Simulation-Based Secure Functional Encryption in the Random Oracle Model

Vincenzo Iovino[1] and Karol Żebroski[2]

University of Luxembourg, `vincenzo.iovino@uni.lu`
University of Warsaw, `kz277580@students.mimuw.edu.pl`

Abstract. One of the main lines of research in functional encryption (FE) has consisted in studying the security notions for FE and their achievability. This study was initiated by [Boneh et al. – TCC'11, O'Neill – ePrint'10] where it was first shown that for FE the indistinguishability-based (IND) security notion is not sufficient in the sense that there are FE schemes that are provably IND-Secure but concretely insecure. For this reason, researchers investigated the achievability of Simulation-based (SIM) security, a stronger notion of security. Unfortunately, the above-mentioned works and others [e.g., Agrawal et al. – CRYPTO'13] have shown strong impossibility results for SIM-Security. One way to overcome these impossibility results was first suggested in the work of Boneh et al. where it was shown how to construct, in the Random Oracle (RO) model, SIM-Secure FE for restricted functionalities and was asked the generalization to more complex functionalities as a challenging problem in the area. Subsequently, [De Caro et al. – CRYPTO'13] proposed a candidate construction of SIM-Secure FE for all circuits in the RO model assuming the existence of an IND-Secure FE scheme for circuits *with RO gates*. To our knowledge there are no proposed candidate IND-Secure FE schemes for circuits with RO gates and they seem unlikely to exist. We propose the first constructions of SIM-Secure FE schemes in the RO model that overcome the current impossibility results in different settings. We can do that because we resort to the two following models:
  - In the public-key setting we assume a bound on the number of queries but this bound only affects the *running-times* of our encryption and decryption procedures. We stress that our FE schemes in this model are SIM-Secure and have ciphertexts and tokens of *constant-size*, whereas in the standard model, the current SIM-Secure FE schemes for general functionalities [De Caro et al., Gorbunov et al. – CRYPTO'12] have ciphertexts and tokens of size growing as the number of queries.
  - In the symmetric-key setting we assume a timestamp on both ciphertexts and tokens. In this model, we provide FE schemes with short ciphertexts and tokens that are SIM-Secure against adversaries asking an *unbounded* number of queries.
Both results also assume the RO model, but not functionalities with RO gates and rely on extractability obfuscation [Boyle et al. – TCC'14] (and other standard primitives) secure only in the *standard model*.

Keywords. Functional Encryption, Random Oracle Model, Simulation-Based Security, Obfuscation.

1   Introduction

The study of simulation-based (SIM) notions of security for functional encryption were initiated only recently by Boneh, Sahai, and Waters [1] and O'Neill [2]. Quite interestingly, they show there exists clearly insecure FE schemes for certain functionalities that are nonetheless deemed secure by IND-Security, whereas these schemes do not meet the stronger notion of SIM-Security.

For this reason, researchers have started a further theoretical study of FE that includes either negative results showing SIM-Security is not always achievable [1, 3, 4] or alternative models overcoming the impossibility results [5, 6]. On the positive direction, Boneh *et al.* [1] showed the existence of SIM-Secure FE schemes in the Random Oracle (RO, in short) [7] model for restricted functionalities (i.e., Attribute-based Encryption), and at the same time they left as a challenging problem the construction of FE for more sophisticated functionalities that satisfy SIM-Security in the RO model. More recently, De Caro *et al.* [8] showed how to overcome all known impossibility results assuming the existence of IND-Secure schemes for circuits with *random oracle gates*. This is a very strong assumption for which we do *not* know any candidate scheme and their existence seems unlikely.[1]

Furthermore, their scheme incurs the following theoretical problem. First of all, recall that in a FE system for functionality $F : K \times X \to \Sigma$, defined over *key space K*, *message space X* and *output space Σ*, for every *key $k \in K$*, the owner of the master secret key Msk associated with master public key Pk can generate a secret key Tok_k that allows the computation of $F(k, x)$ from a ciphertext of x computed under master public key Pk. Thus, in a standard FE scheme the functionality is fixed in advance, and the scheme should allow to compute over encrypted data accordingly to this functionality. Instead, in the scheme for the RO model of De Caro *et al.* [8], the functionality does *depend* on the RO, and thus, even their implicit definition of functionality and FE scheme is not standard. Therefore, their scheme is not satisfactory.

This leads to the main question that we study in this work:

Can we achieve standard FE schemes in the (conventional) Programmable RO model from reasonable assumptions?

Our results answer affirmatively to this question demonstrating the existence of SIM-Secure schemes in the RO model with short parameters.

---

[1] This issue was first noticed by several researchers [9] and personally communicated by Jonathan Katz to the authors of the work [8].

The impossibility result of [4] shows that in a SIM-Secure FE scheme, the size of the ciphertexts has to grow as the number of token queries (see also [10] and [5]). On the other hand, our results also provide schemes for the RO model that are SIM-Secure but in which the size of the tokens and the ciphertexts is constant.[2] Before presenting our positive results in more detail, we prefer to first sketch our techniques so to highlight the technical problems that led to our constructions and models.

Our techniques. We recall (a simplified version of) the transformation of De Caro *et al.* [8] to bootstrap an IND-Secure scheme for Boolean circuits to a SIM-Secure scheme for the same functionality. For sake of simplicity we focus on the non-adaptive setting, specifically on SIM-Security against adversaries asking q non-adaptive queries.[3] The idea of their transformation is to replace the original circuit with a "trapdoor" one that the simulator can use to program the output in some way. In the transformed scheme, they put additional "slots" in the plaintexts and secret keys that will only be used by the simulator. A plaintext has $2 + 2q$ slots and a secret key will have one. In the plaintext, the first slot is the actual message m and the second slot will be a bit flag indicating whether the ciphertext is in trapdoor mode. and the last $2q$ slots will be q pairs (r_i, z_i), where r_i is a random string and z_i is a programmed string. These $2q$ slots are used to handle q non-adaptive queries. On the other hand, in a secret key for circuit C, the slot is a random string r, that will be equal to one of the r_i in the challenge ciphertext. For evaluation, if the ciphertext is not in trapdoor mode (flag $= 0$) then the new circuit simply evaluates the original circuit C of the message m. If the ciphertext is in trapdoor mode, if $r = r_i$ for some $i \in [q]$ then the transformed circuit outputs z_i.

