New Techniques for Obfuscating Conjunctions

James Bartusek[1,⋆], Tancrède Lepoint[2,⋆⋆], Fermi Ma[1,⋆], and Mark Zhandry[1]

[1] Princeton University
{bartusek.james,fermima1}@gmail.com,
mzhandry@princeton.edu
[2] SRI International, New York, NY, USA
tancrede@google.com

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 placing 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 is known from LWE [47, 31], 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 handles conjunctions over exponential size alphabets. We give a significantly simpler 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.

1 Introduction

Program obfuscation [7] scrambles a program in order to hide its implementation details, while still preserving its functionality. Much of the recent attention on obfuscation focuses on obfuscating *general* programs. Such obfuscation is naturally the most useful [43, 15], but currently the only known constructions are extremely inefficient and rely on new uncertain complexity assumptions about cryptographic multilinear maps [27, 23, 29]. Despite recent

⋆ This work was done while these authors were interns at SRI International.
⋆⋆ Now at Google.

advances [1, 37, 28, 35, 2, 13, 36, 39, 8], obfuscating general programs remains out of reach.

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 [21, 38, 46, 24, 48, 10]. Brakerski, Vaikuntanathan, Wee, and Wichs [19] 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 [31, 47].

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 [12], and in related contexts is known as bit-fixing [14].

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 programs $f_{\mathsf{pat},m}$ which output a secret message $m \in \{0,1\}^\ell$ on an accepting input, and conjunctions over arbitrary size alphabets.[3]

In particular, we consider the notion of *distributional virtual black-box obfuscation* (VBB), 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, the polynomial time simulator cannot find an accepting input except with negligible probability. Thus our goal will be to produce obfuscations that are easily simulatable without any information about the sampled pattern other than its distribution.

Recently, Bishop, Kowalczyk, Malkin, Pastro, Raykova, and Shi [12] gave a simple and elegant obfuscation scheme for conjunctions, which they prove secure in the generic group model [44]. Unfortunately, they did not prove security relative to any concrete (efficiently falsifiable [41, 30]) 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) [47, 31]. However, for the restricted setting of

[3] Conjunctions over boolean/binary inputs naturally generalize to alphabets $[\ell]$ for $\ell \geq 2$. In this setting, each $x_i \in [\ell]$, and ℓ_i specifies the setting on the ith character. Positions not fixed by the ℓ_i are the wildcards.

conjunctions, the Bishop et al. [12] construction is significantly simpler and more efficient.

Our Results. In this work, we show how to alter the Bishop et al. construction in various ways, obtaining the following results.

- *A New Generic Group Construction.* We give a new group-based construction that can be viewed as "dual" to the construction of Bishop et al [12]. Our construction offers significant efficiency improvements by removing the dependence on alphabet size from the construction.
  We also improve upon the generic group security analysis of Bishop et al. [12] by simplifying the proof steps and extending the argument to handle a larger class of distributions.
- *Security from LPN.* We show that a few modifications to the group-based construction allows us to remove groups from the scheme entirely. We prove security of the resulting construction under the (constant-rate) Learning Parity with Noise (LPN) assumption. Along the way, we give a reduction from standard LPN to a specific, structured-error version of LPN, which we believe may be of independent interest.
- *Information-Theoretic Security.* Finally, we show how to extend our techniques to the information-theoretic setting if the number of wildcards is sub-linear. We stress that this requires considering a weaker notion of functionality preservation. We also give an alternative information theoretic scheme that achieves an intermediate "computational" notion of functionality preservation assuming discrete log.

In Table 1, we compare our results with prior works on conjunction obfuscation achieving distributional-VBB security (we omit the [19] and [17] constructions from entropic-ring-LWE and multilinear maps).

	Assumption	Alphabet	Distribution	FP
[47, 31]	LWE	Exponential	$H_\infty(\mathbf{b}\|\mathsf{pat}^{-1}(*)) \geq \log(n)$	Strong
Bishop et. al.	GGM	Binary	$\mathcal{U}_{w,n}$ for $w < .774n$	Strong
This work	GGM	Exponential	$\mathcal{U}_{w,n}$ for $w < n - \omega(\log(n))$ [4]	Strong
This work	LPN	Binary	$\mathcal{U}_{w,n}$ for $w = cn, c < 1$	Weak
This work	None	Binary	$H_\infty(\mathbf{b}\|\mathsf{pat}^{-1}(*)) \geq n^{1-\gamma}$ [5]	Weak

Table 1: A comparison between our constructions and prior work. Let $\mathcal{U}_{n,w}$ be the uniform distribution over all patterns in $\{0,1,*\}^n$ with exactly w wildcards. 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. When we say the alphabet is exponential, we mean any alphabet with size at most exponential in the security parameter. FP refers to functionality preservation.

1.2 Technical Overview

Review of the Bishop et al. Construction [12]. We first recall the Bishop et al. scheme for obfuscating a pattern $\mathsf{pat} \in \{0,1,*\}^n$. Begin by fixing a prime q exponential in n. Then sample uniformly random $s_1, \ldots, s_{n-1} \leftarrow \mathbb{Z}_q$ and define the polynomial $s(t) \coloneqq \sum_{k=1}^{n-1} s_k t^k \in \mathbb{Z}_q[t]$. Note that $s(t)$ is a uniformly random degree $n-1$ polynomial conditioned on $s(0) = 0$.

