = 0$ for all $i \in [n]$, and the n non-zero elements of v are Lagrange coefficients. Notice that for any input x (even ones not corresponding to accepting inputs), the Lagrange coefficients ensure v satisfies $v^\top \cdot A = 0 \in \mathbb{F}_q^{2n}$ and the corresponding scalar equation $v^\top \cdot A \cdot s = 0$. This means an input x is only accepted if $v^\top \cdot (A \cdot s + e) = v^\top \cdot e = 0$. Indeed, we can verify that if there exists a position $i \in [n]$ where $x_i \neq \mathsf{pat}_i$ (note that if $\mathsf{pat}_i = *$ we take this to mean $x_i = \mathsf{pat}_i$), this corresponds to an entry where v is non-zero and e is uniformly random, making $v^\top \cdot e$ non-zero with overwhelming probability.

1.2.2 Interpretation 2: The Dual

Notice that evaluation only required the A matrix and e vector. The random degree $n-1$ polynomial $s(t)$ generated in the [BKM+18] scheme, whose coefficients form the random s vector, does not play a role in functionality. This suggests performing the following "dual" transformation to the $A \cdot s + e$ scheme. Let B be an $(n+1) \times 2n$ dimensional matrix whose rows span the left kernel of A. Since $B \cdot A = \mathbf{0} \in \mathbb{F}_q^{(n+1)\times(n-1)}$, multiplying $B \cdot (A \cdot s + e)$ yields the $n+1$ dimensional vector $B \cdot e$. We claim this dual $g^{B \cdot e}$ scheme captures all the information needed for secure generic group obfuscation, but with $n+1$ group elements rather than $2n$.

Evaluation in the Dual. A similar evaluation procedure works for the dual scheme. On input x, the evaluator solves for a vector $k \in \mathbb{F}_q^{n+1}$ so that the $2n$-dimensional vector $k^\top \cdot B$ is 0 at position $2i + x_i - 1$ for each $i \in [n]$. Note that such a k exists since we only place n constraints on $n+1$ variables. $k^\top \cdot B$ will play exactly the same role as $v^\top$ vector from the $A \cdot s + e$ evaluation. On accepting input, $k^\top \cdot B$ will be 0 in all the positions where e is non-zero, so $k^\top \cdot B \cdot e = 0$. On rejecting inputs, $(k^\top \cdot B)$ will have a non-zero entry where the corresponding entry of e is uniformly random, so $k^\top \cdot B \cdot e \neq 0$ with overwhelming probability.

Solving for k to ensure $k^\top \cdot B$ is zero in n positions gives an evaluation procedure that in general requires more than $O(n^2)$ time, making it slower to evaluate than the [BKM+18] scheme. However, we can choose B to enable fast evaluation. In particular, if we simply choose B to be a Vandermonde style matrix whose (i,j)th entry is j^i, solving for k becomes the problem of computing the coefficients of a polynomial given specified roots. In Appendix B, we show that this scheme, as well as the [BKM+18] scheme, can be optimized to $O(n \log^2 n)$ evaluation.

Most importantly, our security analysis will only require that $B \in \mathbb{F}_q^{(n+1)\times 2n}$ satisfies the property that every $(n+1) \times (n+1)$ submatrix is full rank. The above choice of B guarantees this for sufficiently large q.

Proving Generic Group Security. Some of the formalism in the [BKM+18] security proof is dedicated to handling the random polynomial s, which is cleaned out of our dual construction. We briefly sketch an intuitive security argument for the following theorem, which extends the [BKM+18] result.

Theorem 2. *Let $\mathcal{U}_{n,w}$ be the uniform distribution over all patterns in $\{0,1,*\}^n$ with exactly w wildcards. For any $w \leq n - \omega(\log n)$, our dual construction attains distributional virtual black box security in the generic group model.*

Roughly speaking, a generic group model adversary can only learn information about the obfuscated pat by taking linear combinations of the encoded elements and testing whether the result is g^0. In the dual scheme, this means that the adversary can learn whether or not $g^{k^\top \cdot (B \cdot e)} = g^0$ for arbitrary $k \in \mathbb{F}_q^{n+1}$.

We argue that for any k the adversary tries, it will fail to get $k^\top \cdot (B \cdot e) = 0$. Intuitively, this means the model leaks no information; with a bit more generic group formalism, we can show this implies distributional VBB security. To show this, we note that if there exists a position $j \in [2n]$ where $(k^\top \cdot B)_j \neq 0$ and $e_j \neq 0$, the scalar $k^\top \cdot (B \cdot e) = 0$ is a sum containing a uniformly random element and will be non-zero with overwhelming probability.

Since the adversary does not know where the non-zero entries of $e \in \mathbb{F}_q^{2n}$ are, a natural attack strategy would be to find k such that $k^\top \cdot B$ has many zeros. However, any choice of $n+1$ columns of B are linearly independent, and so for any k, $k^\top \cdot B \in \mathbb{F}_q^{2n}$ has at least n non-zero elements.

The remainder of the proof is a purely combinatorial argument. A uniformly random pattern with $c(n) \coloneqq n - w$ *non-wildcard* bits will produce an e with $c(n)$ random non-zero elements. The only requirement on the positions of these non-zero elements is that in any index pair $(2i-1, 2i)$ for $i \in [n]$, at most one of $\{e_{2i-1}, e_{2i}\}$ can be non-zero (if both are non-zero, the pattern will not match either of $x_i = 0$ or $x_i = 1$).

The n non-zero positions of $k^\top \cdot B$ must inhabit at least $n/2$ of these index pairs. In expectation, at least $c(n)/2$ of these $n/2$ index pairs $(2i-1, 2i)$ *also* correspond to indices where either e_{2i-1} or e_{2i} is a random non-zero value. A Chernoff bound argument proves this value must be close to its expectation; concretely, with overwhelming probability at least $c(n)/8$ index pairs correspond to both a non-zero position of $k^\top \cdot B$ and a non-zero position of e.[2]

In each of these $c(n)/8$ pairs, we know at least one of $(k^\top \cdot B)_{2i-1}$ or $(k^\top \cdot B)_{2i}$ is non-zero, and at least one of e_{2i-1} or e_{2i} is uniformly random. Since the bits of the pattern are picked uniformly at random, with probability at least $1/2$ a non-zero position of $(k^\top \cdot B)$ and a uniformly random

[2] In our full proof, we have to be slightly careful applying a Chernoff bound since the associated random variables are not independent.

position of e coincide. To have any non-negligible chance of getting 0, the adversary must avoid this scenario in each of these $c(n)/8$ pairs, which happens with probability at most $1/2^{c(n)/8}$. For $c(n) = \omega(\log n)$, this is negligible.

We can also use similar arguments (minus the Chernoff bound analysis) to obtain security for distributions satisfying a more general min-entropy requirement. In particular, we consider distributions over patterns with some fixed number of wildcards w. For any w, our argument gives a lower bound b_w such that obfuscation for a pattern drawn from any distribution over patterns with w wildcards and min-entropy at least b_w is secure in the generic group model. For any $w \leq 0.75n$, the bound b_w is less than the min-entropy of the uniform distribution with w wildcards. However, when w gets too large this is no longer the case and the bound becomes meaningless.

Conjunctions over Large Alphabets. If we go beyond binary alphabets, the dual scheme actually reduces the obfuscation size by far more than a factor of 2. Suppose the alphabet is $[\ell]$ for some integer ℓ, so a conjunction is specified by a length n pattern $\mathsf{pat} \in \{[\ell] \cup \{*\}\}^n$. $f_{\mathsf{pat}}(x) = 1$ only if $x_i = \mathsf{pat}_i$ on all non-wildcard positions.

There is a natural generalization of the $A \cdot s + e$ scheme. For an alphabet of size ℓ, we partition e into the n blocks of length ℓ, corresponding the ith block with the ith pattern position. As in the binary case, if $\mathsf{pat}_i = *$, we set every entry of e in the ith block to 0. If $\mathsf{pat}_i = j$ for $j \in [\ell]$, we set the jth position in the ith block of e to a uniformly random value in $\mathbb{F}_q$, and set the remaining $\ell - 1$ entries in the ith block to 0. To evaluate on $x \in [\ell]^n$, we solve for $v^\top \cdot A = 0$ where v is restricted to be non-zero only at $v_{(i-1)\ell + x_i}$ for each $i \in [n]$. It is easy to verify correctness of evaluation.[3] However, this scheme is fundamentally stuck at polynomial-size alphabets, since $A \cdot s + e$ contains $n\ell$ elements.

If we switch to the dual view, this same scheme can be implemented as $g^{B \cdot e}$ where $B \in \mathbb{F}_q^{(n+1) \times n\ell}, e \in \mathbb{F}_q^{n\ell}$. But the number of group elements in $g^{B \cdot e}$ is simply $n + 1$, which has *no* dependence on the alphabet size. Of course B will have dimension $(n+1) \times n\ell$, but by once again choosing B to be a Vandermonde style matrix, we can demonstrate that neither the evaluator nor the obfuscator ever have to store B or e in entirety, since e is sparse for large ℓ. We simply need q to grow with $\log \ell$ to ensure this implicit B satisfies the rank conditions needed for security.

1.2.3 Moving Out of the Exponent

Returning to the $A \cdot s + e$ view of the scheme for a moment, we see that its form begs an interesting question:

Can the Vandermonde matrix be replaced with other matrices?

In [BKM+18], the Vandermonde matrix plays at least two crucial roles: it allows for evaluation by polynomial interpolation and at the same time is vital for their security analysis. However, the structure of the Vandermonde matrix is what leads to Reed-Solomon decoding attacks on the plain scheme, necessitating encoding the values in a cryptographic group. Furthermore, observe that our abstract evaluation procedure described in Section 1.2.1 made no reference to the specific *structure* of A; in particular, it works for *any* public matrix A. In the case of the Vandermonde matrix, applying this abstract procedure results in the Lagrange coefficients [BKM+18] use, but we can easily perform evaluation for other matrices.

[3] We note that if we set $\ell = 2$, this generalization flips the role of 0 and 1, but is functionally equivalent.

Furthermore, the matrix form of the scheme is strongly reminiscent of the Learning Parity with Noise (LPN) problem and in particular its extension to $\mathbb{F}_q$, known as the Random Linear Codes (RLC) problem [IPS09].

We recall the form of the RLC problem over $\mathbb{F}_q$ for noise rate ρ and n^c samples. Here, we have a uniformly random matrix $A \leftarrow \mathbb{F}_q^{n^c \times n}$, a uniformly random column vector $s \in \mathbb{F}_q^n$, and an error vector $e \in \mathbb{F}_q^{n^c}$ generated as follows. For each $i \in [n^c]$, set $e_i = 0$ with independent probability $1-\rho$, and otherwise draw $e_i \leftarrow \mathbb{F}_q$ uniformly at random. The search version of this problem is to recover the secret vector s given $(A, A \cdot s + e)$, and the decision version is to is to distinguish $(A, A \cdot s + e)$ from (A, v) for uniformly random $v \leftarrow \mathbb{F}_q^{n^c}$. The standard search RLC and decisional RLC assumptions are that these problems are intractable for any computationally bounded adversary for constant noise rate $0 < \rho < 1$.

This suggests the following approach to obtaining a secure obfuscation scheme from the original scheme: simply replace A with a *random matrix*. A would be publicly output along with $A \cdot s + e$. To prove security, the hope would be that we could invoke the RLC assumption to show that even given A, the obfuscation $A \cdot s + e$ is computationally indistinguishable from a vector v of $2n$ random elements.

Structured Error Distributions. At first glance, there are some difficulties with this approach. We can immediately see a mismatch between the error vector in the RLC problem and the error vector in our matrix scheme. To begin with, the entries of the RLC error vector e are chosen so that each entry is non-zero with independent random probability ρ. In our matrix view, the number of non-zero terms in the error vector is exactly equal to the number of fixed bits, which is $n - w$. Setting $\rho = (n-w)/n$ does not quite simulate the correct error distribution, since the actual number of non-zero terms in the error is unlikely to be exactly $n - w$, as needed for our distribution. Fortunately, this issue can be resolved by considering a decisional "exact RLC" problem (decisional xRLC for short) in which the number of non-zero error terms is fixed to be ρn. This is completely analogous to the "decisional exact LPN" problem (decisional xLPN) considered by Jain, Krenn, Pietrzak and Tentes [JKPT12].

However, even this decisional xRLC problem is insufficient to argue security. The problem lies in the fact that the error vector in our setting is *structured*: for any pair of positions e_{2i-1}, e_{2i} for $i \in [n]$, the construction ensures that at least one of e_{2i-1} or e_{2i} is 0. Recall that if the ith bit of the pattern is b, then $e_{2i-b} = 0$ while $e_{2i-(1-b)}$ is drawn randomly from $\mathbb{F}_q$. If the ith bit of the pattern is $*$, then $e_{2i-1} = e_{2i} = 0$. But if both e_{2i-1} and e_{2i} are random elements from $\mathbb{F}_q$, this corresponds to a position where the input string can be neither 0 nor 1, which can never arise in the obfuscation construction.

To the best of our knowledge, the only work that considers this particular structured error distribution is the work of Arora and Ge [AG11], which shows that this problem is actually *insecure* in the binary case (corresponding to a structured error version of LPN). Their attack uses re-linearization and it is easy to see that it extends to break the problem we would like to assume hard as long as A has $\Omega(n^2)$ rows.

This leaves some hope for security, as our construction only requires that A have $2n$ rows. Thus, one of the core technical components of this work is a reduction that proves hardness of the structured error RLC assumption with $2n$ samples from the hardness of the standard RLC assumption for polynomially many samples. We note that our reductions handle both the search and decision variants, both the exact and non-exact variants, and both LPN and RLC. We give a

high-level overview of our reduction below.

1.2.4 The Reduction to Structured Error

For our reduction, we return to the $B \cdot e$ view of the scheme and consider the equivalent "dual" version of the decisional RLC problem[4], where the goal is to distinguish (B, e) from (B, v) for $B \leftarrow \mathbb{F}_q^{(n^c-n)\times n^c}$, $v \leftarrow \mathbb{F}_q^{n^c-n}$, and e as drawn previously.

Note that the problem of distinguishing between $(B, B \cdot e)$ and (B, u) for $n^c - n$ samples and error vector e of dimension n^c is equivalent to the setting where the number of samples is $n - n^{1/c}$ and the error vector is of dimension n. Since this problem is conjectured hard for any constant c, we set $\epsilon = 1/c$ and assume hardness for any $0 < \epsilon < 1$.

We show how to turn an instance of this problem into a structured error RLC instance, where the challenge is to distinguish between $(B, B \cdot e)$ and (B, u) for uniformly random $B \leftarrow \mathbb{F}_q^{(n+1)\times 2n}$, a *structured* error vector $e \in \mathbb{F}_q^{2n}$ with noise rate ρ, and uniformly random $u \in \mathbb{F}_q^{n+1}$.

To perform this transformation, we need to somehow inject the necessary structure into the error vector e, which means introducing a zero element in each pair. The most natural way to do this given the regular RLC instance $(B, B \cdot e)$ is to draw n new uniformly random columns and insert them into B at random locations to produce the matrix B', without changing the value of $B \cdot e$. We now have a structured error instance with dimension $(n - n^\epsilon) \times 2n$, which appears very close to our goal, but not quite there as we need B to have $n + 1$ rows. We would like to simply add n^ϵ uniformly random rows to B. Unfortunately, this appears impossible, as it is not clear how to simulate the extra entries of $B' \cdot e$ without knowledge of e.

Instead, we add rows $b_i^\top$ that are *statistically* close to uniformly random but are such that we know the value of $b_i^\top \cdot e$. Observe that we already know $n - n^\epsilon$ equations over the elements of e. We can generate new equations by taking random linear combinations of these; however, the resulting coefficient vectors $b_i^\top$ will certainly not be statistically close to uniform since they will be in the row space of public matrix B. The final observation is that the reduction algorithm actually knows the location of n elements of e that are definitely zero (since they were chosen by the algorithm itself). We can then replace the elements at the corresponding locations of each $b_i^\top$ with uniformly random elements without changing the product $b_i^\top \cdot e$. This gives us enough entropy on the $b_i^\top$'s to show that they are statistically close to uniform via a collision probability argument.

1.2.5 Distributional VBB Security

The above reduction implies our "random linear code" obfuscation scheme (outside of any group exponent) is computationally indistinguishable from random. However, this does not quite give distributional VBB security, as the definition requires indistinguishability from a simulated obfuscation even given any one bit predicate on the circuit. In fact, indistinguishability from random does not necessarily provide *any* security at all. Consider for example the distributional point obfuscator that simply outputs the single accepting point in the clear as the "obfuscation." To evaluate, we simply compare the input point with the accepting point. Notice this trivially insecure obfuscation is perfectly indistinguishable from random for point functions drawn from the uniform distribution.

[4] In the context of LWE this duality/transformation has been observed a number of times, see e.g. [MM11]. For RLC, this is essentially syndrome decoding.

In order to achieve meaningful distributional VBB security, we need to ensure that no predicate on the hidden circuit is leaked by the obfuscation (note this property is not satisfied by the above example, which leaks the hidden circuit entirely). For us, it will suffice to prove that our obfuscation $B \cdot e$ is indistinguishable from random even if the adversary knows a one bit predicate on the pattern hidden within e. Mapping this back to RLC, what we want is a version of the above decisional xRLC problem that is still hard even if the adversary is given a predicate on the error vector e.

A natural attempt to establishing pseudorandomness in this setting is to consider the corresponding (dual) search xRLC problem, where the goal is to find e given $B, e, \mathcal{P}(e)$ for some one bit predicate $\mathcal{P}$. This is clearly as hard as the search problem without the predicate since a reduction algorithm can simply guess the value of the predicate and be correct with probability at least $1/2$. Since the search problem remains hard under leakage of a one bit predicate, we can try to show hardness of the decisional version with a compatible search-to-decision reduction.[5]

Unfortunately, as pointed out in [IPS09], search to decision reductions are only known for the (regular) RLC problem over polynomially large fields; we would ideally like to make use of exponentially large fields to keep the same correctness guarantees that we had for the scheme encoded in a group. We instead restrict our construction to the binary field and make use of the LPN assumption to establish distributional VBB security. For LPN, sample-preserving search to decision reductions are known [AIK09] and, combined with the Goldreich-Levin theorem, give the necessary search to decision reduction for LPN with a one bit predicate [Döt16].

However, now that we are restricted to a binary field, we have to slightly tweak the scheme in order to maintain correctness. It is straightforward to achieve (weak) functionality preservation by adding a few rows to B, and we explain how to achieve the stronger form (where the obfuscated circuit maintains functionality on all inputs simultaneously) with an additional tweak. We note that our modified LPN-based scheme achieving strong functionality preservation achieves *expected* polynomial evaluation time. We leave the question of whether or not one can simultaneously achieve deterministic polynomial evaluation time and strong functionality preservation from this LPN approach unresolved. Finally, using the reduction from standard LPN to structured error LPN, we prove the distributional VBB security of our construction assuming the standard constant-noise LPN assumption.

1.2.6 Information Theoretic Security

Bishop et al. [BKM+18] motivate the design of their scheme by explaining how an even simpler idea seems to fail. They informally give the following scheme for point obfuscation. To accept on input $x = \texttt{0101}$, we simply draw uniformly random elements from $\mathbb{F}_q$ conditioned on the elements in grey cells corresponding to `0101` summing to 0.

r_1	r_3	r_5	r_7
r_2	r_4	r_6	r_8

This can easily be done by setting $r_7 = -r_2 - r_3 - r_6$. While [BKM+18] do not explore this idea further, distributional virtual black-box security of this scheme follows from the leftover hash lemma, and this scheme is actually *statistically* secure.[6]

[5] We thank Daniel Wichs for pointing us in this direction; see full acknowledgement details in Section 6.

[6] We remark that this scheme does not actually satisfy strong functionality preservation, which may preclude its use in certain settings.

However, [BKM+18] point out that the moment we have two gray cells in the same column, this fails. If we want to extend this idea to 01*1, we have no choice but to set $r_5 = r_6$ if we want to preserve functionality. Then the scheme is trivially insecure, since a wildcard position will correspond to a column with two repeated elements.

This barrier appears inherent if we are limited to evaluating the scheme by simply summing a set of elements in $\mathbb{F}_q$ and checking if the result is 0. But what if we use matrices in $\mathbb{F}_q$ instead of scalar elements? Evaluation could now involve checking the *rank* of the resulting matrix sum.

Our construction takes the following form. An obfuscation of a pattern $\mathsf{pat} \in \{0,1,*\}^n$ consists of a single full rank matrix $F \in \mathbb{F}_q^{k \times k}$, and n rank 1 matrices $A^{(i)} \in \mathbb{F}_q^{k \times k}$ for $i \in [n]$. To evaluate on an n bit string x, we simply compute the sum

$$F + \sum_{i|x_i=1} A^{(i)}.$$

We choose F and $A^{(i)}$ so that if the determinant of this sum is 0, then x must have matched pat.

Observe that if F and $\{A^{(i)}\}_{i\in[n]}$ were scalars, the $A^{(i)}$ terms corresponding to wildcard positions i would have to be 0, since the sum must remain unchanged whether or not those $A^{(i)}$ are included. But since these are matrices, we can set the $A^{(i)}$ corresponding to wildcard positions to be in the *column span* of the matrix

$$F + \sum_{i|\mathsf{pat}_i=1} A^{(i)}.$$

Concretely, the construction is the following. To obfuscate pat, we first sample a uniformly random rank $k-1$ matrix B. Then for each i where $\mathsf{pat}_i = *$, we sample a random rank one matrix $A^{(i)}$ in the column span of B. Finally, we sample random rank one matrices $A^{(i)}$ for all positions where $\mathsf{pat}_i = 0$ or 1, and give out $F = B - \sum_{i|\mathsf{pat}_i=1} A^{(i)}$. With overwhelming probability, the resulting F will be full rank.[7]

We prove security of this scheme by applying the leftover hash lemma (LHL), which shows that as long as the non-wildcard bits of pat have sufficient min-entropy, the matrix F is statistically close to a uniformly random matrix. Then the rank deficient matrix B is statistically hidden from view, so if there are fewer than k wildcards, the $A^{(i)}$ matrices are distributed as uniformly random rank 1 matrices.[8]

The number of wildcards this scheme can handle is $k-1$ where k is the dimension of the matrices we use. The limitation on k arises in our LHL analysis, which only works for k as large as n^δ (for any $\delta < 1$), so we obtain statistical security for patterns with a sublinear number of wildcards.

