Multi-Input Functional Encryption with Unbounded-Message Security

Vipul Goyal * Aayush Jain † Adam O' Neill‡

Abstract

Multi-input functional encryption (MIFE) was introduced by Goldwasser *et al.* (EUROCRYPT 2014) as a compelling extension of functional encryption. In MIFE, a receiver is able to compute a joint function of multiple, independently encrypted plaintexts. Goldwasser *et al.* (EUROCRYPT 2014) show various applications of MIFE to running SQL queries over encrypted databases, computing over encrypted data streams, etc.

The previous constructions of MIFE due to Goldwasser *et al.* (EUROCRYPT 2014) based on indistinguishability obfuscation had a major shortcoming: it could only support encrypting an *a priori bounded* number of message. Once that bound is exceeded, security is no longer guaranteed to hold. In addition, it could only support *selective-security*, meaning that the challenge messages and the set of "corrupted" encryption keys had to be declared by the adversary up-front.

In this work, we show how to remove these restrictions by relying instead on *sub-exponentially secure* indistinguishability obfuscation. This is done by carefully adapting an alternative MIFE scheme of Goldwasser *et al.* that previously overcame these shortcomings (except for selective security wrt. the set of "corrupted" encryption keys) by relying instead on differing-inputs obfuscation, which is now seen as an implausible assumption. Our techniques are rather generic, and we hope they are useful in converting other constructions using differing-inputs obfuscation to ones using sub-exponentially secure indistinguishability obfuscation instead.

*Microsoft Research, India. Email: vipul@microsoft.com.

†UCLA, USA. Email: aayushjainiitd@gmail.com. Work done while at Microsoft Research, India.

‡Georgetown University, USA. Email: adam@cs.georgetown.edu. Work done in part while visiting Microsoft Research, India.

Contents

1 Introduction

In traditional encryption, a receiver in possession of a cipher-text either has a corresponding decryption key for it, in which case it can recover the underlying message, or else it can get no information about the underlying message. *Functional encryption* (FE) [SW05, BSW11, GKP+13, GGH+13] is a vast new paradigm for encryption in which the decryption keys are associated to *functions*, whereby a receiver in possession of a cipher-text and a decryption key for a particular function can recover that function of the underlying message. Intuitively, security requires that it learns nothing else. Due to both theoretical appeal and practical importance, FE has gained tremendous attention in recent years.

In particular, this work concerns a compelling extension of FE called *multi-input functional encryption* (MIFE), introduced by Goldwasser *et al.* [GGG+14]. In MIFE, decryption operates on *multiple cipher-texts*, such that a receiver with some decryption key is able to recover the associated function applied to all of the underlying plaintexts (*i.e.*, the underlying plaintexts are all arguments to the associated function). MIFE enables an number of important applications not handled by standard (single-input) FE. On the theoretical side, MIFE has interesting applications to non-interactive secure multiparty computation [BGI+14]. On the practical side, we reproduce the following example from [GGG+14]

Running SQL queries over encrypted data: Suppose we have an encrypted database. A natural goal in this scenario would be to allow a party Alice to perform a certain class of general SQL queries over this database. If we use ordinary functional encryption, Alice would need to obtain a separate secret key for every possible valid SQL query, a potentially exponentially large set. Multi-input functional encryption allows us to address this problem in a flexible way. We highlight two aspects of how Multi-Input Functional Encryption can apply to this example:

- Let f be the function where $f(q, x)$ first checks if q is a valid SQL query from the allowed class, and if so $f(q, x)$ is the output of the query q on the database x. Now, if we give the decryption key corresponding to f *and* the encryption key ek_1 (corresponding to the first input of the function f) to Alice, then Alice can choose a valid query q and encrypt it under her encryption key EK_1 to obtain ciphertext c_1. Then she could use her decryption key on ciphertexts c_1 and c_2, where c_2 is the encrypted database, to obtain the results of the SQL query.

- Furthermore, if our application demanded that multiple users add or manipulate different entries in the database, the most natural way to build such a database would be to have different ciphertexts for each entry in the database. In this case, for a database of size n, we could let f be an $(n+1)$-ary function where $f(q, x_1, \ldots, x_n)$ is the result of a (valid) SQL query q on the database $(x_1, \ldots, x_n)$.

Goldwasser *et al.* [GGG+14] discuss various other application of MIFE to non-interactive differentially private data release, delegation of computation, and, computing over encrypted streams, etc. We refer the reader to [GGG+14] for a more complete treatment. Besides motivating the notion, Goldwasser *et al.* [GGG+14] gave various flavors of definitions for MIFE and its security, as well as constructions based on different forms of program obfuscation. First of all, we note a basic observation about MIFE: in the public-key setting, functions for which one can hope to have any security at all are limited. In particular, a dishonest decryptor in possession of public key PP, a secret key SK_f for (say) a binary function f, and cipher-text CT encrypting message m, can try to learn m by repeatedly choosing some m' and learning $f(m, m')$, namely by encrypting m' under PP to get CT' and decrypting C, C' under SK_f. This means one can only hope for a very weak notion of security in such a case. As a result, in this work we focus on a more general setting where the functions have say a fixed arity n and there are encryption keys $\mathsf{EK}_1, \ldots, \mathsf{EK}_n$ corresponding to each index (*i.e.*, EK_i is used to encrypt a message which can then be used as an i-th argument in any function via decryption with the appropriate key). Only some subset of these keys (or maybe none of them) are known to the adversary. Note that this subsumes both the public key and the secret key setting (in which a much more meaningful notion of security maybe possible). In this setting, [GGG+14] presented an MIFE scheme based on indistinguishability obfuscation (iO) [BGI+01, GGH+13].

Bounded-message security: The construction of Goldwasser *et al.* [GGG+14] based on iO has a severe shortcoming namely that it could only support security for an encryption of an *a priori bounded* number of

messages[1]. This bound is required to be fixed at the time of system setup and, if exceeded, would result in the guarantee of semantic security not holding any longer. In other words, the number of challenge messages chosen by the adversary in the security game needed to be *a priori* bounded. The size of the public parameters in [GGG+14] grows linearly with the number of challenge messages.

Now we go back to the previous example of running SQL queries over encrypted databases where each entry in the database is encrypted individually. This bound would mean that the number of entries in the database would be bounded at the time of the system setup. Also, the number of updates to the database would be bounded as well. Similar restrictions would apply in other applications of MIFE: e.g., while computing over encrypted data streams, the number of data streams would have to be a priori bounded, etc. In addition, the construction of Goldwasser *et al.* [GGG+14] could only support *Selective-security:* The challenge messages and the set of "corrupted" encryption keys needed by the adversary is given out at the beginning of the experiment.[2]

Let us informally refer to an MIFE construction that does not have these shortcomings as unbounded-message secure or simply *fully-secure*. In addition to the main construction based on iO, Goldwasser *et al.* [GGG+14] also showed a construction of adaptively-secure MIFE (except wrt. the subset of encryption keys given to the advesary, so we still do not call it fully-secure) that relies on a stronger form of obfuscation called *differing-inputs obfuscation* ($di\mathcal{O}$) [BGI+01, ABG+13, BCP14].[3] Roughly, $di\mathcal{O}$ says that for any two circuits C_0 and C_1 for which it is hard to find an input on which their outputs differ, it should be hard to distinguish their obfuscations, and moreover given such a distinguisher one can extract such a differing input. Unfortunately, due to recent negative results [GGHW14], $di\mathcal{O}$ is now viewed as an implausible assumption. The main question we are concerned with in this work is: *Can fully-secure MIFE can be constructed from* $i\mathcal{O}$*?*

1.1 Our Contributions

Our main result is a fully-secure MIFE scheme from *sub-exponentially secure* $i\mathcal{O}$. More specifically, we use the following primitives: (1) sub-exponentially secure $i\mathcal{O}$, (2) sub-exponentially secure injective one-way functions, and (3) standard public-key encryption (PKE). Here "sub-exponential security" refers to the fact that advantage of any (efficient) adversary should be sub-exponentially small. For primitive (2), this should furthermore hold against adversaries running in sub-exponential time.

A few remarks about these primitives are in order. First, the required security will depend on the function arity, but *not* on the number of challenge messages. Indeed, Goldwasser *et al.* already point out that selective-security (though not bounded-message security, which instead has to do with their use of statistically sound non-interactive proofs) of their MIFE scheme based on $i\mathcal{O}$ can be overcome by standard complexity leveraging. However, in that case the required security level would depend on the the number of challenge messages. As in most applications we expect the number of challenge messages to be orders of magnitude larger than the function arity, this would result in much larger parameters than our scheme. Second, we only use a sub-exponentially secure injective one-way function (*i.e.*, primitive (2)) in our *security proof*, not in the scheme itself. Thus it suffices for such an injective one-way function to simply *exist* for security of our MIFE scheme, even if we do not know an explicit candidate.

1.2 Our Techniques

The starting point of our construction is the fully-secure construction of MIFE based on $di\mathcal{O}$ due to Goldwasser *et al.* [GGG+14] mentioned above. In their scheme, the encryption key for an index $i \in [n]$ (where n is the function arity) is a pair of public keys (pk_i^0, pk_i^1) for an underlying PKE scheme, and a ciphertext for index i consists of encryptions of the plaintext under pk_i^0, pk_i^1 respectively, along with a simulation-sound non-interactive zero knowledge proof that the two ciphertexts are well-formed (*i.e.*, both encrypting the

[1] We note that, since we do not work in the public-key setting, there is no generic implication of single-message to multi-message security.

[2] Corruption of encryption keys $\mathsf{EK}_1, \ldots, \mathsf{EK}_n$ is an aspect of MIFE security not present for single-input FE; note that in [GGG+14], some subset of these keys could not be requested *adaptively* by the adversary - they were to be chosen even before the setup was done.

[3] Actually, [GGG+14] required even a stronger form of $di\mathcal{O}$ called strong differing-inputs obfuscation or differing-inputs obfuscation secure in presence of an oracle.

same underlying message). The secret key for a function f is an obfuscation of a program that takes as input n ciphertext pairs with proofs $(c_1^0, c_1^1, \pi_1), \ldots, (c_n^0, c_n^1, \pi_n)$, and, if the proofs verify, decrypts the first ciphertext from each pair using the corresponding secret key, and finally outputs f applied to the resulting plaintexts. Note that it is important for the security proof to assume $di\mathcal{O}$, since one needs to argue when the function keys are switched to decrypting the second ciphertext in each pair instead, an adversary who detects the change can be used to extract a false proof.

We will develop modifications that this scheme so that we can instead leverage a result of [BCP14] that any indistinguishability obfuscator is in fact a differing-inputs obfuscator on circuits which differ on polynomially many points. In fact, we we will only need to use this result for circuits which differ on a *single* point. But, we will need to require the extractor to work given an adversary with even exponentially-small distinguishing gap on the obfuscations of two such circuits, due to the exponential number of hybrids in our security proof. Fortunately, [CGJS15] recently showed the result of [BCP14] extends to this case of we start with an indistinguishability obfuscator that is sub-exponentially secure.