Now visualize a $2 \times n$ grid with columns indexed as $i = 1, \ldots, n$ and rows indexed as $j = 0, 1$. To obfuscate $\mathsf{pat} \in \{0,1,*\}^n$, for each (i,j) such that $\mathsf{pat}_i \in \{j,*\}$, we place $s(2i+j)$ in grid cell (i,j) and otherwise, we place $r_{2i+j} \leftarrow \mathbb{Z}_q$. For example, if the pattern is $\mathsf{pat} = \texttt{11*0}$, we write

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

Bishop et al. [12] 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. However, all known error-correction algorithms for Reed-Solomon codes are *non-linear*. Thus, the final step is to place the $2n$ field elements in the exponent of a group $\mathbb{G} = \langle g \rangle$ of order q. The crucial observation in [12] is that we can perform honest evaluation on an input $x \in \{0,1\}^n$ with linear operations in the exponent. For example, to evaluate on input $x = \texttt{1110}$, we generate Lagrange reconstruction coefficients L_3, L_5, L_7, L_8 for the cells corresponding to x and reconstruct

$$g^{L_3 s(3) + L_5 s(5) + L_7 s(7) + L_8 s(8)} = g^{s(0)} = g^0.$$

Evaluation accepts if and only if the result is g^0. Notice that if a single element from a white cell is included in the reconstruction, the evaluator fails to recover g^0 with overwhelming probability $(q-1)/q$. For security, they prove the following:

Theorem ([12]). *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.*

Bishop et al. [12] do not address whether the scheme becomes insecure for $0.774n < w < n - \omega(\log n)$, or if the bound is a limitation of their analysis.[6]

[4] In a concurrent work [11], Beullens and Wee achieved the same improvement in parameters and show how to base security on a new knowledge assumption secure in the generic group model. In the full version [9], we also obtain security for more general distributions that satisfy a certain min-entropy requirement.

[5] For patterns with n^δ wildcards, and $\gamma < 1 - \delta$.

[6] 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.

This Work. We provide several new interpretations of the [12] scheme. Through these interpretations, we are able to obtain improved security, efficiency, and generality, as well as novel constructions secure under standard cryptographic assumptions. We summarize the properties of these new constructions in Table 1

Interpretation 1: The Primal. Our first observation is that the $2n$ field elements generated by the [12] construction can be rewritten as a product of a transposed Vandermonde matrix A and a random vector s, plus a certain "error vector" e. So if the pattern is $\mathsf{pat} = \texttt{11*0}$, instead of writing the elements in grid form as above, we can stack them in a column as

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

So far, nothing has changed — the Bishop et al. [12] obfuscation scheme 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 $v \in \mathbb{Z}_q^{2n}$ where $v_{2i+x_i-1} = 0$ for all $i \in [n]$, and the n non-zero elements of v are Lagrange coefficients. For any input x (even ones not corresponding to accepting inputs), the Lagrange coefficients ensure v satisfies $v^\top \cdot A = 0 \in \mathbb{Z}_q^{2n}$ 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 $\mathsf{pat}_i = \texttt{*}$ 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.

Interpretation 2: The Dual. Observe that evaluation only required the A matrix and e vector. The random degree $n-1$ polynomial $s(t)$ generated in the [12] 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 $B \cdot A = \mathbf{0} \in \mathbb{Z}_q^{(n+1)\times(n-1)}$, 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 the $v^\top$ 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.

Proving Generic Group Security. The Bishop et al. [12] proof of distributional VBB security uses over 10 pages of generic group and combinatorial analysis. They derive their bound of $0.774n$ on the number of wildcards by numerically solving a non-linear equation arising from their analysis of a certain combinatorial problem.

Our first contribution is to show that by analyzing our dual scheme, we can give a short and extremely intuitive proof of generic group model security from a linear independence argument (Section 3). Part of the simplification arises from the fact that our dual scheme completely removes the random polynomial from the construction. Our generic model proof steps end with the same combinatorial problem Bishop et al. [12] consider, but instead of deferring to their analysis, we give a simple combinatorial argument from a Chernoff bound. This allows us to improve their $0.774n$ wildcard bound to $n - \omega(\log n)$. This bound is optimal; for $n - O(\log n)$ wildcards, a polynomial time adversary can guess an accepting input and learn the pattern entirely. We remark that our new combinatorial analysis implies the original Bishop et al. [12] scheme is also secure up to $n - \omega(\log n)$ wildcards. In the full version [9], we show how to generalize this analysis to certain distributions with sufficient entropy.

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 $\mathsf{pat} \in \{[\ell] \cup \{\star\}\}^n$. $f_{\mathsf{pat}}(x) = 1$ only if $x_i = \mathsf{pat}_i$ on all non-wildcard positions.