1.2.7 Functionality Preservation Notions

Statistical security arguments such as the one above can only hold for schemes that fall short of strong functionality preservation (i.e. except with negligible probability, correctness holds on all inputs simultaneously). If we do not relax correctness, statistical virtual black box security is

[7] A reader familiar with the Learning Subspace with Noise (LSN) problem introduced by Dodis, Kalai, and Lovett [DKL09] might notice similarities. However, if we map our scheme to their setting, we obtain an LSN/LPN instance where the number of samples is so restricted that information theoretic security is actually possible.

[8] In the actual security proof, we need to be a bit more careful, since again, a proof that our scheme is indistinguishable from random does *not* imply distributional VBB security.

impossible since a computationally unbounded adversary can recover pat from the truth table of the obfuscated function.

Thus, our basic sum-of-matrices scheme achieves "weak functionality preservation" (considered in [GR07, BR17, BKM+18]), which requires that for any pattern pat and any input x, correctness (i.e., $\mathcal{O}(f_{\mathsf{pat}})(x) = f_{\mathsf{pat}}(x)$) holds with overwhelming probability over the randomness of the obfuscation.

A Motivating Scenario from [WZ17]. A natural question to ask is whether weak functionality preservation is "good enough." To shed light on this, we take a step back and recall a motivating example for general evasive circuit obfuscation. Even this might not be immediately obvious: *what good is an obfuscated circuit if a user can never find an accepting input?* Wichs and Zirdelis [WZ17] address precisely this question with the following scenario. Suppose we have a set of users where a subset of them has access to additional privileged information. If we publicly give out an obfuscated circuit containing this privileged information, then security assures us that the un-privileged users cannot find accepting inputs. For them, functionality preservation is unimportant since the circuit may as well be the all 0's circuit.[9] However, it does matter for the privileged users who may actually find accepting inputs (for these users, security does not hold).

In this example, a secure obfuscation that only achieves weak functionality preservation is good enough to ensure the un-privileged users never learn anything about the hidden circuit. However, it might not be enough to ensure the privileged users are actually given the correct circuit. Weak functionality preservation does not explicitly rule out the possibility that a user with privileged information can detect that the obfuscated circuit functionality differs from the intended circuit functionality.

Computational Functionality Preservation. To address this gap, we introduce a new notion (between weak and strong) we call *computational functionality preservation*. To the best of our knowledge, this notion has not explicitly been used in the context of obfuscation, but it is essentially the same definition considered by Brakerski and Vaikuntanathan [BV15] for constrained PRFs. For us, computational functionality preservation guarantees that even a user who knows the real circuit (in this work, "real circuit" means the obfuscated pattern) cannot find a point x on which the obfuscated circuit and the real circuit differ, provided they are computationally bounded.

In Section 5.3, we describe a simple modification of our basic sum-of-matrices scheme that allows us to achieve computational functionality preservation from discrete log. We note that the resulting construction is still information theoretically secure. Mapping this to the above example, this means even computationally unbounded un-privileged users cannot learn any predicate on the hidden pattern. This is only possible because our obfuscated circuit computes the *wrong output* on exponentially many inputs. Despite this, a computationally bounded user (who might even know the hidden pattern) cannot even find one of these incorrect inputs assuming discrete log.

While it might at first seem strange to simultaneously make statistical and computational claims, this makes sense in the setting where we have two distinct classes of users: the un-privileged users are modeled as computationally unbounded, but the privileged users as computationally bounded.

[9] This is slightly informal, since it requires a notion of input-hiding obfuscation [BBC+14].

1.3 Related Work

Conjunction Obfuscation. Previously, Brakerski and Rothblum had shown how to obfuscate conjunctions using multilinear maps [BR13]. This was followed by a work of Brakerski et al. which showed how to obfuscate conjunctions under entropic ring LWE [BVWW16]. More recently, Wichs and Zirdelis showed how to obfuscate compute-and-compare programs under LWE [WZ17]. Goyal, Koppula, and Waters concurrently and independently introduced lockable obfuscation and proved security under LWE [GKW17]. Both of these works easily imply secure obfuscation of conjunctions under LWE, though with a complicated construction that encodes branching programs in a manner reminiscent of the GGH15 multilinear map [GGH15]. The main contribution of [BKM+18] then was the simplicity and efficiency of their conjunction obfuscation scheme. In this work, we provide constructions and proofs that maintain these strengths while addressing the major weaknesses of the [BKM+18] construction — lack of generality (to more wildcards, more distributions, and more alphabet sizes) and lack of security based on a falsifiable assumption.

LWE vs LPN. Given the similarities between LPN and LWE (LWE can be framed as a generalization of LPN), it might seem that our conjunction obfuscation result was essentially known as a consequence of lockable obfuscation [WZ17, GKW17]. To address this point, we briefly recall some of the deeper qualitative differences between LPN-based and LWE-based cryptography.

Over the past decade, LWE has been shown to imply a rich class of powerful cryptographic tools such as fully homomorphic encryption [Gen09], inner-product encryption [AFV11], and private constrained PRFs [CC17, BTVW17, PS18], just to name a few. In comparison, the applications of LPN have been somewhat limited. Applebaum, Avron, and Brzuska [AAB15] investigate this discrepancy and observe that LPN-based constructions tend to admit natural "arithmetic" generalizations. They demonstrate that arithmetic constructions are provably limited in many settings where LWE is not, suggesting that the limitations of LPN-based cryptography might be inherent.

The current state of affairs in obfuscation is no exception. Prior to this work, extremely little was known from LPN; Yu and Zhang constructed a point obfuscator from sub-exponentially hard LPN [YZ16], but to the best of our knowledge this is the only obfuscation construction for any non-trivial functionality known directly from LPN.[10] On the other hand, lockable obfuscation [WZ17, GKW17] allows us to obfuscate a large and expressive class of circuits under LWE.

In this work, we demonstrate that conjunction obfuscation is a problem where LPN-based obfuscation is not only possible, but actually provides a more natural and intuitive solution than the known LWE approach. The particular structure of our conjunction obfuscation schemes corresponds precisely to a structured LPN error vector, allowing us to avoid a construction relying on more technical machinery (such as branching programs and lattice trapdoors, as required in lockable obfuscation).

Reader's Guide In Section 2, we briefly recall security notions for evasive circuit obfuscation, the definition of the generic group model, and the LPN problem. After that point, Section 3, Section 4, and Section 5 are self-contained and can be read in any order, though we recommend familiarity with the introduction. All of our generic group constructions and security arguments are confined to Section 3, Appendix B, and Appendix C. In Section 4, we formally define the

[10] Yu and Zhang [YZ16] improve upon Dodis, Kalai, and Lovett [DKL09], who constructed a point obfuscator from a related Learning Subspace with Noise (LSN) problem.

LPN error distributions we consider, and proceed to prove security of our constructions assuming hardness of decisional "structured-error" exact LPN. We prove that computational intractibility of this problem follows from the standard constant-noise LPN assumption in Appendix A. This section is written more generally to handle RLC, as we believe the reduction might be of independent interest (restricting the field size to $q = 2$ gives the LPN result needed for our security proof in Section 4). Our sum-of-matrices construction achieving information theoretic VBB security is in Section 5, plus an extension that achieves computational functionality preservation.

2 Preliminaries

Notation. Let $\mathbb{Z}, \mathbb{N}$ be the set of integers and positive integers. For $n \in \mathbb{N}$, we let $[n]$ denote the set $\{1, \ldots, n\}$. For $q \in \mathbb{N}$, denote $\mathbb{Z}/q\mathbb{Z}$ by $\mathbb{Z}_q$, and denote the finite field of order q by $\mathbb{F}_q$. A vector v in $\mathbb{F}_q$ (represented in column form by default) is written as a lower-case letter and its coefficients $v_i \in \mathbb{F}_q$ are indexed by i; a matrix A is written as a capital letter and its columns $(A)_j$ are indexed by j. We denote by $0^{n\times m}$ the (n, m)-dimensional matrix filled with zeros. For any matrix M, let colspan(M) denote the column span of M.

If D is a distribution, we denote $\mathsf{Supp}(D) = \{x : D(x) \neq 0\}$ its support. For a set S of finite weight, we let $U(S)$ denote the uniform distribution on S. The statistical distance between two distributions D_1 and D_2 over a countable support S is $\Delta(D_1, D_2) \coloneqq \frac{1}{2}\sum_{x\in S} |D_1(x) - D_2(x)|$. We naturally extend those definitions to random variables.

We use the usual Landau notations. A function $f(n)$ is said to be negligible if it is $n^{-\omega(1)}$ and we denote it by $f(n) \coloneqq \mathsf{negl}(n)$. A probability $p(n)$ is said to be overwhelming if it is $1 - n^{-\omega(1)}$.

Let $\epsilon > 0$. We say that two distributions D_1 and D_2 are ϵ-statistically close if $\Delta(D_1, D_2) \leq \epsilon$. We say that D_1 and D_2 are statistically close, and denote $D_1 \approx_s D_2$, if there exists a negligible function ϵ such that D_1 and D_2 are $\epsilon(n)$-statistically close.

The distinguishing advantage of an algorithm $\mathcal{A}$ between two distributions D_0 and D_1 is defined as $\mathsf{Adv}_{\mathcal{A}}(D_0, D_1) \coloneqq |\Pr_{x\leftarrow D_0}[\mathcal{A}(x) = 1] - \Pr_{x\leftarrow D_1}[\mathcal{A}(x) = 1]|$, where the probabilities are taken over the randomness of the input x and the internal randomness of $\mathcal{A}$. We say that D_1 and D_2 are computationally indistinguishable, and denote $D_1 \approx_c D_2$, if for any non-uniform probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, there exists a negligible function ϵ such that $\mathsf{Adv}_{\mathcal{A}} = \epsilon(n)$.

Finally, we let $x \leftarrow X$ denote drawing x *uniformly at random* from the space X, and define $\mathcal{U}_{n,w}$ to be the uniform distribution over $\{0, 1, *\}^n$ with a fixed w number of $*$ (wildcard) characters.

The min-entropy of a random variable X is $H_\infty(X) \coloneqq -\log(\max_x \Pr[X = x])$. The (average) conditional min-entropy of a random variable X conditioned on a correlated variable Y, denoted as $H_\infty(X|Y)$, is defined by

$$H_\infty(X|Y) \coloneqq -\log\left(\mathbb{E}_{y\leftarrow Y}\left[\max_x \Pr[X = x|Y = y]\right]\right).$$

We recall the leftover hash lemma below.

Lemma 1 (Leftover hash lemma)**.** *Let $\mathcal{H} = \{h\colon \mathcal{X} \to \mathcal{Y}\}$ be a 2-universal hash function family. Then for any random variable $X \in \mathcal{X}$ and Z, for $\epsilon > 0$ such that $\log(|\mathcal{Y}|) \leq H_\infty(X|Z) - 2\log(1/\epsilon)$, the distributions $(h, h(X), Z)$ and $(h, U(\mathcal{Y}), Z)$ are ϵ-statistically close.*

2.1 Security Notions for Evasive Circuit Obfuscation

We recall the definition of a distributional virtual black-box (VBB) obfuscator. We roughly follow the definition of Brakerski and Rothblum [BR13], but we include a new computational functionality preservation definition.

Definition 1 (Distributional VBB Obfuscation)**.** *Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a family of polynomial-size circuits, where $\mathcal{C}_n$ is a set of boolean circuits operating on inputs of length n, and let Obf be a PPT algorithm which takes as input an input length $n \in \mathbb{N}$ and a circuit $C \in \mathcal{C}_n$ and outputs a boolean circuit $\mathsf{Obf}(C)$ (not necessarily in $\mathcal{C}$). Let $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be an ensemble of distribution families $\mathcal{D}_n$ where each $D \in \mathcal{D}_n$ is a distribution over $\mathcal{C}_n$.*

Obf is a distributional VBB obfuscator for the distribution class $\mathcal{D}$ over the circuit family $\mathcal{C}$ if it has the following properties:

1. *Functionality Preservation: We give three variants:*
   - *(Weak) Functionality Preservation: For every $n \in \mathbb{N}$, $C \in \mathcal{C}_n$, and $x \in \{0,1\}^n$, there exists a negligible function μ such that*
   $$\Pr[\mathsf{Obf}(C, 1^n)(x) = C(x)] = 1 - \mu(n)\,.$$
   - *(Computational) Functionality Preservation: For every PPT adversary $\mathcal{A}$, $n \in \mathbb{N}$, and $C \in \mathcal{C}_n$, there exists a negligible function μ such that*
   $$\Pr[x \leftarrow \mathcal{A}(C, \mathsf{Obf}(C, 1^n)) : C(x) \neq \mathsf{Obf}(C, 1^n)(x)] = \mu(n)\,.$$
   - *(Strong) Functionality Preservation: For every $n \in \mathbb{N}$, $C \in \mathcal{C}_n$, there exists a negligible function μ such that*
   $$\Pr[\mathsf{Obf}(C, 1^n)(x) = C(x)\ \forall x \in \{0,1\}^n] = 1 - \mu(n)\,.$$
2. *Polynomial Slowdown: For every $n \in \mathbb{N}$ and $C \in \mathcal{C}_n$, the evaluation of $\mathsf{Obf}(C, 1^n)$ can be performed in time $\mathrm{poly}(|C|, n)$.*
3. *Distributional Virtual Black-Box: For every PPT adversary $\mathcal{A}$, there exists a (non-uniform) polynomial size simulator $\mathcal{S}$ such that for every $n \in \mathbb{N}$, every distribution $D \in \mathcal{D}_n$ (a distribution over $\mathcal{C}_n$), and every predicate $\mathcal{P} : \mathcal{C}_n \to \{0,1\}$, there exists a negligible function μ such that*
   $$\left| \Pr_{C\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathsf{Obf}(C, 1^n)) = \mathcal{P}(C)] - \Pr_{C\leftarrow\mathcal{D}_n}[\mathcal{S}^C(1^{|C|}, 1^n) = \mathcal{P}(C)] \right| = \mu(n)\,.$$

Both weak functionality preservation [GR07, BR17, BKM+18] and strong functionality preservation [BGI+01] have been considered numerous times before in the obfuscation literature.

To the best of our knowledge, the above computational functionality preservation definition has not specifically appeared in the literature in the context of obfuscation, although it is essentially the same as the functionality preservation notion considered in Definition 3.1 of [BV15] in the context of constrained PRFs. We motivate and discuss this definition in Section 1.2.7, and demonstrate an obfuscation scheme achieving it in Section 5.3.

We now recall the definition of *perfect-circuit hiding*, introduced by Barak, Bitansky, Canetti, Kalai, Paneth, and Sahai [BBC+14].

Definition 2 (Perfect Circuit-Hiding [BBC+14])**.** *Let $\mathcal{C}$ be a collection of circuits. An obfuscator Obf for a circuit collection $\mathcal{C}$ is perfect circuit-hiding if for every PPT adversary $\mathcal{A}$ there exists a negligible function μ such that for every balanced predicate $\mathcal{P}$, every $n \in \mathbb{N}$ and every auxiliary input $z \in \{0,1\}^{\mathsf{poly}(n)}$ to $\mathcal{A}$:*

$$\Pr_{C \leftarrow \mathcal{C}_n}[\mathcal{A}(z, \mathsf{Obf}(C)) = \mathcal{P}(C)] \leq \frac{1}{2} + \mu(n)\,,$$

where the probability is also over the randomness of Obf.

Barak et al. [BBC+14] prove that perfect-circuit hiding security is equivalent to distributional virtual black-box security, i.e. property 3 in Definition 1 is equivalent to Definition 2. We rely on this equivalence to simplify the proof of Theorem 6.

2.2 The Generic Group Model

We analyze one of the presented obfuscation constructions in the generic group model [Sho97], which assumes that the adversary interacts with the group elements that comprise the scheme in a *generic* way. To model this, it is common to associate each group element with an independent and uniformly random string (drawn from a sufficiently large space) with we refer to as a "handle." The adversary has access to a generic group oracle which maintains the mapping between group elements and handles. The adversary is initialized with the handles corresponding to the group elements that comprise the scheme in question. It can query its generic group oracle with two handles, after which the oracle performs the group operation on the associated group elements and returns the handle associated with the resulting group element.

It will be convenient to associate each of these group operation queries performed by the adversary to a linear combination over the initial handles that it receives. The adversary can also request a "ZeroTest" operation on a handle, to which the oracle replies with a bit indicating whether or not that handle is associated with the identity element of the group. See [BBG05] for examples and more details.

There is a natural extension of the notion of distributional VBB security to the generic group model. In Definition 1, we simply give the obfuscation Obf and adversary $\mathcal{A}$ access to the generic group oracle $\mathcal{G}$. We refer to this definition as *Distributional VBB Obfuscation in the Generic Group Model*.

2.3 Learning Parity with Noise

We give the precise definition of the Learning Parity with Noise (LPN) problem in its dual formulation. Let $\rho > 0$ and m be an integer. Let $\mathcal{B}_\rho^m$ denote the distribution on $\mathbb{F}_2^m$ for which each component of the output independently takes the value 1 with probability ρ and 0 with probability $1 - \rho$ (i.e., each component is sampled according to the Bernoulli distribution with parameter ρ).

Definition 3. *Let n, m be integers and $\rho \in [0,1]$. The Decisional Learning Parity with Noise (DLPN) problem with parameters n, m, ρ, denoted $\mathsf{DLPN}(n, m, \rho)$, is hard if, for every probabilistic polynomial-time (in n) algorithm $\mathcal{A}$, there exists a negligible function μ such that*

$$\left| \Pr_{B,e}[\mathcal{A}(B, B \cdot e) = 1] - \Pr_{B,u}[\mathcal{A}(B, B \cdot u) = 1 \right| \leq \mu(n),$$

where $B \leftarrow \mathbb{F}_2^{(m-n)\times m}$, $e \leftarrow \mathcal{B}_\rho^m$, and $u \leftarrow \mathbb{F}_2^{m-n}$.

Remark 1. *The primal version of the above problem is, for* $A \leftarrow \mathbb{F}_2^{m\times n}, s \leftarrow \mathbb{F}_2^n, e \leftarrow \mathcal{B}_\rho^m$*, and* $v \leftarrow \mathbb{F}_2^m$*, to distinguish between* $(A, As+e)$ *and* (A, v)*. These problems are equivalent for* any *error distribution when* $m = n + \omega(\log n)$*, as discussed for example in [MM11, Sec. 4.2].*

3 Obfuscating Conjunctions in the Generic Group Model

In this section, we present our generalized dual scheme for obfuscating conjunctions in the generic group model. We then show a simple proof of security in the generic group model that applies to the uniform distribution over binary patterns with *any* fixed number of wildcards. In particular, our distributional VBB security result holds for up to $n - \omega(\log n)$ wildcards, but distributional VBB security is vacuously satisfied for $w > n - O(\log n)$ wildcards. This extends the generic model analysis of [BKM+18] that proved security up to $w < .774n$. We note that the combinatorial argument we give can be used to show that the original [BKM+18] construction achieves security for all values of w as well.

We then extend our security analysis to handle more general distributions with sufficient min-entropy, which had not been considered in [BKM+18]. We then extend our scheme to handle exponential size alphabets, and demonstrate other efficiency improvements resulting from our "dual" scheme.

Here and throughout the remainder of paper, the length n of the pattern will double as the security parameter.

3.1 Generic Group Construction

Throughout this section, we will refer to a fixed matrix B.

Definition 4. *Let* $B_{n+1,k,q} \in \mathbb{Z}_q^{(n+1)\times k}$ *be the matrix whose* (i,j)*th entry is* j^i*:*

$$B_{n+1,k,q} = \begin{pmatrix} 1 & 2 & \dots & k \\ 1 & 2^2 & \dots & k^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 2^{n+1} & \dots & k^{n+1} \end{pmatrix}.$$

Construction.

- $\mathsf{Setup}(n)$. Let $\mathbb{G}$ be a group of prime order $q > 2^n$ with generator g. We let $B := B_{n+1,2n,q}$ where $B_{n+1,2n,q}$ is as in Definition 4.
- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n)$. Set $e \in \mathbb{Z}_q^{2n\times 1}$ as follows. For each $i \in [n]$:
  - If $\mathsf{pat}_i = *$, set $e_{2i-1} = e_{2i} = 0$.
  - If $\mathsf{pat}_i = b$, sample $e_{2i-b} \leftarrow \mathbb{Z}_q$ and set $e_{2i-(1-b)} = 0$.

  Output
  $$g^{B\cdot e} \in \mathbb{G}^{n+1}.$$

- $\mathsf{Eval}(v \in \mathbb{G}^{n+1}, x \in \{0,1\}^n)$. Define B_x to be the $(n+1) \times n$ matrix where column j is set as

$$(B_x)_j := (B)_{2j-x_j}.$$

Solve $tB_x = 0$ for a non-zero $t \in \mathbb{Z}_q^{1\times(n+1)}$ (see Appendix B for a description of how to do this in $O(n \log^2(n))$ time). Compute

$$\prod_{i=1}^{n+1} v_i^{t_i}$$

and accept if and only if the result is g^0.

Alternative Setup. For concreteness (and efficiency), we define Obf and Eval to use the matrix $B_{n+1,2n,q}$. However, Setup can be modified to output any $B \in \mathbb{Z}_q^{(n+1)\times 2n}$ with the property that any $n+1$ columns of B form a full rank matrix (with overwhelming probability), and Obf and Eval will work as above with the matrix B (albeit, the efficiency gain of Appendix B may no longer apply).

Functionality Preservation. We first state a useful lemma.