Specifically, we need to make the proofs of well-formedness described above *unique* for every ciphertext pair, so that there is only one differing input point in the corresponding hybrids in our security proof. To achieve this, we design novel "special-purpose" proofs built from $i\mathcal{O}$ and punctured pseudorandom functions (PRFs) [BW13, BGI13, KPTZ13],[4] which works as follows. We include in the public parameters an obfuscated program that takes as input two cipher-texts and a witness that they are well-formed (*i.e.*, the message and randomness used for both the cipher-texts), and, if this check passes, outputs a (puncturable) PRF evaluation on those ciphertexts. Additionally, the secret key for a function f will now be an obfuscation of a program which additionally has this PRF key hardwired keys and verifies the "proofs" of well-formedness by checking that PRF evaluations are correct. Interestingly, in the security proof, we will switch to doing this check via an injective one-way function applied to the PRF values (*i.e.*, the PRF values themselves are not compared, but rather the outputs of an injective one-way function applied to them). This is so that extracting a differing input at this step in the security proof will correspond to inverting an injective one-way function; otherwise, the correct PRF evaluation would still be hard-coded in the obfuscated function key and we do not know how to argue security.

We now sketch the sequence of hybrids in our security proof. The proof starts from a hybrid where each challenge ciphertext encrypts m_i^0 for $i \in [n]$. Then we switch to a hybrid where each c_i^1 is an encryption of m_i^1 instead. These two hybrids are indistinguishable due to security of the PKE scheme. Let ℓ denote the length of a ciphertext. For each index $i \in [n]$ we define hybrids indexed by x, for all $x \in [2^{2n\ell}]$, in which function key SK_f decrypts the first ciphertext in the pair using SK_i^0 when $(c_1^0, c_1^1, .., c_n^0, c_n^1) < x$ and decrypts the second ciphertext in the pair using SK_i^1 otherwise. Parse $x = (x_1^0, x_1^1, .., x_n^0, x_n^1)$. Hybrids indexed by x and $x+1$ can be proven indistinguishable as follows: We first switch to sub-hybrids that puncture the PRF key at $\{x_i^0, x_i^1\}$, changes a function key SK_f to check correctness of an PRF value by applying an injective one-way function as described above, and hard-coded the output of the injective one-way function at the PRF evaluation at the punctured point. Now if the two hybrids differ at an input of the form $(x_1^0, x_1^1, \alpha_1, .., x_n^0, x_n^1, \alpha_n)$ where α_i is some fixed value (a PRF evaluation of (x_i^0, x_i^1)), extracting the differing input can be used to invert the injective one-way function on random input (namely the α_i).

Finally, we note that exponentially many hybrids are indexed by all possible ciphertext vectors that could be input to decryption (i.e., vectors of length the arity of the functionality) and *not* all possible challenge ciphertext vectors. This allows us to handle any unbounded (polynomial) number of ciphertexts for every index.

Our techniques further demonstrate the power of the exponentially-many hybrids technique, together with the iO=>one-point-diO, which have also been used recently in works such as [CGJS15, BPR15].

1.3 Related Work, Open Problems

In this work we focus on an *indistinguishability-based* security notion for MIFE. This is justified as Goldwasser *et al.* [GGG+14] show that an MIFE meeting a stronger simulation-based security definition in general implies black-box obfuscation [BGI+01] and hence is impossible. They also point out that in the secret-key setting with small function arity, an MIFE scheme meeting indistinguishability-based security notion can

[4] Due to the number of hybrids in our proof, we will also need the punctured PRFs to be sub-exponentially secure, but this already follows from a sub-exponentially secure injective one-way function.

be "compiled" into a simulation-secure one, following the work of De Caro *et al.* [CIJ+13]; in such a setting we can therefore achieve simulation-based secuirty as well. We note that a main problem left open by our work is whether $i\mathcal{O}$ without sub-exponential security implies MIFE, which would in some sense show these two primitives are equivalent (up to the other primitives used in the construction). Another significant open problem is removing the bound a function's arity in our construction, as well as the bound on the message length, perhaps by building on recent work in the setting of single-input FE [KLW15].

Initial constructions of single-input FE from $i\mathcal{O}$ [GGH+13] also had the shortcomings we are concerned with removing for constructions of MIFE in this work, namely selective and bounded-message security. These restrictions were similarly first overcome using differing-inputs obfuscation [ABG+13, BCP14], and later removed while only relying on $i\mathcal{O}$ [ABSV14, Wat14]. Unfortunately, we have not been able to make the techniques of these works apply to the MIFE setting, which is why we have taken a different route. If they could, this would be a path towards solving the open problem of relying on $i\mathcal{O}$ with standard security mentioned above.

[BKS15] construct an adaptively secure multi-input functional encryption scheme in the secret key setting for any number of cipher-texts from any secret key functional encryption scheme. Their construction builds on a clever observation that function keys of a secret-key function-hiding functional encryption can be used to hide any message. This provides a natural 'arity amplification' procedure that allows us to go from a t arity secret key MIFE to a $t+1$ arity MIFE. However, because the arity is amplified one by one, it leads to a blow up in the scheme, so the arity of the functions had to be bounded by $O(loglogk)$. [AJ15] builds on similar techniques but considers construction of secret key MIFE from a different view-point (i.e. building iO from functional encryption).

Finally, we note that the source of trouble in achieving differing-inputs obfuscation is the *auxiliary input* provided to the distinguisher. Another alternative to using differing-inputs obfuscation is *public-coin* diO [IPS15], where this auxiliary input is simply a uniform random string as done in [BGJS] (they however achieve selective security). There are no known implausibility results for public-coin $di\mathcal{O}$, and it is interesting to give an alternative construction of fully-secure MIFE based on it. Our assumption seems incomparable, as we only need $i\mathcal{O}$ but also sub-exponential security.

1.4 Organisation

The rest of this paper is organized as follows: In Section 2, we recall some definitions and primitives used in the rest of the paper. In Section 3 we formally define MIFE and present our security model. Finally in Section 4, we present our construction and its security proof.

2 Preliminaries

In this section we recall various concepts on which the paper is built upon. We assume the familiarity of a reader with concepts such as public key encryption, one way functions and omit formal description in the paper. For the rest of the paper, we denote by $\mathbb{N}$ the set of natural numbers $\{1, 2, 3, ..\}$. Sub-exponential indistinguishability obfuscation and sub-exponentially secure puncturable pseudo-random functions have been used a lot recently such as in the works of [CLTV15, BV15, KLW15]. For completeness, we present these notions below:

2.1 Indistinguisability Obfuscation

The following definition has been adapted from [GGH+13]:

Definition 1. *A uniform PPT machine $i\mathcal{O}$ is an indistinguishability obfuscator for a class of circuits $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ if the following properties are satisfied.*
Correctness: *For every $k \in \mathbb{N}$, for all $\{\mathcal{C}_k\}_{k\in\mathbb{N}}$, we have*

$$Pr[C' \leftarrow i\mathcal{O}(1^k, C) : \forall x, C'(x) = C(x)] = 1$$

Security: *For any pair of functionally equivalent equi-sized circuits $C_0, C_1 \in \mathcal{C}_k$ we have that: For every non uniform PPT adversary $\mathcal{A}$ there exists a negligible function ϵ such that for all $k \in \mathbb{N}$,*

$$| Pr[\mathcal{A}(1^n, i\mathcal{O}(1^k, C_0), C_0, C_1, z) = 1] - Pr[\mathcal{A}(1^k, i\mathcal{O}(1^k, C_1), C_0, C_1, z) = 1] | \leq \epsilon(k)$$

We additionally say that $i\mathcal{O}$ is sub-exponentially secure if there exists some constant $\alpha > 0$ such that for every non uniform PPT $\mathcal{A}$ the above indistinguishability gap is bounded by $\epsilon(k) = O(2^{-k^\alpha})$

Definition 2 (Indistinguishability obfuscation for P/poly)**.** *$i\mathcal{O}$ is a secure indistinguishability obfuscator for P/Poly, if it is an indistinguishability obfuscator for the family of circuits $\{\mathcal{C}_k\}_{k\in\mathbb{N}}$ where $\mathcal{C}_k$ is the set of all circuits of size k.*

2.2 Puncturable Psuedorandom Functions

A PRF $F : \mathcal{K}_{k\in\mathbb{N}} \times \mathcal{X} \to \mathcal{Y}_{k\in\mathbb{N}}$ is a puncturable pseudorandom function if there is an additional key space $\mathcal{K}_p$ and three polynomial time algorithms $(F.\mathsf{setup}, F.\mathsf{eval}, F.\mathsf{puncture})$ as follows:

- $F.\mathsf{setup}(1^k)$ a randomized algorithm that takes the security parameter k as input and outputs a description of the key space $\mathcal{K}$, the punctured key space $\mathcal{K}_p$ and the PRF F.
- $F.\mathsf{puncture}(K, x)$ is a randomized algorithm that takes as input a PRF key $K \in \mathcal{K}$ and $x \in \mathcal{X}$, and outputs a key $K\{x\} \in \mathcal{K}_p$.
- $F.\mathsf{Eval}(K, x')$ is a deterministic algorithm that takes as input a punctured key $K\{x\} \in \mathcal{K}_p$ and $x' \in \mathcal{X}$. Let $K \in \mathcal{K}$, $x \in \mathcal{X}$ and $K\{x\} \leftarrow F.\mathsf{puncture}(K, x)$.

The primitive satisfies the following properties:

1. **Functionality is preserved under puncturing:** For every $x^* \in \mathcal{X}$,

   $$Pr[F.\mathsf{eval}(K\{x^*\}, x) = F(K, x)] = 1$$

   here probability is taken over randomness in sampling K and puncturing it.

2. **Psuedo-randomness at punctured point:** For any poly size distinguisher D, there exists a negligible function $\mu(\cdot)$, such that for all $k \in \mathbb{N}$ and $x^* \in \mathcal{X}$,

   $$| Pr[D(x^*, K\{x^*\}, F(K, x^*)) = 1] - Pr[D(x^*, K\{x^*\}, u) = 1] | \leq \mu(k)$$

   where $K \leftarrow F.\mathsf{Setup}(1^k)$, $K\{x^*\} \leftarrow F.\mathsf{puncture}(K, x^*)$ and $u \xleftarrow{\$} \mathcal{Y}_k$

We say that the primitive is sub-exponentially secure if μ is bounded by $O(2^{-k^{c_{PRF}}})$, for some constant $0 < c_{PRF} < 1$. We also abuse the notation slightly and use $F(K, \cdot)$ and $F.\mathsf{Eval}(K, \cdot)$ to mean one and same thing irrespective of whether key is punctured or not.