A natural approach to shorten the size of the ciphertexts in the RO model would be the following. Recall that a Multi-Input FE (MI-FE) scheme [11–13] is a FE over multiple ciphertexts. Let our starting scheme

---

[2] Specifically, in our main transformation the size of the tokens is constant if we employ a collision-resistant hash function of variable-length, otherwise their size only depends on the encoding of the value and thus can be sub-logarithmic. Similarly, for the timestamp model of Section 3.3, both tokens and ciphertexts need to encode a temporal index that being a number at most equal to the number of queries issued by any PPT adversary, will be at most super-logarithmic, and thus can be encoded with a string of poly-logarithmic size. For simplicity, henceforth, we will claim that our constructions have tokens of constant size omitting to specify this detail.

[3] Henceforth, we mean by non-adaptive queries the queries that the adversary asks before seeing the challenge ciphertext and adaptive queries the queries the adversary asks after seeing it.

be a MI-FE over 2-inputs. Then, instead of encoding the slots (r_i, z_i)'s in the ciphertext, we could add to the ciphertext a tag $\mathsf{tag_c}$ (that is, the encryption would consist of a ciphertext plus the tag in clear) such that the simulator can program the RO on this point to output the values (r_i, z_i)'s. Now the ciphertext would consist of only a short ciphertext ct and the short tag $\mathsf{tag_c}$. At decryption time, we could "decompress" $\mathsf{tag_c}$ to get some string y, encrypt it with the public-key of the of the MI-FE scheme to produce $\mathsf{ct_2}$ and finally feed $\mathsf{ct_1}$ and $\mathsf{ct_2}$ to the multi-input token. Then, the simulator could program the RO so to output the values (r_i, z_i)'s on the point $\mathsf{tag_c}$. The functionality would be thus modified so that, in trapdoor mode, the output would be taken by the values (r_i, z_i)'s (specifically, choosing the values z_i corresponding to the string r_i in the token). Therefore, any token applied to the simulated ciphertext (that is in trapdoor mode) should decrypt the same output as the real ciphertext would do.

This simple approach incurs more problems, the most serious being that the adversary could feed the multi-input token with a *different* ciphertext that does not correspond to $\mathcal{RO}(\mathsf{tag_c})$, thus detecting whether the first ciphertext is in normal or trapdoor mode. In fact, notice that in normal mode the second ciphertext does not affect the final output, whereas in trapdoor mode, the output only depends on the second ciphertext fed to the multi-input token. Another problem here is that $\mathcal{RO}(\mathsf{tag_c})$ should not contain the values (r_i, z_i)'s in clear, but this is easily solved by letting $\mathcal{RO}(\mathsf{tag_c})$ be an encryption of them and putting the corresponding secret-key in the first ciphertext. So, the main technical problem is:

> How can we force the adversary to feed the 2-inputs token with a second ciphertext that encrypts $\mathcal{RO}(\mathsf{tag_c})$?

Note that in the case of FE schemes that support functionalities with RO gates, this can be easily done by defining a new functionality that first tests whether the second input equals $\mathcal{RO}(\mathsf{tag_c})$, but in the "pure" RO model this solution can not be applied. Our patch is to add a new slot h of *short* size in the first ciphertext. Such value h is set to the hash of $\mathcal{RO}(\mathsf{tag_c})$ with respect to a Collision-Resistant Hash Function (CRHF) Hash, i.e., $h = \mathsf{Hash}(\mathcal{RO}(\mathsf{tag_c}))$. Furthermore, we modify the transformed functionality so that it first checks whether $\mathsf{Hash}(\mathcal{RO}(\mathsf{tag_c})) = h$. If this test fails, the functionality outputs an error $\perp$. The intuition is that with this modification, the adversary is now forced to use a second ciphertext that encrypts $\mathcal{RO}(\mathsf{tag_c})$ since otherwise it gets $\perp$ on both real or simulated ciphertext, and so, under the IND-Security of the MI-FE scheme, it seems

that the adversary can not tell apart a real ciphertext from a simulated ciphertext. Unfortunately, we are not able to prove the security of this transformation assuming only the standard notion of IND-Security for MI-FE. In fact, notice that there *exist* second inputs for the modified functionality that distinguish whether the first input has the flag set to normal or trapdoor mode, namely inputs that correspond to collisions of Hash with respect to h and $\mathcal{RO}(\mathsf{tag_c})$. That is, any another second ciphertext that encrypt a value $y \neq \mathcal{RO}(\mathsf{tag_c})$ such that $\mathsf{Hash}(y) = h$ allows to distinguish whether the first ciphertext is in normal or trapdoor mode. Furthermore, it is not possible to make direct use of the security of the CRHF. The problem is that the definition of (2-inputs) MI-FE is too strong in that it requests the adversary to output two challenge message pairs (x_0, y) and (x_1, y) such that for any function f for which the adversary asked a query $f(x_0, \cdot) = f(x_1, \cdot)$. In our case, this does not hold: there exists a set of collisions C such that for any $z \in C$, $\mathsf{Hash}(z) = h$ and $f(x_0, z) \neq f(x_1, z)$. However, notice that it seems difficult for the adversary to find such collisions.

Our assumptions and CRIND-Security. For these reasons, we need to extend the notion of MI-FE to what we call *collision-resistant indistinguishability* (CRIND, in short)[4]. In Section 4 we provide an instantiation of this primitive from extractability obfuscation w.r.t. distributional auxiliary input [14] (cf. Remark 3). We think that this definition can be of independent interest since it is more tailored for the applicability of MI-FE to other primitives. The reader may have noticed that the security of the second ciphertext guaranteed by the underlying MI-FE is not *necessary*. That is, our transformation would work even assuming 2-inputs MI-FE systems that take the second input *in clear*. In fact, CRIND-Security does *not* imply IND-Security for MI-FE schemes but this suffices for our scopes.

Roughly speaking, in CRIND-Security the security is quantified only with respect to *valid* adversaries[5], where an adversary is considered valid if it only submits challenges m_0, m_1 and asks a set of queries K that satisfy some "hardness" property called *collision-resistance compatibility*, namely that it is difficult to find a second input m_2 such that $F(k, m_0, m_2) \neq F(k, m_1, m_2)$ for some $k \in K$. Since in the reductions to CRIND-Security

---

[4] Maybe, a better name would have been "differing-inputs indistinguishability" but we do not adopt this name to not overlap with differing-inputs obfuscation and because it recalls the reason to advocate this stronger notion for our transformations.

[5] We can recast IND-Security in a similar way by defining valid adversaries that only ask queries and challenges satisfying the compatibility property.

it is not *generally* possible to directly check the hardness of (K, m_0, m_1), the definition dictates (1) the existence of an efficient *checker* algorithm that approves *only* (but possibly not all) valid triples (K, m_0, m_1) (i.e., the checker can detect efficiently if a triple is collision-resistant compatible) and (2) that an adversary is valid if it only asks triples approved by the checker. We defer the details of the definition to Section 2.1.