Lemma 2. *If $k < q$, any set of $n+1$ columns of $B_{n+1,k,q}$ are linearly independent over $\mathbb{Z}_q$.*

Proof. This follows from inspecting the form of the determinant of the Vandermonde matrix, and noting that none of the factors of the determinant will divide q as long as $k < q$. □

Fix an x which matches pat and let t be the row vector computed in the Eval procedure. By construction, the vector tB is zero in all of the positions for which e is non-zero and thus

$$\prod_{i=1}^{n+1} v_i^{t_i} = g^{tBe} = g^0 .$$

On the other hand, for an x which does not match pat, by construction there is at least one index $i \in [2n]$ such that $(B)_i$ is not part of B_x and e_i is a uniformly random field element. Then appealing to Lemma 2, $t(B)_i \neq 0$ since otherwise the $n+1$ columns B_x and $(B)_i$ would be linearly dependent. Then the product $t(B)_i e_i$ is distributed as a uniformly random field element, which means that tBe is as well. Thus x is only accepted with probability $1/q = \mathsf{negl}(n)$.[11]

Security. We prove the distributional virtual black-box security of our construction.

Theorem 3. *Fix any function $w(n) \leq n$. The above construction is a distributional VBB obfuscator in the generic group model for the distribution $\mathcal{U}_{n,w(n)}$ over strings $\{0,1,*\}^n$.*

Proof. First we consider the case where $w(n) = n - \omega(\log(n))$. Let $c(n) = n - w(n) = \omega(\log(n))$. Let $\mathcal{H}$ be the space of handles used in the generic group instantiation of the obfuscation and let $|\mathcal{H}| > 2^n$ so that two uniformly drawn handles collide with negligible probability. For any adversary $\mathcal{A}$, we consider the simulator $\mathcal{S}$ that acts as the generic group model oracle and initializes $\mathcal{A}$ with

[11] As noted in [BKM+18], we can boost this to strong functionality preservation by setting $q > 2^{2n}$.

$n+1$ uniformly random handles. On a group operation query by $\mathcal{A}$, $\mathcal{S}$ responds with a uniformly random handle unless $\mathcal{A}$ had previously requested the same linear combination of initial elements, in which case $\mathcal{S}$ responds with the same handle as before. $\mathcal{S}$ can easily implement this with a lookup table. We assume without loss of generality that $\mathcal{A}$ only submits linear combinations over initial elements that are not identically zero. On any ZeroTest query by $\mathcal{A}$, $\mathcal{S}$ will return "not zero". Finally, $\mathcal{S}$ will output whatever $\mathcal{A}$ outputs after it has finished interacting with the generic group model simulation.

We show that with all but negligible probability, $\mathcal{A}$'s view of the generic group model oracle that is honestly implementing the obfuscation is *identical* to its view of the simulated oracle, which completes the proof of security. Observe that the only way that $\mathcal{A}$'s view diverges is if when interacting with the honest oracle, $\mathcal{A}$ either gets a successful ZeroTest, or receives the same handle on two group operation queries corresponding to different linear combinations of the initial handles. In the first case, $\mathcal{A}$ has formed a non-trivial linear combination of the initial $n+1$ group elements that evaluates to zero. Likewise, in the second case, if we subtract these two linear combinations, we see that $\mathcal{A}$ has also formed a non-trivial linear combination of the initial $n+1$ elements that evaluates to zero. Consider the first time that this occurs. We show that the probability of this happening over the randomness of the pattern and of the obfuscation is negligible.

Let $e \in \mathbb{Z}_q^{2n\times 1}$ be the vector drawn in the Obf procedure on input a pattern pat drawn from $\mathcal{U}_{n,n-c(n)}$. Denote the vector of coefficients in the adversary's linear combination as $k = (k_1, \ldots, k_{n+1}) \in \mathbb{Z}_q^{1\times(n+1)}$, so the resulting evaluation is equal to kBe. Since these coefficients are specified by $\mathcal{A}$ before its view has diverged from the simulated view, we can treat k as completely independent of e. Now by Lemma 2, any $n+1$ columns of B form a full rank matrix, so the vector $kB \in \mathbb{Z}_q^{1\times 2n}$ is 0 in at most n positions. If there exists $i \in [2n]$ for which $(kB)_i$ is non-zero and e_i is uniformly random, then with overwhelming probability $kBe \neq 0$ over the randomness of the obfuscation.

To complete the proof, we show that for any fixed set $S \subset [2n]$ of n indices, there exists an $i \in S$ for which e_i is uniformly random with overwhelming probability (over the randomness of the pattern). Partition e into the n pairs $\{e_{2j-1}, e_{2j}\}_{j\in[n]}$. At least $n/2$ of these pairs must contain at least one e_i such that $i \in S$. Sampling pat from $\mathcal{U}_{n,n-c(n)}$ corresponds to uniformly randomly picking $c(n)$ of the pairs to have one uniformly random e component, and then within each of these $c(n)$ sets, picking either e_{2j-1} or e_{2j} with probability $1/2$ to be the uniformly random component.

So among these $n/2$ pairs, an expected $c(n)/2$ of them have a uniformly random e component. This random variable is an instance of a *hypergeometric* random variable, and in Lemma 3 we use a Chernoff bound to show that it is greater than $c(n)/8$ except with negligible probability. Now for each of these $n/2$ pairs that contains a uniformly random component e_i, we have that $i \in S$ with probability $1/2$. Then the probability that there does not exist any $i \in S$ such that e_i is uniformly random is at most $(1/2)^{c(n)/8} + \mathsf{negl}(n)$ which is $\mathsf{negl}(n)$ for $c(n) = \omega(\log n)$.

Now we handle the case where $w(n) = n - O(\log(n))$. In this parameter regime, distributional VBB security is a vacuous security notion since a random input will satisfy the pattern with $1/\mathsf{poly}(n)$ probability. Thus a polynomial time simulator $\mathcal{S}$ can find an accepting input with overwhelming probability. Then it simply varies the accepting input one bit at a time in queries to the function oracle, and recovers the pattern in full. At this point it can run the obfuscation itself and simulate $\mathcal{A}$ on the honest obfuscation. □

We now state and prove Lemma 3. While tail bounds are known for hypergeometric random

variables, we were unable to find bounds strong enough for our parameter settings. In particular, plugging in the bounds summarized by Skala [Ska13] into the proof of Theorem 3 imply security when $c(n)$ is as small as $1/n^{\epsilon}$ for $\epsilon < 1/2$. Using Lemma 3, we obtain $c(n) = \omega(\log n)$. We note that our bound is specifically tailored for our application and should not be misinterpreted as a strengthening of known bounds on hypergeometric random variables.

Lemma 3. *A bag initially contains n balls, of which $c(n)$ are black and $n - c(n)$ are white. If $n/2$ balls are randomly drawn without replacement, then*

$$\Pr\left[\#\text{ black balls drawn } \geq \frac{c(n)}{8}\right] \geq 1 - e^{-c(n)/12}.$$

Proof. Instead of considering randomly selected white and black balls, consider an alternative setup where white balls are randomly painted black. Starting from n white balls, draw $n/2$ of them and call this set of balls D. Now consider two procedures to paint the balls:

- Procedure 1. For $c(n)$ iterations, pick a white ball at random and paint it black.
- Procedure 2. While the number of black balls in D is less than $c(n)/8$, pick a white ball at random and paint it black.

$\Pr[\#$ black balls drawn $\geq c(n)/8]$ is equivalent to the probability that at least $c(n)/8$ balls in D are colored black at the conclusion of Procedure 1. Observe that Procedure 1 and Procedure 2 are equivalent except for their stopping conditions. The probability that fewer than $c(n)/8$ balls in D are black at the conclusion of Procedure 1 is equal to the probability that fewer than $c(n)/8$ balls in D are black after $c(n)$ steps of the second procedure. In other words, the probability that fewer than $c(n)/8$ balls in D are black is equal to the probability the second procedure requires more than $c(n)$ steps.

For each $i = 1, \ldots, c(n)$, let B_i be a binary random variable that equals 1 if on the ith iteration of Procedure 2, a white ball in D is painted black. For the same range of i, let C_i be a binary random variable that equals 1 with probability $3/8$. Define the random variable $C \coloneqq \sum_{i=1}^{c(n)} C_i$ with expectation $E[C] = 3c(n)/8$.

First we show that $\Pr[B_i = 1] \geq \Pr[C_i = 1]$. To see this, note that at most $c(n)/8$ balls in D are black at any point during Procedure 2, so at least $n/k - c(n)/8 \geq 3n/8$ balls in D are always white. Thus, the probability a ball in D is selected at any step in Procedure 2 is at least $3/8$.

Collecting all of the above claims, we have

$$\begin{aligned}\Pr\left[\#\text{ black balls drawn } < \frac{c(n)}{8}\right] &= \Pr\left[\sum_{i=1}^{c(n)} B_i < \frac{c(n)}{8}\right] \leq \Pr\left[\sum_{i=1}^{c(n)} C_i < \frac{c(n)}{8}\right] \\ &= \Pr\left[C < (1 - \frac{2}{3})E[C]\right] \leq e^{-(1/2)(2/3)^2 E[C]} = e^{-c(n)/12},\end{aligned}$$

where the final inequality follows from a Chernoff bound. □

3.2 General Min-Entropy Distributions

The above proof shows security for a specific distribution $\mathcal{U}_{n,w}$, which has min-entropy $\log \binom{n}{w} + n - w$. We show here (again considering distributions over patterns with a fixed number of wildcards w),

that we can obtain security for any distribution over $\{0,1,*\}^n$ that satisfies a certain min-entropy requirement. We then note that this bound gives an improvement over $\mathcal{U}_{n,w}$ for any $w \leq cn$ where c is some constant in the range $[0.75, 0.821]$.

Lemma 4. *The above construction is a distributional VBB obfuscator in the generic group model for any distribution $\mathcal{D}$ over strings $\{0,1,*\}^n$ with exactly w wildcards such that $\mathcal{D}$ has min-entropy at least*

- $\log \binom{n}{w} + \omega(\log(n))$ *if* $w \leq n/2$;
- $\log \binom{2(n-w)}{n-w} + 2w - n + \omega(\log(n))$ *if* $3n/4 \geq w > n/2$;
- $\log \binom{n/2}{w-n/2} + n/2 + \omega(\log(n))$ *if* $w > 3n/4$.

Proof. Fix a distribution $\mathcal{D}$ over patterns with $w \leq n$ wildcards that satisfies the above min-entropy requirement. Let e be as drawn in $\mathsf{Obf}(\mathsf{pat})$ for $\mathsf{pat} \leftarrow \mathcal{D}$. Assume towards contradiction that there exists some set $S \subset [2n]$ of n indices such that $e_i = 0$ for all $i \in S$ with probability $1/p(n)$ for some polynomial p (where the probability is over $\mathsf{pat} \leftarrow \mathcal{D}$). Note that if there does not exist such a set then we get security in the generic group model by the same arguments as in the proof of Theorem 3. Partition e into the n pairs $\{e_{2j-1}, e_{2j}\}_{j\in[n]}$. Say that $2k$ of the n indices in S are in the same pair, for some $k \in [0, ..., w]$. We want to calculate the maximum number of possible patterns which satisfy $e_i = 0$ for all $i \in S$. This fixes k indices of pat to be $*$, $n-2k$ indices to be either in the set $\{0,*\}$ or the set $\{1,*\}$, and leaves k indices to be in the set $\{0,1,*\}$. So we can pick any $w-k$ wildcard locations among $n-k$ indices and then (overcounting slightly) pick any length k binary string. So we want to find

$$\max_{k\in[0,\ldots,w]} \binom{n-k}{w-k} 2^k .$$

Just as in the analysis in [BKM+18, Lemma 14], since $\frac{w-k}{n-k}$ is monotonically decreasing in k, the quantity is maximized for the largest k such that $\frac{w-k}{n-k} \geq 1/2$. For $w \leq n/2$, this maximum is attained at $k = 0$, for $n/2 < w \leq 3n/4$, the maximum is attained at $k = 2w - n \leq n/2$, and for $w > 3n/4$, the maximum is obtained at $k = n/2$.

For $w \leq n/2$, this shows that the maximum number of matching patterns over all possible sets of n elements is $\binom{n}{w}$. By averaging, there must be some pattern that occurs with probability at least $\frac{1}{p(n)\binom{n}{w}}$ which implies the min-entropy of the distribution is $\log \binom{n}{w} + O(\log(n))$. The same averaging argument gives the results for the cases where $w > n/2$. □

Remark 2. *The proof of [BKM+18, Lemma 14] shows that our min-entropy bound is an improvement over the uniform distribution for fixed number of wildcards $w = cn$ when $c \leq 3/4$. Note that the bound improves (gets farther from the uniform distribution) as w decreases. Since $\log(\binom{n}{cn}) \sim H(c)n$ (where H is Shannon entropy) for constant c, we get an improvement over the uniform distribution by a factor of about $\frac{1-c+H(c)}{H(c)}$ for $c \leq 1/2$. Furthermore, when w is a constant, the bound becomes $\omega(log(n))$, which is optimal.*

Remark 3. *For $c \geq 0.821$, the proposed bound becomes higher than the min-entropy of the uniform distribution. Indeed, it holds that*

$$\begin{aligned}\log\binom{n/2}{w-n/2} - \log\binom{n}{w} &= \log\left(\frac{w(w-1)\cdots(w-n/2+1)}{n(n-1)\cdots(n/2+1)}\right)\\ &\geq \frac{n}{2}\log\left(\frac{w-n/2}{n/2}\right)\\ &= \frac{n}{2}\log(2c-1)\,,\end{aligned}$$

and for $c \geq 0.821$, it holds that $\log(2c-1) > (1-2c)$; hence we get

$$\log\binom{n/2}{w-n/2} - \log\binom{n}{w} > \frac{n}{2}(1-2c) = (n-w) - \frac{n}{2}\,.$$

Hence for that parameter regime, we need to appeal to the Chernoff bound analysis above to get security for the uniform distribution.

3.3 Extension to Larger Alphabets

We generalize the construction to obfuscate patterns over any alphabet Σ of size ℓ. For a pattern length n, we choose the field size q to be a prime of size at least $\max\{n\ell, 2^n\}$. Then the size of the obfuscation is $O(n\log(q))$ and the running time of Obf and Eval is $O(n^2\log(q))$. We let $\Sigma_* := \Sigma \cup \{*\}$.

Construction.

- $\mathsf{Setup}(n, \Sigma)$. Let $\mathbb{G}$ be a group of prime order $q > \max\{n|\Sigma|, 2^n\}$ with generator g.
- $\mathsf{Obf}(\mathsf{pat} \in \Sigma_*^n)$: We associate the characters in Σ with the integers $[\ell]$. Let w be the number of wildcards in pat, and for $j \in [n-w]$, let $\mathsf{ind}(j)$ be the index of the jth non-wildcard element of pat. Let the integer $\alpha_j := (\mathsf{ind}(j)-1)\ell + \mathsf{pat}_{\mathsf{ind}(j)}$ for $j \in [n-w]$. Define $B_{\mathsf{pat}} \in \mathbb{Z}_q^{(n+1)\times(n-w)}$ as follows:

$$B_{\mathsf{pat}} := \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{n-w} \\ (\alpha_1)^2 & (\alpha_2)^2 & \dots & (\alpha_{n-w})^2 \\ \vdots & \vdots & \dots & \vdots \\ (\alpha_1)^{n+1} & (\alpha_2)^{n+1} & \dots & (\alpha_{n-w})^{n+1} \end{pmatrix}.$$

  Draw $e \leftarrow \mathbb{Z}_q^{n-w}$ and output

$$g^{B_{\mathsf{pat}}\cdot e} \in \mathbb{G}^{n+1}\,.$$

- $\mathsf{Eval}(v \in \mathbb{G}^{n+1}, x \in \Sigma^n)$: Define $B_x \in \mathbb{Z}_q^{(n+1)\times n}$ as follows:

$$B_x := \begin{pmatrix} x_1 & \ell + x_2 & \dots & (n-1)\ell + x_n \\ (x_1)^2 & (\ell + x_2)^2 & \dots & ((n-1)\ell + x_n)^2 \\ \vdots & \vdots & \dots & \vdots \\ (x_1)^{n+1} & (\ell + x_2)^{n+1} & \dots & ((n-1)\ell + x_n)^{n+1} \end{pmatrix}.$$

Solve for a non-zero $t \in \mathbb{Z}_q^{1\times(n+1)}$ such that $t \cdot B_x = 0$. Compute

$$\prod_{i=1}^{n+1} v_i^{t_i}$$

and accept if and only if the result is g^0.

Functionality Preservation. Fix an x which matches pat, let B_{pat} and e be as computed in the Obf procedure, and let t and B_x be as computed in the Eval procedure. By construction, each column of B_{pat} is a column of B_x, so since $t \cdot B_x = 0$, $t \cdot B_{\mathsf{pat}} = 0$ and thus $g^{t \cdot B_{\mathsf{pat}} \cdot e} = g^0$.

Now fix an x which does not match pat. Then by construction there is a column $(B_{\mathsf{pat}})_i$ of B_{pat} such that $(B_{\mathsf{pat}})_i$ is not a column of B_x. Then $t \cdot (B_{\mathsf{pat}})_i \neq 0$, otherwise the $n+1$ columns $[B_x|(B_{\mathsf{pat}})_i]$ would not be linearly independent and violate Lemma 2. Thus, $t \cdot (B_{\mathsf{pat}})_i \cdot e_i$ is a uniformly random field element, which implies that $t \cdot B_{\mathsf{pat}} \cdot e$ is as well, so Eval will only accept with probability $1/q = \mathsf{negl}(n)$.

Security. We first note that there is an equivalent (yet inefficient) representation of the Obf procedure, and we use it to prove distributional virtual black-box security of our construction.

Remark 4. *We can define an equivalent (inefficient)* Obf *procedure. On input a* $\mathsf{pat} \in \Sigma_*^n$*, where* $|\Sigma| = \ell$*,* Obf *draws the error vector* $\widehat{e} \in \mathbb{Z}_q^{n\ell}$ *as follows (again associating characters in* Σ *with* $[\ell]$*). For each* $i \in [n]$*:*

- *If* $\mathsf{pat}_i = *$*, set* $\widehat{e}_{(i-1)\ell+1} = \widehat{e}_{(i-1)\ell+2} = \cdots = \widehat{e}_{i\ell} = 0$
- *Otherwise, set* $\widehat{e}_{(i-1)\ell+\mathsf{pat}_i} \leftarrow \mathbb{Z}_q$ *and* $\widehat{e}_{(i-1)\ell+a} = 0$ *for* $a \in [\ell] \setminus \{\mathsf{pat}_i\}$*.*

The output of Obf *will be* $g^{(B_{n+1,n\ell,q})\cdot\widehat{e}}$*, where* $B_{n+1,n\ell,q}$ *is the* $(n+1) \times n\ell$ *matrix defined at the beginning of this section. Note that this procedure now has running time linear in the alphabet size rather than logarithmic. We will use the fact that this procedure produces an equivalent obfuscation as the procedure in the construction, namely,* $B_{\mathsf{pat}} \cdot e = B_{n+1,n\ell,q} \cdot \widehat{e}$ *where* B_{pat} *and* e *are as drawn in the* Obf *procedure above on input* pat*.*

Theorem 4. *Let* Σ *be an alphabet of arbitrary size and let* $\mathcal{U}_{n,w}^{\Sigma,\alpha}$ *be the set of distributions over* Σ_*^n *with the following properties:*

- *There are a fixed number of wildcard locations* w*;*
- *The* w *wildcard locations are uniformly distributed;*
- *Each non-wildcard character can be guessed independently with probability at most* $1 - \alpha$*.*

Then the above construction is a distributional VBB obfuscator in the generic group model for any distribution $\mathcal{D} \in \mathcal{U}_{n,w(n)}^{\Sigma,\alpha}$ *for any* $w(n) = n - \omega(\log(n))$ *and constant* $\alpha > 0$*.*

Proof. We follow the same argument as in the proof of Theorem 3 up until the point where the adversary $\mathcal{A}$ has specified $n+1$ coefficients $k = (k_1, \ldots, k_{n+1}) \in \mathbb{Z}_q^{1\times(n+1)}$ (while interacting with the generic group oracle implementing the honest obfuscation). Following the remark above, we view the obfuscation as $B_{n+1,n\ell,q} \cdot \widehat{e}$ and again by Lemma 2, $k \cdot B_{n+1,n\ell,q} \in \mathbb{Z}_q^{1\times n\ell}$ is 0 in at most

n positions. Now we argue in the same manner that there must exist some $i \in [n\ell]$ for which $(k \cdot B_{n+1,n\ell,q})_i$ is non-zero and $\widehat{e}_i$ is uniformly random.

We show that for any fixed set $S \subset [n\ell]$ of $n\ell - n$ indices, there exists an $i \in S$ for which $\widehat{e}_i$ is uniformly random with overwhelming probability. This follows by partitioning $\widehat{e}$ into n sets $\{\widehat{e}_{(j-1)\ell+1}, \ldots, \widehat{e}_{j\ell}\}_{j\in[n]}$ and noting that at least $n/2$ of these sets must contain at least $n-1$ elements $\widehat{e}_i$ for which $i \in S$. Then since we are considering distributions that induce a uniform distribution over wildcard positions, the same hypergeometric tail bound from Lemma 3 applies. So with overwhelming probability, we have that at least $c(n)/8$ of these sets correspond to a uniformly random e_i (for $c(n) = n - w(n)$).

Guessing the position of the uniformly random e_i in each of these sets is equivalent to guessing the character in a particular non-wildcard position. Therefore, there does not exist any $i \in [n\ell]$ such that $(k \cdot B_{n,n\ell,q})$ is non-zero and $\widehat{e}_i$ is uniformly random with probability at most $(1-\alpha)^{c(n)/8} + \mathsf{negl}(n) = \mathsf{negl}(n)$ for $c(n) = \omega(\log(n))$. □

In Appendix C, we show how to extend our result for general min-entropy distributions to the case where the alphabet size is $\ell > 2$; in particular we show that Lemma 4 works for any size alphabet ℓ.