2.3 Injective One-Way Function

A one-way function with security (s, ϵ) is an efficiently evaluable function $P : \{0,1\}^* \to \{0,1\}^*$ and $Pr_{x \xleftarrow{\$} \{0,1\}^n}[P(A(P(x))) = P(x)] < \epsilon(n)$ for all circuits A of size bounded by $s(n)$. It is called an injective one-way function if it is injective in the domain $\{0,1\}^n$ for all sufficiently large n.
In this work we require that there *exists*[5] (s, ϵ) injective one-way function with $s(n) = 2^{n^{c_{owp1}}}$ and $\epsilon = 2^{-n^{c_{owp2}}}$ for some constants $0 < c_{owp1}, c_{owp2} < 1$. This assumption is well studied, [Hol06, Wee07] have used $(2^{cn}, 1/2^{cn})$ secure one-way functions and permutations for some constant c.
This is a reasonable assumption due to following result from [GGKT05]

Lemma 1. *Fix $s(n) = 2^{n/5}$. For all sufficiently large n, a random permutation π is $(s(n), 1/2^{n/5})$ secure with probability at least $1 - 2^{-2^{n/2}}$.*

[5]We however do not require that the injective one-way function can be sampled efficiently

Such assumptions have been made and discussed in works of [Hol06, Wee05, Wee07]. In particular, we require the following assumption:

Assumption 1: For any adversary A with running time bounded by $s(n) = O(2^{n^{c_{owp1}}})$, for any apriori bounded polynomial $p(n)$ there exists an injective one-way function P such that,

$$Pr[r_i \sim \{0,1\}^n \forall i \in [p], \mathcal{A}^{\mathcal{O}}(P(r_1),..,P(r_p)) = (r_1,..,r_p)] < O(2^{-n^{c_{owp2}}})$$

for some constant $0 < c_{owp_1}, c_{owp_2} < 1$. Here, oracle $\mathcal{O}$ can reveal at most $p-1$ values out of $r_1,..,r_p$. Note that this assumption follows from the assumption described above with a loss p in the security gap.

2.4 (d,δ)-Weak Extractability Obfuscators

The concept of weak extractability obfuscator was first introduced in [BCP14] where they claimed that if there is an adversary that can distinguish between indistinguishability obfuscations of two circuits that differ on polynomial number of inputs with noticable probability, then there is a PPT extractor that extracts a differing input with overwhelming probability. [CGJS15] generalised the notion to what they call (d,δ) weak extractability obfuscator, where they require that if there is any PPT adversary that can distinguish between obfuscations of two circuits (that differ on at most d inputs) with atleast $\epsilon > \delta$ probability, then there is an explicit extractor that extracts a differing input with overwhelming probability and runs in time $poly(1/\epsilon, d, k)$ time. Such a primitive can be constructed from a sub-exponentially secure indistinguishability obfuscation. $(1, 2^{-k})$ weak extractability obfuscation will be crucially used in our construction for our $MIFE$ scheme. We believe that in various applications of differing inputs obfuscation, it may suffice to use this primitive along with other sub-exponentially secure primitives.

Definition 3. *A uniform transformation $we\mathcal{O}$ is a (d,δ) weak extractability obfuscator for a class of circuits $\mathcal{C} = \{\mathcal{C}_k\}$ if the following holds. For every PPT adversary $\mathcal{A}$ running in time $t_{\mathcal{A}}$ and $1 \geq \epsilon(k) > \delta$, there exists a algorithm E for which the following holds. For all sufficiently large k, and every pair of circuits on n bit inputs, $C_0, C_1 \in \mathcal{C}_k$ differing on at most $d(k)$ inputs, and every auxiliary input z,*

$$| Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_0), C_0, C_1, z) = 1] - Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_1), C_0, C_1, z) = 1] | \geq \epsilon$$

$$\Rightarrow Pr[x \leftarrow E(1^k, C_0, C_1, z) : C_0(x) \neq C_1(x) \geq 1 - negl(k)$$

and the expected runtime of E is $O(p_E(1/\epsilon, d, t_{\mathcal{A}}, n, k))$ for some fixed polynomial p_E. In addition, we also require the obfuscator to satisfy correctness.

Correctness: *For every $n \in \mathbb{N}$, for all $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$, we have*

$$Pr[C' \leftarrow we\mathcal{O}(1^n, C) : \forall x, C'(x) = C(x)] = 1$$

We now construct a $(1, 2^{-k})$ input weak extractability obfuscator from sub-exponentially secure indistinguishability obfuscation. Following algorithm describes the obfuscation procedure.

$we\mathcal{O}(1^k, C)$: The procedure outputs $C' \leftarrow i\mathcal{O}(1^{k^{1/\alpha}}, C)$. Here, $\alpha > 0$ is a constant chosen such that any polynomial time adversary against indistinguishability obfuscation has security gap upper bounded by $2^{-k}/4$.

The proof of the following theorem is given in the appendix A.

Theorem 1. *Assuming sub-exponentially secure indistinguishability obfuscation, there exists $(1,\delta)$ weak obfuscator for P/poly for any $\delta > 2^{-k}$, where k is the size of the circuit.*

In general, assuming sub-exponential security one can construct (d,δ) extractability obfuscator for any $\delta > 2^{-k}$. Our construction is as follows:
$we\mathcal{O}(C)$: Let α be the security constant such that $i\mathcal{O}$ with parameter $1^{k^{1/\alpha}}$ has security gap upper bounded by $O(2^{-3k})$. This can be found due to sub exponential security of indistinguishability obfuscation. The procedure outputs $C' \leftarrow i\mathcal{O}(1^{k^{1/\alpha}}, C)$.
We cite [BCP14] for the proof of the following theorem.

Theorem 2 ([BCP14])**.** *Assuming sub-exponentially secure indistinguishability obfuscation, there exists (d,δ) weak extractability obfuscator for P/poly for any $\delta > 2^{-k}$.*

3 Multi-Input Functional Encryption

Let $\mathcal{X} = \{\mathcal{X}_k\}_{k\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_k\}_{k\in\mathbb{N}}$ denote ensembles where each $\mathcal{X}_k$ and $\mathcal{Y}_k$ is a finite set. Let $\mathcal{F} = \{\mathcal{F}_k\}_{k\in\mathbb{N}}$ denote an ensemble where each $\mathcal{F}_k$ is a finite collection of n-ary functions. Each $\mathsf{f} \in \mathcal{F}_k$ takes as input n strings $\mathsf{x_1}, .., \mathsf{x_n}$ where each $\mathsf{x_i} \in \mathcal{X}_k$ and outputs $\mathsf{f}(\mathsf{x_1}, .., \mathsf{x_n}) \in \mathcal{Y}_k$. We now describe the algorithms.

- $\mathsf{MIFE.Setup}(1^\kappa, \mathsf{n})$: is a PPT algorithm that takes as input the security parameter κ and the function arity n. It outputs n encryption keys $\mathsf{EK_1}, .., \mathsf{EK_n}$ and a master secret key MSK.
- $\mathsf{MIFE.Enc}(\mathsf{EK}, \mathsf{m})$: is a PPT algorithm that takes as input an encryption key $\mathsf{EK_i} \in (\mathsf{EK_1}, .., \mathsf{EK_n})$ and an input message $m \in \mathcal{X}_k$ and outputs a ciphertext $\mathsf{CT_i}$ which denotes that the encrypted plaintext constitutes an i^{th} input to a function f.
- $\mathsf{MIFE.Keygen}(\mathsf{MSK}, \mathsf{f})$: is a PPT algorithm that takes as input the master secret key MSK and a $\mathsf{n}-$ary function $\mathsf{f} \in \mathcal{F}_k$ and outputs a corresponding decryption key $\mathsf{SK_f}$.
- $\mathsf{MIFE.Dec}(\mathsf{SK_f}, \mathsf{CT_1}, .., \mathsf{CT_n})$: is a deterministic algorithm that takes as input a decryption key $\mathsf{SK_f}$ and n ciphertexts $\mathsf{CT_i}, .., \mathsf{CT_n}$ and outputs a string $\mathsf{y} \in \mathcal{Y}_k$.

The scheme is said to satisfy *correctness* if for honestly generated encryption and function key and any tuple of honestly generated ciphertexts, decryption of the cipher-texts with function key for f outputs the joint function value of messages encrypted inside the ciphertexts with overwhelming probability.

Definition 4. *Let $\{\mathsf{f}\}$ be any set of functions $f \in \mathcal{F}_\kappa$. Let $[n] = \{1, .., n\}$ and $I \subseteq [n]$. Let $\mathbf{X^0}$ and $\mathbf{X^1}$ be a pair of input vectors, where $\mathbf{X^b} = \{x^b_{1,j}, .., x^b_{n,j}\}_{j=1}^q$. We define $\mathcal{F}$ and (X^0, X^1) to be I-compatible if they satisfy the following property: For every $f \in \{f\}$, every $I' = \{i_1, .., i_t\} \subseteq I$, every $j_1, .., j_{n-t} \in [q]$ and every $x'_{i_1}, .., x'_{i_t} \in \mathcal{X}_\kappa$,*

$$f(< x^0_{i_1,j_1}, .., x^0_{i_{n-t},j_{n-t}}, x'_{i_1}, .., x'_{i_t} >) = f(< x^1_{i_1,j_1}, .., x^1_{i_{n-t},j_{n-t}}, x'_{i_1}, .., x'_{i_t} >)$$

where $< y_{i_1}, ..., y_{i_n} >$ denotes a permutation of the values $y_{i_1}, .., y_{i_n}$ such that the value y_{i_j} is mapped to the l^{th} location if y_{i_j} is the l^{th} input (out of n inputs) to f.

IND-Secure MIFE: Security definition in [GGG+14] was parameterized by two parameters (t, q) where t denotes the number of encryption keys known to the adversary, and q denotes the number of challenge messages per encryption key. Since, our scheme can handle any unbounded polynomial q and any $t \leq n$, we present a definition independent of these parameters.

Definition 5. *(Indisitinguishability based security). We say that a multi-input functional encryption scheme MIFE for for n ary functions $\mathcal{F}$ is fully IND-secure if for every PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ defined as*

$$\mathsf{Adv}^{\mathsf{MIFE,IND}}_{\mathcal{A}}(1^\kappa) = |Pr[\mathsf{IND}^{\mathsf{MIFE}}_{\mathcal{A}}] - 1/2|$$

is $negl(\kappa)$, where:

Experiment $\mathsf{IND}^{\mathsf{MIFE}}_{\mathcal{A}}(1^\kappa)$

$(\mathbf{EK}, MSK) \leftarrow \mathsf{MIFE.Setup}(1^\kappa, n)$

$b \leftarrow \{0, 1\}$

$b' \leftarrow \mathcal{A}^{\mathsf{MIFE.Keygen}(\mathsf{MSK},\cdot),\mathcal{O}(\mathbf{EK},\cdot),\mathcal{E}(\mathbf{EK},b,\cdot)}(1^\kappa)$

Output $(b = b')$

Figure 1: Security Game

Valid adversaries: In the above experiment, $\mathcal{O}(\mathbf{EK}, \cdot)$ is an oracle that takes an index i and outputs EK_i. Let I be the set of queries to this oracle. $\mathcal{E}(\mathbf{EK}, b, \cdot)$ on a query $(x^0_{1,j}, .., x^0_{n,j}), (x^1_{1,j}, .., x^1_{n,j})$ (where j denotes

the query number) outputs $\mathsf{CT}_{i,j} \leftarrow \mathsf{MIFE.Enc}(\mathsf{EK}_i, x^b_{i,j})\ \forall i \in [n]$. *If* q *is the total number of queries to this oracle then let* $\mathbf{X^1} = \{x^l_{1,j}, .., x^l_{n,j}\}_{j=1}^q$ *and* $l \in \{0,1\}$. *Also, let* $\{f\}$ *denote the entire set of function key queries made by* $\mathcal{A}$. *Then, the challenge message vectors* $\mathbf{X^0}$ *and* $\mathbf{X^1}$ *chosen by* $\mathcal{A}$ *must be* $I-$*compatible with* $\{f\}$. *The scheme is said to be secure if for any valid adversary* $\mathcal{A}$ *the advantage in the game described above is negligible.*

4 Our MIFE Construction

Notation: Let k denote the security parameter and $n = n(k)$ denote the bound on arity of the function for which the keys are issued. By $\mathsf{PRF} = (\mathsf{PRF.Setup}, \mathsf{PRF.Puncture}, \mathsf{PRF.Eval})$ denote a sub-exponentially secure puncturable PRF with security constant c_{PRF} and PKE denote a public key encryption scheme. Let P be any one-one function (in the security proof we instantiate with a sub-exponentially secure injective one-way function with security constants c_{owp1} and c_{owp2}). Finally, let $\mathcal{O}$ denote a $(1, 2^{-3nl-k})$ weak extractability obfuscator (here l is the length of the cipher-text of PKE). In particular, for any two equivalent circuits security gap of the obfuscation is bounded by 2^{-3nl-k} (any algorithm that distinguishes obfuscations of two circuits with more than this gap will yield an algorithm that extracts a differing point).