Next, in a security reduction to CRIND-Security (i.e., when we need to prove the indistinguishability of two hybrid experiments assuming the CRIND-Security), the main task is to define an appropriate checker and prove that triples that are not collision-resistant compatible are rejected by it. This is usually done by checking that messages and keys satisfy an appropriate format. For instance, in the above case, the checker will check whether the machine (corresponding to the token) uses as sub-routine the specified CRHF and that the challenge messages and such machine have the right format. The construction of CRIND-Secure schemes follows the lines of the construction of fully IND-Secure FE schemes of Boyle *et al.* Namely, the encryption of the first input m_1 will be an obfuscation of a machine that has embedded m_1 and a verification key for a signature scheme and takes as input a signature of a machine M and a second input m_2 and (1) checks the validity of the signature and (2) if such test passes outputs $M(m_1, m_2)$. For the same resonas of Boyle *et al.* we need to resort to functional signatures. Details along with a broader overview can be found in Section 4.

The above presentation is an oversimplification and we defer the reader to the Sections 3.1 and 3.2 for more details.

Our models and results. The reader may have noticed that the output of $\mathcal{RO}$ has to be "big", i.e., its size depends on the number of queries. Of course, we could assume that its range has constant size and replace a single invocation of the RO with range of size $> q$ with many invocation of a RO with range of constant size, but also in this case the *running-time* of the encryption and decryption procedures would have to depend on the number of queries. Here, it is the novelty of our approach. All the parameters of our SIM-Secure public-key FE scheme (including ciphertexts and tokens) have *constant* size but the cost of the "expansion" is moved from the length of ciphertexts and tokens to the running-time of the encryption and decryption procedures. That is, our SIM-Secure public-key FE scheme stills depends on q in the setup and running-time, but the *size* of the ciphertexts and tokens is *constant*. The results we achieve can be summarized as follows:

- $(q_1, q_c, \mathsf{poly})$-SIM-Security with ciphertext of constant size and tokens of size q_c. That is, SIM-Security against adversaries asking bounded non-adaptive token queries, bounded ciphertext queries, and unbounded adaptive token queries. In this case the size of the ciphertexts is constant but the size of the tokens grows as the number of ciphertext queries (and thus is constant in the case of 1 ciphertext query). This is known to be impossible in the standard model due to the impossibility of Agrawal *et al.* [4] (precisely this impossibility does not rule out the existence of schemes that satisfy this notion of security but it rules out the existence of schemes that satisfy both this notion of security *and* have short ciphertexts). Moreover, in this case the encryption and decryption procedures have running-times depending on q_1.
- (q_1, q_c, q_2)-SIM-Security with both ciphertexts and tokens of constant size. That is, SIM-Security against adversaries asking bounded token (both non-adaptive and adaptive) and ciphertext queries but with both ciphertext and token of constant size. In the standard model this is known to be impossible due to the impossibility result of De Caro and Iovino [5] for SIM-Security against adversaries asking unbounded ciphertext queries and bounded adaptive token queries (this impossibility is essentially an adaptation of the impossibility of Agrawal *et al.* [4]). In this case, the encryption and decryption procedures have running-times depending on $\max\{q_1, q_c, q_2\}$.
- We show how to remove the afore-mentioned limitation in a variant of the symmetric-key model where ciphertexts and tokens are tagged with a timestamp that imposes an order on their generation (i.e., the i-th token/ciphertext generated in the system is tagged with the value i). This model is reasonable because in the symmetric-key setting, the user that set-up the system is the same entity that generates tokens and ciphertexts as well. [6] We defer the reader to Section 3.3 for more details.

For the sake of providing constructions with optimal parameters we work in the *Turing Machine* model of computation.[7]

---

[6] The same considerations also hold in applications where many users share the same secret-key. Indeed, in this case one needs to assume that such users trust each other, and thus they will tag the ciphertexts and tokens with the correct timestamp.

[7] The main focus of the work of Goldwasser *et al.* [13] is for the circuit model but they sketch how to extend it to the Turing Machine model. Similar considerations hold for the schemes of Gordon *et al.* [12]. Further details will be given in the Master's Thesis of the second author.

The optimality and the soundness of our results. It is easy to see
that SIM-Security in the standard model but for schemes with procedures
of running-time dependent on the number of queries is impossible as well.
Moreover, we think that SIM-Security in the RO model with a constant
number of RO calls is impossible to achieve as well, though we were
not able to prove an impossibility result for it and leave to future work
to set positive or negative results. Anyway, one could object that if we
instantiate the RO with any concrete hash function, the resulting scheme
is *not* "SIM-Secure" due to the impossibility results. This problem is also
shared with the constructions for the RO model of De Caro *et al.* [8] and
Boneh *et al.* [1]. What does it mean?

What the impossibility results say is that there are adversaries for
which there exists no simulator though we do not know any concrete at-
tacks on these schemes. This is different from the counter-examples of
Canetti *et al.* [15] where they were presented signature schemes provably
secure in the RO model but *concretely* insecure when instantiated with
any hash function. In our view, this could merely mean that general def-
initions of SIM-Security are too strong. Along this direction, the works
of De Caro and Iovino [5] and Agrawal *et al.* [6] provide another way to
overcome this limitation.

2   Definitions

Due to space constraints we defer to the full version [16] the definitions
of functional signature scheme, collision-resistant hash function, pseudo-
random function family, single and multi-inputs functional encryption
scheme.

2.1   Collision-Resistant Indistinguishability Security for MI-FE

As we mentioned in the construction overview sketched in Section 1, we
need a different notion of MI-FE security. Here, we consider only the 2-
inputs case, since this is suited for our main transformation but it is
straightforward to extend it to the n-ary case.

Furthermore, in Section 4 we will show how to construct a CRIND-
Secure MI-FE scheme from extractability obfuscation w.r.t. distributional
auxiliary input [14] (cf. Remark 3). We presented an informal discussion
of the definition in Section 1. We now present the formal definition.

The collision-resistant indistinguishability-based notion of security for
a multi-input functional encryption scheme MI-FE = (Setup, KeyGen,

Enc, Eval) for functionality F defined over (K, M) is formalized by means of the following experiment $\mathsf{CRIND}_{\mathcal{A}}^{\mathsf{MI\text{-}FE}}$ with an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$. Below, we present the definition for only one message; it is easy to see the definition extends naturally for multiple messages.