3.4 Efficiency Improvements

Observe that our generalized construction for arbitrary alphabet size has an equivalent interpretation in the primal $(A \cdot s + e)$ or "grid" view of the original [BKM+18] construction. However, this gives an A of dimension $n\ell \times (n-1)$ where ℓ is the alphabet size, or equivalently a grid of total size $n\ell$. Thus, viewing the [BKM+18] construction as a dual scheme $(B \cdot e)$ allows us to reduce the size of the generalized construction from $n\ell$ group elements to $n+1$ group elements, and reduce the running time of Obf and Eval from linear in ℓ to logarithmic in ℓ. In the binary alphabet case (the only case considered by [BKM+18]), it may appear that this savings in space (from $2n$ group elements to $n+1$) comes at the cost of slower evaluation time. The [BKM+18] construction makes no mention of running time, though following their exact evaluation procedure to compute n Lagrange reconstruction coefficients suggests an $O(n^2)$ runtime.

On the other hand, evaluation of our scheme requires solving a system of n linear equations, which is $O(n^\omega)$ time in general (where here ω refers to the matrix multiplication constant, for which the best known bound is $\omega < 2.373$ [Wil14]). However, we show in Appendix B that evaluation in both the [BKM+18] construction and our construction can be done in $O(n \log^2 n)$. The key insight to improving the evaluation time of our construction is that when

$$B \coloneqq \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2^1 & \cdots & (2n)^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^n & \cdots & (2n)^n \end{pmatrix},$$

solving $t \cdot B = 0$ for t becomes the problem of finding the coefficients of a polynomial given its roots.

4 Obfuscating Conjunctions from Constant-Noise LPN

In this section, we consider a second construction: we show that replacing the fixed B matrix over $\mathbb{F}_q$ with a random matrix allows us to take our construction out of the group exponent and prove security in the standard model. Our security will be based on the standard constant-noise LPN assumption.

LPN vs. RLC. We note that under the Random Linear Codes (RLC) assumption (i.e., a generalization of LPN to $\mathbb{F}_q$ for $q \geq 2$; see Appendix A or [IPS09]), we could use the techniques from this section to prove that our construction over large fields is indistinguishable from random. But as explained in Section 1.2.5, indistinguishability from random *does not* imply distributional VBB security.[12] The problem arises from the fact that distributional VBB security requires indistinguishability from a simulated obfuscation even if the adversary knows a one bit predicate on the circuit (the pattern/conjunction in our case). This requires us to prove that the decisional "structured error" LPN/RLC problem is indistinguishable from random even if the adversary knows a predicate on the positions of the non-zero error vector entries, which encode the pattern. We show how to do this by modifying an appropriate search-to-decision reduction. Unfortunately, no search-to-decision reductions are known for RLC with super-polynomial modulus q, preventing our approach from extending beyond polynomial size q.

Our presentation is therefore structured as follows. We give a core technical reduction from standard RLC to structured error RLC that works for q up to size 2^{n^γ} in Appendix A. We then argue in Section 4.3 that we can modify this result to hold even if arbitrary predicates on the error vector are known, provided $q = 2$. Thus, the results for larger q given in Appendix A are not strictly necessary for our construction; we simply state them for maximum generality, as we believe this structured-error hardness may be of independent interest. As discussed in the introduction, we give all our results for the dual versions of the LPN and RLC problems in order to achieve a more efficient construction. For an in-depth discussion on the equivalence between the dual and primal formulations, refer to Micciancio and Mol [MM11] (they consider the LWE setting, but their transformations apply here).

Strong Functionality Preservation. We note that simply plugging our reduction into our obfuscation scheme only gives us weak functionality preservation (Definition 1). Other works such as [BKM+18] address this issue by increasing the size of the field, but this will not work here since LPN restricts us to $q = 2$. Fortunately, we can still boost our scheme and satisfy strong functionality preservation by making use of additional *regular* (as opposed to structured) LPN samples (as we describe in Section 4.4). This modification has one caveat: the evaluation is polynomial-time *in expectation*, requiring a slight relaxation of the polynomial slowdown requirement in Definition 1.

Multi-bit Output. Even after achieving strong functionality preservation, our scheme suffers from a noticeable weakness: it can handle random conjunctions where a constant fraction ρ of the bits are wildcards, but it cannot handle a sub-constant fraction of wildcards. This is surprising, since obfuscation for evasive functionalies should intuitively get *easier* as we reduce the number of

[12] In the generic group model, indistinguishability from random *does* imply distributional VBB. However as discussed in Section 1.2.5, trivial counterexamples to this claim exist in the standard model.

accepting inputs. However, our construction is completely broken if there are no wildcards, and in fact there is an attack on our scheme for any $\rho = 1 - O(\log n/n)$.[13]

We fix this problem by extending our obfuscator to support multi-bit output. In this setting, the obfuscator can embed a fixed message into the obfuscation, which an evaluator recovers upon finding an accepting input. We show how to leverage well-known leakage-resilience properties of LPN to extend our construction to the multi-bit setting, which then allow us to handle conjunctions with a sub-constant (or even zero) fraction of wildcards. The rough idea is to arbitrarily set some of the non-wildcard bits to be wildcards, and then use the multi-bit output to specify the true settings of those bits.

4.1 Exact Structured Learning Parity with Noise

Exact (Unstructured) Learning Parity with Noise. We begin by recalling the decisional Exact Learning Parity with Noise (DxLPN) problem considered by Jain et al. [JKPT12]. The word "exact" modifies the standard decisional Learning Parity with Noise (DLPN) problem by changing the sampling procedure for the error vector. Instead of setting each component of $e \in \mathbb{F}_q^m$ to be 1 with independent probability ρ, we sample e uniformly from the set of error vectors with exactly $\lfloor \rho m \rfloor$ entries set to 1 (we refer to these as vectors of weight $\lfloor \rho m \rfloor$). DLPN is polynomially equivalent to the exact version following the search to decision reduction given in [AIK09], as noted in [JKPT12, Döt16]. We give the precise definition in its dual formulation.

Let $\rho \in [0, 1]$ and $m > 0$ be an integer. Let χ_ρ^m denote the distribution on $\mathbb{F}_2^m$ which outputs uniformly random vectors in $\mathbb{F}_2^m$ of weight $\lfloor \rho m \rfloor$.

Definition 5 (Exact Learning Parity with Noise)**.** *Let n, m be integers and $\rho \in [0, 1]$. The (dual) Decisional Exact Learning Parity with Noise (DxLPN) problem with parameters n, m, ρ, denoted DxLPN(n, m, ρ), is hard if, for every probabilistic polynomial-time (in n) algorithm $\mathcal{A}$, there exists a negligible function μ such that*

$$\left| \Pr_{B,e}[\mathcal{A}(B, B \cdot e) = 1] - \Pr_{B,u}[\mathcal{A}(B, u) = 1] \right| \leq \mu(n)$$

where $B \leftarrow \mathbb{F}_2^{(m-n)\times m}$, $e \leftarrow \chi_\rho^m$, and $u \leftarrow \mathbb{F}_2^{m-n}$.

Exact Structured LPN. We now introduce a modification of the Exact Learning Parity with Noise (DxLPN) problem where we enforce that the error vector is *structured*. Concretely, the error vector e is now $2m$-dimensional, and we enforce that in any of the pairs $(2i - 1, 2i)$ for $i \in [m]$, at least one of e_{2i-1} and e_{2i} is 0. As we are considering the exact version of the problem, we enforce that $\lfloor \rho m \rfloor$ components of e are non-zero. Note that while the error vector has doubled in size, the number of non-zero components is unchanged. As described in Section 1.2.3, we take advantage of this particular error structure to encode a conjunction over a binary alphabet.

[13] The idea of the no wildcard attack is easy to see in the $A \cdot s + e$ view of the scheme: if there are no wildcards, the error vector is guaranteed to have exactly one 1 entry and one 0 entry in each pair of indices $(2i - 1, 2i)$. Then by summing every pair of equations $(2i-1, 2i)$, the error is completely known and we can solve for s. For $\rho = 1-O(^{log n}/_n)$, we can guess which pairs of equations do not have this structured error with noticeable probability.

We first introduce some notation. For a distribution $\mathcal{D}$ on $\mathbb{F}_2^m$, we define the distribution

$$\sigma(\mathcal{D}) = \left\{ \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ \vdots \\ s_{2m-1} \\ s_{2m} \end{pmatrix} \middle| \begin{array}{l} x \leftarrow \{0,1\}^m \\ e' \leftarrow \mathcal{D} \\ \text{for all } i \in [m], \begin{cases} s_{2i-x_i} = e'_i \\ s_{2i-(1-x_i)} = 0 \end{cases} \end{array} \right\}.$$

Definition 6 (Exact Structured LPN)**.** *The (dual) Decisional Exact Structured Learning Parity with Noise (DxSLPN) problem with parameters $n, 2m, \rho$, denoted $\mathsf{DxSLPN}(n, 2m, \rho)$, is hard if for every probabilistic polynomial-time (in n) algorithm $\mathcal{A}$, there exists a negligible function μ such that*

$$\left| \Pr_{B,e}[\mathcal{A}(B, B \cdot e) = 1] - \Pr_{B,u}[\mathcal{A}(B, u) = 1] \right| \leq \mu(n)$$

where $B \leftarrow \mathbb{F}_2^{(2m-n)\times 2m}, e \leftarrow \sigma(\chi_\rho^m)$, and $u \leftarrow \mathbb{F}_2^{2m-n}$.

In other words, the error vector $e \in \mathbb{F}_2^{2m}$ in the DxSLPN problem can be derived from the error vector $e' \in \mathbb{F}_2^m$ of the DxLPN problem; for each $i \in [m]$, randomly set one of e_{2i-1} or e_{2i} to e'_i and the other to 0.

We prove the following theorem in Appendix A. We stress that $\mathsf{DxLPN}(n^\epsilon, n, \rho)$ hardness follows from standard LPN hardness for polynomially many samples as noted in [JKPT12, Döt16], so this theorem implies $\mathsf{DxSLPN}(n - n^\delta, 2n, \rho)$ hardness from the standard LPN assumption.

Theorem 5. *Fix constants $\epsilon, \delta, \in [0, 1/2)$ and constant $\rho \in (0, 1)$. If $\mathsf{DxLPN}(n^\epsilon, n, \rho)$ is hard, then $\mathsf{DxSLPN}(n - n^\delta, 2n, \rho)$ is hard.*

4.2 Construction

The following construction is parameterized by a pattern length n and a constant $\delta \in [0, 1/2)$.

- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n)$: Draw $B \leftarrow \{0,1\}^{(n+n^\delta)\times 2n}$ and $e \in \{0,1\}^{2n}$ as follows. For each $i \in [n]$
  - If $\mathsf{pat}_i = *, e_{2i-1} = e_{2i} = 0$
  - If $\mathsf{pat}_i = b, e_{2i-b} = 1, e_{2i-(1-b)} = 0$

  Output (B, Be).
- $\mathsf{Eval}((B, v), x)$: Define B_x to be the $(n + n^\delta) \times n$ matrix where column j is set as $(B_x)_j := (B)_{2j-x_j}$. Solve for a full rank matrix $T \in \{0,1\}^{n^\delta \times (n+n^\delta)}$ such that $T \cdot B_x = 0$. Compute $T \cdot v$ and if the result is $0^{n^\delta \times 1}$ output 1 and otherwise output 0.

Weak Functionality Preservation. We show that for all $\mathsf{pat} \in \{0,1,*\}^n$ and $x \in \{0,1\}^n$, it holds that

$$\Pr[\mathsf{Eval}(\mathsf{Obf}(\mathsf{pat}), x) = f_{\mathsf{pat}}(x)] = 1 - \mathsf{negl}(n)\,,$$

over the randomness of the Obf procedure. Let B, e be drawn as in the Obf procedure. Let T, B_x be as defined in the Eval procedure and $B_{\overline{x}}$ be the n columns of B not in B_x. Let $e_{\overline{x}}$ be defined analogously. First, if $f_{\mathsf{pat}}(x) = 1$, then $e_{\overline{x}} = 0$ by construction. Then

$$T \cdot v = T \cdot B \cdot e = (T \cdot B_{\overline{x}}) \cdot e_{\overline{x}} = 0\,.$$

Hence, $\mathsf{Eval}(\mathsf{Obf}(\mathsf{pat}), x) = 1$ with probability 1. Now if $f_{\mathsf{pat}}(x) = 0$, then $e_{\overline{x}} \neq 0$ by construction. Since $T \cdot B_{\overline{x}}$ is a uniformly random rank n^δ matrix independent of $e_{\overline{x}}$, it holds that

$$\Pr[T \cdot v = 0] = \frac{1}{2^{n^\delta}} = \mathsf{negl}(n)\,.$$

4.3 Security

Lemma 5. *Fix any predicate $\mathcal{P}\colon \{0,1,*\}^n \to \{0,1\}$. If $\mathsf{DxSLPN}(n, 2m, \rho)$ is computationally hard, then the problem of distinguishing $(B, Be, \mathcal{P}(e))$ from $(B, u, \mathcal{P}(e))$ is computationally hard, where $B \leftarrow \mathbb{F}_2^{(2m-n)\times 2m}, e \leftarrow \sigma(\chi_\rho^m)$, and $u \leftarrow \mathbb{F}_2^{2m-n}$.*

Proof. First, we introduce the two search variants of the problems considered in the lemma statement. Search xSLPN (or SxSLPN) will be defined as the problem of returning e with non-negligible probability, given (B, Be) as drawn in the definition of DxSLPN. This problem is clearly at least as hard as DxSLPN. Next, we define "SxSLPN with a one bit predicate", where the adversary must find e with non-negligible probability, given $(B, Be, \mathcal{P}(e))$ for any fixed predicate $\mathcal{P}$. This problem is at least as hard as regular SxSLPN since the adversary for SxSLPN, which receives (B, Be) as input, can simply guess the value of $\mathcal{P}(e)$, be correct with probability at least $1/2$, and forward $(B, Be, \mathcal{P}(e))$ as input to the adversary for SxSLPN with a one bit predicate. To complete the proof of the lemma, we argue that DxSLPN with a one bit predicate is at least as hard as SxSLPN with a one bit predicate.

The proof of this fact follows from the proof of Lemma 5 in [Döt16] (equivalence of search and decision "leaky LPN")[14], where we let the underlying problem be structured LPN rather than regular LPN and consider the special case of leakage functions corresponding to one bit predicates. We can use this proof, which relies on the Goldreich-Levin theorem, to conclude that an adversary with advantage ϵ in breaking DxSLPN with a one bit predicate can be used to produce an adversary with advantage $\epsilon^2/8$ in breaking SxSLPN with a one bit predicate. In particular, an adversary with non-negligible advantage in breaking DxSLPN with a one bit predicate implies an adversary with non-negligible advantage in breaking SxSLPN with a one bit predicate. □

Theorem 6. *Fix any constant $\rho \in (0,1)$. Assuming the computational hardness of $\mathsf{DLPN}(n^\epsilon, n, \rho)$[15] for some $\epsilon < 1/2$, the above obfuscation with parameters (n, δ) for $\delta < 1/2$ is Distributional-VBB secure for patterns $\mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n}$.*

[14] The main component of this proof is a very slight tweak of the search-to-decision reduction presented in [AIK09]. We note that this proof is presented for the $As + e$ version of LPN, but the same technique works for the dual Be version, as shown for example in the proof of Lemma 2.3 in [HOSS18].

[15] As previously noted, the DLPN and DxLPN problems are polynomially equivalent

Proof. We show that the above obfuscator satisfies the definition of Perfect Circuit-Hiding (Definition 2), which implies Distributional VBB security [BBC+14]. We want to show that for any probabilistic polynomial-time adversary $\mathcal{A}$ and any *balanced* predicate $\mathcal{P}\colon \{0,1,*\}^n \to \{0,1\}$ (that is, $\mathcal{P}$ takes the values 0 and 1 with probability $1/2$ over the randomness of $\mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n}$),

$$\Pr_{\mathsf{pat}\leftarrow\mathcal{U}_{n,n-\rho n}}[\mathcal{A}(\mathsf{Obf}(\mathsf{pat})) = \mathcal{P}(\mathsf{pat})] = \frac{1}{2} + \mathsf{negl}(n)\,.$$

We know by assumption and from Theorem 5 and Lemma 5 that, for any predicate $\mathcal{P}\colon \{0,1,*\} \to \{0,1\}$ and for all probabilistic polynomial-time $\mathcal{B}$,

$$\left|\Pr_{\mathsf{pat}\leftarrow\mathcal{U}_{n,n-\rho n}}[\mathcal{B}(\mathsf{Obf}(\mathsf{pat}),\mathcal{P}(\mathsf{pat})) = 1] - \Pr_{\mathsf{pat}\leftarrow\mathcal{U}_{n,n-\rho n}}[\mathcal{B}((B,u),\mathcal{P}(\mathsf{pat})) = 1]\right| = \mathsf{negl}(n)\,,$$

where $B \leftarrow \{0,1\}^{(n+n^\delta)\times 2n}$ and $u \leftarrow \{0,1\}^{n+n^\delta}$.

Now assume that there exists a balanced predicate $\mathcal{P}$ such that there exists a probabilistic polynomial-time adversary $\mathcal{A}$ that breaks the above Perfect Circuit-Hiding definition for predicate $\mathcal{P}$ with non-negligible advantage $\mu(n)$. Consider an adversary $\mathcal{B}$ that receives $((B,u),\mathcal{P}(\mathsf{pat}))$, runs $\mathcal{A}$ on (B,u) and outputs 1 if $\mathcal{A}(B,u) = \mathcal{P}(\mathsf{pat})$ and 0 otherwise. If (B,u) was an honest obfuscation, then $\mathcal{B}$ outputs 1 with probability $\frac{1}{2} + \mu(n)$. If (B,u) was uniformly random, then $\mathcal{A}(B,u)$ is independent of $\mathcal{P}(\mathsf{pat})$, so since $\mathcal{P}$ is balanced, $\mathcal{B}$ outputs 1 with probability exactly $1/2$. Thus, $\mathcal{B}$'s advantage in distinguishing $(\mathsf{Obf}(\mathsf{pat}),\mathcal{P}(\mathsf{pat}))$ from $((B \leftarrow \{0,1\}^{(n+n^\delta)\times 2n}, u \leftarrow \{0,1\}^{n+n^\delta}),\mathcal{P}(\mathsf{pat}))$ is $\mu(n)$ which is non-negligible. This is a contradiction, which completes the proof. □

4.4 Boosting to Strong Functionality Preservation

In this section, we describe how to tweak our obfuscation construction to satisfy the notion of strong functionality preservation from Definition 1. To do this, we had to modify our construction and evaluation scheme so that evaluation is polynomial-time *in expectation*, requiring a slight relaxation of the polynomial slowdown requirement in Definition 1 (namely that, for any input x, the expected running time of the evaluation procedure is polynomial *over the randomness of the obfuscation*).

Indeed, while our weak functionality preservation argument guarantees correctness on any input x with overwhelming probability, the chance of error is $1/2^{n^\delta}$. Since there are 2^n possible inputs, we cannot simply apply a union bound to achieve the stronger notion, as done in [BR17, BKM+18]. Roughly speaking, it stems from the fact that, given $(B, B\cdot e)$ as generated above where e is determined by some pattern pat with w wildcards, there will be some other e' determined by pattern pat' with w wildcards such that $B\cdot e = B\cdot e'$. Then given $v = B\cdot e = B\cdot e'$, the obfuscation of pat is *identical* to an obfuscation of pat'. The most natural way to circumvent this issue would be to increase the number of rows of B until $B\cdot e$ uniquely determines e with overwhelming probability. Unfortunately, making this argument work would require about $\Theta(n)$ additional rows, but we are constrained by a step in our reduction (Lemma 12) that only permits us to introduce n^δ additional rows for $\delta < 1/2$.

Boosting with Exact LPN Samples. However, we observe that we can add rows to B if we correspondingly increase the number of columns. Note that each additional column we add to B requires adding an element to the error vector e. In order to preserve security, these elements will

have to be standard exact LPN samples (with *unstructured* error). For concreteness, we set the error rate to $1/16$ and add $16n$ extra columns; since these are exact LPN samples, note that exactly n of the last $16n$ components of e will be set to 1.

- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n)$: Draw $B \leftarrow \{0,1\}^{17n \times 18n}$. Set $e' \in \{0,1\}^{2n}$ as follows. For each $i \in [n]$:
  - If $\mathsf{pat}_i = *, e'_{2i-1} = e'_{2i} = 0$
  - If $\mathsf{pat}_i = b, e'_{2i-b} = 1, e'_{2i-(1-b)} = 0$

  Draw $\widehat{e} \leftarrow \chi_{1/16}^{16n}$, let $e = \begin{pmatrix} e' \\ \widehat{e} \end{pmatrix}$ and output (B, Be)

- $\mathsf{Eval}((B,v), x \in \{0,1\}^n)$: Define B_x to be the $17n \times n$ matrix where column j is set as $(B_x)_j := (B)_{2j-x_j}$. Define $\widehat{B}$ to be the $17n \times 16n$ matrix consisting of the final $16n$ columns of B. Now solve for the affine space $S \subseteq \{0,1\}^{17n}$ of solutions $[B_x|\widehat{B}]s = v$ for $s \in S$, iterate over all vectors $s \in S$ and accept if and only if the final $16n$ components of some s has Hamming weight at most n.

Security. We can base security for the distribution $\mathcal{U}_{n,n-\rho n}$ on the computational hardness of $\mathsf{DxLPN}(n^\epsilon, 17n, \min(\rho, 1/16))$ for some $\epsilon < 1/2$. This follows from a small tweak to the proof of Theorem 5, where in proving the indistinguishability of distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we transform just the left-most n columns of B into a length $2n$ "structured error" instance (instead of performing the transformation on the entire matrix B).