We can give a natural generalization of the $A \cdot s + e$/Bishop et al. [12] scheme to handle larger alphabets. For an alphabet of size ℓ, we use an error vector $e \in \mathbb{Z}_q^{n\ell}$, which we imagine partitioning into n blocks of length ℓ. The ith block of e corresponds with the ith pattern position. As in the binary case, if $\mathsf{pat}_i = \star$, 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. A is now a transposed Vandermonde matrix of dimension $n\ell \times n\ell - n - 1$, and s is drawn as a random vector from $\mathbb{Z}_p^{n\ell - n - 1}$. To evaluate on $x \in [\ell]^n$, we solve for $v^\top \cdot A = 0$ where v is restricted to be zero only at $v_{(i-1)\ell + x_i}$ for each $i \in [n]$.[7] However, this scheme

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

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 $B \in \mathbb{Z}_q^{(n+1) \times n\ell}, e \in \mathbb{Z}_q^{n\ell}$. 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 if we choose B to be a Vandermonde matrix, we can demonstrate that neither the evaluator nor the obfuscator ever have to store B or e in their entirety, since e is sparse for large ℓ. In particular, we set the (i, j)th entry of B to j^i. We simply need q to grow with $\log \ell$ to ensure this implicit B satisfies the certain linear independence conditions that arise from our security analysis.

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 (transposed) Vandermonde matrix A be replaced with other matrices?

In [12], the transposed Vandermonde matrix A 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 transposed 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 for our primal interpretation made no reference to the specific *structure* of A; in particular, it works for *any* public matrix A. In the case of the transposed Vandermonde matrix, applying this abstract procedure results in the Lagrange coefficients used in [12], but we can easily perform evaluation for other matrices.

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 [33].

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* over $\mathbb{F}_q$. A would be publicly output along with $A \cdot s + e$. 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. This

would allow us to simultaneously avoid encoding in a group exponent *and* obtain security under a standard assumption.

Structured Error Distributions. However, we cannot invoke security of RLC right away. The main 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 [5], 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, we give a reduction that proves hardness of the structured error RLC assumption with $2n$ samples assuming the hardness of the standard RLC assumption for polynomially many samples. We note that our reductions handle both the search and decision variants, and both LPN and RLC. We give a high-level overview of our reduction below.

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,[8] where the goal is to distinguish $(B, B \cdot e)$ from (B, u) for $B \leftarrow \mathbb{F}_q^{(n^c-n)\times n^c}$, $u \leftarrow \mathbb{F}_q^{n^c-n}$, and e as drawn previously. The advantage of considering the dual version is that the resulting technical steps of the reduction are slightly easier to see, and we stress that our proof implies the hardness of structured error RLC in its primal $A \cdot s + e$ form.

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 the standard RLC 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 standard RLC error vector e, which means introducing a

[8] In the context of LWE this duality/transformation has been observed a number of times, see e.g. [40]. For RLC, this is essentially syndrome decoding.

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 in random locations to produce the structured matrix B'. Now $B \cdot e = B' \cdot e'$, where e' is a structured error vector with a 0 element in every $(2i-1, 2i)$ index pair. This immediately gives us a structured error RLC instance with a matrix B' of dimension $(n - n^\epsilon) \times 2n$. However, we require B to have $n+1$ rows to enable evaluation of the corresponding obfuscation. We would like to simply extend the $(n-n^\epsilon) \times 2n$-dimensional B' to an $(n+1) \times 2n$-dimensional B'' by appending $n^\epsilon + 1$ newly generated uniformly random rows, but this appears impossible since we will be unable to fill in the $n^\epsilon + 1$ additional entries of $B'' \cdot e'$ without knowledge of e'.

As a first attempt, we can try to extend B' to B'' by appending random linear combinations of the $n - n^\epsilon$ rows of B'. This would allow us to properly generate $B'' \cdot e'$ by extending $B' \cdot e'$ with the corresponding linear combinations. Unfortunately, this is not quite sufficient since the matrix B'' is distinguishable from random, since its bottom $n^\epsilon + 1$ rows are in the row span of the first $n - n^\epsilon$.

We now appeal to the fact that the reduction algorithm itself chose the locations of the newly generated columns in B', and thus it knows the location of n elements of e' set to 0. The reduction can therefore introduce randomness into the last $n^\epsilon + 1$ rows of B'' by modifying only the entries in these n columns, since any changes it makes will not affect the dot product with e'. After this process, the last $n^\epsilon + 1$ rows of B'' are no longer restricted to being in the row span of the top $n - n^\epsilon$ rows. By appealing to leftover hash lemma arguments, we can prove the resulting $(n + 1) \times 2n$ dimensional B'' matrix is statistically close uniform and that $B'' \cdot e'$ is correctly distributed.

A Note on Functionality Preservation. Some previous works on conjunction obfuscation [12, 17] explicitly prove a weak notion of functionality preservation, where on any given input the obfuscation is required to be correct with overwhelming probability. This is in contrast to strong functionality preservation, which requires simultaneous correctness on all inputs with overwhelming probability. Both [12] and [17] remark that if desired, their constructions can be boosted to achieve the stronger notion by scaling parameters until the error probability on any given input can be union bounded over all inputs.[9]

A notable weakness of our analysis is that the above argument used for proving the B'' matrix is statistically close to uniform does not work for q as large as 2^n. Further complications arise when we attempt to equip a search-to-decision reduction with a predicate (for more detail, see Section 4), and thus we limit $q = 2$ for our formal obfuscation construction.[10] Our reduction allows us to add slightly more than $n^\epsilon + 1$ additional rows, and it turns out these rows can be used to boost correctness — to a point. On any input, our final construction has an error probability of $1/2^{n^\delta}$ (for any $\delta < 1/2$), and therefore settles for weak functionality preservation.