$\mathsf{CRIND}_{\mathcal{A}}^{\mathsf{MI\text{-}FE}}(1^\lambda)$

1. $(\mathsf{Mpk}, \mathsf{Msk}) \leftarrow \mathsf{MI\text{-}FE.Setup}(1^\lambda)$;
2. $r \xleftarrow{R} \{0,1\}^\lambda$;
3. $(x_0, x_1, \mathtt{st}) \leftarrow \mathcal{A}_0^{\mathsf{MI\text{-}FE.KeyGen}(\mathsf{Msk}, \cdot)}(\mathsf{Mpk}, r)$ where we require that $|x_0| = |x_1|$;
4. $b \xleftarrow{R} \{0,1\}$;
5. $\mathsf{Ct}_1 \leftarrow \mathsf{MI\text{-}FE.Encrypt}(\mathsf{Mpk}, (x_b \| r))$;
6. $b' \leftarrow \mathcal{A}_1^{\mathsf{MI\text{-}FE.KeyGen}(\mathsf{Msk}, \cdot)}(\mathsf{Mpk}, \mathsf{Ct}_1, \mathtt{st})$;
7. **Output:** $(b = b')$.

We make the following additional requirements:

– Collision-resistance compatibility. Let K denote a set of keys. We say that a pair of messages x_0 and x_1 is *collision-resistant compatible* with K if it holds that: any (possibly, non-uniform) PPT algorithm $\mathcal{B}$ given the security parameter 1^λ and (r, x_0, x_1) for r uniformly distributed in $\{0,1\}^\lambda$, can find y satisfying inequality $F(k, (x_0 \| r), y) \neq F(k, (x_1 \| r), y)$ for some $k \in K$ with at most negligible (in λ) probability, where probability is taken over $r \xleftarrow{R} \{0,1\}^\lambda$ and the random coins of $\mathcal{B}$.[8]
– Efficient checkability. We assume that there exist efficient checker algorithm Checker, which takes as input (k, x_0, x_1) and outputs false if x_0 and x_1 are *not* collision-resistant compatible with $\{k\}$ (i.e., the singleton set containing the key k).
– Validity. We say that an adversary $\mathcal{A}$ in the above game is *valid* with respect to a checker Checker with efficient checkability if during the execution of the above game, $\mathcal{A}$ outputs challenge messages x_0 and x_1 of the same length and asks a set of queries K in the key space of

---

[8] It can appear that the definition be not well-defined because we do not specify how the key k is related to the security parameter. To understand this, you may imagine that k be the code of some algorithm P (ant thus of constant-size) to compute a keyed hash function $\mathsf{Hash}(\cdot, \cdot)$. The program P takes an hashing key s computed with respect to an arbitrarily long security parameter λ and an input x and computes $\mathsf{Hash}(s, x)$. Therefore in the above definition, k (along with the functionality F) plays the role of P and thus can have constant size whereas r plays the role of the hashing key s that depends instead on the security parameter.

the functionality such that for any $k \in K$, $\mathsf{Checker}(k, x_0, x_1) = \mathsf{true}$ (i.e., the adversary only asks queries and challenges approved by the checker).

The advantage of a valid adversary $\mathcal{A}$ in the above game is defined as

$$\mathsf{Adv}^{\mathsf{MI\text{-}FE,IND}}_{\mathcal{A}}(1^\lambda) = |\mathrm{Prob}[\mathsf{CRIND}^{\mathsf{MI\text{-}FE}}_{\mathcal{A}}(1^\lambda) = 1] - 1/2|.$$

Definition 1 We say that a 2-inputs MI-FE scheme is *collision-resistant indistinguishably secure* (CRIND-Secure, in short) if for any checker Checker satisfying efficient checkability, it holds that all PPT adversaries $\mathcal{A}$ *valid* with respect to Checker have at most negligible advantage in the above game.

2.2 Extractability obfuscation w.r.t. distributional auxiliary input

Boyl *et al.* [14] defined obfuscators secure against general distributional auxiliary input. We recall their definition (cf. Remark 3).

Definition 2 A uniform PPT machine $\mathsf{e}\mathcal{O}$ is an *extractability obfuscator w.r.t. (general) distributional auiliary input* for the class of Turing Machines $\{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ if it satisfies the following properties:

- (Correctness): For all security parameter λ, all $M \in \mathcal{M}$, all inputs x, we have
$$\Pr\left[M' \leftarrow \mathsf{e}\mathcal{O}(1^\lambda, M) \colon M'(x) = M(x) \right] = 1.$$

- (Polynomial slowdown): There exist a universal polynomial p such that for any machine M, we have $|M'| \leq p(|M|)$ for all $M' = \mathsf{e}\mathcal{O}(1^\lambda, M)$ under all random coins.

- (Security): For every non-uniform PPT adversary $\mathcal{A}$, any polynomial $p(\lambda)$, and efficiently sampleable distribution $\mathcal{D}$ over $\mathcal{M}_\lambda \times M_\lambda \times \{0,1\}^\star$, there exists a non-uniform PPT extractor E and polynomial $q(\lambda)$ and negligible function $\mathsf{negl}(\lambda)$ such that, for every $\lambda \in \mathbb{N}$, with probability $\geq 1 - \mathsf{negl}(\lambda)$ over $(M_0, M_1, z) \leftarrow \mathcal{D}(1^\lambda)$, it holds that:
If $\Pr\left[b \xleftarrow{R} \{0,1\}; M' \leftarrow \mathsf{e}\mathcal{O}(1^\lambda, M_b) : \mathcal{A}(1^\lambda, M', M_0, M_1, z) = b \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)}$,
then $\Pr\left[w \leftarrow \mathsf{E}(1^\lambda, M_0, M_1, z) : M_0(w) \neq M_1(w) \right] \geq \frac{1}{q(\lambda)}$

Remark 3 In light of the recent "implausibility" results on extractability obfuscation with auxiliary input [17, 18], we would like to point out that our results are based on *specific* distributions for which no implausibility result is known. Same considerations were also made in [14].

3 Our Transformations

3.1 $(q_1, q_c, \mathsf{poly})$-SIM-Security

In this section, we show that assuming a CRIND-Secure (in the standard model) MI-FE scheme for $p\text{-}\mathsf{TM}_2$ (2-inputs Turing machines with run time equal to a polynomial p) for any polynomial p, it is possible to construct a SIM-Secure functional encryption scheme in the RO model for functionality $p\text{-}\mathsf{TM}$ for any polynomial p. Moreover, this is possible also for FE schemes with input-specific run time. The resulting scheme is secure for a bounded number of messages and non-adaptive token queries, and unbounded number of adaptive key queries. Moreover, it enjoys ciphertexts and tokens of size not growing with the number of non-adaptive queries, overcoming the impossibility result of Agrawal *et al.* [4] for the standard model. In Section 4 we will show how to construct a CRIND-Secure MI-FE from extractability obfuscation w.r.t. distributional auxiliary input [14] (cf. Remark 3).