Strong Functionality Preservation. We begin by showing that over the random choice of $B \leftarrow \{0,1\}^{17n \times 18n}$, with overwhelming probability, there does not exist two vectors $e^{(1)}, e^{(2)} \in \{0,1\}^{18n}$ such that $Be^{(1)} = Be^{(2)}$ and such that the final $16n$ components of each has Hamming weight at most n. If this were the case, then there must exist some $e^* \in \{0,1\}^{18n}$ with Hamming weight at most $2n$ among its final $16n$ components such that $Be^* = 0$. Let B' be the first $2n$ columns of B and $\widehat{B}$ be the final $16n$ columns. Then there must be $2n$ columns B^* of $\widehat{B}$ such that the matrix $[B'|B^*]$ is not full rank. There are $\binom{16n}{2n}$ choices for B^*, and each choice gives a $17n \times 4n$ binary matrix, which by a union bound is not full rank with probability at most $4n/2^{13n}$. Then applying a union bound over all choices for B^* gives that a non-full rank matrix $[B'|B^*]$ exists with probability at most

$$\frac{4n\binom{16n}{2n}}{2^{13n}} \leq \frac{4n(8e)^{2n}}{2^{13n}} = 2^{(2\log(8e)-13)n+2\log(n)} \leq 2^{2\log(n)-4n} = \mathsf{negl}(n)\,.$$

Now, let B be drawn as in the Obf procedure, and assume that the above property is true. Let $e = \begin{pmatrix} e' \\ \widehat{e} \end{pmatrix}$ be as drawn in the Obf procedure for pattern pat and fix an x matching pat. Let B_x and $\widehat{B}$ be as defined in the Eval procedure and $B_{\overline{x}}$ be the n out of the first $2n$ columns of B not in B_x. Now let e_x be the n components of e' corresponding to the columns in B_x and $e_{\overline{x}}$ be the other n components of e'. Then for an x that matches pat, $e_{\overline{x}} = 0$ by construction, so

$$v = B\begin{pmatrix} e' \\ \widehat{e} \end{pmatrix} = [B_x|\widehat{B}]\begin{pmatrix} e_x \\ \widehat{e} \end{pmatrix}\,,$$

and thus $\begin{pmatrix} e_x \\ \widehat{e} \end{pmatrix}$ is in the space of vectors solved for during Eval and $\widehat{e}$ has Hamming weight n. Thus x will be accepted.

Now fix an x that does *not* match the pattern. For x to be (incorrectly) accepted, there must exist $s \in \{0,1\}^{17n}$ with Hamming weight at most n among its final $16n$ components such that

$$[B_x \mid \widehat{B}]s = v\,.$$

From s, we construct a vector $s' \in \{0,1\}^{18n}$ as follows. On the n indices corresponding to B_x, place the first n entries of s. On the n indices corresponding to $B_{\overline{x}}$, place 0's. Then let the last $16n$ components of s' equal the last $16n$ components of s. By construction, we have that $Bs' = [B_x \mid \widehat{B}]s = v = Be$.

On the other hand, we know that $s' \neq e$, since $e_{\overline{x}} \neq 0$, whereas $s_{\overline{x}} = 0$. Since these are two distinct vectors such that the Hamming weight of the final $16n$ components of each is at most n, we obtain a contradiction.

Expected Polynomial Slowdown. Note that, during Eval, we can solve for the space S by first finding a matrix K which spans the kernel of (the square matrix) $[B_x|\widehat{B}]$ and then shifting the space by some solution s that satisfies $[B_x|\widehat{B}]s = v$. These steps are clearly polynomial time, but now we have to iterate over the all vectors in the column space of K, of which there are exactly 2^d, where d is the rank deficiency of $[B_x|\widehat{B}]$. We now show that the expected value of this number is at most $2n^2 + n$ over the randomness of the Obf procedure.

For each value of $d \in [17n]$, we can union bound to derive an upper bound on the probability that $[B_x|\widehat{B}]$ is rank deficient by exactly d:

$$\frac{\binom{17n}{d}}{2^{d^2}} \leq \left(\frac{17en}{d2^d}\right)^d.$$

Then, we sum to calculate the expected number of vectors 2^d, upper bounding the above probability by 1 for all $d \leq 2\log(n)$:

$$\sum_{d=0}^{2\log(n)} 2^d + \sum_{d=2\log(n)+1}^{n} 2^d \left(\frac{17en}{d2^d}\right)^d \leq 2n^2 + n\left(\frac{17e}{n\log(n)}\right)^{\log(n)} \leq 2n^2 + n\,,$$

where the last inequalities hold for n sufficiently large.

4.5 Multi-Bit Output

We show how to extend our construction to obfuscate functions of the form $f_{\mathsf{pat},m}(x) = m$ if x matches pat and $\perp$ otherwise. To obtain this, we XOR the message m with the output of a polynomial stretch PRG, whose seed we embed in the error vector of the structured LPN instance.

The following construction is parameterized by a pattern length n, a message length function $\ell(n)$ which is polynomial in n, a constant $\delta \in [0, 1/2)$, and a constant κ such that $\kappa < \delta$. We let $G\colon \{0,1\}^{n^\kappa} \to \{0,1\}^{\ell(n)}$ be a polynomial stretch PRG.

- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n, m \in \{0,1\}^{\ell(n)})$: Sample a PRG seed $s \leftarrow \{0,1\}^{n^\kappa}$ and $B \leftarrow \{0,1\}^{(n+n^\delta)\times(2n+n^\kappa)}$. Fix $e' \in \{0,1\}^{2n}$ as follows. For each $i \in [n]$

– If $\mathsf{pat}_i = *, e'_{2i-1} = e'_{2i} = 0$
– If $\mathsf{pat}_i = b, e'_{2i-b} = 1, e'_{2i-(1-b)} = 0$

Now let $e = \begin{pmatrix} e' \\ s \end{pmatrix}$ and output $(B, Be, G(s) \oplus m)$.

- $\mathsf{Eval}((B, v, c), x)$: Define $\widehat{B}$ to be the last n^κ columns of B. Define B_x to be the $(n + n^\delta) \times n$ matrix where column j is set as $(B_x)_j := (B)_{2j-x_j}$. Solve for a full rank matrix $T \in \{0,1\}^{(n^\delta - n^\kappa) \times (n+n^\delta)}$ such that $T \cdot [B_x | \widehat{B}] = 0$. If $T \cdot v = 0^{(n^\delta - n^\kappa) \times 1}$, solve for an e such that $[B_x | \widehat{B}] \cdot e = v$, let s be the last n^κ elements of e, and output $G(s) \oplus c$; otherwise output $\perp$.

Functionality Preservation. We show that for all $\mathsf{pat} \in \{0,1,*\}^n, m \in \{0,1\}^{\ell(n)}$, and $x \in \{0,1\}^n$,

$$\Pr[\mathsf{Eval}(\mathsf{Obf}(\mathsf{pat}, m)) = f_{\mathsf{pat},m}(x)] = 1 - \mathsf{negl}(n)\,.$$

Let B_x be as defined in the Obf procedure. Define $B_{\overline{x}}$ to be the other n columns out of the first $2n$ columns of B and $e_{\overline{x}}$ to be the corresponding elements of e. If $f_{\mathsf{pat},m}(x) = m$, then $e_{\overline{x}} = 0^n$ by construction. Thus, $B \cdot e = v = [B_x | \widehat{B}] \cdot e$. Moreover, by a union bound, with probability at least $1 - \frac{n+n^\kappa}{2^{n^\delta - n^\kappa}} = 1 - \mathsf{negl}(n)$ (for $\delta > \kappa$), the matrix $[B_x | \widehat{B}]$ is full rank. Then e is the unique solution to this equation, so the evaluator will successfully recover the PRG seed s and thus the message m.

Now if $f_{\mathsf{pat},m}(x) = \perp$, then $e_{\overline{x}} \neq 0^{n \times 1}$ by construction. Then since $T \cdot B_{\overline{x}}$ is a uniformly random rank $n^\delta - n^\kappa$ matrix independent of $e_{\bar{x}}$,

$$\Pr[T \cdot u = 0] = \frac{1}{2^{n^\delta - n^\kappa}} = \mathsf{negl}(n)$$

for $\delta > \kappa$.

Lemma 6. *Assuming the computational hardness of $\mathsf{DxLPN}(n^\epsilon, n, \rho)$ for some $\epsilon < 1/2$, the above obfuscation with parameters $(n, \ell(n), \delta, \kappa)$ is Distributional-VBB secure for functions $f_{\mathsf{pat},m}$ where $\mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n}$ and m is drawn from an arbitrary distribution $\mathcal{M}$ over $\{0,1\}^{\ell(n)}$, independently of pat.*

Proof. We introduce a series of three distributions. Let $\mathcal{P}\colon \{0,1\}^{n+n^\kappa} \to \{0,1\}$ be any fixed predicate.

$\mathcal{D}_0$ is the uniform distribution over the set

$$\left\{ \Big(\mathcal{O}, \mathcal{P}(\mathsf{pat}, m)\Big) \;\middle|\; \begin{array}{l} \mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n} \\ m \leftarrow \mathcal{M} \\ \mathcal{O} \leftarrow \mathsf{Obf}(\mathsf{pat}, m) \end{array} \right\}$$

$\mathcal{D}_1$ is the uniform distribution over the set

$$\left\{ \Big((B, u, G(s) \oplus m), \mathcal{P}(\mathsf{pat}, m)\Big) \;\middle|\; \begin{array}{l} \mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n} \\ m \leftarrow \mathcal{M} \\ B \leftarrow \{0,1\}^{(n+n^\delta) \times (2n+n^\kappa)} \\ u \leftarrow \{0,1\}^{n+n^\delta} \\ s \leftarrow \{0,1\}^{n^\kappa} \end{array} \right\}$$

$\mathcal{D}_2$ is the uniform distribution over the set

$$\left\{ \Big((B,u,v),\mathcal{P}(\mathsf{pat},m)\Big) \;\middle|\; \begin{array}{l} \mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n} \\ m \leftarrow \mathcal{M} \\ B \leftarrow \{0,1\}^{(n+n^\delta)\times(2n+n^\kappa)} \\ u \leftarrow \{0,1\}^{n+n^\delta} \\ v \leftarrow \{0,1\}^{\ell(n)} \end{array} \right\}$$

Note that computational indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_2$ is sufficient to prove Distributional-VBB via the same arguments as in the proof of Theorem 6. Now, for some predicate $\mathcal{P}$, assume that there exists an adversary $\mathcal{A}$ that distinguishes $\mathcal{D}_0$ and $\mathcal{D}_1$ with non-negligible advantage ϵ. Then there must exist some fixed m such that $\mathcal{A}$'s distinguishing advantage given this fixed m rather than m drawn from $\mathcal{M}$ is at least ϵ. Hard-coding m into $\mathcal{P}$ produces a predicate $\mathcal{P}'$ over patterns in $\{0,1,*\}^n$. Now $\mathcal{A}$ can be used to distinguish between $(\mathsf{Obf}(\mathsf{pat}),\mathcal{P}'(\mathsf{pat}))$ and $(B,u,\mathcal{P}'(\mathsf{pat}))$ with advantage ϵ (where Obf refers to the construction in 4.3 with parameters (n,δ)). The reduction simply draws s for itself, appends $\widehat{B} \leftarrow \{0,1\}^{(n+n^\delta)\times(n^\kappa)}$ to B, and adds $\widehat{B}\cdot s$ to u. However, this is ruled out by Theorem 5 and Lemma 5, assuming the computational hardness of $\mathsf{DxLPN}(n^\epsilon,n,\rho)$ for some $\epsilon < 1/2$. Distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are indistinguishable by the security of the PRG G, which completes the proof. □

Security for Patterns with Sub-constant Error Rate. Note that Theorem 3 gives security for patterns with a fixed $\lfloor \rho n \rfloor$ wildcards for any constant ρ (assuming the hardness of the appropriate LPN instance). However, we would like to have security for patterns with less wildcards, which intuitively should be no more difficult to obfuscate. This follows from the above construction, where to obfuscate a pattern with $w(n)$ wildcards where $w(n)$ is a sub-constant function of n, we draw $\mathsf{pat}' \leftarrow \mathcal{U}_{n,n-\rho n}$ and $m \leftarrow \mathcal{U}_{n-\rho n,w(n)}$. Letting m correspond to the wildcard locations of pat' defines a pattern pat with $w(n)$ wildcards. If the evaluator gets an accepting input x for pat', he recovers m and can see if m matches x on its non wildcard positions.

5 Information-Theoretic Security

In this section, we consider a third construction, which relies on subset sums of random rank one matrices. To obfuscate a pattern $\mathsf{pat} \in \{0,1,*\}^n$, we sample a uniformly random $B \in \mathbb{F}_q^{k\times k}$ with rank $k-1$. Then for each $i \in [n]$ where $\mathsf{pat}_i \neq *$, we sample a uniformly random rank one matrix $A^{(i)} \in \mathbb{F}_q^{k\times k}$, and for all i where $\mathsf{pat}_i = *$, we sample a random rank one matrix $A^{(i)} \in \mathbb{F}_q^{k\times k}$ with its columns in the column span of B. Finally, we set $F \coloneqq B - \sum_{i|\mathsf{pat}_i=1} A^{(i)}$ and give out $F, \{A^{(i)}\}_{i\in[n]}$. To evaluate on x, we simply compute $F + \sum_{i|x_i=1} A^{(i)}$ and accept if the determinant is 0.

Correctness follows from the fact that on an accepting input, this sum is simply B plus a sum of $A^{(i)}$ matrices corresponding to wildcard positions; these $A^{(i)}$ matrices are chosen in the column span of B, so its rank remains at most $k-1$. On a rejecting input, we rely on the fact that a rank $k-1$ matrix plus an independently random rank one matrix will be full rank with overwhelming probability. For security, we can support n^δ wildcards by setting $k = n^\delta + 1$. We show that if the pattern has sufficient min-entropy on the non-wildcard positions, B is information theoretically hidden. With a bit more effort, we turn this into a formal proof of statistical virtual black box security.

Remark 5. *This scheme is reminiscent of the Learning Subspace with Noise problem introduced by Dodis et al. [DKL09] where we correspond the column span of B with the hidden subspace and the wildcard matrices $A^{(i)}$ with samples from the subspace. Dodis et al. [DKL09] observe that this problem is equivalent to LPN if the hidden subspace is rank deficient by 1 (since q is large in our scheme, it is more analogous to the Random Linear Code (RLC) problem over $\mathbb{F}_q$). However, we are able to obtain statistical security arguments by limiting the number of samples to only give out $k-1$ vectors from the $(k-1)$-dimensional subspace.*

However, statistical security requires us to sacrifice strong functionality preservation, i.e., we cannot ensure $\mathsf{Obf}(f_{\mathsf{pat}})(x) = f_{\mathsf{pat}}(x)$ holds simultaneously for all x. This follows from the fact that if strong functionality preservation holds, then a computationally unbounded adversary can simply recover the entire truth table and learn pat in the clear. Thus, security relies on the fact that $\mathsf{Obf}(f_{\mathsf{pat}})(x) \neq f_{\mathsf{pat}}(x)$ on *exponentially* many x. However, we still satisfy weak functionality preservation, which ensures that for any x, $\mathsf{Obf}(f_{\mathsf{pat}})(x) = f_{\mathsf{pat}}(x)$ with overwhelming probability.

In Section 5.3, we show how to modify this base construction to achieve an intermediate notion we call *computational functionality preservation*, assuming the (computational) discrete log assumption. The resulting scheme has the curious property of being distributional-VBB secure against computationally unbounded adversaries, but functionality preserving in the view of any computationally bounded adversary (even those who know pat).

5.1 Construction

We now present the construction informally sketched above. We note that we will draw the rank deficient matrix B by choosing its first $k-1$ rows at random, and then picking its last row in the row span of the first $k-1$. It is easy to see that this is statistically indistinguishable from sampling a uniformly random B with rank $k-1$. However, "pushing" the rank deficiency to the last row of B will simplify both the security analysis and the modified construction in Section 5.3.

Notation. We will frequently write a matrix M as

$$\begin{pmatrix} \overline{M} \\ \underline{M} \end{pmatrix},$$

where $\overline{M}$ is the submatrix of M consisting of everything except the bottom row, and $\underline{M}$ denotes the bottom row.

Construction. The following is parameterized by a pattern length n and field size $q = 2^{n^\gamma}$ for a $\gamma > 0$. We let $\mathbb{F}_q$ denote a field of size q.

- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n)$. Partition $[n]$ into $S_0 \cup S_1 \cup S_*$ so that
$$S_0 = \{i \mid \mathsf{pat}_i = 0\}, \quad S_1 = \{i \mid \mathsf{pat}_i = 1\}, \quad S_* = \{i \mid \mathsf{pat}_i = *\}.$$
  - Let $k = |S_*| + 1$;
  - Draw $\overline{B} \leftarrow \mathbb{F}_q^{(k-1)\times k}$, $r \leftarrow \mathbb{F}_q^{1\times(k-1)}$ and let[16]
$$B := \begin{pmatrix} \overline{B} \\ r \cdot \overline{B} \end{pmatrix}$$

[16] We could instead simply draw B as a uniformly distributed rank $k-1$ matrix, but for ease of presentation we draw it with this structure.

– For each $i \in S_0 \cup S_1$, sample a uniformly random rank 1 matrix $A^{(i)} \in \mathbb{F}_q^{k \times k}$;

– For each $i \in S_*$, sample a uniformly random rank 1 matrix $\overline{A}^{(i)} \in \mathbb{F}_q^{(k-1) \times k}$, and let[17]

$$A^{(i)} := \begin{pmatrix} \overline{A}^{(i)} \\ r \cdot \overline{A}^{(i)} \end{pmatrix}$$

– Define

$$F \coloneqq B - \sum_{i \in S_1} A^{(i)},$$

and output $(F, A^{(1)}, \ldots, A^{(n)})$.

- $\mathsf{Eval}((F, A^{(1)}, \ldots, A^{(n)}), x \in \{0,1\}^n)$. Output 1 if

$$\det \left(F + \sum_{i | x_i = 1} A^{(i)} \right) = 0,$$

and 0 otherwise.

Weak Functionality Preservation. By construction, for an x that matches pat, we have that

$$\operatorname{colspan} \left(F + \sum_{i | x_i = 1} A^{(i)} \right) = \operatorname{colspan} \left(B + \sum_{i | x_i = 1 \wedge \mathsf{pat}_i = *} A^{(i)} \right) \subseteq \operatorname{colspan}(B)\,.$$

It then follows that $\det(F + \sum_{i|x_i=1} A^{(i)}) = 0$ since B has rank $k-1$. For an x that does not match pat, consider the matrix

$$\begin{aligned} F + \sum_{i | x_i = 1} A^{(i)} &= \underbrace{B + \sum_{i | x_i = 1 \wedge \mathsf{pat}_i = *} A^{(i)}}_{B'} + \underbrace{\sum_{i | x_i = 1 \wedge \mathsf{pat}_i = 0} A^{(i)} - \sum_{i | x_i = 0 \wedge \mathsf{pat}_i = 1} A^{(i)}}_{A'} \\ &= \begin{pmatrix} \overline{B}' + \overline{A}' \\ \underline{B}' + \underline{A}' \end{pmatrix}. \end{aligned}$$

Since the first $k-1$ rows of B are all uniformly random, the same is true of $\overline{B}'$. Furthermore, we know by construction that there exists at least one i such that $\mathsf{pat}_i \neq x_i$ and $\mathsf{pat}_i \in \{0,1\}$, so A' contains at least one of these $A^{(i)}$ matrices. Note that the last row of $A^{(i)}$ (and hence $\underline{A}'$) is uniformly random and independent of $\overline{B}'$. Thus $F + \sum_{i|x_i=1} A^{(i)}$ is distributed as a uniformly random matrix, so its determinant is non-zero with overwhelming probability $1 - k/q = 1 - \mathsf{negl}(n)$ by the Schwartz–Zippel lemma.

[17] We could instead draw $A^{(i)}$ as a random rank 1 matrix whose columns are in the column span of B, but again for ease of presentation we draw it with this structure

5.2 Security

First we give precise definitions of statistical security for obfuscated functionalities. The following definition is mostly the same as the definition for Distributional VBB Security so we just describe the differences with Definition 1.

Definition 7 (ϵ-Statistical Distributional VBB Obfuscation)**.** *Let* Obf *be as defined in Definition 1 and* $\epsilon(n)$ *be a function of* n*. We require the same notions of Functionality Preservation and Polynomial Slowdown, but alter the definition of Distributional Virtual Black-Box as follows:*

3. *ϵ-Statistical Distributional Virtual Black-Box: For every (unbounded) adversary $\mathcal{A}$, there exists a PPT simulator $\mathcal{S}$ such that for every $n \in \mathbb{N}$, every distribution $D \in \mathcal{D}_n$ (a distribution over $\mathcal{C}_n$), and every predicate $\mathcal{P} \colon \mathcal{C}_n \to \{0,1\}$:*

$$\left| \Pr_{C \leftarrow \mathcal{D}_n, \mathcal{A}}[\mathcal{A}(\mathsf{Obf}(C, 1^n)) = \mathcal{P}(C)] - \Pr_{C \leftarrow \mathcal{D}_n, \mathcal{S}}[\mathcal{S}^C(1^{|C|}, 1^n) = \mathcal{P}(C)] \right| \leq \epsilon(n)\,.$$

We say a construction achieves statistical distributional VBB security if it is ϵ-statistical distributional VBB secure for some $\epsilon(n) = \mathsf{negl}(n)$.

Remark 6. *As discussed in the introduction, if strong functionality preservation holds, then a computationally unbounded adversary can learn the entire truth table of the original function. Thus, distributional VBB security is only possible if we consider strictly weaker notions of correctness such as weak functionality preservation or computational functionality preservation (cf. Section 5.3).*

For any pattern $\mathsf{pat} \in \{0,1,*\}^n$, define $\mathsf{pat}^{-1}(*) := \{j \mid \mathsf{pat}_j = *\}$ the *positions* of the wildcards and let $\mathbf{b} \in \{0,1\}^{n-w}$ denote the fixed bits of pat.

Theorem 7. *The above construction with field size q is $\epsilon(n)$-Statistically Distributional VBB secure for any distribution over patterns with $w \leq n$ wildcards such that $H_\infty(\mathbf{b}|\mathsf{pat}^{-1}(*)) \geq (w+1)\log(q) + 2\log(1/\epsilon(n)) + 1$*

Corollary 1. *Fix any $\delta \in [0,1)$. The above construction can be used to satisfy Statistical Distributional VBB security for any distribution over patterns with $w = n^\delta$ wildcards such that $H_\infty(\mathbf{b}|\mathsf{pat}^{-1}(*)) \geq n^{1-\gamma}$ for some $\gamma < 1 - \delta$.*

Proof. Let $k = w + 1$ and $\mathbb{F}_q$ be a field of size q. It suffices to show the following two distributions are $\epsilon(n)$-statistically close

$$(F, A^{(1)}, \ldots, A^{(n)}, \mathcal{P}(\mathsf{pat})) \quad \text{and} \quad (U, U_1, \ldots, U_n, \mathcal{P}(\mathsf{pat}))\,,$$

where $\mathcal{P}$ is any fixed one bit predicate over patterns, $F, A^{(1)}, \ldots, A^{(n)}$ are the matrices output by Obf with field size q on pattern pat where pat has w wildcard positions, U is a uniformly random matrix in $\mathbb{F}_q^{k \times k}$, and $U_1, \ldots, U_n$ are uniformly random rank 1 matrices in $\mathbb{F}_q^{k \times k}$.