[9] This holds for our generic group model constructions as well.
[10] RLC for field size $q = 2$ is equivalent to LPN.

Information Theoretic Security. Our third and final contribution is a new *statistically* secure conjunction obfuscator. As a starting point, we recall a simple proposal for distributional VBB secure point obfuscation informally discussed by Bishop et al. [12]. The idea of their proposal (modified slightly for our setting) is roughly the following. To obfuscate a point $\mathsf{p} \in \{0,1\}^n$, output n uniformly random elements $a^{(1)}, \ldots, a^{(n)}$ from $\mathbb{F}_q$ conditioned on $\sum_{i|\mathsf{p}_i=1} a^{(i)} = 0$. Equality checking on an input $x \in \{0,1\}^n$ would be done by checking whether $\sum_{i|x_i=1} a^{(i)} = 0$.[11]

While this idea seems like a plausible starting point for point obfuscation, there is no room to support conjunctions. Any wildcard element must be set to 0 to preserve functionality, and thus the obfuscation trivially leaks information on the underlying pattern. This barrier appears inherent if we are limited to 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.

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, all of the $A^{(i)}$ matrices are distributed as uniformly random $k \times k$ rank 1 matrices. The number of wildcards this scheme can handle is $k-1$, but we cannot make the matrices arbitrarily large. The limitation arises from our statistical security arguments which only work for k as large as n^δ (for any $\delta < 1$), so we obtain statistical distributional VBB security for patterns with a sublinear number of wildcards.

Computational Functionality Preservation. Although we obtain weak functionality preservation with the above construction, it *necessarily* falls short of strong functionality preservation. Without relaxing correctness, statistical VBB security is impossible since a computationally unbounded adversary can recover pat from the truth table of the obfuscated function.

A Motivating Scenario from [47]. A natural question to ask is when 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 [47] 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,

[11] To the best of our knowledge, this scheme had not appeared in the literature before [12]. However, most prior work on point obfuscation considers stronger correctness, security, and functionality requirements (such as multi-bit output) that this scheme falls short of, which may preclude its use in certain settings.

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.[12] 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 for certain applications. 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. In addition, it does not rule out the possibility that a user (privileged or not) can find an input that causes the obfuscated circuit to wrongly accept. For example, in many cases the hash of a password can be viewed as an "obfuscation" of a point function for that password; simply accept if the input hashes to the stored hash [38]. Even if we guarantee that a computationally unbounded adversary cannot learn any information about the original password just given the hash, this does not rule out the possibility that an attacker can find a different string that causes the obfuscated password checker to accept.

An Intermediate Definition. To address this gap, we use a notion (between weak and strong) we refer to as *computational functionality preservation*. In the context of point obfuscation, this notion is essentially equivalent to the correctness definition for oracle hashing[13] considered by Canetti [21] (also achieved by Canetti, Micciancio, and Reingold [22] and Dodis and Smith [25]), as observed by Wee [46]. It is also roughly the same definition considered by Brakerski and Vaikuntanathan [18] 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 to 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.

1.3 Related Work

Conjunction Obfuscation. Previously, Brakerski and Rothblum had shown how to obfuscate conjunctions using multilinear maps [16]. This was followed by a

[12] This is slightly informal, since it requires a notion of input-hiding obfuscation [6].
[13] This was re-named to "perfectly one-way functions" in [22].

work of Brakerski et al. which showed how to obfuscate conjunctions under entropic ring LWE [19]. More recently, Wichs and Zirdelis showed how to obfuscate compute-and-compare programs under LWE [47]. Goyal, Koppula, and Waters concurrently and independently introduced lockable obfuscation and proved security under LWE [31]. 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 [29]. The main contribution of [12] 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 [12] construction — lack of generality (to more wildcards, more distributions, and more alphabet sizes) and lack of security based on a falsifiable assumption.

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 $\mathsf{colspan}(M)$ denote the column span of M.

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 $1 - n^{-\omega(1)}$.

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. 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

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

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. For any random variable $X\in\mathcal{X}$ and Z, for $\epsilon>0$ such that $\log(|\mathcal{Y}|)\le H_\infty(X|Z)-2\log(1/\epsilon)$, 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 [16], but we include a 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 $\mathsf{poly}(|C|,n)$.*
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 $\mathcal{P}\colon\mathcal{C}_n\to\{0,1\}$, there exists a negligible function μ such that*
   $$\left|\Pr_{C\leftarrow\mathcal{D}_n}[\mathcal{A}(\mathsf{Obf}(C,1^n))=\mathcal{P}(C)]-\Pr_{C\leftarrow\mathcal{D}_n}[\mathcal{S}^C(1^{|C|},1^n)=\mathcal{P}(C)]\right|=\mu(n).$$

We note that computational functionality preservation has appeared before in the obfuscation literature [46, 25], and our definition is also the same as the functionality preservation notion considered in Definition 3.1 of [18] in the context of constrained PRFs. We motivate and discuss this definition in Section 1.2, and demonstrate an obfuscation scheme achieving it in Section 5.3.