Trapdoor Machines The idea of our transformations is to replace the original machine with a "trapdoor" one that the simulator can use to program the output in some way. This approach is inspired by the FLS paradigm introduced by Feige, Lapidot and Shamir [19] to obtain zero-knowledge proof systems from witness indistinguishable proof systems. Below we present the construction of trapdoor machine, which works in standard model.

Definition 4 [Trapdoor Machine] Fix $q > 0$. Let M be a Turing machine with one input. Let $\mathsf{SE} = (\mathsf{SE.Enc}, \mathsf{SE.Dec})$ be a symmetric-key encryption scheme with key-space $\{0,1\}^\lambda$, message-space $\{0,1\}^{\lambda+1}$, and ciphertext-space $\{0,1\}^\nu$. We require for simplicity that SE has pseudorandom ciphertexts (see the full version [16]) and can encrypt messages of variable length (at most $\lambda + 1$). Let $\mathsf{Hash}\colon \{0,1\}^\lambda \times \{0,1\}^{q\cdot\nu} \to \{0,1\}^\lambda$ be a collision resistant hash function [9]. For $\mathsf{tag_k} = (id_k, c) \in \{0,1\}^{\lambda+\nu}$ define the corresponding *trapdoor machine* $\mathsf{Trap}[M, \mathsf{Hash}, \mathsf{SE}]^{\mathsf{tag_k}}$ on two inputs as follows:

---

[9] For sake of simplicity as Hash key we will use a random string of length λ, instead of key generated by Gen. Alternatively, we could feed the Gen algorithm with this randomness.

$\textbf{Machine } \mathsf{Trap}[M, \mathsf{Hash}, \mathsf{SE}]^{\mathsf{tag_k}}(m', R)$
$\quad (m, \mathsf{flag}, sk, h, k_{\mathsf{H}}) \leftarrow m'$
$\quad \text{If } \mathsf{Hash}(k_{\mathsf{H}}, R) \neq h \text{ then return } \bot$
$\quad \text{If } \mathsf{flag} = 0 \text{ then return } M(m)$
$\quad (id_k, c) \leftarrow \mathsf{tag_k}$
$\quad (R_1, \ldots, R_q) \leftarrow R$
$\quad \text{For } i = 1, \ldots, q \text{ do}$
$\quad\quad (id_k', v) \leftarrow \mathsf{SE.Dec}(sk, R_i)$
$\quad\quad \text{If } id_k' = id_k \text{ then return } v$
$\quad \text{return } \mathsf{SE.Dec}(sk, c)$

RO-based Transformation

Overview. In Section 1 we sketched a simplified version of our transformation. Here, we present an overview with more details. The idea is to put additional "slots" in the plaintexts and secret keys that will only be used by the simulator. A plaintext contains five slots and a secret key contains two slots. In the plaintext, the first slot is the actual message m. The second slot is a bit flag indicating whether the ciphertext is in trapdoor mode. The third slot is a random key sk used by SE scheme, the fourth slot is a hash h of $\mathcal{RO}(\mathsf{tag_c})$ (computed with respect to a CRHF) attached to the ciphertext and finally the fifth slot contains a hash function key k_{H}.

In the secret key, the first slot encodes the actual machine M and the second slot is a random tag $\mathsf{tag_k} = (id_k, c)$. Slot id_k is used by simulator to identify pre-challenge tokens and c is used to convey programmed output value in post-challenge tokens. For evaluation, if the ciphertext is not in trapdoor mode (i.e., $\mathsf{flag} = 0$) then the functionality simply evaluates the original machine M of the message m. If the ciphertext is in trapdoor mode, depending on the nature of the secret key (non-adaptive or adaptive), for $\mathsf{tag_k} = (id_k, c)$, if for some $i \in [q]$, $(id_k, v) = \mathsf{SE.Dec}(sk, R_i)$, then the functionality outputs v, otherwise it outputs $\mathsf{SE.Dec}(sk, c)$. Here R_i is the i-th element in the string R that the machine takes as second input, and is set by the (honest) evaluation procedure to $\mathcal{RO}(\mathsf{tag_c})$.

For sake of simplicity we assume that TM functionality, for which our scheme is constructed, has output space $\{0, 1\}$. The construction can be easily extended to work for any TM functionality with bounded output length.

Definition 5 [RO-Based Transformation] Let $p(\cdot)$ be any polynomial. Let $\mathsf{SE} = (\mathsf{SE.Enc}, \mathsf{SE.Dec})$ be a symmetric-key encryption scheme with key-space $\{0,1\}^\lambda$, message-space $\{0,1\}^{\lambda+1}$, and ciphertext-space $\{0,1\}^\nu$. We require for simplicity that SE has pseudo-random ciphertexts (see the full version [16]) and can encrypt messages of variable length (at most $\lambda+1$). Let $\mathsf{Hash} \colon \{0,1\}^\lambda \times \{0,1\}^{q\cdot\nu} \to \{0,1\}^\lambda$ be a collision-resistant hash function (not modeled as a random oracle). Assuming that the running time of machine M equals exactly $p(|m|)$ on input m, the running time of trapdoor machine $\mathsf{Trap}[M, \mathsf{Hash}, \mathsf{SE}]^{\mathsf{tag_k}}$ is bounded by some polynomial $p'(\cdot)$. Let $\mathsf{MI\text{-}FE} = (\mathsf{MI\text{-}FE.Setup}, \mathsf{MI\text{-}FE.Enc}, \mathsf{MI\text{-}FE.KeyGen}, \mathsf{MI\text{-}FE.Eval})$ be a multi-input functional encryption scheme for the functionality p'-TM_2.

In our construction we assume that the output length of the programmable random oracle $\mathcal{RO}$ equals $q \cdot \nu$.

We define a new (single-input) functional encryption scheme $\mathsf{SimFE}[\mathsf{Hash}, \mathsf{SE}] = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for functionality p-TM as follows.