We start with the following lemma.

Lemma 7. *Let $\epsilon(n)$ be a function. For any one bit predicate $\mathcal{P}$ over patterns, the following distributions are $\epsilon(n)$-statistically close*

- *Distribution $\mathcal{D}_0$. Sample* pat *with w wildcards such that the fixed coefficients $\mathbf{b} \in \{0,1\}^{n-w}$ (where b_i denotes the ith bit of $\mathbf{b}$) of* pat *have min-entropy $H_\infty(\mathbf{b}|\mathsf{pat}^{-1}(*)) \geq (w+1)\log(q) + 2\log(1/\epsilon(n)) + 1$. For $i = 1, \ldots, n-w$, sample independently random vectors $v_i \in \mathbb{F}_q^k$. Output*

$$v_1, \ldots, v_{n-w}, \sum_{i \in [n-w]} b_i v_i, \mathsf{pat}^{-1}(*), \mathcal{P}(\mathsf{pat}).$$

- *Distribution $\mathcal{D}_1$. This is the same as $\mathcal{D}_0$, except replace $\sum_{i \in [n-w]} b_i v_i$ with uniformly random $u \leftarrow \mathbb{F}_q^k$. Output*

$$v_1, \ldots, v_{n-w}, u, \mathsf{pat}^{-1}(*), \mathcal{P}(\mathsf{pat}).$$

Proof. We show indistinguishability of $\mathcal{D}_0$ and $\mathcal{D}_1$ by introducing two additional distributions, $\mathcal{F}_0$, and $\mathcal{F}_1$. We claim that for any fixed Boolean predicate $\mathcal{P}'$, the following two distributions are $\epsilon(n)$-statistically close:

- Distribution $\mathcal{F}_0$. Sample a random $\mathbf{b} \in \{0,1\}^{n-w}$ with $H_\infty(\mathbf{b}) \geq (w+1)\log(q) + 2\log(1/\epsilon(n)) + 1$. For $i = 1, \ldots, n-1$, sample independently random vectors $v_i \in \mathbb{F}_q^k$. Output

$$v_1, \ldots, v_{n-w}, \sum_{i \in [n-w]} b_i v_i, \mathcal{P}'(\mathbf{b}).$$

- Distribution $\mathcal{F}_1$. Same as $\mathcal{F}_0$, except $\sum_{i \in [n-w]} b_i v_i$ is replaced with random $u \in \mathbb{F}_q^k$. Output

$$v_1, \ldots, v_{n-w}, u, \mathcal{P}'(\mathbf{b}).$$

Define the hash function family

$$h_{v_1, \ldots, v_{n-w}}(\mathbf{b}) = \sum_{i \in [n-w]} b_i v_i.$$

This hash function family is 2-universal, since for $\mathbf{b} \neq \mathbf{b}'$,

$$\Pr_{v_1, \ldots, v_{n-w}}[h_{v_1, \ldots, v_{n-w}}(\mathbf{b}) = h_{v_1, \ldots, v_{n-w}}(\mathbf{b}')] = \frac{1}{q^k}.$$

Furthermore, its range size is q^k. Then since $\mathcal{P}'(\mathbf{b})$ leaks at most 1 bit of the entropy of $\mathbf{b}$, and $H_\infty(\mathbf{b}) - 1 \geq k\log(q) + 2\log(1/\epsilon(n))$, by the leftover hash lemma (Lemma 1), it holds that the distributions $\mathcal{F}_0$ and $\mathcal{F}_1$ are $\epsilon(n)$-statistically close.

Now, we use these distributions $\mathcal{F}_i$ to show that $\mathcal{D}_0$ and $\mathcal{D}_1$ are $\epsilon(n)$-statistically close. Assume towards contradiction that an adversary can distinguish between $\mathcal{D}_0$ and $\mathcal{D}_1$ with non-negligible advantage for some predicate $\mathcal{P}$. Note that by simply hard-coding in the wildcard positions, we can turn predicate $\mathcal{P}\colon \{0,1,*\}^n \to \{0,1\}$ (where the number of $*$ is fixed to be w) into a predicate into a predicate $\mathcal{P}'\colon \{0,1\}^{n-w} \to \{0,1\}$. Then, there must exist some setting of the wildcard positions for which the resulting $\mathcal{P}'$ is a distinguishing predicate for $\mathcal{F}_0$ and $\mathcal{F}_1$, which is a contradiction. □

To complete the proof, we show how to post-process the distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ from Lemma 7. The following procedure Proc outputs $(F, A^{(1)}, \ldots, A^{(n)}, \mathcal{P}(\mathsf{pat}))$ on input distribution $\mathcal{D}_0$, and outputs $(U, U_1, \ldots, U_n, \mathcal{P}(\mathsf{pat}))$ on input distribution $\mathcal{D}_1$.

Before explaining Proc, we re-write the sampling procedure in Obf, giving each element an explicit name. First, we can re-write B as

$$B \coloneqq \begin{pmatrix} b_{1,1} & \cdots & b_{1,k} \\ \vdots & \ddots & \vdots \\ b_{k-1,1} & \cdots & b_{k-1,k} \\ \sum_{j=1}^{k-1} r_j b_{j,1} & \cdots & \sum_{j=1}^{k-1} r_j b_{j,k} \end{pmatrix}$$

Now, for each $i \in S_0 \cup S_1$, we can imagine sampling $c_1^{(i)}, \ldots, c_k^{(i)}, d_1^{(i)}, \ldots, d_k^{(i)}$ all uniformly at random from $\mathbb{F}_q$ and letting

$$A^{(i)} \coloneqq \begin{pmatrix} c_1^{(i)} \\ \vdots \\ c_k^{(i)} \end{pmatrix} \begin{pmatrix} d_1^{(i)} & \cdots & d_k^{(i)} \end{pmatrix}, \qquad \forall i \in S_0 \cup S_1.$$

Finally, for each $i \in S_*$, we can imagine sampling $c_1^{(i)}, \ldots, c_{k-1}^{(i)}, d_1^{(i)}, \ldots, d_k^{(i)}$ all uniformly at random from $\mathbb{F}_q$ and letting

$$A^{(i)} \coloneqq \begin{pmatrix} c_1^{(i)} \\ \vdots \\ c_{k-1}^{(i)} \\ \sum_{j=1}^{k-1} r_j c_j^{(i)} \end{pmatrix} \begin{pmatrix} d_1^{(i)} & \cdots & d_k^{(i)} \end{pmatrix}, \qquad \forall i \in S_*.$$

It will be convenient to write $F = \begin{pmatrix} (F)_1 & \cdots & (F)_k \end{pmatrix}$, where the column $(F)_t$ for $t \in [k]$ is as follows:

$$(F)_t = \begin{pmatrix} b_{1,t} - \sum_{i \in S_1} c_1^{(i)} d_t^{(i)} \\ \vdots \\ b_{k-1,t} - \sum_{i \in S_1} c_{k-1}^{(i)} d_t^{(i)} \\ \sum_{j=1}^{k-1} r_j b_{j,t} - \sum_{i \in S_1} c_k^{(i)} d_t^{(i)} \end{pmatrix} = \begin{pmatrix} b'_{1,t} \\ \vdots \\ b'_{k-1,t} \\ \sum_{j=1}^{k-1} r_j b'_{j,t} + \sum_{j=1}^{k-1} r_j \sum_{i \in S_1} c_j^{(i)} d_t^{(i)} - \sum_{i \in S_1} c_k^{(i)} d_t^{(i)} \end{pmatrix}.$$

where the $b'_{s,t}$ are uniformly random field elements.

We will now describe the procedure Proc. The input to Proc is a sample from $\mathcal{D}_b$ of the form

$$v_1, \ldots, v_{n-w}, v, \mathsf{pat}^{-1}(*), \mathcal{P}(\mathsf{pat}),$$

where v is either $\sum_{i \in [n-w]} b_i v_i$ (if drawn from $\mathcal{D}_0$) or the uniform vector u (if drawn from $\mathcal{D}_1$). The procedure is as follows:

1. Sample $r_1, \ldots, r_{k-1} \leftarrow \mathbb{F}_q$ uniformly at random.

2. Let $p[i]$ denote the index $j \in [n]$ of the i-th fixed bit of pat. Then for all $i \in [n-w]$, we let

$$A^{(p[i])} = \begin{pmatrix} c_1^{(p[i])} \\ \vdots \\ c_k^{(p[i])} \end{pmatrix} \begin{pmatrix} d_1^{(p[i])} & \cdots & d_k^{(p[i])} \end{pmatrix} \qquad \forall i \in [n-w].$$

   where $c_1^{(p[i])}, \ldots, c_k^{(p[i])}$ are sampled uniformly at random from $\mathbb{F}_q$, and for each $t \in [k]$, setting

$$d_t^{(p[i])} \coloneqq (v_i)_t \left(\sum_{j=1}^{k-1} r_j c_j^{(p[i])} - c_k^{(p[i])} \right)^{-1}.$$

   Not that since each $(v_i)_t$ is sampled uniformly at random, each $d_t^{(p[i])}$ is distributed uniformly.

3. Set $S_* \coloneqq \mathsf{pat}^{-1}(*)$. For each wildcard slot $i \in S_*$, we sample $c_1^{(i)}, \ldots, c_{k-1}^{(i)}$ and $d_1^{(i)}, \ldots, d_k^{(i)}$ uniformly at random. Then we set

$$A^{(i)} \coloneqq \begin{pmatrix} c_1^{(i)} \\ \vdots \\ c_{k-1}^{(i)} \\ \sum_{j=1}^{k-1} r_j c_j^{(i)} \end{pmatrix} \begin{pmatrix} d_1^{(i)} & \cdots & d_k^{(i)} \end{pmatrix} \qquad \forall i \in S_*.$$

4. Finally, generate uniformly random $b'_{s,t}$ for all $s \in [k-1], t \in [k]$ and set B' (column t highlighted) as

$$B' = \begin{pmatrix} \cdots & b'_{1,t} & \cdots \\ & \vdots & \\ \cdots & b'_{k-1,t} & \cdots \\ \cdots & \sum_{j=1}^{k-1} r_j b'_{j,t} + (v)_t & \cdots \end{pmatrix}.$$

5. Output $B', A^{(1)}, \ldots, A^{(n)}, \mathcal{P}(\mathsf{pat})$.

When $v = \sum_{i \in [n-w]} b_i v_i$, the matrix B' is distributed exactly as F since we have that

$$(v)_t = \sum_{i \in [n-w]} b_i (v_i)_t = \sum_{i \in [n-w]} b_i d_t^{(p[i])} \left(\sum_{j=1}^{k-1} r_j c_j^{(p[i])} - c_k^{(p[i])} \right) = \sum_{j=1}^{k-1} r_j \sum_{i \in S_1} d_t^{(i)} c_j^{(i)} - \sum_{i \in S_1} d_t^{(i)} c_k^{(i)}.$$

Furthermore, the matrices $A^{(1)}, \ldots, A^{(n)}$ are generated exactly as they are in the honest procedure.

When $v = u$, F is a uniformly random matrix U. Furthermore, there are at most $w \leq k-1$ wildcards and the only information about the $r_1, \ldots, r_{k-1}$ is now in the $A^{(i)}$ matrices for $i \in \mathsf{pat}^{-1}(*)$. Since we only see $k-1$ samples of the form $\sum_{j=1}^{k-1} r_j c_j^{(i)}$, we can replace these values with uniformly random values $c_k^{(i)}$ without any change in the distribution. □

5.3 Computational Functionality Preservation

We now consider the notion of computational functionality preservation from Definition 1, which is strictly weaker than strong functionality preservation, and strictly stronger than weak functionality preservation.[18] Refer to Section 1.2.7 for general discussion motivating this definition.

Remark 7. *For the setting of conjunction obfuscation, computational functionality preservation combined with distributional VBB security imply that a computationally bounded adversary can never find an accepting input to the obfuscated program.*[19] *If the adversary can find an accepting input to the program that actually matches the hidden pattern* pat*, the adversary can learn a predicate on* pat*, violating distributional VBB. If they find an accepting input to the program that does not match the hidden pattern, they violate computational functionality preservation.*

We show how to achieve computational functionality preservation under the discrete log assumption (for patterns with bounded number of wildcards) with a small tweak to the above obfuscation.

Placing a Row in the Exponent. We show that if we place the bottom row of each matrix in the exponent of some group $\mathbb{G} = \langle g \rangle$, then we can base computational functionality preservation of this scheme on the hardness of solving discrete log in $\mathbb{G}$. The only difference between this construction and our previous construction is that we move some elements into the exponent, so it will still satisfy statistical distributional VBB security. We will slightly modify the evaluation procedure, but we note that this will not change the actual matrices and will therefore not affect security. We only describe the differences between this construction and our standard information-theoretic construction.

- **Modification 1:** All of the matrices $F, A^{(1)}, \ldots, A^{(n)}$ have their last row encoded in the exponent of the group.
- **Modification 2:** On evaluation, we first check if $\text{rank}(\overline{F} + \sum_{i|x_i=1} \overline{A}^{(i)}) = k - 1$ (i.e. the first $k-1$ rows have full row rank), and if not, immediately reject.

Our security proof will use a reduction from the *representation problem*, introduced by Brands [Bra94], which we denote as FIND-REP following [Pei06].

> **Instance:** A group $\mathbb{G}$ of order q, and uniformly random $g^{s_1}, \ldots, g^{s_n} \leftarrow \mathbb{G}$.
>
> **Problem:** Find non-trivial $d_1, \ldots, d_n \in \mathbb{Z}_q$ such that $g^{\sum_{i=1}^n d_i s_i} = g^0$.

Brands [Bra94] proves that solving FIND-REP in $\mathbb{G}$ is as hard as solving discrete log in $\mathbb{G}$. For completeness, we recall Brands's proof.

Proposition 1 (Proposition 3, [Bra94])**.** *If there exists a PPT algorithm $\mathcal{A}$ that solves* FIND-REP *with non-negligible probability in $\mathbb{G}$, then there exists a PPT algorithm $\mathcal{A}'$ that solves discrete log in $\mathbb{G}$ with non-negligible probability.*

[18] To see this informally, consider any obfuscation scheme for an evasive functionality given by $(\mathsf{Obf}, \mathsf{Eval})$ that achieves weak functionality preservation. Now define $(\mathsf{Obf}', \mathsf{Eval}')$ where $\mathsf{Obf}'(C)$ samples a random y from the input space and then outputs $\mathsf{Obf}(C), y$. Then $\mathsf{Eval}(\mathsf{Obf}', x)$ returns $\mathsf{Eval}(\mathsf{Obf}, x)$ if $x \neq y$, but returns 1 if $x = y$. It is not hard to see that this scheme still satisfies weak functionality preservation, but now an adversary can easily tell that functionality preservation is violated at y, so computational functionality preservation is violated.

[19] We note that this is reminiscent of the notion of input-hiding obfuscation [BBC+14], but different in that we require the adversary cannot find an accepting input for the *obfuscated* circuit rather than the original circuit.

Proof. On a discrete log challenge g^a, $\mathcal{A}'$ samples uniformly random $s_1, \ldots, s_n, t_1, \ldots, t_n \leftarrow \mathbb{Z}_q$ and runs the FIND-REP algorithm $\mathcal{A}$ on $g^{s_1+at_1}, \ldots, g^{s_n+at_n}$. If $\mathcal{A}$ is successful, it outputs $d_1, \ldots, d_n$ satisfying $g^{\sum_{i=1}^n d_i(s_i+at_i)} = g^0$. Then $\mathcal{A}'$ outputs $-(\sum_{i=1}^n d_i s_i)/(\sum_{i=1}^n d_i t_i)$.

Conditioned on $\mathcal{A}$ solving FIND-REP, $\mathcal{A}'$ solves discrete log with overwhelming probability. We note that this algorithm only fails if $\sum_{i=1}^n d_i t_i = 0$, but for any $t_1, \ldots, t_n$ we can pick a corresponding $s_1, \ldots, s_n$ that leaves the view of $\mathcal{A}$ unchanged. Thus $\sum_{i=1}^n d_i t_i = 0$ is a distributed uniformly in $\mathbb{Z}_q$, and is non-zero with probability $(q-1)/q$. □

Now we prove a theorem similar to Theorem 7, but with different parameters than Corollary 1.

Theorem 8. *Fix any $\delta \in [0, \frac{1}{2})$. Assuming discrete log, this construction satisfies computational functionality preservation for any distribution over patterns with $w = n^\delta$ wildcards such that $H_\infty(\mathbf{b}|\mathsf{pat}^{-1}(*)) \geq n^{1-\epsilon}$ for some $\epsilon < 1 - 2\delta$.*

Proof. We prove that a PPT adversary that can find some point x for which $f_{\mathsf{pat}}(x) \neq \mathsf{Obf}(f_{\mathsf{pat}})(x)$, even given $\mathsf{Obf}(f_{\mathsf{pat}})$, can solve discrete log in $\mathbb{G}$. We break up the analysis into two cases: we denote inputs x for which $f_{\mathsf{pat}}(x) = 1$ and $\mathsf{Obf}(f_{\mathsf{pat}})(x) = 0$ as false negatives, and denote inputs for which $f_{\mathsf{pat}}(x) = 0$ and $\mathsf{Obf}(f_{\mathsf{pat}})(x) = 1$ as false positives.

For $\delta \in [0, 1/2)$ we pick $\delta' > \delta$ and set the field size q used in the construction to $2^{n^{\delta'}}$.

Lemma 8. *For our choice of parameters $q = 2^{n^{\delta'}}$ and $w = n^\delta$ where $\delta' > \delta$, with overwhelming probability our construction has no false negatives.*

Proof. For any x where $f_{\mathsf{pat}}(x) = 1$, $\mathsf{Obf}(f_{\mathsf{pat}})(x)$ can only evaluate to 0 if

$$\text{rank}\left(\overline{B} + \sum_{i|x_i=1,\mathsf{pat}_i=*} \overline{A}^{(i)}\right) < k-1.$$

Recall from the construcion that $\overline{B}$ is sampled as a uniformly random matrix, and for i where $\mathsf{pat}_i = *$, $\overline{A}^{(i)}$ is sampled as a uniformly random rank 1 matrix. Thus, each of the 2^{n^δ} possible $(k-1) \times k$ subset sums is distributed as uniformly random $(k-1) \times k$ matrix, and is thus rank deficient with probability at most $\frac{k-1}{q^2}$. Since we set q to be at least $2^{n^{\delta'}}$ for $\delta' > \delta$, the probability that any of these subset sum matrices is rank deficient is at most $\frac{(k-1)\cdot 2^{n^\delta}}{q^2} = \mathsf{negl}(n)$. □

Thus with overwhelming probability, an adversary that finds an x where $f_{\mathsf{pat}}(x) \neq \mathsf{Obf}(f_{\mathsf{pat}})(x)$ must return a false positive. We show that finding a false positive is as hard as solving FIND-REP.

Lemma 9. *If there exists an algorithm $\mathcal{A}$ that finds a false positive with non-negligible probability, there exists an algorithm $\mathcal{A}'$ that solves FIND-REP with non-negligible probability.*

Proof. On input $g^{s_1}, \ldots, g^{s_n}$, $\mathcal{A}'$ constructs an obfuscation for a pattern pat with $w = n^\delta$ wildcards drawn from an arbitrary distribution. Given pat, define the same sets S_0, S_1, and S_* and as before, let $k = w + 1$. Note that throughout this proof, when we add/subtract matrices that include group elements, we multiply/divide the group element components of the matrices. Likewise, when we multiply a vector of group elements by a scalar, we actually raise each group element to the appropriate power. $\mathcal{A}$ constructs the obfuscation as follows.

- Let $r \in g^{\mathbb{Z}_q^{1\times(k-1)}} = [\dots g^{s_j} \dots]$ for $j \in S_*$, draw $\overline{B} \leftarrow \mathbb{Z}_q^{(k-1)\times k}$, and let

$$B := \begin{pmatrix} \overline{B} \\ r \cdot \overline{B} \end{pmatrix}$$

- For each $i \in S_0 \cup S_1$, sample a uniformly random rank 1 matrix $\overline{A}^{(i)} \in \mathbb{F}_q^{(k-1)\times k}$, and let

$$A^{(i)} := \begin{pmatrix} \overline{A}^{(i)} \\ g^{s_i} \cdot \overline{A}_1^{(i)} \end{pmatrix}$$

- For each $i \in S_*$, sample $c_i \leftarrow \mathbb{F}_q^{k-1}$ and $d_i \leftarrow \mathbb{F}_q^{1\times k}$, and let

$$A^{(i)} := \begin{pmatrix} c_i \\ r \cdot c_i \end{pmatrix} \cdot d_i$$

- Define

$$F := B - \sum_{i \in S_1} A^{(i)}$$

  and output $(F, A^{(1)}, \dots, A^{(n)})$

So $\mathcal{A}'$ sends $(F, A^{(1)}, \dots, A^{(n)}, \mathsf{pat})$ to $\mathcal{A}$ and if $\mathcal{A}$ is successful, $\mathcal{A}'$ receives back a set T with the following properties:

- $\det(F + \sum_{i \in T} A^{(i)}) = 0$
- $\det(\overline{F} + \sum_{i \in T} \overline{A}^{(i)}) \neq 0$
- $T \setminus S_* \neq S_1$

The determinant polynomial reduces to a linear combination of the elements in the last row of $F + \sum_{i \in T} A^{(i)}$. By the second property above, this linear combination is not identically zero. Now $\mathcal{A}'$ will multiply by the random values it multiplied the group elements by to recover a linear combination over $s_1, \dots, s_n$ that evaluates to zero, by the first property above. It then submits this linear combination to the FIND-REP challenger.