We now extend the above definition to give the notion of *statistical* security in the context of average-case obfuscation.

Definition 2 ($\epsilon(n)$-Statistical Distributional VBB Obfuscation). *Let $\mathcal{C}$, Obf, and $\mathcal{D}$, be as in Definition 1. Obf is a $\epsilon(n)$-statistical distributional VBB obfuscator if it satisfies the notions of Functionality Preservation and Polynomial Slowdown and a modified notion of Distributional Virtual Black-Box where for any* unbounded *adversary $\mathcal{A}$, the distinguishing advantage is bounded by $\epsilon(n)$.*

We recall the definition of *perfect-circuit hiding*, introduced by Barak, Bitansky, Canetti, Kalai, Paneth, and Sahai [6].

Definition 3 (Perfect Circuit-Hiding [6]). *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. [6] 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 3. We rely on this equivalence to simplify the proof of Theorem 3.

2.2 The Generic Group Model

Part of our analysis occurs in the generic group model [44], which assumes that an adversary interacts with group elements 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.

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 $\rho \in (0,1)$ 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 $1-\rho$.

Definition 4. *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 [40, 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 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 [12] that proved security up to $w < .774n$. We note that the combinatorial argument we give can be used to show that the original [12] construction achieves security for all values of w as well.

In the full version [9], we show how to extend these generic group model results in a number of ways. In particular, we prove security for general distributions with sufficient min-entropy (over a fixed number of wildcards). We then give a formal description of how to extend our construction to large alphabets, though we stress the construction is essentially the one sketched in Section 1.2. We also prove that our min-entropy results extend to the large alphabet setting.

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 5. *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 5.
- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n)$. 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[14] $tB_x = 0$ for a non-zero $t \in \mathbb{Z}_q^{1\times(n+1)}$. 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. We note that if B is viewed as the generator matrix for a linear code (of length $2n$ and rank $n+1$), then this property is equivalent to the code having distance n. This requirement on B is sufficient to prove Theorem 1 below.

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 $k < q$. □

[14] See the full version [9] for a description of how to do this in $O(n\log^2(n))$ time

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 $1/q = \mathsf{negl}(n)$.[15]

Security. We prove the distributional VBB security of our construction.

Theorem 1. *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 $n+1$ 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. If we subtract these two linear combinations, we see that in both cases $\mathcal{A}$ has formed a non-trivial linear combination of the initial $n+1$ group elements that evaluates to zero. Consider the first time that this occurs and denote the vector of coefficients as $k = (k_1, \ldots, k_{n+1}) \in \mathbb{Z}_q^{1\times(n+1)}$. Let $e \in \mathbb{Z}_q^{2n\times 1}$ be the vector drawn in the Obf procedure on input a pattern pat drawn from $\mathcal{U}_{n,n-c(n)}$, so the resulting evaluation is equal to kBe. We show that the probability that $kBe = 0$ over the randomness of the pattern and of the obfuscation is negligible.

[15] As noted in [12], we can boost this to strong functionality preservation by setting $q > 2^{2n}$.

Since the coefficients of k 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. Now 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.

Partition e into the n pairs $\{e_{2j-1}, e_{2j}\}_{j\in[n]}$. 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.

Let $S \subset [2n]$ be any fixed set of n indices. At least $n/2$ of these pairs must contain at least one e_i such that $i \in S$, and among them, an expected $c(n)/2$ number 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 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 [45] into the proof of Theorem 1 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*

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

This claim follows from Chernoff bounds; a detailed proof is given in the full version [9].

4 Obfuscating Conjunctions from Constant-Noise LPN

In this section, we present our second obfuscation construction. As described in Section 1.2, this is our "dual" construction instantiated with a random matrix

B over $\mathbb{F}_2$ and taken out of the group exponent. 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 the full version [9] or [33]), we could use the techniques from this section to prove that our construction over large fields is indistinguishable from random. However, indistinguishability from random *does not* imply distributional VBB security.[16] 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 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, which can be accomplished 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 [3]. Since no (asymptotic) improvements to our construction result from considering polynomial size q, we restrict to $q = 2$ for concreteness and prove security from LPN.

In Section 4.1, we define the relevant LPN variants we consider for our construction, which we formally describe in Section 4.2. We then observe in Section 4.3 that prior work implies hardness of our structured error LPN notion still holds even if an arbitrary predicate on the error vector is known.

In the full version [9], we give a core technical reduction from standard RLC to structured error RLC that works for q up to size 2^{n^γ}. Plugging in $q = 2$ suffices for our constructions, but we state our result for maximal generality as the reduction may be of independent interest.

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 [12] address this issue by increasing the size of the field, but this will not work here since LPN restricts us to $q = 2$. 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 the full version [9]). However, this modification has one caveat: the evaluation is polynomial-time *in expectation*, requiring a relaxation of the polynomial slowdown requirement in Definition 1.