$\mathsf{MIFE.Setup}(1^\mathsf{k}, \mathsf{n})$: Sample $K_i \leftarrow \mathsf{PRF.Setup}(1^\lambda)$ and $\{(PK^b_i, SK^b_i)\}_{b\in\{0,1\}} \leftarrow \mathsf{PKE.Setup}(1^k)$. Let PP_i be the circuit as in figure 2. EK_i is declared as the set $EK_i = \{PK^0_i, PK^1_i, \tilde{PP_i} = \mathcal{O}(PP_i), P\}$ and

Hard-wired: PK^0_i, PK^1_i, K_i.
Input: $c^0_i, c^1_i, m, r^0_i, r^1_i$
The program does the following:

- Check that $c^0_i = \mathsf{PKE.Enc}(PK^0_i, m; r^0)$ and $c_i = \mathsf{PKE.Enc}(PK^1_i, m; r^1)$. If the check fails output $\perp$.
- Output $\mathsf{PRF.Eval}(K_i, c^0_i, c^1_i)$

Figure 2: Program Encrypt

$MSK = \{SK^0_i, SK^1_i, K_i, P\}_{i\in[n]}$. Here injective function P takes as input elements from the co-domain the PRF. λ is set greater than $(3nl + k)^{1/c_{PRF}}$ and so that the length of output of the PRF is at least $max\{(5nl + 2k)^{1/c_{owp1}}, (3nl + k)^{1/c_{owp2}}\}$ long.

$\mathsf{MIFE.Enc}(EK_i, m)$: To encrypt a message m, encryptor computes $c^0_i = \mathsf{PKE.Enc}(PK^0_i, m; r^0)$ and $c^1_i = \mathsf{PKE.Enc}(PK^1_i, m; r^1)$ and computes $\pi_i \leftarrow \tilde{PP_i}(c^0_i, c^1_i, m, r^0, r^1)$. Output $CT_i = (c^0_i, c^1_i, \pi_i)$.

$\mathsf{MIFE.KeyGen}(MSK, f)$: Let G^0_f be the circuit described below. Key for f is output as $K_f \leftarrow \mathcal{O}(G^0_f)$

Hard-wired: $\{SK^0_i, K_i, P\}_{i\in[n]}$.
Input: $\{c^0_i, c^1_i, \pi_i\}_{i\in[n]}$
The program does the following:

- For all $i \in [n]$, check that $P(\mathsf{PRF.Eval}(K_i, c^0_i, c^1_i)) = P(\pi_i)$. If the check fails output $\perp$.
- Output $f(\mathsf{PKE.Dec}(SK^0_1, c^0_1), .., \mathsf{PKE.Dec}(SK^0_n, c^0_n))$.

Figure 3: Program G^0_f

$\mathsf{MIFE.Decrypt}(K_f, \{c^0_i, c^1_i, \pi_i\}_{i\in[n]})$: Output $K_f(c^0_1, c^1_1, \pi_1, .., c^0_n, c^1_n, \pi_n)$.
Remark:

1. We also assume that the circuits are padded appropriately before they are obfuscated.

2. Note that in the scheme, circuit for the key for a function f, G_f^0 is instantiated with any one-one function (denoted by P). In the proofs we replace it with a sub-exponentially secure injective one-way function. We see that the input output behaviour of G_f^0 do not change when it is instantiated with any one-one function, hence we can switch to a hybrid when it is instantiated by sub-exponentially secure injective one way function and due to the security of obfuscation these two hybrids are close.

4.1 Proof Overview

The starting point of our construction is the fully-secure construction of MIFE based on $di\mathcal{O}$ due to Goldwasser *et al.* [GGG+14] mentioned above. In their scheme, the encryption key for an index $i \in [n]$ (where n is the function arity) is a pair of public keys (pk_i^0, pk_i^1) for an underlying PKE scheme, and a ciphertext for index i consists of encryptions of the plaintext under pk_i^0, pk_i^1 respectively, along with a simulation-sound non-interactive zero knowledge proof that the two ciphertexts are well-formed (*i.e.*, both encrypting the same underlying message). The secret key for a function f is an obfuscation of a program that takes as input n ciphertext pairs with proofs $(c_1^0, c_1^1, \pi_1), \ldots, (c_n^0, c_n^1, \pi_n)$, and, if the proofs verify, decrypts the first ciphertext from each pair using the corresponding secret key, and finally outputs f applied to the resulting plaintexts. Note that it is important for the security proof to assume $di\mathcal{O}$, since one needs to argue when the function keys are switched to decrypting the second ciphertext in each pair instead, an adversary who detects the change can be used to extract a false proof.

We develop modifications that this scheme so that we can instead leverage a result of [BCP14] that any indistinguishability obfuscator is in fact a differing-inputs obfuscator on circuits which differ on polynomially many points. In fact, we we will only need to use this result for circuits which differ on a *single* point. But, we will need to require the extractor to work given an adversary with even exponentially-small distinguishing gap on the obfuscations of two such circuits, due to the exponential number of hybrids in our security proof. We make use of sub-exponentially secure obfuscation to achieve this.

Specifically, we make the proofs of well-formedness described above *unique* for every ciphertext pair, so that there is only one differing input point in the corresponding hybrids in our security proof. To achieve this, we design novel "special-purpose" proofs built from $i\mathcal{O}$ and punctured pseudorandom functions (PRFs) [BW13, BGI13, KPTZ13],[6] which works as follows. We include in the public parameters an obfuscated program that takes as input two cipher-texts and a witness that they are well-formed (*i.e.*, the message and randomness used for both the cipher-texts), and, if this check passes, outputs a (puncturable) PRF evaluation on those ciphertexts. Additionally, the secret key for a function f will now be an obfuscation of a program which additionally has this PRF key hardwired keys and verifies the "proofs" of well-formedness by checking that PRF evaluations are correct. Interestingly, in the security proof, we will switch to doing this check via an injective one-way function applied to the PRF values (*i.e.*, the PRF values themselves are not compared, but rather the outputs of injective one-way function applied to them). This is so that extracting a differing input at this step in the security proof will correspond to inverting a injective one-way function; otherwise, the correct PRF evaluation would still be hard-coded in the obfuscated function key and we do not know how to argue security.

We now sketch the sequence of hybrids in our security proof. The proof starts from a hybrid where each challenge ciphertext encrypts m_i^0 for $i \in [n]$. Then we switch to a hybrid where each c_i^1 is an encryption of m_i^1 instead. These two hybrids are indistinguishable due to security of the PKE scheme. Let ℓ denote the length of a ciphertext. For each index $i \in [n]$ we define hybrids indexed by x, for all $x \in [2^{2n\ell}]$, in which function key SK_f decrypts the first ciphertext in the pair using SK_i^0 when $(c_1^0, c_1^1, .., c_n^0, c_n^1) < x$ and decrypts the second ciphertext in the pair using SK_i^1 otherwise. Parse $x = (x_1^0, x_1^1, .., x_n^0, x_n^1)$. Hybrids indexed by x and $x+1$ can be proven indistinguishable as follows: We first switch to sub-hybrids that puncture the PRF key at $\{x_i^0, x_i^1\}$, changes a function key SK_f to check correctness of an PRF value by applying an injective one-way function as described above, and hard-coded the output of the injective one-way function at the punctured point. Now if the two hybrids differ at an input of the form $(x_1^0, x_1^1, \alpha_1, .., x_n^0, x_n^1, \alpha_n)$ where α_i is some fixed value (a PRF evaluation of (x_i^0, x_i^1)), extracting the differing input can be used to invert the injective one-way

[6] Due to the number of hybrids in our proof, we will also need the punctured PRFs to be sub-exponentially secure, but this already follows from sub-exponentially secure injective one-way functions.

function on random input (namely the α_i). As in [BCP14], this inverter runs in time inversely proportional to the distinguishing gap between the two consecutive hybrids (which is sub-exponentially small). Hence, we require a sub-exponential secure injective one-way function to argue security.

Finally, we note that exponentially many hybrids are indexed by all possible ciphertext vectors that could be input to decryption (i.e., vectors of length the arity of the functionality) and *not* all possible challenge ciphertext vectors. This allows us to handle any unbounded (polynomial) number of ciphertexts for every index.

4.2 Proof of Security

Theorem 3. *Assuming an existence of a sub-exponentially secure indistinguishability obfuscator, injective one-way function and a polynomially secure public-key encryption scheme there exists a fully IND secure multi-input functional encryption scheme for any polynomially apriori bounded arity n.*

Proof. We start by giving a lemma that will be crucial to the proof.

Lemma 2. *Let X and Y denote two (possibly correlated) random variables from distribution $\mathcal{X}$ and $\mathcal{Y}$, with support $|\mathcal{X}|$ and $|\mathcal{Y}|$, and $U(X,Y)$ denote an event that depends on X, Y. We say that $U(X,Y) = 1$ if the event occurs, and $U(X,Y) = 0$ otherwise. Suppose $\Pr_{(X,Y)\sim\mathcal{X},\mathcal{Y}}[U(X,Y) = 1] = p$. We say that a transcript $\mathbb{X}$ falls in the set 'good' if $Pr_{Y\sim\mathcal{Y}}[U(X,Y|X = \mathbb{X}) = 1] \geq p/2$. Then, $Pr_{X\sim\mathcal{X}}[X \in \mathsf{good}] \geq p/2$.*

Proof. We prove the lemma by contradiction. Suppose $Pr_{X\sim\mathcal{X}}[X \in \mathsf{good}] = c < \frac{p}{2}$. Then,

$$\begin{aligned} Pr_{(X,Y)\sim(\mathcal{X},\mathcal{Y})}[U(X,Y) = 1] = Pr_{(X,Y)\sim(\mathcal{X},\mathcal{Y})}[U(X,Y) = 1 | X \in \mathsf{good}] \cdot \Pr_{X\sim\mathcal{X}}[X \in \mathsf{good}] \\ + Pr_{(X,Y)\sim(\mathcal{X},\mathcal{Y})}[U(X,Y) = 1 | X \notin \mathsf{good}] \cdot Pr_{X\sim\mathcal{X}}[X \notin \mathsf{good}] \end{aligned}$$

By definition of the set good, $Pr_{(X,Y)\sim(\mathcal{X},\mathcal{Y})}[U(X,Y) = 1 | X \notin \mathsf{good}] < \frac{p}{2}$. Then, $p = \Pr[U(X,Y) = 1] < 1 \cdot c + (1-c) \cdot p/2$. Then, if $c < \frac{p}{2}$, we will have that $p < \frac{p}{2} + \frac{p}{2}$, which is a contradiction. This proves our lemma. □

We proceed listing hybrids where the first hybrid corresponds to the hybrid where the challenger encrypts message $m^0_{i,j}$ for all $i \in [n]$ and the last hybrid corresponds to the hybrid where the challenger encrypts $m^1_{i,j}$. We then prove that each consecutive hybrid is indistinguishable from each other. Then, we sum up all the advantages between the hybrids and argue that the sum is negligible.