So it just remains to show that this final linear combination is not identically zero. As in our weak functionality preservation proof, we can re-write the summation as

$$F + \sum_{i \in T} A^{(i)} = \underbrace{B + \sum_{i \in T \cap S_*} A^{(i)}}_{B'} + \underbrace{\sum_{i \in T \cap S_0} A^{(i)} - \sum_{i \in ([n] \setminus T) \cap S_1} A^{(i)}}_{A'}.$$

By the third property above, there exists some i such that A' includes the matrix $A^{(i)}$. We show that with overwhelming probability, this implies that there is some setting of $s_1, \dots, s_n$ that produces a non-zero evaluation, which shows that the final linear combination must not be identically zero.

We condition on the fact that with overwhelming probability, for each of the 2^{n^δ} possible sets $T \cap S_*$, and each $i \notin S_*$, the row span of $A^{(i)}$ is outside of the row span of $\overline{B}'$. Indeed, this fails to happen with probability at most

$$\frac{n2^{n^\delta}}{q} = \mathsf{negl}(n)$$

Thus since $\underline{A}'$ must include a row from some $A^{(i)}$, we conclude that the row $\underline{B}' + \underline{A}'$ could be anything in the entire k dimensional space, depending on the values of $s_1, \ldots, s_n$. In particular it could be outside of the $k-1$ dimensional space spanned by $\overline{A}' + \overline{B}'$, in which case the determinant polynomial would evaluate to non-zero.

□

Together, Lemma 8, Lemma 9, and Corollary 1 imply that any adversary that breaks computational functionality preservation can solve discrete log in $\mathbb{G}$. □

6 Acknowledgements

We thank Daniel Wichs for sharing his proof of decisional LPN hardness (with an arbitrary one-bit predicate on the error vector) from hardness of search LPN. Wichs's proof does not appear in the paper, as we later discovered the claim follows from the proof of Lemma 5 in a work of Döttling [Döt16].

We thank Allison Bishop for providing suggestions and feedback on an early draft of this work. We also thank Zvika Brakerski, Brent Carmer, Benjamin Fuller, Yuval Ishai, Aayush Jain, Luke Johnson, Tal Malkin, and Mariana Raykova for helpful discussions. This material is based upon work supported by the ARO and DARPA under Contract No. W911NF-15-C-0227. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ARO and DARPA.

References

[AAB15] Benny Applebaum, Jonathan Avron, and Christina Brzuska. Arithmetic cryptography: Extended abstract. In Tim Roughgarden, editor, *ITCS 2015*, pages 143–151. ACM, January 2015.

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 528–556. Springer, Heidelberg, March 2015.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Heidelberg, December 2011.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *ICALP 2011, Part I*, volume 6755 of *LNCS*, pages 403–415. Springer, Heidelberg, July 2011.

[AGIS14] Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 14*, pages 646–658. ACM Press, November 2014.

[AIK09] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. *Journal of Cryptology*, 22(4):429–469, October 2009.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 308–326. Springer, Heidelberg, August 2015.

[AS17] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 152–181. Springer, Heidelberg, April / May 2017.

[BBC+14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 26–51. Springer, Heidelberg, February 2014.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. Cryptology ePrint Archive, Report 2005/015, 2005. https://ia.cr/2005/015.

[BFM15] Christina Brzuska, Pooya Farshim, and Arno Mittelbach. Random-oracle uninstantiability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 428–455. Springer, Heidelberg, March 2015.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[BGMZ18] James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In *TCC 2018*, 2018.

[BISW17] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 247–277. Springer, Heidelberg, April / May 2017.

[BKM+18] Allison Bishop, Lucas Kowalczyk, Tal Malkin, Valerio Pastro, Mariana Raykova, and Kevin Shi. A simple obfuscation scheme for pattern-matching with wildcards. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 731–752. Springer, Heidelberg, August 2018.

[BP15] Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427. Springer, Heidelberg, March 2015.

[BR13] Zvika Brakerski and Guy N. Rothblum. Obfuscating conjunctions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 416–434. Springer, Heidelberg, August 2013.

[BR17] Zvika Brakerski and Guy N. Rothblum. Obfuscating conjunctions. *Journal of Cryptology*, 30(1):289–320, January 2017.

[Bra94] Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 302–318. Springer, Heidelberg, August 1994.

[BS16] Mihir Bellare and Igors Stepanovs. Point-function obfuscation: A framework and generic constructions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 565–594. Springer, Heidelberg, January 2016.

[BTVW17] Zvika Brakerski, Rotem Tsabary, Vinod Vaikuntanathan, and Hoeteck Wee. Private constrained PRFs (and more) from LWE. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 264–302. Springer, Heidelberg, November 2017.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 1–30. Springer, Heidelberg, March 2015.

[BVWW16] Zvika Brakerski, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Obfuscating conjunctions under entropic ring LWE. In Madhu Sudan, editor, *ITCS 2016*, pages 147–156. ACM, January 2016.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Heidelberg, December 2013.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 480–499. Springer, Heidelberg, August 2014.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 455–469. Springer, Heidelberg, August 1997.

[CC17] Ran Canetti and Yilei Chen. Constraint-hiding constrained PRFs for NC^1 from LWE. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 446–476. Springer, Heidelberg, April / May 2017.

[CCC+16] Yu-Chi Chen, Sherman S. M. Chow, Kai-Min Chung, Russell W. F. Lai, Wei-Kai Lin, and Hong-Sheng Zhou. Cryptography for parallel RAM from indistinguishability obfuscation. In Madhu Sudan, editor, *ITCS 2016*, pages 179–190. ACM, January 2016.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 489–508. Springer, Heidelberg, April 2008.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493. Springer, Heidelberg, August 2013.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 468–497. Springer, Heidelberg, March 2015.

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 72–89. Springer, Heidelberg, February 2010.

[DGL+16] Dana Dachman-Soled, S. Dov Gordon, Feng-Hao Liu, Adam O'Neill, and Hong-Sheng Zhou. Leakage-resilient public-key encryption from obfuscation. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part II*, volume 9615 of *LNCS*, pages 101–128. Springer, Heidelberg, March 2016.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 621–630. ACM Press, May / June 2009.

[Döt16] Nico Döttling. Low noise LPN: key dependent message secure public key encryption an sample amplification. *IET Information Security*, 10(6):372–385, 2016.

[FRS17] Rex Fernando, Peter M. R. Rasmussen, and Amit Sahai. Preventing CLT attacks on obfuscation with linear overhead. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 242–271. Springer, Heidelberg, December 2017.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527. Springer, Heidelberg, March 2015.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 230–240. Tsinghua University Press, January 2010.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th FOCS*, pages 612–621. IEEE Computer Society Press, 2017.

[GMM+16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 241–268. Springer, Heidelberg, October / November 2016.

[GPS16] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 579–604. Springer, Heidelberg, August 2016.

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 194–213. Springer, Heidelberg, February 2007.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

[HOSS18] Carmit Hazay, Emmanuela Orsini, Peter Scholl, and Eduardo Soria-Vazquez. TinyKeys: A new approach to efficient multi-party computation. Cryptology ePrint Archive, Report 2018/208, 2018. https://ia.cr/2018/208.

[IPS09] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Secure arithmetic computation with no honest majority. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 294–314. Springer, Heidelberg, March 2009.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 663–680. Springer, Heidelberg, December 2012.

[KY18] Ilan Komargodski and Eylon Yogev. Another step towards realizing random oracles: Non-malleable point obfuscation. Cryptology ePrint Archive, Report 2018/149, 2018. https://ia.cr/2018/149.

[lec] Lecture notes: Extractors and the leftover hash lemma. https://www.cs.bu.edu/~reyzin/teaching/s11cs937/notes-leo-1.pdf.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 28–57. Springer, Heidelberg, May 2016.

[LPS04] Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 20–39. Springer, Heidelberg, May 2004.

[LT17] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 630–660. Springer, Heidelberg, August 2017.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In Irit Dinur, editor, *57th FOCS*, pages 11–20. IEEE Computer Society Press, October 2016.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Heidelberg, August 2011.

[MO14] Antonio Marcedone and Claudio Orlandi. Obfuscation $\rightarrow$ (IND-CPA security $\not\rightarrow$ circular security). In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 77–90. Springer, Heidelberg, September 2014.

[MPS16] Antonio Marcedone, Rafael Pass, and Abhi Shelat. Bounded KDM security from iO and OWF. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 571–586. Springer, Heidelberg, August / September 2016.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 629–658. Springer, Heidelberg, August 2016.

[MZ18] Fermi Ma and Mark Zhandry. The mmap strikes back: Obfuscation and new multilinear maps immune to CLT13 zeroizing attacks. In *TCC 2018*, 2018.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003.

[Pei06] Chris Peikert. On error correction in the exponent. In Shai Halevi and Tal Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 167–183. Springer, Heidelberg, March 2006.

[PS18] Chris Peikert and Sina Shiehian. Privately constraining and programming PRFs, the LWE way. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 675–701. Springer, Heidelberg, March 2018.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.

[Ska13] Matthew Skala. Hypergeometric tail inequalities: ending the insanity. *arXiv preprint arXiv:1311.5939*, 2013.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

[Wee05] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 523–532. ACM Press, May 2005.

[Wil14] Virginia Vassilevska Williams. Multiplying matrices in $O(n^{2.373})$ time. 2014.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th FOCS*, pages 600–611. IEEE Computer Society Press, 2017.

[YZ16] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise LPN. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 214–243. Springer, Heidelberg, August 2016.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 439–467. Springer, Heidelberg, April 2015.

A Reduction for Structured Error LPN/RLC

In this section, we establish the hardness of decisional structured exact LPN based on the regular exact LPN.

Theorem (Restatement of Theorem 5)**.** *Fix constants $\epsilon, \delta \in [0, 1/2)$ and constant $\rho \in [0, 1)$. If $\mathsf{DxLPN}(n^\epsilon, n, \rho)$ is hard, then $\mathsf{DxSLPN}(n + n^\delta, 2n, \rho)$ is hard.*

We will actually prove a more general reduction between extensions of DxLPN and DxSLPN to larger fields (Theorem 9, cf. below)[20]. We believe this reduction to be of independent interest, which is why we state it for random linear codes rather than LPN; the LPN result we use for security follows from plugging in $q = 2$.

A.1 Random Linear Code Problems

Let us define the exact Random Linear Code (RLC) and exact structured RLC problems. Let m, q be integers and $\rho \in [0, 1]$. We denote $\chi_\rho^m(\mathbb{F}_q)$ the distribution on $\mathbb{F}^n$ where a uniformly randomly selected set of $\lfloor \rho m \rfloor$ coordinates are set to a uniformly random non-zero field element and the rest of the coordinates are zero. In the case where $q = 2$, we have $\chi_\rho^m(\mathbb{F}_2) = \chi_\rho^m$ as defined in Section 2.3.

Definition 8 (Exact RLC)**.** *The (dual) Decisional Exact Random Linear Code (DxRLC) problem with parameters n, m, γ, ρ, denoted $\mathsf{DxRLC}(n, m, \gamma, \rho)$, is hard if, for every probabilistic polynomial-time (in n) algorithm $\mathcal{A}$ running, there exists a negligible function μ such that*

$$\left| \Pr_{B,e}[\mathcal{A}(B, B \cdot e) = 1] - \Pr_{B,u}[\mathcal{A}(B, u) = 1] \right| \leq \mu(n)\,,$$

where $q = 2^{n^\gamma}$, $\mathbb{F}_q$ is a field of size q, $B \leftarrow \mathbb{F}_q^{(m-n)\times m}$, $e \leftarrow \chi_\rho^m(\mathbb{F}_q)$, and $u \leftarrow \mathbb{F}_q^{m-n}$.

Remark 8. *We can define a non-exact variant of the above problem, where the error vector does not have a fixed ρm number of non-zero entries, but rather each entry is non-zero independently with probability ρ. This problem has been considered previously in the literature under the name RLC [IPS09, AAB15].*

Definition 9 (Structured error RLC)**.** *The (dual) Decisional Exact Structured Random Linear Code (DxSLPN) problem with parameters $n, 2m, \gamma, \rho$, denoted $\mathsf{DxSLPN}(n, 2m, \gamma, \rho)$, is hard if for every probabalistic polynomial-time (in n) algorithm $\mathcal{A}$, there exists a negligible function μ such that*

$$\left| \Pr_{B,e}[\mathcal{A}(B, B \cdot e) = 1] - \Pr_{B,u}[\mathcal{A}(B, u) = 1] \right| \leq \mu(n)$$

where $q = 2^{n^\gamma}$, $\mathbb{F}_q$ is a field of size q, $B \leftarrow \mathbb{F}_q^{(2m-n)\times 2m}$, $e \leftarrow \sigma(\chi_\rho^m(\mathbb{F}_q))$, and $u \leftarrow \mathbb{F}_q^{2m-n}$.

The rest of this Section proves the following theorem.

Theorem 9. *Fix constants $\epsilon, \delta, \gamma, \rho \in [0, 1)$ such that $2\epsilon + \gamma < 1$ and $2\delta + \gamma < 1$. If $\mathsf{DxRLC}(n^\epsilon, n, \gamma, \rho)$ is hard, then $\mathsf{DxSLPN}(n + n^\delta, 2n, \gamma, \rho)$ is hard.*

[20] Setting $\gamma = 0$ in Theorem 9 gives the exact statement of Theorem 5.

A.2 Preliminary Lemmas

The proof will use the following technical lemmatas.

Lemma 10. *Let $i, j, k \in \mathbb{N}$. It holds that*

$$\Pr\left[AB = 0 \middle| \begin{array}{l} A \leftarrow \mathbb{F}_q^{i\times j} \\ B \leftarrow \mathbb{F}_q^{j\times k} \end{array}\right] \leq \frac{1}{q^{ki}} + \sum_{\ell=0}^{i-1} \frac{\binom{j}{\ell}}{q^{k\ell} q^{(i-\ell)(j-\ell)}} \,.$$

Proof. Note that if A has full rank, then each column of B is in the kernel of A independently with probability $1/q^i$. We split the probability into two terms, depending on whether or not A has full rank i.

$$\Pr\left[AB = 0 \middle| \begin{array}{l} A \leftarrow \mathbb{F}_q^{i\times j} \\ B \leftarrow \mathbb{F}_q^{j\times k} \end{array}\right] = \left(\frac{1}{q^i}\right)^k \cdot \Pr\left[\mathsf{rank}(A) = i \middle| A \leftarrow \mathbb{F}_q^{i\times j}\right] + \Pr\left[AB = 0 \middle| \begin{array}{l} A \leftarrow \mathbb{F}_q^{i\times j} \\ \mathsf{rank}(A) < i \\ B \leftarrow \mathbb{F}_q^{j\times k} \end{array}\right] .$$

Now, if $\mathsf{rank}(A) = \ell$ for $\ell \in [0, i-1]$, we can bound the probability that $AB = 0$ for a random B by $(1/q^\ell)^k$. It follows that

$$\Pr\left[AB = 0 \middle| \begin{array}{l} A \leftarrow \mathbb{F}_q^{i\times j} \\ \mathsf{rank}(A) = \ell \\ B \leftarrow \mathbb{F}_q^{j\times k} \end{array}\right] \leq \frac{1}{q^{k\ell}} \cdot \Pr\left[\mathsf{rank}(A) = \ell \middle| A \leftarrow \mathbb{F}_q^{i\times j}\right] .$$

Finally, for any $\ell \in [0, i-1]$, we can bound $\Pr\left[\mathsf{rank}(A) = \ell \middle| A \leftarrow \mathbb{F}_q^{i\times j}\right]$ by using the following argument. If the rank of A is ℓ, there must be some set S of ℓ linearly independent columns of A such that each of the other $j - \ell$ columns are in the column span of S (of cardinality $\leq q^\ell$); we then union bound over all the possibilities for S:

$$\Pr\left[\mathsf{rank}(A) = \ell \middle| A \leftarrow \mathbb{F}_q^{i\times j}\right] \leq \sum_{\substack{S\subset[j] \\ |S|=\ell}} \left(\frac{q^\ell}{q^i}\right)^{j-\ell} = \frac{\binom{j}{\ell}}{q^{(i-\ell)(j-\ell)}} \,.$$

Note that the previous equation accounts for the case where $j < \ell$ using the convention that $\binom{j}{\ell} = 0$ for $\ell > j$. □

Lemma 11. *Let $\epsilon > 0$ and X_ϵ be a random variable over a countable set $\mathcal{Y}$ such that $\Pr_{x,y\leftarrow X_\epsilon}[x = y] = (1 + \epsilon^2)/|\mathcal{Y}|$. Then it holds that the distribution induced by X_ϵ is $\epsilon/2$-statistically close to the uniform distribution over $\mathcal{Y}$.*

Adapted from [lec]. Let Y be the uniformly distributed random variable over $\mathcal{Y}$, and denote $Z = X_\epsilon - Y$. By Cauchy–Schwarz, it holds that $\|Z\|_1 \leq \|Z\|_2 \cdot \|\mathsf{sign}(Z)\|_2 \leq \|Z\|_2 \cdot |\mathcal{Y}|^{1/2}$. Now

$$\begin{aligned} \|Z\|_2^2 &= \sum_{x\in\mathcal{Y}} (\Pr[X_\epsilon = x] - \Pr[Y = x])^2 \\ &= \sum_{x\in\mathcal{Y}} \left(\Pr[X_\epsilon = x]^2 - \frac{2}{|\mathcal{Y}|}\Pr[X_\epsilon = x] + \frac{1}{|\mathcal{Y}|^2}\right) \leq \frac{\epsilon^2}{|\mathcal{Y}|} \,. \end{aligned}$$

The proof follows by observing that $\Delta(X_\epsilon, Y) = 1/2\|Z\|_1$. □

Before giving our next lemma, we have the following definitions. First, let $S \subseteq [n]$ be a subset of cardinality $k \leq n$ and $y \in \{0,1\}^k$. Let $x|_S \in \{0,1\}^{|S|}$ denote the substring formed by taking the indices of x corresponding to the elements of S. We define $X|_{S,y} \subseteq \{0,1\}^n$ to be the set of all binary strings

$$\{x \in \{0,1\}^n : x|_S = y\}$$

with the substring corresponding to the indices in S fixed to be y. Second, let $Y_0, Y_1 \in F_2^{n \times m}$ be two matrices and let $x \in \{0,1\}^n$ be a bitstring, and define

$$\sigma_x(Y_0, Y_1) \coloneqq \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ (Y_{x_1})_1 & (Y_{1-x_1})_1 & (Y_{x_2})_2 & (Y_{1-x_2})_2 & \cdots & (Y_{x_n})_n & (Y_{1-x_n})_n \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \end{pmatrix},$$

that is a matrix in $\mathbb{F}_2^{n \times 2m}$ where its $(2i-1)$th column is the ith column of Y_{x_i} and its $2i$th column if the ith column of Y_{1-x_i}. Note that the distribution σ introduced in Section 4.1 is such that

$$\sigma(\mathcal{D}) = \left\{ \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ \vdots \\ s_{2n-1} \\ s_{2n} \end{pmatrix} \middle| \begin{array}{l} x \leftarrow \{0,1\}^m \\ e' \leftarrow \mathcal{D} \\ \text{for all } i \in [n], \begin{cases} s_{2i-x_i} = e'_i \\ s_{2i-(1-x_i)} = 0 \end{cases} \end{array} \right\} = \left\{ (\sigma_x(0^{1\times m}, e'^{\top}))^{\top} \middle| \begin{array}{l} x \leftarrow \{0,1\}^m \\ e' \leftarrow \mathcal{D} \end{array} \right\}.$$

Lemma 12. *Fix constants $\epsilon, \gamma, \delta \geq 0$ such that $2\epsilon + \gamma < 1$ and $2\delta + \gamma < 1$, and let $q = 2^{n^\gamma}$. Let $c \in (0,1]$ and fix a subset $S \subset [n]$ of cardinality $n - cn$ and a string $y \in \{0,1\}^{n-cn}$. Denote by $\mathcal{V}_{S,y}$ the set*

$$\mathcal{V}_{S,y} \coloneqq \left\{ V \middle| \begin{array}{l} V \leftarrow \sigma_x(Y, 0^{(n^\epsilon + n^\delta) \times n}) \\ Y \leftarrow \mathbb{F}_q^{(n^\epsilon + n^\delta) \times n} \\ x \leftarrow X|_{S,y} \end{array} \right\}.$$

The following distributions are statistically indistinguishable:

$$\left\{ \left(\begin{bmatrix} B \\ RB \end{bmatrix} + \begin{bmatrix} 0 \\ V \end{bmatrix}, S, y \right) \middle| \begin{array}{l} B \leftarrow \mathbb{F}_q^{(n-n^\epsilon) \times 2n} \\ R \leftarrow \mathbb{F}_q^{(n^\epsilon + n^\delta) \times (n-n^\epsilon)} \\ V \leftarrow \mathcal{V}_{S,y} \end{array} \right\},$$

and

$$\left\{ (U, S, y) \middle| \; U \leftarrow \mathbb{F}_q^{(n+n^\delta) \times 2n} \right\}.$$

Proof. By Lemma 11, it suffices to show that there exists a negligible function ϵ such that

$$\Pr\left[\begin{bmatrix} B_0 \\ R_0 B_0 + V_0 \end{bmatrix} = \begin{bmatrix} B_1 \\ R_1 B_1 + V_1 \end{bmatrix} \right] = \frac{1 + \epsilon(n)}{q^{2n(n+n^\delta)}},$$

where the probability is over the choices of $B_0, B_1 \leftarrow \mathbb{F}_q^{(n-n^\epsilon) \times 2n}$, $R_0, R_1 \leftarrow \mathbb{F}_q^{(n^\epsilon + n^\delta) \times (n-n^\epsilon)}$, $V_0, V_1 \leftarrow \mathcal{V}_{S,y}$.