Multi-bit Output. As a consequence of the reduction from constant noise LPN, our scheme can handle random conjunctions where a constant fraction ρ of the

[16] 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. However, we note that in the generic group model, indistinguishability from random *does* imply distributional VBB.

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

In the full version [9], we show how to adapt our construction 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. This allows us to handle conjunctions with a sub-constant (or even zero) fraction of wildcards. The idea is to 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

We begin by recalling the decisional Exact Learning Parity with Noise (DxLPN) problem considered by Jain et al. [34]. 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 [4], as noted in [34, 26]. 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 6 (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.

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

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

Definition 7 (Exact Structured LPN). *Let n, m be integers and $\rho \in (0,1)$. 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 the full version [9].

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

- $\mathsf{Obf}(\mathsf{pat} \in \{0,1,*\}^n)$: Draw $B \leftarrow \mathbb{F}_2^{(n+n^\delta)\times 2n}$ and $e \in \mathbb{F}_2^{2n}$ 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 $T \in \mathbb{F}_2^{n^\delta \times (n+n^\delta)}$ 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 4. *Fix any predicate $\mathcal{P}\colon \{0,1,*\}^n \to \{0,1\}$. Assuming the hardness of $\mathsf{DxSLPN}(n, 2m, \rho)$ implies that for all probabilistic polynomial-time $\mathcal{A}$,*

$$\left| \Pr_{B,e}[\mathcal{A}(B, Be, \mathcal{P}(e)) = 1] - \Pr_{B,u}[\mathcal{A}(B, u, \mathcal{P}(e)) = 1] \right| = \mathsf{negl}(n)$$

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

Proof. The hardness of $\mathsf{DxSLPN}(n, 2m, \rho)$ immediately implies that for all probabilistic polynomial-time $\mathcal{A}'$,

$$\Pr_{B,e}[\mathcal{A}'(B, Be, \mathcal{P}(e)) = e] = \mathsf{negl}(n)\,,$$

where B, e are drawn as in the lemma statement. This follows since the reduction can simply guess the value of $\mathcal{P}(e)$ and be correct with probability at least $1/2$. Thus we just need to show a search to decision reduction for structured LPN with a one bit predicate. This follows from the proof of Lemma 5 in [26] (equivalence of search and decision "leaky LPN"), which is a slight tweak of the search-to-decision reduction presented in [4]. We can easily adapt the proof to our case by letting the underlying problem be structured LPN rather than regular LPN and considering the special case of leakage functions corresponding to one bit predicates. 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 [32]. □

Theorem 3. *Fix any constant $\rho \in (0,1)$. Assuming the hardness of $\mathsf{DLPN}(n^\epsilon, n, \rho)$ for some $\epsilon < 1/2$, the above obfuscation with parameters (n, δ) for $\delta < 1/2$ is Distributional-VBB secure for patterns $\mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n}$.*

Proof. We show that the above obfuscator satisfies the definition of Perfect Circuit-Hiding (Definition 3), which implies Distributional VBB security [6]. 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 2 and Lemma 4 that, for any predicate $\mathcal{P}\colon \{0,1,*\} \to \{0,1\}$ and for all probabilistic polynomial-time $\mathcal{B}$,

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

where $\mathsf{pat} \leftarrow \mathcal{U}_{n,n-\rho n}$, $B \leftarrow \mathbb{F}_2^{(n+n^\delta)\times 2n}$, and $u \leftarrow \mathbb{F}_2^{n+n^\delta}$.

Now assume that there exists a balanced predicate $\mathcal{P}$ such that there exists a probabilistic polynomial-time adversary $\mathcal{A}$ with non-negligible advantage $\mu(n)$ in the above Perfect Circuit-Hiding definition. 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 distinguishing advantage is $\mu(n)$, which is non-negligible. □

5 Information-Theoretic Security

In this section, we consider a third construction, which relies on subset sums of random rank one matrices. We prove this construction attains a notion of statististical distributional VBB security, as well as weak functionality preservation. In order to achieve statistical security, however, we must limit the number of wildcards to at most n^δ for any $\delta < 1$. In Section 5.3, we show how to modify this base construction to achieve an intermediate notion of *computational functionality preservation*, assuming the 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 begin by drawing a $k \times k$ matrix B by choosing its first $k-1$ rows at random, and then picking its last row to be in the row span of the first $k-1$. We could also have drawn B as a uniformly random rank $k-1$ matrix; 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 every row but the last, and $\underline{M}$ denotes the last 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 $S_0 = \{i \mid \mathsf{pat}_i = 0\}$, $S_1 = \{i \mid \mathsf{pat}_i = 1\}$, and $S_* = \{i \mid \mathsf{pat}_i = *\}$, and 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 $B \coloneqq \begin{pmatrix} \overline{B} \\ r \cdot \overline{B} \end{pmatrix}$
- For all $i \in S_0 \cup S_1$, sample a uniformly random rank 1 $A^{(i)} \in \mathbb{F}_q^{k\times k}$.
- For all $i \in S_*$, sample a uniformly random rank 1 $\overline{A}^{(i)} \in \mathbb{F}_q^{(k-1)\times k}$. Let

$$A^{(i)} \coloneqq \begin{pmatrix} \overline{A}^{(i)} \\ r \cdot \overline{A}^{(i)} \end{pmatrix}.$$

- 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 at most $k-1$. For an x that does not match pat, consider the matrix

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

Since the first $k-1$ rows of B are all uniformly random, the same is true of first $k-1$ rows of B', denoted as $\overline{B}'$. 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.