- $\mathsf{Setup}(1^\lambda)$: runs $(\mathsf{Mpk}, \mathsf{Msk}) \leftarrow \mathsf{MI\text{-}FE.Setup}(1^\lambda)$ and chooses random $r \xleftarrow{R} \{0,1\}^\lambda$ and returns a pair (Mpk, r) as public key Pk and Msk as master secret key.
- $\mathsf{Enc}(\mathsf{Pk}, m)$: on input $\mathsf{Pk} = (\mathsf{Ek_1}, \mathsf{Ek_2}, r)$ and $m \in \{0,1\}^*$, the algorithm chooses random $sk \xleftarrow{R} \{0,1\}^\lambda$, $\mathsf{tag_c} \xleftarrow{R} \{0,1\}^\lambda$ and sets $k_\mathsf{H} = r$, then it takes $m' = (m, 0, sk, \mathsf{Hash}(k_\mathsf{H}, \mathcal{RO}(\mathsf{tag_c}), k_\mathsf{H})$, computes $c \leftarrow \mathsf{MI\text{-}FE.Enc}(\mathsf{Ek_1}, m')$ and returns a pair $(c, \mathsf{tag_c})$ as its own output.
- $\mathsf{KeyGen}(\mathsf{Msk}, M)$: on input Msk and a machine M, the algorithm chooses random $id_k \xleftarrow{R} \{0,1\}^\lambda$, $c \xleftarrow{R} \{0,1\}^\nu$ and returns $(\mathsf{Tok}, \mathsf{tag_k})$ where
  $\mathsf{Tok} \leftarrow \mathsf{MI\text{-}FE.KeyGen}(\mathsf{Msk}, \mathsf{Trap}[M, \mathsf{Hash}, \mathsf{SE}]^{\mathsf{tag_k}})$ and $\mathsf{tag_k} = (id_k, c)$.
- $\mathsf{Eval}(\mathsf{Pk}, \mathsf{Ct}, \mathsf{Tok})$: on input $\mathsf{Pk} = (\mathsf{Ek_1}, \mathsf{Ek_2}, r)$, $\mathsf{Ct} = (c, \mathsf{tag_c})$ and $\mathsf{Tok} = (\mathsf{Tok'}, \mathsf{tag_k})$, computes $c' = \mathsf{MI\text{-}FE.Enc}(\mathsf{Ek_2}, \mathcal{RO}(\mathsf{tag_c}))$ and returns the output $\mathsf{MI\text{-}FE.Eval}(\mathsf{Tok'}, c, c')$.

Theorem 6 Suppose $\mathsf{MI\text{-}FE}$ is CRIND-Secure in the standard model. Then SimFE is $(q, 1, \mathsf{poly})$-SIM-Secure in the random oracle model. Furthermore, this can be extended to $(q_1, q_c, \mathsf{poly})$-$\mathsf{SIM}$-Security (see the full version [16]) and if $\mathsf{MI\text{-}FE}$ satisfies the properties of succinctness and input-specific time, so SimFE does.

Security proof overview. We conduct the security proof of our construction by a standard hybrid argument. To move from real world experiment to ideal one we use the following hybrid experiments:

– The first hybrid experiment corresponds to the real experiment.
– The second hybrid experiment is identical to the previous one except
  that the random oracle is programmed at point $\mathsf{tag_c}$ so to output the
  encryption of the desired output values on pre-challenge queries, and
  post-challenge queries are answered with tokens that have embedded
  appropriate encrypted output values. Moreover, *all* these values are
  encrypted using the underlying SE scheme with a *randomly chosen*
  secret-key sk'. Notice that in this experiment the secret-key sk' is
  uncorrelated to the secret-key sk embedded in ciphertext.
– The third hybrid experiment is identical to the previous one except
  that the ciphertext contains the same secret-key sk' used to program
  the RO.
– In last step we switch the flag slot in the ciphertext to 1 indicating the
  trapdoor mode. At the same time we change the content of message
  slot m to $0^{|m|}$. This is necessary due to the fact that simulator only
  knows the challenge message length, but not the message itself.

One can reduce the security of first two transitions to the ciphertext
pseudo-randomness of SE scheme and to the CRIND-Security of underlying $\mathsf{MI\text{-}FE}$ scheme. The proof in these cases is pretty straightforward.

One could be tempted to reduce the indistinguishability security of
last two hybrids to both collision resistance of used hash function and
IND-Security on $\mathsf{MI\text{-}FE}$. However, the security reduction is not obvious.
The adversary could recognize the simulation by finding a string R different from $\mathcal{RO}(\mathsf{tag_c})$ for which $\mathsf{Hash}(k_\mathsf{H}, R) = \mathsf{Hash}(k_\mathsf{H}, \mathcal{RO}(\mathsf{tag_c}))$, and
applying the evaluation algorithm to this value as second input. The output of evaluation algorithm in this case would be different than expected.
Although the adversary would contradict the collision resistance of Hash,
we are not able to construct algorithm based on that adversary, which
breaks the hash function security.

Therefore we need to rely on the CRIND-Security of $\mathsf{MI\text{-}FE}$. Moreover,
for completeness we will only assume CRIND-Security and never IND-Security.

We defer the proof of Theorem 6 to the full version [16].

3.2 (q_1, q_c, q_2)-SIM-Security with short tokens

The previous transformation suffers from the problem that the size of the
tokens grows as the number of ciphertext queries q_c. In this Section we
show how to achieve (q_1, q_c, q_2)-SIM-Security. Notice that in the standard
model constructions satisfying this level of security like the scheme of

Gorbunov *et al.* [20] have short ciphertexts and tokens. Moreover, De Caro and Iovino [5] showed an impossibility result for this setting. Our transformation assumes a 3-inputs MI-FE scheme (CRIND-Secure in the standard model). The resulting scheme is (q_1, q_c, q_2)-SIM-Secure according to the definition with simulated setup (see the full version [16]). The idea is very similar to the transformation presented in Section 3.1, but due to space constraints, we defer the sketch of the transformation to the full version [16].

3.3 **(poly, poly, poly)-SIM-Security in the Timestamp model**

We recall that any known impossibility results in the standard model also apply to the symmetric-key setting. In this Section we show how to achieve unbounded SIM-Security in the RO model in a variant of the symmetric-key setting that we call the *timestamp* model. Moreover, our scheme enjoys ciphertexts and tokens of constant size. The timestamp model is identical to the symmetric-key mode except for the following changes:

- The encryption and key generation procedures also take as input a *temporal index* or *timestamp*. The security of this index is not required. The security experiments are identical to those of the symmetric-key model except that the queries are answered by providing tokens and ciphertexts with *increasing* temporal index (the exact value does not matter as long as they are ordered in order of invocation). Roughly speaking, this is equivalent to saying that the procedures are stateful. Notice that in the symmetric-key model, this change has no cost since the user who set-up the system can keep the value of the current timestamp and guarantee that ciphertexts and tokens are generated with timestamps of increasing order.
- For simplicity, we also assume that there is a decryption key. Precisely, the evaluation algorithm takes as input a token, a ciphertext and a decryption key. It is easy to see that this decryption key can be removed at the cost of including it in any token or ciphertext.