First, note that $\Pr_{B_0,B_1}[B_0 = B_1] = 1/q^{2n(n-n^\epsilon)}$. Hence, it holds that

$$\Pr\left[\begin{bmatrix} B_0 \\ R_0B_0 + V_0 \end{bmatrix} = \begin{bmatrix} B_1 \\ R_1B_1 + V_1 \end{bmatrix}\right] = \Pr\left[\begin{bmatrix} (B_0 - B_1) \\ (R_0B_0 - R_1B_1) + (V_0 - V_1) \end{bmatrix} = 0\right]$$
$$= \frac{1}{q^{2n(n-n^\epsilon)}} \cdot \underbrace{\Pr\left[RB + (V_0 - V_1) = 0\right]}_{:=p(n)},$$

where the probabilities are over the choices of $B_0, B_1, R_0, R_1, V_0, V_1$ with the same distributions as above, $B \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times 2n}$, and $R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)}$.

Denote $V^- := \{V_0 - V_1 : V_0, V_1 \leftarrow \mathcal{V}_{S,y}\}$. Let $i \in [n]$, and consider the distribution of the columns V_{2i-1}, V_{2i} of the elements of V^-. By definition, if $i \in S$, the $(2i - y_i)$th column of the matrices in $\mathcal{V}_{S,y}$ is 0 and the $(2i - (1 - y_i))$th column is randomly distributed. If $i \notin S$, denote $x_i^{(0)}$ and $x_i^{(1)}$ the ith bit of the bitstrings used to sample $V_0, V_1 \leftarrow \mathcal{V}_{S,y}$. If $x_i^{(0)} = x_i^{(1)}$, then the $(2i - x_i^{(0)})$th column of $V := V_0 - V_1$ is 0 and the $(2i - (1 - x_i^{(0)}))$th column is uniformly randomly distributed. If $x_i^{(0)} \neq x_i^{(1)}$, both the $(2i - 1)$th and $2i$th columns of V are uniformly randomly distributed. Therefore, the matrices in V^- have $n - cn + k$ zero columns where k is distributed as a binomial distribution of probability $1/2$ with $cn = |[n] \setminus S|$ repetitions. Hence, we split the probability $p(n)$ we are interested in according to k and the $n - cn + k$ zero columns:

$$p(n) = \sum_{k=0}^{cn} \frac{\binom{cn}{k}}{2^{cn}} \Pr\left[\begin{array}{l|l} RB^{(0)} = 0 \wedge RB^{(1)} = V & \begin{array}{l} R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)} \\ B^{(0)} \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times(n-cn+k)} \\ B^{(1)} \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times(n+cn-k)} \\ V \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n+cn-k)} \end{array} \end{array}\right]$$
$$= \frac{1}{2^{cn}} \sum_{k=0}^{cn} \frac{\binom{cn}{k}}{q^{(n+cn-k)(n^\epsilon+n^\delta)}} \Pr\left[RB = 0 \,\middle|\, \begin{array}{l} B \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times(n-cn+k)} \\ R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)} \end{array}\right].$$

We use Lemma 10 to bound the probability that the product RB is 0:

$$\Pr[RB = 0] \leq \frac{1}{q^{(n-cn+k)(n^\epsilon+n^\delta)}} + \sum_{\ell=0}^{n^\epsilon+n^\delta-1} \frac{\binom{n-n^\epsilon}{\ell}}{q^{(n-cn+k)\ell} q^{(n^\epsilon+n^\delta-\ell)(n-n^\epsilon-\ell)}}.$$

where the distribution is over $B \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times(n-cn+k)}$ and $R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)}$. Finally, this gives that

$$p(n) \leq \frac{1 + \mu(n)}{q^{2n(n^\epsilon+n^\delta)}},$$

where

$$\mu(n) := \frac{q^{2n(n^\epsilon+n^\delta)}}{2^{cn}} \sum_{k=0}^{cn} \frac{\binom{cn}{k}}{q^{(n+cn-k)(n^\epsilon+n^\delta)}} \sum_{\ell=0}^{n^\epsilon+n^\delta-1} \frac{\binom{n-n^\epsilon}{\ell}}{q^{(n-cn+k)\ell+(n^\epsilon+n^\delta-\ell)(n-n^\epsilon-\ell)}}.$$

To conclude the proof, it remains to show that μ is a negligible function. First, let us focus on the term $\binom{n-n^\epsilon}{\ell}$. Since $\epsilon, \delta < 1$, for n sufficiently large and any $\ell \in [0, n^\epsilon + n^\delta - 1]$, it holds that

$$\binom{n - n^\epsilon}{\ell} \leq \binom{n - n^\epsilon}{n^\epsilon + n^\delta}.$$

We assume n sufficiently large in what follows. Let $t \in [0, cn]$ to be determined later. We get

$$\mu(n) \leq \underbrace{\frac{\binom{n-n^\epsilon}{n^\epsilon+n^\delta}}{2^{cn}} \sum_{k=0}^{cn-t} \binom{cn}{k} \sum_{\ell=0}^{n^\epsilon+n^\delta-1} q^{Q_k(\ell)}}_{:=\mu_1(n)} + \underbrace{\frac{\binom{n-n^\epsilon}{n^\epsilon+n^\delta}}{2^{cn}} \sum_{k=0}^{cn} \binom{cn}{k} \sum_{\ell=0}^{n^\epsilon+n^\delta-1} q^{Q_k(\ell)}}_{:=\mu_2(n)},$$

where

$$\begin{aligned} Q_k(\ell) &\coloneqq 2n(n^\epsilon + n^\delta) - (n + cn - k)(n^\epsilon + n^\delta) - (n - cn + k)\ell - (n^\epsilon + n^\delta - \ell)(n - n^\epsilon - \ell) \\ &= (n^\epsilon + n^\delta)(2n - n - cn + k - n + n^\epsilon) - \ell(n - cn + k - n^\epsilon - n^\delta - n + n^\epsilon + \ell) \\ &= (n^\epsilon + n^\delta)(n^\epsilon - cn + k) + \ell(cn - k + n^\delta - \ell)\,. \end{aligned}$$

We will prove that μ_1 and μ_2 are negligible.

In order to bound μ_1, we will upper bound $Q_k(\ell)$. The derivative of Q_k is $Q_k'(\ell) = 2\ell - cn + k - n^\delta$. In order for Q_k to be nondecreasing on $[0, n^\epsilon + n^\delta - 1]$, we solve $Q_k'(n^\epsilon + n^\delta) \geq 0$, which gives the condition $cn - 2n^\epsilon - n^\delta \geq k$. Assume that $t \geq 2n^\epsilon + n^\delta$. For all $k \leq cn - t$, we have that $k \leq cn - 2n^\epsilon - n^\delta$, and hence that

$$\begin{aligned} Q_k(\ell) &\leq (n^\epsilon + n^\delta)(n^\epsilon - cn + k) + (n^\epsilon + n^\delta - 1)(cn - k - n^\epsilon + 1) \\ &= (n^\epsilon + n^\delta) - (cn - k - n^\epsilon + 1) \\ &= 2n^\epsilon + n^\delta - cn + k - 1\,. \end{aligned}$$

Since $\binom{n-n^\epsilon}{n^\epsilon+n^\delta} \leq n^{n^\epsilon+n^\delta}$, it follows that

$$\begin{aligned} \mu_1(n) &\leq \frac{n^{n^\epsilon+n^\delta}}{2^{cn}} \sum_{k=0}^{cn-t} \binom{cn}{k} (n^\epsilon + n^\delta) q^{2n^\epsilon+n^\delta-cn+k-1} \\ &\leq q^{2n^\epsilon+n^\delta+t-1} \frac{(n^\epsilon + n^\delta) n^{n^\epsilon+n^\delta}}{2^{cn}} \sum_{k=0}^{cn-t} \binom{cn}{k} \\ &\leq 2^{n^\gamma(2n^\epsilon+n^\delta-t-1)+\log(n^\epsilon+n^\delta)+(n^\epsilon+n^\delta)\log(n)}\,. \end{aligned}$$

For this to be negligible, it suffices to set $t = n^{2\epsilon} + n^{2\delta}$.

In order to bound μ_2, for $cn - k \geq 0$, we bound $Q_{k,\ell}$ by

$$Q_{k,\ell} \leq (n^\epsilon + n\delta)(n^\epsilon - cn + k) + (n^\epsilon + n^\delta)(cn - k + n^\delta) \leq (n^\epsilon + n^\delta)^2\,.$$

Now, we use the fact that $\binom{cn}{k} = \binom{cn}{cn-k} \leq n^t$ because $cn - k \leq t$, and by evaluating in $t = n^{2\epsilon} + n^{2\delta}$, we get

$$\begin{aligned} \mu_2(n) &\leq \frac{n^{n^\epsilon+n^\delta}}{2^{cn}} \sum_{k=cn-t+1}^{cn} \binom{cn}{k} (n^\epsilon + n^\delta) q^{(n^\epsilon+n^\delta)^2} \\ &\leq \frac{n^{n^\epsilon+n^\delta}}{2^{cn}} t n^t (n^\epsilon + n^\delta) q^{(n^\epsilon+n^\delta)^2} \\ &\leq 2^{-cn+n^\gamma\cdot(n^\epsilon+n^\delta)^2+\log(n^\epsilon+n^\delta)+\log(n^{2\epsilon}+n^{2\delta})+(n^\epsilon+n^\delta+n^{2\epsilon}+n^{2\delta})\log(n)}\,. \end{aligned}$$

which is negligible for $2\epsilon + \gamma < 1$ and $2\delta + \gamma < 1$. □

A.3 Proof of Theorem 9

Let $q = 2^{n^\gamma}$. We will show by introducing hybrid distributions that, assuming the hardness of $\mathsf{DxRLC}(n^\epsilon, n, \gamma, \rho)$, the distribution $D_{n+n^\delta,2n,q,\rho}$ is computationally indistinguishable from the uniform distribution over $\mathbb{F}_q^{n\times 2m} \times \mathbb{F}_q^{2m}$. The result follows directly from Lemma 13, Lemma 14, and Lemma 15 below.

Let us introduce the following distributions:

D_0 is the uniform distribution over the set

$$\left\{ \left(\left[B\right], \left[Be\right] \right) \middle| \begin{array}{l} B \leftarrow \mathbb{F}_q^{(n+n^\delta)\times 2n} \\ e \leftarrow \sigma(\chi_\rho^n(\mathbb{F}_q)) \end{array} \right\}.$$

D_1 is the uniform distribution over the set

$$\left\{ \left(\begin{bmatrix} \hat{B} \\ R\hat{B}+V \end{bmatrix}, \begin{bmatrix} \hat{B} \\ R\hat{B} \end{bmatrix} \cdot \left[e\right] \right) \middle| \begin{array}{l} \hat{B} \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times 2n} \\ e \leftarrow \sigma(\chi_\rho^n(\mathbb{F}_q)) \\ R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)} \\ V \leftarrow \sigma_x(Y, 0^{(n^\epsilon+n^\delta)\times n}) \\ Y \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times n} \\ x \leftarrow \{0,1\}^n \end{array} \right\}.$$

D_2 is the uniform distribution over the set

$$\left\{ \left(\begin{bmatrix} \hat{B} \\ R\hat{B}+V \end{bmatrix}, \begin{bmatrix} u' \\ Ru' \end{bmatrix} \right) \middle| \begin{array}{l} \hat{B} \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times 2n} \\ u' \leftarrow \mathbb{F}_q^{n-n^\epsilon} \\ R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)} \\ V \leftarrow \sigma_x(Y, 0^{(n^\epsilon+n^\delta)\times n}) \\ Y \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times n} \\ x \leftarrow \{0,1\}^n \end{array} \right\}.$$

D_3 is the uniform distribution over the set

$$\left\{ \left(\left[B\right], \left[Bu\right] \right) \middle| \begin{array}{l} B \leftarrow \mathbb{F}_q^{(n+n^\delta)\times 2n} \\ u \leftarrow \mathbb{F}_q^{n+n^\delta} \end{array} \right\}.$$

Lemma 13. *It holds that $D_0 \approx_s D_1$.*

Proof. Let $S \subset [n]$ be a uniformly randomly chosen subset of size ρn and $y \leftarrow \{0,1\}^{\rho n}$. Then Lemma 12 shows that

$$\left(\begin{bmatrix} \hat{B} \\ \hat{B}' \end{bmatrix}, S, y \right) \approx_s \left(\begin{bmatrix} \hat{B} \\ R\hat{B}+V \end{bmatrix}, S, y \right)$$

where $\hat{B} \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times 2n}, \hat{B}' \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times 2n}, R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)}$ and $V \leftarrow \mathcal{V}_{S,y}$. We can post-process these distributions to obtain D_0 and D_1. First we draw $e' \in \mathbb{F}_q^n$ as follows: $e'_i \leftarrow \mathbb{F}_q \setminus \{0\}, \forall i \in S$

and $e'_i = 0, \forall i \notin S$. Note that e' is distributed according to $\chi_\rho^n(\mathbb{F}_q)$. Then we draw $x \leftarrow X^{(S,y)}$, and set $e = \sigma_x(0^{1\times n}, e')$. Note that e is distributed according to $\sigma(\chi_\rho^n(\mathbb{F}_q))$ and that all of the indices of e that contain a non-zero element correspond to columns of V that are zero (this follows from how V is drawn according to the statement of Lemma 12). Then

$$\left(\begin{bmatrix}\hat{B}\\ \hat{B}'\end{bmatrix}, \begin{bmatrix}\hat{B}\\ \hat{B}'\end{bmatrix} \cdot [e]\right)$$

is distributed identically to D_0 and

$$\left(\begin{bmatrix}\hat{B}\\ R\hat{B}\end{bmatrix} + \begin{bmatrix}0\\ V\end{bmatrix}, \begin{bmatrix}\hat{B}\\ R\hat{B}\end{bmatrix} \cdot [e]\right)$$

is distributed identically to D_1 since $Ve = 0$. □

Lemma 14. *Assume there exists a polynomial-time algorithm $\mathcal{A}$ that distinguishes between D_1 and D_2 with advantage $\epsilon(n)$. Then, there exists a polynomial-time algorithm that solves $\mathsf{DxRLC}(n^\epsilon, n, \gamma, \rho)$ with advantage $\epsilon(n)$.*

Proof. Let $(B, v) \in \mathbb{F}_q^{(n-n^\epsilon)\times n} \times \mathbb{F}_q^{n-n^\epsilon}$ be the $\mathsf{DxRLC}(n^\epsilon, n, \gamma, \rho)$ sample. Draw $x \leftarrow \{0,1\}^n, B' \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times n}$, and R, V as in the distributions D_2 and D_3. Send

$$\left(\begin{bmatrix}\sigma_x(B', B)\\ R\sigma_x(B', B)\end{bmatrix} + \begin{bmatrix}0\\ V\end{bmatrix}, \begin{bmatrix}v\\ Rv\end{bmatrix}\right)$$

to $\mathcal{A}$ and outputs what $\mathcal{A}$ outputs. If v was uniform, the distribution is identical to D_2. If $v = Be$ for a $e \leftarrow \chi_\rho^m(\mathbb{F}_q)$, then if we set $\hat{B} := \sigma_x(B', B)$ and $e' := (\sigma_x(0^{1\times n}, e^\top))^\top$, we have $v = Be = \hat{B}e'$ and the distribution is identical to D_1. □

Lemma 15. *It holds that $D_2 \approx_s D_3$.*

Proof. Lemma 12 shows that

$$\left(\begin{bmatrix}B\\ RB\end{bmatrix} + \begin{bmatrix}0\\ V\end{bmatrix}, \{n+1\}, 1\right) \approx_s \left(\begin{bmatrix}B\\ B'\end{bmatrix}, \{n+1\}, 1\right),$$

where $B \leftarrow \mathbb{F}_q^{(n-n^\epsilon)\times(2n+2)}, R \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(n-n^\epsilon)}, B' \leftarrow \mathbb{F}_q^{(n^\epsilon+n^\delta)\times(2n+2)}, V \leftarrow \mathcal{V}^{(\{n+1\},1)}$. The result follows from the fact that the first $2n+1$ columns of $\begin{bmatrix}B\\ RB+V\end{bmatrix}$ are distributed identically to D_2 and the first $2n+1$ columns of $\begin{bmatrix}B\\ B'\end{bmatrix}$ are distributed identically to D_3. □

B Efficient Evaluation in [BKM+18] and Our Generic Construction

In this section, we rely on the fact that computing the coefficients of a (monic) degree n polynomial given n of its roots can be done in $O(n \log^2 n)$ operations, and similarly, evaluating a degree n

polynomial at n arbitrary points can also be done in $O(n \log^2 n)$ operations.

Evaluating [BKM+18] in $O(n \log^2 n)$ time. Recall that the task is to compute, given $x_1, \ldots, x_n$, the coefficients for reconstructing $f(x_0)$ from $f(x_1), \ldots, f(x_n)$:

$$L_i = \prod_{j \in [n], j \neq i} \frac{-x_j}{x_i - x_j}$$

All n numerators can be computed in $O(n)$ operations by simply computing $N = \prod_{j \in [n]} -x_j$, and then letting the ith numerator as $\frac{N}{-x_j}$.

The n denominators can be computed in $O(n \log^2 n)$ operations as follows. Observe that the term in the denominator of L_i is the evaluation of $p(x) = \sum_{i \in [n]} \prod_{j \in [n], j \neq i} (x - x_j)$ at $x = x_i$. Computing the coefficients of $p(x)$ can be done in $O(n \log^2 n)$ time by observing that $p(x)$ is $\frac{d}{dx} \prod_{j \in [n]} (x - x_j)$. We can compute the coefficients of $\prod_{j \in [n]} (x - x_j)$ in $O(n \log^2 n)$ operations since we know its roots are $x_1, \ldots, x_n$. Deriving the coefficient vector of the derivative is then an additional $O(n)$ operations. Finally, given the coefficients of $p(x)$, we can evaluate it at $x_1, \ldots, x_n$ in $O(n \log^2 n)$ time.

Evaluating Our Construction in $O(n \log^2 n)$ time. Similarly, our scheme can be evaluated in $O(n \log^2 n)$ non-group operations followed by $O(n)$ group operations. Recall that we set

$$B := \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2^1 & \cdots & (2n)^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2^n & \cdots & (2n)^n \end{pmatrix}.$$

and for any $x \in \{0,1\}^n$, evaluation is done by selecting the $(n+1) \times (n+1)$ submatrix B_x, and solving for $t \in \mathbb{F}_q^{1 \times (n+1)}$ satisfying $t \cdot B_x = 0$. Writing out the equation explicitly as

$$\begin{pmatrix} t_1 & \cdots & t_{n+1} \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ (1+x_1)^1 & (3+x_2)^1 & \cdots & (2n-1+x_n)^1 \\ \vdots & \vdots & \ddots & \vdots \\ (1+x_1)^n & (3+x_2)^n & \cdots & (2n-1+x_n)^n \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 \end{pmatrix},$$

shows that this is equivalent to finding the coefficients of a degree n polynomial polynomial $p_t(z) = \sum_{i=0}^{n} t_{i+1} z^i$ given that it has roots at

$$1 + x_1, 3 + x_2, \ldots, 2n - 1 + x_n.$$

So given n roots $r_1, \ldots, r_n$, we compute the coefficients of a (monic) degree n polynomial with these roots in $O(n \log^2 n)$ time. Finally we evaluate a dot product in $O(n)$ group operations.

C Extending Min-Entropy Arguments to Larger Alphabets

We demonstrate that for any alphabet Σ of arbitrary size ℓ, security in the generic group model actually holds for any distribution over Σ_*^n satisfying the same min-entropy bounds derived in

Section 3.2. Note that these bounds are *independent* of the alphabet size ℓ. Hence, for $\ell \gg n$ we get significant improvements (in all three wildcard regimes outlined above) over the uniform distribution over Σ_*^n with w wildcards, which has min-entropy $\log \binom{n}{w} + (n-w)\log(\ell)$.

Claim 1 (Informal)**.** *Lemma 4 works for any size alphabet ℓ.*

Proof. We show how to modify the proof of Lemma 4 to work for conjunctions of arbitrary size alphabets ℓ. Assume towards contradiction that there exists some set $S \subset [n\ell]$ of $n\ell - n$ indices such that $\widehat{e}_i = 0$ for all $i \in S$ with inverse polynomial probability. Again we partition the indices of $\widehat{e}$ into n sets $\{\widehat{e}_{(j-1)\ell+1}, \ldots, \widehat{e}_{j\ell}\}_{j\in[n]}$ so that we associate the jth set with the jth position of the pattern.

We can imagine adversarially choosing how many of the n (non-zero) indices $\overline{S} := [n\ell] \setminus S$ are a part of each of the n sets just defined and deriving the number of patterns that can give rise to an error vector $\widehat{e}$ that has all of its $n-w$ non-zero indices contained in these n indices. Note that since the pattern is drawn from a distribution with a fixed number of w wildcards, there can be at most w sets (out of the n sets) that contain *none* of the indices in $\overline{S}$. Let k represent the number of sets that contain no indices in $\overline{S}$ and we maximize the number of satisfying patterns over $k \in [0, \ldots, w]$. So we have fixed k positions of the pattern to be wildcards and assign one index in $\overline{S}$ to each of the remaining positions of the pattern. Once we fix these k positions to wildcards, we have $\binom{n-k}{w-k}$ choices for the positions of the remaining wildcards. After fixing some set of these remaining wildcard positions, we have at most k indices in $\overline{S}$ to assign to the remaining $n-w$ sets, so we overcount and assume that we have exactly k. Before assigning these k indices, we have exactly one pattern that matches the non-zero error vector indices since we have fixed all w wildcards and assigned just one non-zero index to each of the remaining $n-w$ sets. Each index we add to one of the $n-w$ sets increases the number of possible characters at that position by one. So we want to maximize the expression

$$\prod_{i=1}^{k}(a_i+1) \text{ such that } \sum_{i=1}^{k} a_i = k \text{ and } a_i \geq 0, \forall i\,.$$

We can apply the arithmetic mean-geometric mean inequality to show that

$$\prod_{i=1}^{k}(a_i+1) \leq \left(\frac{\sum_{i=1}^{k}(a_i+1)}{k}\right)^k = \left(\frac{2k}{k}\right)^k = 2^k\,.$$

Hence we derive the same expression as in the proof of Lemma 4, that there are at most

$$\max_{k\in[0,\ldots,w]} \binom{n-k}{w-k} 2^k,$$

patterns which match the n chosen non-zero indices. The remainder of the proof is then identical. □