H_0

1. Challenger does setup to compute encryption keys $EK_i \forall i \in [n]$ and MSK as described in the algorithm.
2. $\mathcal{A}$ may query for encryption keys EK_i for some $i \in [n]$, function keys for function f and ciphertext queries in an interleaved fashion.
3. If it asks for an encryption key for index i, it is given EK_i.
4. When $\mathcal{A}$ queries keys for n ary function f_j and challenger computes keys honestly using MSK.
5. $\mathcal{A}$ may also ask encryptions of message vectors $M^h = \{(m^h_{1,j}, .., m^h_{n,j})\}$ where $h \in \{0,1\}$, where j denotes the encryption query number. The message vectors has to satisfy the constraint as given in the security definition.
6. For all queries j, challenger encrypts $CT_{i,j} \forall i \in [n]$ as follows: $c^0_{i,j} = \mathsf{PKE.Enc}(PK^0_i, m^0_{i,j})$ and $c^1_{i,j} = \mathsf{PKE.Enc}(PK^1_i, m^0_{i,j})$ and $\pi_{i,j} \leftarrow \mathsf{PRF.Eval}(K_i, c^0_{i,j}, c^1_{i,j})$. Then the challenger outputs $CT_{i,j} = (c^0_{i,j}, c^1_{i,j}, \pi_{i,j})$.
7. $\mathcal{A}$ can ask for function keys for functions f_j, encryption keys EK_i's and cipher-texts as long as they satisfy the constraint given in the security definition.
8. $\mathcal{A}$ now outputs a guess $b' \in \{0,1\}$.

H_1 : Let q denote the number of cipher-text queries. This hybrid is same as the previous one except that for all indices $i \in [n], j \in [q]$ challenge cipher-text cipher-text component $c^1_{i,j}$ is set as $c^1_{i,j} = \mathsf{PKE.Enc}(PK^1_i, m^1_{i,j})$.

$\mathsf{H}_{x \in [2, 2^{2ln}+2]}$: This hybrid is same as the previous one except key for every function query f is generated as an obfuscation of program 4 by hard-wiring x (along with SK^0_i, SK^1_i, K_i, P).

Hard-wired: $\{SK^0_i, SK^1_i, K_i, x, P\}_{i\in[n]}$.
Input: $\{c^0_i, c^1_i, \pi_i\}_{i\in[n]}$
The program does the following:

- For all $i \in [n]$, check that $P(\mathsf{PRF.Eval}(K_i, c^0_i, c^1_i)) = P(\pi_i)$. If the check fails output $\perp$.
- If $(c^0_1, c^1_1, .., c^0_n, c^1_n) < x-2$, output $f(\mathsf{PKE.Dec}(SK^1_1, c^1_1), .., \mathsf{PKE.Dec}(SK^1_n, c^1_n))$ otherwise output $f(\mathsf{PKE.Dec}(SK^0_1, c^0_1), .., \mathsf{PKE.Dec}(SK^0_n, c^0_n))$.

Figure 4: Program $G_{f,x}$

$\mathsf{H}_{2^{2ln}+3}$: This hybrid is same as the previous one except that function keys for any function f is generated by obfuscating program 5.

Hard-wired: $\{SK^1_i, K_i, P\}_{i\in[n]}$.
Input: $\{c^0_i, c^1_i, \pi_i\}_{i\in[n]}$
The program does the following:

- For all $i \in [n]$, check that $P(\mathsf{PRF.Eval}(K_i, c^0_i, c^1_i)) = P(\pi_i)$. If the check fails, output $\perp$.
- Output $f(\mathsf{PKE.Dec}(SK^1_1, c^1_1), .., \mathsf{PKE.Dec}(SK^1_n, c^1_n))$.

Figure 5: Program G^1_f

$\mathsf{H}_{2^{2ln}+4}$: Let q denote the number of cipher-text queries made by the adversary. This hybrid is same as the previous one except that for all indices $i \in [n], j \in [q]$, challenge cipher-text component $c^0_{i,j}$ is generated as $c^0_{i,j} = \mathsf{PKE.Enc}(PK^0_i, m^1_{i,j})$.

$\mathsf{H}_{2^{2ln}+4+x|x\in[2^{2ln}+1]}$: This hybrid is same as the previous one except key for a function f is generated by obfuscating program 4 by hard-wiring $2^{2ln} + 3 - x$ (along with SK^0_i, SK^1_i, K_i, P).

$\mathsf{H}_{2.2^{2ln}+6}$: This hybrid corresponds to the real security game when $b = 1$.

We now argue indistinguishability by describing following lemmas.

Lemma 3. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_0) = 1] - Pr[D(\mathsf{H}_1) = 1] \mid < negl(k)$.*

Proof. This lemma follows from the security of the encryption scheme PKE. In these hybrids, all function keys only depend on one secret key SK^0_i for all $i \in [n]$ and SK^1_i never appears in the hybrids. If there is a distinguisher D that distinguishes between the hybrids then there exists an algorithm $\mathcal{A}$ that breaks the security of the encryption scheme with the same advantage. $\mathcal{A}$ gets set of public keys $PK_1, .., PK_n$ from the encryption scheme challenger and samples public keys $(PK^0_i, SK^0_i) \forall i \in [n]$ himself and sets $PK^1_i = PK_i \forall i \in [n]$. It also samples PRF keys $K_i \forall i \in [n]$. Using these keys, it generates encryption keys $EK_i \forall i \in [n]$. Then, it invokes D and answers queries for encryption keys EK_i's and function keys. $\mathcal{A}$ generates function keys using only as obfuscation of G^0_f. Finally, D declares $M^b = \{(m^b_{1,j}, .., m^b_{n,j})\}_{j\in[q]}$. $\mathcal{A}$ sends (M^0, M^1) to the encryption challenger and gets $c_{i,j} \forall i \in [n], j \in [q]$ from the challenger. $\mathcal{A}$ computes $c^0_{i,j} \leftarrow \mathsf{PKE.Enc}(\mathsf{PK^0_i}, \mathsf{m^0_{i,j}})$. Then evaluates $\pi_{i,j} \leftarrow \mathsf{PRF.Eval}(K_i, c^0_{i,j}, c_{i,j})$. Then it sets, $CT_{i,j} = (c^0_{i,j}, c_{i,j}, \pi_{i,j})$ and sends it to D. After that D may query keys for functions and encryption keys and the response is given as before. D now submits

a guess b' which is also output by $\mathcal{A}$ as its guess for the encryption challenge. If $c_{i,j}$ is an encryption of $m^0_{i,j}$ then D's view is identical to the view in H_1 otherwise its view is identical to the view in H_2. Hence, distinguishing advantage of D in distinguishing hybrids is less than the advantage of $\mathcal{A}$ in breaking the security of the encryption scheme. □

Lemma 4. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_1) = 1] - Pr[D(\mathsf{H}_2) = 1] \mid < negl(k)$.*

Proof. For simplicity, we consider the case when there is only single function key query f. General case can be argued by introducing v many intermediate hybrids where v is the number of keys issued to the adversary. Indistinguishability of these hybrids follows from the fact that circuit G^0_f and $G_{f,x=2}$ are functionally equivalent. Hence, due to the security of indistinguishability obfuscation property of the weak extractability obfuscator the lemma holds. For completeness, we describe the reduction. Namely, we construct an adversary $\mathcal{A}$ that uses D to break the security of weak extractability obfuscator. $\mathcal{A}$ invokes D and does setup (by sampling PKE encryption key pairs and PRF keys for all indices) and answers cipher-text queries as in the previous hybrid H_1. On query f from D, it sends G^0_f and $G_{f,x}$ to the obfuscation challenger. It receives K_f and sends it to $\mathcal{A}$. $\mathcal{A}$ sends it to D. It replies to the encryption key queries to D using the sampled PKE keys and PRF keys. Then it outputs whatever D outputs. Note that view of D is identical to the view in H_1 (if K_f is an obfuscation of G^0_f) or H_2 (if K_f is an obfuscation of $G_{f,x=2}$). Hence, advantage of $\mathcal{A}$ is atleast the advantage of D in distinguishing hybrids. Due to security of obfuscation claim holds. □

Lemma 5. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{2^{2ln}+2}) = 1] - Pr[D(\mathsf{H}_{2^{2ln}+3}) = 1] \mid < negl(k)$.*

Proof. This follows from the indistinguishability obfuscator $\mathcal{O}$. For any function f, G^1_f is functionally equivalent to $G_{f,x=2^{2ln}+2}$. Proof of the lemma is similar to the proof of lemma 4. □

Lemma 6. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{2^{2ln}+3}) = 1] - Pr[D(\mathsf{H}_{2^{2ln}+4}) = 1] \mid < negl(k)$.*

Proof. This follows from the security of encryption scheme PKE. Note that in both the hybrids SK^0_i is not used anywhere. Proof is similar to the proof of lemma 3. □

Lemma 7. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{2^{2ln}+4}) = 1] - Pr[D(\mathsf{H}_{2^{2ln}+5}) = 1] \mid < negl(k)$.*

Proof. This follows from the security of indistinguishability obfuscator $\mathcal{O}$. Proof is similar to the proof of lemma 4. □

Lemma 8. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{2.2^{2ln}+5}) = 1] - Pr[D(\mathsf{H}_{2.2^{2ln}+6}) = 1] \mid < negl(k)$.*

Proof. This follows from the security of indistinguishability obfuscator $\mathcal{O}$. Proof is similar to the proof of lemma 4. □

Lemma 9. *For any PPT distinguisher D and $x \in [2, 2^{2ln} + 1]$, $\mid Pr[D(\mathsf{H}_x) = 1] - Pr[D(\mathsf{H}_{x+1}) = 1] \mid < O(v \cdot 2^{-2ln-k})$ for some polynomial v.*

Proof. We now list following sub hybrids and argue indistinguishability between these hybrids.

$\mathsf{H}_{x,1}$

1. Challenger samples key pairs $(PK^0_i, SK^0_i), (PK^1_i, SK^1_i)$ for each $i \in [n]$.
2. Parses $x - 2 = (x^0_1, x^1_1, .., x^0_n, x^1_n)$ and computes $(a^0_i, a^1_i) \leftarrow (\mathsf{PKE.Dec}(SK^0_i, x^0_i), \mathsf{PKE.Dec}(SK^1_i, x^1_i))$.
3. Samples puncturable PRF's keys $K_i \forall i \in [n]$.
4. Denote by set $Z \subset [n]$ such that $i \in Z$ if $a^0_i \neq a^1_i$. Computes $\alpha_i \leftarrow \mathsf{PRF.Eval}(K_i, x^0_i, x^1_i)$ and derives punctured keys $K'_i \leftarrow \mathsf{PRF.Puncture}(K_i, x^0_i, x^1_i)$ for all $i \in [n]$.
5. If $\mathcal{A}$ queries for encryption keys for any index i, for any i in Z, $\tilde{PP}_i$ is generated as an obfuscation of circuit in figure 2 instantiated with the punctured key K'_i (α_i will never be accessed by the circuit PP_i in this case). For all other indices i, $\tilde{PP}_i$ is constructed by using the punctured key K'_i and hard-coding the value α_i (for input (x^0_i, x^1_i)) as done in figure 6. These $\tilde{PP}_i$ are used to respond to the queries for EK_i.