5.2 Security

We prove our construction attains statistical distributional VBB security, defined in Definition 1.

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

Theorem 4. *The above construction with field size q is a $\epsilon(n)$-Statistical Distributional VBB obfuscator 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 $\epsilon(n)$-Statistical Distributional VBB security for a negligible function $\epsilon(n)$, 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 $\gamma < 1 - \delta$.*

The proof of Theorem 4 follows from standard applications of the leftover hash lemma. We show that as long as there is sufficient entropy on the fixed bits, the leftover hash lemma will imply the matrix F is statistically close to a uniformly random matrix. Then the low rank matrix B is hidden from view, and the $k-1$ random wildcard matrices $A^{(i)}$ drawn from the column space of B are distributed as uniformly random rank 1 matrices, just like all the other $A^{(i)}$ matrices. The formal proof is done in the full version [9].

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.[17] Refer to Section 1.2 for general discussion motivating this definition.

Remark 2. 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.[18] 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 that the following simple tweaks to our scheme allow us to base computational functionality preservation on the hardness of solving discrete log.

- **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 $\text{rank}(\overline{F} + \sum_{i|x_i=1} \overline{A}^{(i)}) = k-1$, and if not, immediately reject.

[17] 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.

[18] This is reminiscent of the notion of input-hiding obfuscation [6], but different in that we require that the adversary cannot find an accepting input for the *obfuscated* circuit rather than the original circuit.

Our functionality proof will use a reduction from the *representation problem*, introduced by Brands [20], which we denote as FIND-REP following [42].

Instance: A group $\mathbb{G}$ of order q, and 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 [20] proves that solving FIND-REP in $\mathbb{G}$ is as hard as solving discrete log in $\mathbb{G}$. Now we prove a theorem similar to Theorem 4, but with different parameters than Corollary 1.

Theorem 5. *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 $\delta \in [0, 1/2)$, pick $\delta' > \delta$ and set the field size q to $2^{n^{\delta'}}$.

Lemma 5. *For $q = 2^{n^{\delta'}}$ 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

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

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 a 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 6. *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 $r \in \mathbb{G}^{1\times(k-1)} = [\ldots g^{s_j} \ldots]$ for $j \in S_*$, draw $\overline{B} \leftarrow \mathbb{Z}_q^{(k-1)\times k}$, and let
$$B \coloneqq \begin{pmatrix} \overline{B} \\ r \cdot \overline{B} \end{pmatrix}$$
- 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 $A^{(i)} \coloneqq \begin{pmatrix} \overline{A}^{(i)} \\ g^{s_i} \cdot \overline{A}_1^{(i)} \end{pmatrix}$
- 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 $A^{(i)} \coloneqq \begin{pmatrix} c_i \\ r \cdot c_i \end{pmatrix} \cdot d_i$.
- Define $F \coloneqq B - \sum_{i\in S_1} A^{(i)}$ and output $(F, A^{(1)}, \ldots, A^{(n)})$.

Then $\mathcal{A}'$ sends $(F, A^{(1)}, \ldots, A^{(n)}, \mathsf{pat})$ 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 plug in the random values it chose in constructing the obfuscation to recover a linear combination over $s_1, \ldots, s_n$ that evaluates to zero, by the first property above. It then submits this linear combination to the $\mathsf{FIND\text{-}REP}$ 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 $s_1, \ldots, s_n$ 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 ${}^{n2^{n^\delta}}/_q = \mathsf{negl}(n)$

Thus since we can assume $\overline{B}'$ has rank $k-1$ for each $T \cap S_*$ (by the arguments from the proof of Lemma 5), and 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, Lemmas 5 and 6 imply that any adversary that breaks computational functionality preservation can solve discrete log in $\mathbb{G}$. □

References

1. Ananth, P., Jain, A.: Indistinguishability obfuscation from compact functional encryption. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 308–326. Springer, Heidelberg (2015)
2. Ananth, P., Sahai, A.: Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part I. LNCS, vol. 10210, pp. 152–181. Springer, Heidelberg (2017)
3. Applebaum, B., Avron, J., Brzuska, C.: Arithmetic cryptography: Extended abstract. In: Roughgarden, T. (ed.) ITCS 2015. pp. 143–151. ACM (2015)
4. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography with constant input locality. Journal of Cryptology 22(4), 429–469 (2009)
5. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 403–415. Springer, Heidelberg (2011)
6. Barak, B., Bitansky, N., Canetti, R., Kalai, Y.T., Paneth, O., Sahai, A.: Obfuscation for evasive functions. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 26–51. Springer, Heidelberg (2014)
7. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)
8. Bartusek, J., Guan, J., Ma, F., Zhandry, M.: Return of GGH15: Provable security against zeroizing attacks. In: TCC 2018, Part II. pp. 544–574. LNCS, Springer, Heidelberg (2018)
9. Bartusek, J., Ma, F., Lepoint, T., Zhandry, M.: New techniques for obfuscating conjunctions. Cryptology ePrint Archive, Report 2018/936 (2018), https://eprint.iacr.org/2018/936
10. Bellare, M., Stepanovs, I.: Point-function obfuscation: A framework and generic constructions. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016-A, Part II. LNCS, vol. 9563, pp. 565–594. Springer, Heidelberg (2016)
11. Beullens, W., Wee, H.: Obfuscating simple functionalities from knowledge assumptions. In: PKC. Lecture Notes in Computer Science, Springer (2019)