Sketch of the transformation. With these changes in mind it is easy to modify the scheme of Section 3.2 to satisfy (poly, poly, poly)-SIM-Security in the RO model. Precisely, the slots for the identifiers will contain the temporal index in *clear*[10]. In the scheme, the tags will be such that the

---

[10] We stress that we could also assume that the temporal index is appended in clear to the final ciphertext.

(both non-programmed and programmed) RO on these input will output a string of size proportional to i, where i is the temporal index. This can be done by assuming that the RO outputs a single bit and invoking it many times. That is, instead of computing $\mathcal{RO}(\mathsf{tag_c})$, the procedures will compute $\mathcal{RO}(\mathsf{tag_c}||j)$ for $j = 1, \ldots, m$ where m is the needed size. For simplicity, henceforth, we assume that the RO outputs strings of variable-length. As byproduct, we need to program the RO on the tag $\mathsf{tag_c}$ (resp. $\mathsf{tag_k}$) of a ciphertext (resp. token) with timestamp i to only output i ciphertexts. Thus, we do not need to fix in *advance* any bound, which was the only limitation of the previous transformations. Notice that the evaluation algorithm needs to encrypt the output of $\mathcal{RO}(\mathsf{tag_c})$ and $\mathcal{RO}(\mathsf{tag_k})$ and this is done by using the encryption keys $\mathsf{Ek_2}$ and $\mathsf{Ek_3}$ for the second and third input. This is the reason why we assume that there is a decryption key that in this scheme consists of the pair $(\mathsf{Ek_2}, \mathsf{Ek_3})$.

Security. The security proof is identical to that of the transformation of Section 3.2 with further simplifications due to the fact that we do not need to have the temporal index encrypted. We first need to switch the slot of the identifiers in both ciphertext and tokens to be encryption of the the temporal index. Then the proof proceeds as before.

4   Constructions of CRIND-Secure MI-FE from e$\mathcal{O}$

Overview of the construction and security reduction. In order to achieve CRIND-security of MI-FE (as defined in Section 2.1), we make use of the following ideas inspired by the construction of fully IND-Secure FE of Boyle *et al.* [14]. We assume a functional signature scheme FS [21]. Namely, our encryption procedure takes as input the first input m_1 and produces an obfuscation of a machine that has embedded m_1 and takes as input a second message m_2 and a functional signature for some function f and (1) verifies the signature and (2) outputs $f(m_1, m_2)$. Roughly speaking, we want prevent the adversary to be able to find distinguishing inputs. To this scope, we need to forbid the adversary from evaluating the machine on functions for which it did not see a signature. For the same reasons as in Boyle *et al.* it is not possible to use a standard signature scheme. This is because, the adversary $\mathcal{A}$ against e$\mathcal{O}$ needs to produce a view to the adversary $\mathcal{B}$ against CRIND-Security, and in particular to simulate the *post-challenge* tokens. To that aim, $\mathcal{A}$ would need to receive an auxiliary input z containing the signing key of the traditional signature scheme but in this case an extractor with access to z could easily

find a distinguishing input. As in Boyle *et al.* we resort to functional signatures. We recall their ideas. In the scheme, they put a functional signing key that allows to sign any function. In the security proof, they use the property of function privacy to show that the original experiment is computationally indistinguishable to an experiment where the post-challenge queries are answered with respect to a *restricted* signing key for the Boolean predicate that is verified on all and only the machines T for which $T(m_0) = T(m_1)$, where m_0 and m_1 are the challenges chosen by the adversary against IND-Security. Thus, putting this restricted signing key in the auxiliary distribution does *not* hurt of the security since an extractor can not make use of it to find a distinguishing input. In the case of CRIND-Security, it is not longer true that $T(m_0) = T(m_1)$ but we will invoke the properties of the checker and we set the Boolean predicate to one that is verified for all machines T approved by the checker with respect to the challenges, i.e., such that $\mathsf{Checker}(T, m_0, m_1) = 1$. Then, by the property of the checker and valid adversaries, it follows that for any machine T for which the valid adversary asked a query, it is difficult to find a second input m_2 such that $T(m_0, m_2) \neq T(m_1, m_2)$. Thus, the existence of an extractor for our distribution would contradict the hypothesis that the adversary is valid and only asked queries for machines and challenges satisfying the collision-resistance compatibility. Precisely, we have the following transformation.

Definition 7 [$e\mathcal{O}$-Based Transformation]
Let $e\mathcal{O}$ be an extractability obfuscator w.r.t. distributional auxiliary input (cf. Remark 3). Let $\mathsf{FS} = (\mathsf{FS.Setup}, \mathsf{FS.KeyGen}, \mathsf{FS.Sign}, \mathsf{FS.Verify})$ be a signature scheme.

For any message m_1 and verification key vk of FS, let us define a machine M_{m_1} (where for simplicity we omit the other parameters in subscript) that takes two inputs, a signature σ_T of machine T and a message m_2, and (1) the machine verifies the signature σ_T according to vk, and if it is an invalid signature, it returns $\perp$; and (2) the machine returns $T(m_1, m_2)$.

We define a new 2-inputs functional encryption scheme
$\mathsf{CRFE}[e\mathcal{O}, \mathsf{FS}] = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for functionality TM_2 as follows[11]

---

[11] For simplicity, henceforth we omit to specify whether the functionality is with respect to machine of fixed time or input-specific. Both cases can be taken in account with small changes.

– Setup(1^λ): chooses a pair (msk, vk) $\leftarrow$ FS.Setup(1^λ) and generates a key sk$_1$ $\leftarrow$ FS.KeyGen(msk, 1) that allows signing all messages (i.e., for the always-accepting predicate $1(T) = T \; \forall \; T$). It sets Ek$_1$ = Ek$_2$ = vk and outputs Mpk = Ek$_1$ and Msk = (sk$_1$, vk).
– Enc(Ek, m): depending on whether Ek is an encryption key for first or second input:
  • if Ek = Ek$_1$ then outputs eØ(M_m) where M_m is defined as above with respect to m and vk (recall that for simiplicity we omit the subscript for vk).
  • if Ek = Ek$_2$ then outputs the message m in clear (recall that we are not interested in the security of the second input and we adopted the formalism of multi-input FE to avoid the need of a new syntax and for sake of generality, e.g., providing in future constructions that satisfy *both* IND-Security and CRIND-Security).
– KeyGen(Msk, T): on input Msk = (sk$_1$, vk) and a machine T, the algorithm generates a signature on T via $\sigma_T \leftarrow$ Signature.Sign(sk$_1$, T) and outputs token σ_T.
– Eval(Mpk, Ct$_1$, Ct$_2$, Tok): on input Mpk = vk, Ct$_1$ which is an obfuscated machine M of machine M_{m_1}, Ct$_2$ = m_2, Tok = σ_T, runs machine M with the other inputs as arguments, and returns the machine output as its own.