6. If $\mathcal{A}$ queries keys for n ary function f_j and challenger computes keys honestly as in H_x using MSK.
7. If $\mathcal{A}$ releases message vectors $M^h = \{(m^h_{1,j}, .., m^h_{n,j})\}$ where $h \in \{0,1\}$, challenger encrypts $CT_{i,j} \forall i \in [n], j \in [q]$ as follows: $c^0_{i,j} = \mathsf{PKE.Enc}(PK^0_i, m^0_{i,j})$ and $c^1_{i,j} = \mathsf{PKE.Enc}(PK^1_i, m^1_{i,j})$. If $(c^0_{i,j}, c^1_{i,j}) = (x^0_i, x^1_i)$ set $\pi_{i,j} = \alpha_i$ otherwise set $\pi_{i,j} \leftarrow \mathsf{PRF.Eval}(K_i, c^0_{i,j}, c^1_{i,j})$. Then the challenger outputs $CT_{i,j} = (c^0_{i,j}, c^1_{i,j}, \pi_{i,j})$. Here q denotes the total number of encryption queries.
8. Challenger can ask for function keys for functions f_j and encryption keys EK_i as long as they satisfy the constraint with the message vectors.
9. $\mathcal{A}$ now outputs a guess $b' \in \{0,1\}$.

Hard-wired: $PK^0_i, PK^1_i, K'_i, \alpha_i, x^0_i, x^1_i$.
Input: $c^0_i, c^1_i, m, r^0_i, r^1_i$
The program does the following:

- Checks that $c^0_i = \mathsf{PKE.Enc}(PK^0_i, m; r^0)$ and $c_i = \mathsf{PKE.Enc}(PK^1_i, m; r^1)$. If the check fails output $\perp$.
- If $(c^0_i, c^1_i) = (x^0_i, x^1_i)$ output α_i otherwise output $\mathsf{PRF.Eval}(K'_i, c^0_i, c^1_i)$

Figure 6: Program Encrypt*

$\mathsf{H}_{x,2}$: This hybrid is similar to the previous one except that function key for any function f is generated as an obfuscation of program 7 by hard-wiring $(SK^0_i, SK^1_i, K^{'}_i, P, P(\alpha_i), x^0_i, x^1_i) \forall i \in [n]$.

Hard-wired: $\{SK^0_i, SK^1_i, K^{'}_i, P, P(\alpha_i), x^0_i, x^1_i\}_{i\in[n]}$.
Input: $\{c^0_i, c^1_i, \pi_i\}_{i\in[n]}$
The program does the following:

- For any $i \in [n]$, if $(c^0_i, c^1_i) = (x^0_i, x^1_i)$ check that $P(\alpha_i) = P(\pi_i)$. If the check fails output $\perp$.
- Otherwise, for $i \in [n]$, check that $P(\mathsf{PRF.Eval}(K_i, c^0_i, c^1_i)) = P(\pi_i)$. If the check fails output $\perp$.
- If $(c^0_1, c^1_1, .., c^0_n, c^1_n) < x-2$, output $f(\mathsf{PKE.Dec}(SK^1_1, c^1_1), .., \mathsf{PKE.Dec}(SK^1_n, c^1_n))$ otherwise output $f(\mathsf{PKE.Dec}(SK^0_1, c^0_1), .., \mathsf{PKE.Dec}(SK^0_n, c^0_n))$.

Figure 7: Program $G^*_{f,x}$

$\mathsf{H}_{x,3}$ This hybrid is similar to the previous hybrid except that for all $i \in [n]$, α_i is chosen randomly from the domain of the injective one way function function P.

$\mathsf{H}_{x,4}$: This hybrid is similar to the previous hybrid except that the function key is generated as an obfuscation program 7 initialised $x+1$.

$\mathsf{H}_{x,5}$: This hybrid is the same as the previous one except that $\alpha_i \forall i \in [n]$ is chosen as actual PRF values at (x^0_i, x^1_i) using the key K_i.

$\mathsf{H}_{x,6}$: This hybrid is the same as the previous one except that key for the function f, keys are generated as obfuscation of program 4 initialised with $x+1$.

$\mathsf{H}_{x,7}$: This hybrid is the same as the previous one except for all $i \in [n]$, $\tilde{PP}_i$ is generated as an obfuscation of 2 initialised with genuine PRF key K_i. This hybrid is identical to the hybrid H_{x+1}

Claim 1. *For any PPT distinguisher D, $| Pr[D(\mathsf{H}_x) = 1] - Pr[D(\mathsf{H}_{x,1}) = 1] | < O(n \cdot 2^{-3nl-k})$.*

Proof. This claim follows from the indistinguishability security of weak extractability obfuscator. We have that circuits for $i \in Z$, circuit in figure 2 initialised with regular PRF key K_i is functionally equivalent to when it is initialised with punctured key K'_i. This is because for $i \in Z$, (x_i^0, x_i^1) never satisfies the check and the PRF is never evaluated at this point and also the fact the punctured key outputs correctly at all points except the point at which the PRF is punctured. For $i \in [n] \setminus Z$, program in figure 2 initialised with K_i is functionally equivalent to the program in 6 initialised with (K'_i, α_i).
From the above observation, we can prove the claim by at most n intermediate hybrids where we switch one by one obfuscation $\tilde{PP}_i$ to use the punctured key and each intermediate hybrid is indistinguishable due to the security of obfuscation. □

Claim 2. *For any PPT distinguisher D, $| Pr[D(\mathsf{H}_{x,1}) = 1] - Pr[D(\mathsf{H}_{x,2}) = 1] | < O(p(k) \cdot 2^{-3nl-k})$. Here, $p(k)$ is some polynomial.*

Proof. This follows from the indistinguishability obfuscation property of the weak extractability obfuscator $\mathcal{O}$. The proof follows by at most p intermediate hybrids where each queried K_f is switched to an obfuscation of program 4 (with hard-wired values $SK_i^0, SK_i^1, K_i, x, P$) to an obfuscation of program 7 (with hard-wired values $SK_i^0, SK_i^1, K'_i, P, P(\alpha_i), x$). Note that in this hybrids, both these programs are functionally equivalent. This reduction is straight forward and we omit details. □

Claim 3. *For any PPT distinguisher D, $| Pr[D(\mathsf{H}_{x,2}) = 1] - Pr[D(\mathsf{H}_{x,3}) = 1] | < O(n \cdot 2^{-2nl-k})$.*

Proof. This claim follows from the property that puncturable PRF's value is psuedo-random at punctured point given the punctured key (sub-exponential security of the puncturable PRF). This proof goes through by a sequence of at most n hybrids where for each index $i \in [n]$, $(K'_i, \alpha_i = \mathsf{PRF.Eval}(K_i, x_i^0, x_i^1))$ is replaced with $(K'_i, \alpha_i \leftarrow \mathcal{R})$ for all $i \in [n]$. This can be done because in both these hybrids, function keys and the encryption keys use only the punctured keys and a the value of the PRF at the punctured point. Here $\mathcal{R}$ is the co-domain of the PRF, which is equal to the domain of the injective one way function P. Since, PRF is sub exponentially secure with parameter c_{PRF} (c_{PRF} be the security constant of the PRF) when PRF is initialised with parameter greater than $(2nl+k)^{1/c_{PRF}}$, distinguishing advantage between each intermediate hybrid is bounded by $O(2^{-2nl-k})$. The reduction is straight forward and we omit the details. □

Claim 4. *For any PPT distinguisher D, $| Pr[D(\mathsf{H}_{x,3}) = 1] - Pr[D(\mathsf{H}_{x,4}) = 1] | < O(p(k).2^{-2nl-k})$. for some polynomial $p(k)$*

Proof. We prove this claim for a simplified case when only one function key is queried. The general case by considering a sequence of intermediate hybrids where function keys are changed one by one, hence the factor $p(k)$. Assume that there is a PPT algorithm D such that $| Pr[D(\mathsf{H}_{x,3}) = 1] - Pr[D(\mathsf{H}_{x,4}) = 1] | > \epsilon > 2^{-2nl-k}$. Note that these hybrids are identical upto the point the adversary asks for a key for a function f. We argue indistinguishability according to following cases.

1. **Case 0:** Circuit given in 7 initialised with x is functionally equivalent to circuit 7 initialised with $x+1$.
2. **Case 1:** This is the case in which the two circuits described above are not equivalent.

Let Q denote the random variable and $Q = 0$ if adversary is in case 0, otherwise $Q = 1$. By $\epsilon_{Q=b}$ denote the value $| Pr[D(\mathsf{H}_{x,3}) = 1/Q = b] - Pr[D(\mathsf{H}_{x,4}) = 1/Q = b] |$. It is known that $Pr[Q = 0]\epsilon_{Q=0} + Pr[Q = 1]\epsilon_{Q=1} > \epsilon$.

Now we analyse both these cases:
$\mathbf{Pr[Q = 0]\epsilon_{Q=0} < 2^{-2nl-k}}$: This claim follows due to the indistinguishability security of $(1, 2^{-3nl-k})$ weak extractability obfuscator. Consider an adversary D with $Q = 0$ and challenger C, we construct an algorithm $\mathcal{A}$ that uses D and breaks the indistinguishability obfuscation of the weak extractability obfuscator with the same advantage. $\mathcal{A}$ works as follows: $\mathcal{A}$ invokes C that invokes D. C does the setup as in the hybrid and responds to the queries of D. D outputs f. $\mathcal{A}$ gives $G^*_{f,x}$ and $G^*_{f,x+1}$ to the obfuscation challenger and gets back K_f in return which is given to D. D's queries are now answered by C. $\mathcal{A}$ outputs whatever D

outputs. $\mathcal{A}$ breaks the indistinguishability obfuscation security of the weak extractability obfuscator with advantage atleast $\epsilon_{Q=0}$ as the view of D is identical to $\mathsf{H}_{x,3}$ if $G^*_{f,x}$ was obfuscated and it is identical to $\mathsf{H}_{x,4}$ otherwise.

$\mathbf{Pr[Q=1]\epsilon_{Q=1} < 2^{-2nl-k}}$: The only point at which the two circuits $G^*_{f,x}$ and $G^*_{f,x+1}$ in this case may differ is $(x^0_1, x^1, \alpha_1, ..., x^0_n, x^n, \alpha_n)$ where α_i is the inverse of a fixed injective one way function value $P(\alpha_i)$. In this case, due to security of weak extractability obfuscator the claim holds. Assume to the contrary $Pr[Q=1]\epsilon_{Q=1} > \delta > 2^{-2nl-k}$. In this case, let τ be the transcript (including the randomness to generate PKE keys, PRF keys along with chosen α'_is) between the challenger and the adversary till the point function key for function f is queried. We denote $\tau \in \mathsf{good}$ if conditioned on τ, $\epsilon_{\tau,Q=1} > \epsilon_{Q=1}/2$. Then, using lemma 2, one can show that $Pr[\tau \in \mathsf{good}] > \epsilon_{Q=1}/2$.