12. Bishop, A., Kowalczyk, L., Malkin, T., Pastro, V., Raykova, M., Shi, K.: A simple obfuscation scheme for pattern-matching with wildcards. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 731–752. Springer, Heidelberg (2018)
13. Boneh, D., Ishai, Y., Sahai, A., Wu, D.J.: Lattice-based SNARGs and their application to more efficient obfuscation. In: Coron, J., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 247–277. Springer, Heidelberg (2017)
14. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 280–300. Springer, Heidelberg (2013)
15. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 480–499. Springer, Heidelberg (2014)
16. Brakerski, Z., Rothblum, G.N.: Obfuscating conjunctions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 416–434. Springer, Heidelberg (2013)
17. Brakerski, Z., Rothblum, G.N.: Obfuscating conjunctions. Journal of Cryptology 30(1), 289–320 (2017)
18. Brakerski, Z., Vaikuntanathan, V.: Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 1–30. Springer, Heidelberg (2015)
19. Brakerski, Z., Vaikuntanathan, V., Wee, H., Wichs, D.: Obfuscating conjunctions under entropic ring LWE. In: Sudan, M. (ed.) ITCS 2016. pp. 147–156. ACM (2016)
20. Brands, S.: Untraceable off-line cash in wallets with observers (extended abstract). In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 302–318. Springer, Heidelberg (1994)
21. Canetti, R.: Towards realizing random oracles: Hash functions that hide all partial information. In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)
22. Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions (preliminary version). In: 30th ACM STOC. pp. 131–140. ACM Press (1998)
23. Coron, J.S., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 476–493. Springer, Heidelberg (2013)
24. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 621–630. ACM Press (2009)
25. Dodis, Y., Smith, A.: Correcting errors without leaking partial information. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 654–663. ACM Press (2005)
26. Döttling, N.: Low noise LPN: key dependent message secure public key encryption an sample amplification. IET Information Security 10(6), 372–385 (2016)
27. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 1–17. Springer, Heidelberg (2013)
28. Garg, S., Miles, E., Mukherjee, P., Sahai, A., Srinivasan, A., Zhandry, M.: Secure obfuscation in a weak multilinear map model. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 241–268. Springer, Heidelberg (2016)
29. Gentry, C., Gorbunov, S., Halevi, S.: Graph-induced multilinear maps from lattices. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 498–527. Springer, Heidelberg (2015)

30. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (2011)
31. Goyal, R., Koppula, V., Waters, B.: Lockable obfuscation. In: 58th FOCS. pp. 612–621. IEEE Computer Society Press (2017)
32. Hazay, C., Orsini, E., Scholl, P., Soria-Vazquez, E.: TinyKeys: A new approach to efficient multi-party computation. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 3–33. Springer, Heidelberg (2018)
33. Ishai, Y., Prabhakaran, M., Sahai, A.: Secure arithmetic computation with no honest majority. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 294–314. Springer, Heidelberg (2009)
34. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 663–680. Springer, Heidelberg (2012)
35. Lin, H.: Indistinguishability obfuscation from constant-degree graded encoding schemes. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 28–57. Springer, Heidelberg (2016)
36. Lin, H., Tessaro, S.: Indistinguishability obfuscation from trilinear maps and block-wise local PRGs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 630–660. Springer, Heidelberg (2017)
37. Lin, H., Vaikuntanathan, V.: Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: Dinur, I. (ed.) 57th FOCS. pp. 11–20. IEEE Computer Society Press (2016)
38. Lynn, B., Prabhakaran, M., Sahai, A.: Positive results and techniques for obfuscation. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 20–39. Springer, Heidelberg (2004)
39. Ma, F., Zhandry, M.: The MMap strikes back: Obfuscation and new multilinear maps immune to CLT13 zeroizing attacks. In: TCC 2018, Part II. pp. 513–543. LNCS, Springer, Heidelberg (2018)
40. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011)
41. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (2003)
42. Peikert, C.: On error correction in the exponent. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 167–183. Springer, Heidelberg (2006)
43. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) 46th ACM STOC. pp. 475–484. ACM Press (2014)
44. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
45. Skala, M.: Hypergeometric tail inequalities: ending the insanity. arXiv preprint arXiv:1311.5939 (2013)
46. Wee, H.: On obfuscating point functions. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 523–532. ACM Press (2005)
47. Wichs, D., Zirdelis, G.: Obfuscating compute-and-compare programs under LWE. In: 58th FOCS. pp. 600–611. IEEE Computer Society Press (2017)
48. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise LPN. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 214–243. Springer, Heidelberg (2016)