Now, let us denote by set Z a set that contains indices in $i \in [n]$ such that $a^0_i \neq a^1_i$. Note that α_i can be requested by the adversary in one of the two following ways: $a^0_i = a^1_i$ and adversary queries for EK_i or adversary queries for an encryption of (a^0_i, a^1_i) and challenger sends encryption as (x^0_i, x^1_i, α_i) with some probability. Let E denote the set of indices for which α_i's queried by the adversary through first method and S denote the set queried through second method. Then it holds that $S \cup E \neq [n]$. This is because adversary cannot query for such cipher-texts and encryption keys in these hybrids since $Q = 1$ and in particular it holds that $f(< \{a^0_i\}_{i\in Z}, \{a^0_i\}_{i\in E} >) \neq f(< \{a^1_i\}_{i\in Z}, \{a^0_i\}_E >)$. Here $<,>$ denotes the permutation which sends a variable with subscript i to index i.
Now we let $T \subsetneq [n]$ denote the set of α_i for $i \in [n]$ requested by D (either by querying cipher-text or by querying for EK_i such that $a^0_i = a^1_i$). We know that conditioned on τ (randomness upto the point f is queried),

$$| Pr[D(\mathsf{H}_{x,3}) = 1/Q = 1, \tau] - Pr[D(\mathsf{H}_{x,4}) = 1/Q = 1, \tau] | > \epsilon_{Q=1}/2$$

For all $t \subsetneq Z$,

$$\Sigma_t \; | Pr[D(\mathsf{H}_{x,3}) = 1 \cap T = t/Q = 1, \tau] - Pr[D(\mathsf{H}_{x,4}) = 1 \cap T = t/Q = 1, \tau] | > \epsilon_{Q=1}/2$$

Since number of proper subsets of $[n]$ is bounded by 2^n, there exists a set t such that

$$| Pr[D(\mathsf{H}_{x,3}) = 1 \cap T = t/Q = 1, \tau] - Pr[D(\mathsf{H}_{x,4}) = 1 \cap T = t/Q = 1, \tau] | > \epsilon_{Q=1}/2^{n+1}$$

Now we construct an adversary $\mathcal{A}$ that breaks the security of injective one way function with probability $Pr[Q=1]\epsilon_{Q=1}/2^{n+1}$ that runs in time $O(2^{2n}/\epsilon^2_{Q=1})$. $\mathcal{A}$ runs as follows:

1. $\mathcal{A}$ invokes D. Then it does setup and generates PKE keys and punctured PRF keys K'_i for all indices in $[n]$ according to hybrid $\mathsf{H}_{x,3}$.
2. $\mathcal{A}$ gets injective one way function values from the injective one way function challenger $(P, P(\alpha_1), .., P(\alpha_n))$.
3. $\mathcal{A}$ now guesses a random proper subset $t \subset [n]$.
4. For all indices in $i \in t$ it gets α_i from the injective one way function challenger.
5. If EK_i is asked for any $i \in t \cup Z$, it is generated as in $\mathsf{H}_{x,3}$ and given out. Otherwise, $\mathcal{A}$ aborts. We call the transcript till here τ.
6. When D asks for a key for f. If f is such that $Q = 0$, $\mathcal{A}$ outputs $\perp$. $\mathcal{A}$ now constructs a distinguisher $\mathcal{B}$ of obfuscation of circuits $G^*_{f,x}$ and $G^*_{f,x+1}$ as follows:
   - $\mathcal{A}$ gets as a challenge obfuscation $\tilde{C}_f$ which is an obfuscation $G^*_{f,x}$ or $G^*_{f,x+1}$.
   - $\mathcal{A}$ gives this obfuscation to $\mathcal{B}$ which invokes D from the point of the transcript τ and gives this obfuscation to D.
   - When D asks for a cipher-text, if the queries are such that $\mathcal{B}$ can generate it using $\alpha_i \forall i \in t$ then answer the cipher-text query. Otherwise, it outputs 0.
   - If EK_i is asked by D for any $i \in t \cup Z$, it is generated as in $\mathsf{H}_{x,3}$ and given out. If any other encryption key is queried, it outputs 0.

- If set of indices for which α_i's used to generate response to the queries (in the transcript τ and the queries asked by D when run by $\mathcal{B}$) equals t it outputs whatever D outputs otherwise, $\mathcal{B}$ outputs 0.

7. If t is correctly guessed as t^*, it is easy to check that $\mid Pr[\mathcal{B}(G^*_{f,x}, G^*_{f,x+1}, \mathcal{O}(G^*_{f,x}), aux) = 1] - Pr[\mathcal{B}(G^*_{f,x}, G^*_{f,x+1}, \mathcal{O}(G^*_{f,x+1}), aux) = 1] \mid > \epsilon_{Q=1}/2^{n+1}$. (Here aux is the information with $\mathcal{A}$ required to run $\mathcal{B}$ including $\alpha_i \forall i \in t$, $P(\alpha_i), PK^0_i, PK^1_i, SK^0_i, SK^1_i, K'_i \forall i \in [n]$ and transcript τ till point 4). This is because,

$$\mid Pr[\mathcal{B}(G^*_{f,x}, G^*_{f,x+1}, \mathcal{O}(G^*_{f,x}), aux) = 1] - Pr[\mathcal{B}(G^*_{f,x}, G^*_{f,x+1}, \mathcal{O}(G^*_{f,x+1}), aux) = 1] \mid =$$

$$\mid Pr[D(\mathsf{H}_{x,3}) = 1 \cap T = t/Q = 1, \tau] - Pr[D(\mathsf{H}_{x,4}) = 1 \cap T = t/Q = 1, \tau] \mid > \epsilon_{Q=1}/2^{n+1}$$

8. We finally run the extractor E of the weak extractability obfuscator using $\mathcal{B}$ to extract a point $(x^0_1, x^1_1, \alpha_1, .., x^0_n, x^1_n, \alpha_n)$. (This extraction can be run as long as $\epsilon_{Q=1}/2^{n+1} > 2^{-3nl}$ implying $\epsilon_{Q=1} > 2^{-2nl-k}$ as otherwise there is nothing to prove and claim trivially goes through). This extractor runs in time $O(t_D.2^{2n}/\epsilon^2_{Q=1})$. Probability of success of this extraction is

$$Pr[Q = 1] \cdot Pr[\tau \text{ is good}] \cdot Pr[\text{ t is guessed correctly}] > Pr[Q = 1] \cdot \epsilon_{Q=1}/2^{n+1}$$

Let μ be the input length for injective one way function. We note the following cases:

Case 0: If $Pr[Q = 1]\epsilon_{Q=1} < O(2^{-2nl-k})$, in this case the claim goes through.

Case 1: If $Pr[Q = 1]\epsilon_{Q=1}/2^{n+1} < O(2^{-\mu^{c_{owp2}}})$, in this case the claim goes through if μ is set to be greater than $(3nl + k)^{1/c_{owp2}}$.

Case 2: If case 1 does not occur, then we must have that $2^{2n}/\epsilon^2_{Q=1} > 2^{\mu^{c_{owp1}}}$, implying that if μ is greater than $(5nl + 2k)^{1/c_{owp1}}$ the claim holds (due to the security of injective one way function P).

Hence, if $\mu > max\{(3nl + k)^{1/c_{owp2}}, (5nl + 2k)^{1/c_{owp1}}\}$, $Pr[Q = 1]\epsilon_{Q=1} < 2^{-2nl-k}$ and the claim holds. □

Claim 5. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{x,4}) = 1] - Pr[D(\mathsf{H}_{x,5}) = 1] \mid < O(n \cdot 2^{-2nl-k})$.*

Proof. This claim follows from the security of the puncturable PRF's. This is similar to the proof of the claim 3. □

Claim 6. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{x,5}) = 1] - Pr[D(\mathsf{H}_{x,6}) = 1] \mid < O(p(k) \cdot 2^{-2nl-k})$. Here $p(\cdot)$ is a some polynomial*

Proof. This claim follows from the indistinguishability obfuscation security of the weak extractability obfuscator. This proof is similar the proof of the claim 2. □

Claim 7. *For any PPT distinguisher D, $\mid Pr[D(\mathsf{H}_{x,6}) = 1] - Pr[D(\mathsf{H}_{x,7}) = 1] \mid < O(n \cdot 2^{-2nl-k})$.*

Proof. This claim follows from the indistinguishability obfuscation security of the weak extractability obfuscator $\mathcal{O}$. This proof is similar the proof of the claim 1. □

Combining all the claims above, we prove the lemma. □

Lemma 10. *For any PPT distinguisher D and $x \in [2^{2ln}]$, $\mid Pr[D(\mathsf{H}_{2^{2ln}+4+x}) = 1] - Pr[D(\mathsf{H}_{2^{2ln}+5+x}) = 1] \mid < O(v(k) \cdot 2^{-2nl-k})$ for some polynomial $v(k)$.*

Proof. Proof of this lemma is similar to the proof of lemma 9. □

Combining all these lemmas above, we get that for any PPT D,

$$\mid Pr[D(\mathsf{H}_0) = 1] - Pr[D(\mathsf{H}_{2.2^{2ln}+6}) = 1] \mid < negl(k) + 2.2^{2nl} O(v(k) \cdot 2^{-2nl-k}) < negl(k)$$

□

References

[ABG+13] Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. *IACR Cryptology ePrint Archive*, 2013:689, 2013. 4, 6

[ABSV14] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. The trojan method in functional encryption: From selective to adaptive security, generically. *IACR Cryptology ePrint Archive*, 2014:917, 2014. 6

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 308–326. Springer, 2015. 6

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 52–73. Springer, 2014. 4, 5, 6, 8, 11

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001. 3, 4, 5

[BGI13] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. *IACR Cryptology ePrint Archive*, 2013:401, 2013. 5, 11

[BGI+14] Amos Beimel, Ariel Gabizon, Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, and Anat Paskin-Cherniavsky. Non-interactive secure multiparty computation. *IACR Cryptology ePrint Archive*, 2014:960, 2014. 3

[BGJS] Saikrishna Badrinarayanan, Divya Gupta, Abhishek Jain, and Amit Sahai. Multi-input functional encryption for unbounded arity functions. *ASIACRYPT*, 2015. 6

[BKS15] Zvika Brakerski, Ilan Komargodski, and Gil Segev. From single-input to multi-input functional encryption in the private-key setting. *IACR Cryptology ePrint Archive*, 2015:158, 2015. 6

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a nash equilibrium. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:1, 2015. 5

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011. 3

[BV15] Nir Bitansky and Vinod Vaikunthanathan. Indistinguishability obfuscation from functional encryption. *IACR Cryptology ePrint Archive*, 2013, 2015. 6

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. *IACR Cryptology ePrint Archive*, 2013:352, 2013. 5, 11

[CGJS15] Nishanth Chandran, Vipul Goyal, Aayush Jain, and Amit Sahai. Functional encryption: Decentralised and delegatable. *IACR Cryptology ePrint Archive*, 2015. 5, 8

[CIJ+13] Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 519–535. Springer, 2013. 5

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Dodis and Nielsen [DN15], pages 468–497. 6

[DN15] Yevgeniy Dodis and Jesper Buus Nielsen, editors. *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*. Springer, 2015. 19, 20

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 578–602. Springer, 2014. 3, 4, 5, 9, 11

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49. IEEE Computer Society, 2013. 3, 6

[GGHW14] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 518–535. Springer, 2014. 4

[GGKT05] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005. 7

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 555–564. ACM, 2013. 3

[Hol06] Thomas Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2006. 7

[IPS15] Yuval Ishai, Omkant Pandey, and Amit Sahai. Public-coin differing-inputs obfuscation and its applications. In Dodis and Nielsen [DN15], pages 668–697. 6

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 419–428, 2015. 6

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 669–684. ACM, 2013. 5, 11

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005. 3

[Wat14] Brent Waters. A punctured programming approach to adaptively secure functional encryption. *IACR Cryptology ePrint Archive*, 2014:588, 2014. 6

[Wee05] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 523–532. ACM, 2005. 7

[Wee07] Hoeteck Wee. One-way permutations, interactive hashing and statistically hiding commitments. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 419–433. Springer, 2007. 7

A Completing proofs of $(1, \delta)$ weak extractability obfuscator

Theorem 4. *Assuming sub-exponentially secure indistinguishability obfuscation, there exists* $(1, \delta)$ *weak obfuscator for P/poly for any* $\delta > 2^{-k}$*, where* k *is the size of the circuit.*

Proof. We now construct a $(1, 2^{-k})$ input weak extractability obfuscator from sub-exponentially secure indistinguishability obfuscation. Following algorithm describes the obfuscation procedure.

$we\mathcal{O}(1^k, C)$: The procedure outputs $C' \leftarrow i\mathcal{O}(1^{k^{1/\alpha}}, C)$. Here, $\alpha > 0$ is a constant chosen such that any polynomial time adversary against indistinguishability obfuscation has security gap upper bounded by $2^{-k}/4$.

Correctness and polynomial slowdown follows from the properties of the indistinguishability obfuscator. Now we formally describe the extractor E. Let us assume that the circuits (C_0, C_1) take as n bit inputs and there is an adversary $\mathcal{A}$ for which $1 \geq \epsilon(k) > 2^{-k}$. Here, $2^{-k}/4$ is the indistinguishability gap corresponding to $we\mathcal{O}$ for equivalent circuits of size k and $n = O(k)$ inputs.

We construct an extractor that runs in $O(n.(2t_{\mathcal{A}} + poly(k)).k.1/\epsilon^2)$ time and extracts the differing input with overwhelming probability. The algorithm takes as input two circuits, auxillary input to the adversary z and distinguishing gap ϵ and outputs the differing input d. For simplicity, lets assume that:

$$Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_0), C_0, C_1, z) = 1] < Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_1), C_0, C_1, z) = 1]$$

Otherwise, the extractor can be run twice using the different sign for this advantage.

Our extractor is described in figure 8. For now, assume that there is a single differing input d. Intuitively, our extractor predicts the differing input bit by bit. For an index $i \in [n]$, E defines a circuit C_{mid} that on input x computes $C_{x_i}(x)$, where x_i is the i^{th} bit of x. One can check that for $b \in \{0, 1\}$, when $d_i = b$, then C_b and C_{mid} are equivalent. Our extractor, knows that the distinguishing advantage of $\mathcal{A}$ satisfies that

$$Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_0), C_0, C_1, z) = 1] + \epsilon < Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_1), C_0, C_1, z) = 1]$$

It estimates the advantage of $\mathcal{A}$ in distinguishing C_0 with C_{mid} and similarly the advantage of $\mathcal{A}$ in distinguishing C_{mid} with C_1 (by repeating experiments many times). Because the advantage of adversary in distinguishing obfuscation of two equivalent circuits is smaller than $2^{-k}/4$ (due to security of indistinguishability obfuscation), it compares the the two advantage and if it finds that advantage of distinguishing obfuscation of C_b from that of C_{mid} is less than the advantage advantage of distinguishing obfuscation of C_{1-b} from that of C_{mid}, it outputs $d_i = b$.

Lemma 11. *Suppose there exists circuits* $C_0, C_1 \in \mathcal{C}_k$ *(set of circuits with size* k*) with* $n = O(k)$ *input bits and disagreeing on at most single input, and auxiliary input* z *for which*

$$| Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_0), C_0, C_1, z) = 1] - Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_1), C_0, C_1, z) = 1] | \geq \epsilon(k)$$

where $2^{-k} < \epsilon \leq 1$ *Then the algorithm* E *on input* $(1^k, C_0, C_1, z, \epsilon)$ *terminates within expected time* $O(n.(2t_{\mathcal{A}} + poly(k)).1/\epsilon^2)$ *and it holds that* $Pr[v \leftarrow E(1^k, C_0, C_1, z, \epsilon) : C_0(v) \neq C_1(v)] \geq 1 - negl(k)$

Proof. Lemma follows from the two claims below..

Input: C_0, C_1, z, ϵ

1. initialise $d = 0$
2. For $i = 1$ to n
   (a) Initialise $L_i = 0, R_i = 0$
   (b) For $j \in [t = k/\epsilon^2]$
      i. Using Figure 9 compute, $L_{i,j} = \mathcal{A}(we\mathcal{O}(C_{Mid}^{C_0,C_1,i}), C_0, C_1, z) - \mathcal{A}(we\mathcal{O}(C_0), C_0, C_1, z)$
      ii. Using Figure 9 compute, $R_{i,j} = \mathcal{A}(we\mathcal{O}(C_1), C_0, C_1, z) - \mathcal{A}(we\mathcal{O}(C_{Mid}^{C_0,C_1,i}), C_0, C_1, z)$
      iii. $L_i = L_i + L_{i,j}$
      iv. $R_i = R_i + R_{i,j}$
   (c) If $L_i < R_i$ set $d_i = 0$ else set $d_i = 1$
3. Output d if $C_0(d) \neq C_1(d)$, $\perp$ otherwise.

Figure 8: Extractor E

Constants: $C_0, C_1, i \in [n]$.
Input: $x \in \{0,1\}^n$

- If $x_i = 0$ output $C_0(x)$.
- If $x_i = 1$ output $C_1(x)$.

Figure 9: Program $C_{Mid}^{C_0,C_1,i}$

Claim 8. *E runs in expected $O(n.(2t_{\mathcal{A}} + poly(k)).k.1/\epsilon^2)$ time where $t_{\mathcal{A}}$ is the expected running time of the adversary $\mathcal{A}$.*

Proof. Let us analyse expected running time of the algorithm E. The algorithm predicts bit by bit the differing input d. For each bit $i \in [n]$, the extractor repeats inner loop $t = k/\epsilon^2$ times. Each execution of the inner loop has an expected termination time of $2t_{\mathcal{A}}+poly(k)$ where $poly(k)$ represents time to compute challenge obfuscation and other computation steps. Hence, time taken to run E is $O(n.(2t_{\mathcal{A}} + poly(k)).1/\epsilon^2)$ □

Claim 9. *E succeeds with an overwhelming probability.*

Proof. Let us analyse the success probability of the extractor. If the circuits are equivalent then differing advantage of the adversary between the obfuscations of the circuits cannot be greater than or equal to ϵ, due to the sub-exponential security of indistinguishability obfuscation. In this case the extractor always outputs $\perp$.

Assuming that there is a differing input d, then let U_i denote the event that d_i is incorrectly predicted by the extractor in the i^{th} loop. We need to show that $Pr[\cup_{i\in[n]} U_i] < negl(k)$. We claim this by bounding $Pr[U_i]$ for any $i \in [n]$ and applying union bound.

Let us calculate the probability that d_i is incorrectly calculated given that $d_i = 0$ (In this case $C_{Mid}^{C_0,C_1,i}$ and C_0 are functionally equivalent). This is probability is equal to $Pr[R_i \leq L_i/d_i = 0]$, where $R_i - L_i$ is calculated by the program E during loop i (i.e. when the program predicts d_i). $Z_i = R_i - L_i = \Sigma_{j\in[t]} R_{i,j} - L_{i,j}$ is a random variable that sums the intermediate summands $(R_{i,j} - L_{i,j})$ during the j^{th} execution of the

inner loop, while predicting d_i. Define $Z_{i,j} = R_{i,j} - L_{i,j}$. We need to bound $Pr[Z_i \leq 0]$.

We now use the following chernoff bound for achieving this. Given $X = X_1 + .. + X_N$ where each X_i $\forall i \in [N]$ are independent random variables in $[0, 1]$ and $\mu = \mathbb{E}(X)$. Then, for any $\alpha \geq 0$ we have that, $Pr[X \leq (1-\alpha)\mu] \leq e^{-\alpha^2\mu/2}$.

In order to apply this chernoff bound, we define $Z'_{i,j} = (Z_{i,j} + 2)/4$ and $Z'_i = \Sigma_j(Z_{i,j} + 2)/4$ and upper bound $Pr[Z_i < 0] = Pr[Z'_i < t/2]$. Let $\mathbb{E}(Z_i) = p.t$ for some $p > 0$ (by assumption that $Pr[\tilde{C} \leftarrow we\mathcal{O}(1^k, C_0) : \mathcal{A}(1^k, \tilde{C}, C_0, C_1, z) = 1] < Pr[\tilde{C} \leftarrow we\mathcal{O}(1^k, C_1) : \mathcal{A}(1^k, \tilde{C}, C_0, C_1, z) = 1]$). On some computation we get that $Pr[Z_i < 0] = e^{-p^2t/8(p+2)}$. Now we compute p.

For any $j \in [t]$, we have that $p = \mathbb{E}(R_{i,j}/d_i = 0) - \mathbb{E}(L_{i,j}/d_i = 0)$. It is easy to see that $\mathbb{E}(R_{i,j}/d_i = 0)$ is the advantage of the adversary in distinguishing the obfuscation of $C^{C_0,C_1,i}_{Mid}$ when $d_i = 0$ from that of C_1 and similarly $\mathbb{E}(L_{i,j}/d_i = 0)$ is the advantage of the adversary in distinguishing the obfuscation of $C^{C_0,C_1,i}_{Mid}$ from that of C_0 when $d_i = 0$. Note that, $\mathbb{E}(R_{i,j}/d_i = 0) + \mathbb{E}(L_{i,j}/d_i = 0) > \epsilon$ (due to the assumption that, $Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_0), C_0, C_1, z) = 1] + \epsilon < Pr[\mathcal{A}(1^k, we\mathcal{O}(1^k, C_1), C_0, C_1, z) = 1]$).

Since $d_i = 0$, $\mid \mathbb{E}(L_{i,j}/d_i = 0) \mid < 2^{-k}/4$ due to subexponential security of the indistinguishability obfuscation. From these observations, we have that $p \geq \epsilon - 2.2^{-k}/4$. Since $\epsilon > 2^{-k}$, $p > \epsilon/2$ hence, when t is set as k/ϵ^2, $Pr[Z_i < 0] \leq e^{-k/16}$.

Similarly when $d_i = 1$, we can upper bound probability of incorrect prediction of d_i as $Pr[R_i > L_i] \leq e^{-k/16}$, when $t = k/\epsilon^2$. This proves $\forall i \in [n], Pr(U_i) < negl(k)$. □

□

□