Correctness. It is easy to see that the scheme satisfies correctness assuming the correctness of eØ and FS.

We now state the security of the constructed scheme.

Theorem 8 *If* eØ *is an extractable obfuscator w.r.t. distributional auxiliary input,* FS *is an unforgeable functional signature scheme with function privacy, then, for any* Checker *satisfying the requirement of the* CRIND-*Security, it holds that* CRFE[eØ, FS] *is* CRIND-*Secure. Furthermore such scheme satisfies input-specific run time and assuming that* FS *is also succinct, so* CRFE *does.*

Due to space constraints, we defer the formal proof of the theorem to Appendix A.

5   Acknowledgments

References

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

2. Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. `http://eprint.iacr.org/`.

3. Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In *Cryptology and Network Security - 12th International Conference, CANS 2013, Paraty, Brazil, November 20-22. 2013. Proceedings*, pages 218–234, 2013.

4. Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *CRYPTO (2)*, pages 500–518, 2013.

5. Angelo De Caro and Vincenzo Iovino. On the power of rewinding simulators in functional encryption. *IACR Cryptology ePrint Archive*, 2013:752, 2013.

6. Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. Function private functional encryption and property preserving encryption : New definitions and positive results. Cryptology ePrint Archive, Report 2013/744, 2013. `http://eprint.iacr.org/`.

7. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

8. 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 Canetti and Garay [22], pages 519–535.

9. Daniel Apon, Dov Gordon, Jonathan Katz, Feng-Hao Liu, Hong-Sheng Zhou, and Elaine Shi. Personal Communication, July 2013.

10. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 162–179, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.

11. 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.

12. S. Dov Gordon, Jonathan Katz, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. *IACR Cryptology ePrint Archive*, 2013:774, 2013.
13. Shafi Goldwasser, Vipul Goyal, Abhishek Jain, and Amit Sahai. Multi-input functional encryption. Cryptology ePrint Archive, Report 2013/727, 2013. `http://eprint.iacr.org/`.
14. 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.
15. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998. ACM Press.
16. Vincenzo Iovino and Karol Żebrowksi. Simulation-based secure functional encryption in the random oracle model. Cryptology ePrint Archive, Report 2014/810, 2014. `http://eprint.iacr.org/`.
17. Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. Cryptology ePrint Archive, Report 2013/860, 2013. `http://eprint.iacr.org/`.
18. Elette Boyle and Rafael Pass. Limits of extractability assumptions with distributional auxiliary input. Cryptology ePrint Archive, Report 2013/703, 2013. `http://eprint.iacr.org/`.
19. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 308–317. IEEE Computer Society, 1990.
20. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2012.
21. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. *IACR Cryptology ePrint Archive*, 2013:401, 2013.
22. 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*. Springer, 2013.

A Proof of Theorem 8

Proof. We define the following hybrids. Let $q(\lambda)$ be a bound on the number of *post-challenge* token queries asked by $\mathcal{B}$ in any execution with security parameter 1^λ. Such bound exists because $\mathcal{B}$ is a PPT algorithm.

- Hybrid $H_0^{\mathcal{B}}$: This is the real experiment $\mathsf{CRIND}_{\mathcal{B}}^{\mathsf{CRFE}[\mathsf{e}\mathcal{O},\mathsf{FS}]}$.
- Hybrid $H_i^{\mathcal{B}}, i = 0, \ldots, q$: Same as the previous hybrid, except that the first i post-challenge token queries are answered with respect to a *restricted* signing key sk_C for the Boolean predicate C that allows one

to sign exactly Turing machines T for which $\mathsf{Checker}(T, m_0, m_1) = 1$. (This is one of the differences with the proof of Boyle *et al.* wherein, being the scope to prove IND-Security, the signing key is for a predicate that allows one to sign exactly the machines T for which $T(m_0) = T(m_1)$. This is not possible in our case, but we make use of the definition of valid adversary that dictates that such adversary will only make queries for machines approved by the checker.). Specifically, at the beginning of the game the challenger generates a restricted signing key $\mathsf{sk_C} \leftarrow \mathsf{FS.KeyGen}(\mathsf{Msk}, \mathsf{C})$. The pre-challenge queries are answered using the standard signing key $\mathsf{sk_1}$ as in hybrid H_0. The first i post-challenge token queries are answered using the restricted key $\mathsf{sk_C}$, that is a token query for machine T is answered with $\sigma_T \leftarrow \mathsf{FS.Sign}(\mathsf{sk_C}, T)$. All remaining token queries are anwered using the standard key $\mathsf{sk_1}$.

Claim 9 For $i = 1, \ldots, q$, the advantage of $\mathcal{B}$ in guessing the bit b in hybrid $H_i^{\mathcal{B}}$ is equal to the advantage of $\mathcal{B}$ in guessing the bit b in hybrid $H_{i-1}^{\mathcal{B}}$ up to a negligible factor.

We prove the claim by using the function privacy property of FS. Namely, for any $i \in [q]$, consider the following adversary $\mathcal{A}_{\mathsf{priv}}(1^\lambda)$ against function privacy of FS.

- $\mathcal{A}_{\mathsf{priv}}^i$ is given keys $(\mathsf{vk}, \mathsf{msk}) \leftarrow \mathsf{FS.Setup}(1^\lambda)$ from the function privacy challenger.
- $\mathcal{A}_{\mathsf{priv}}^i$ submits the all-accepting function 1 as the first of its two challenge functions, and receives a corresponding signing key $\mathsf{sk_1} \leftarrow \mathsf{FS.KeyGen}(\mathsf{msk}, 1)$.
- $\mathcal{A}_{\mathsf{priv}}^i$ simulates interaction with $\mathcal{B}$. First, it forwards vk to $\mathcal{B}$ as the public-key and chooses a random string $r \in \{0, 1\}^\lambda$. For each token query T made by $\mathcal{B}$, it generates a signature on T using key $\mathsf{sk_1}$.
- $\mathcal{A}_{\mathsf{priv}}^i$ At some point $\mathcal{B}$ outputs a pair of messages m_0, m_1. $\mathcal{A}_{\mathsf{priv}}^i$ generates a challenge ciphertext in the CRIND-Security game by sampling a random bit b and encrypting $(m_b || r)$ and sending it to $\mathcal{B}$.
- $\mathcal{A}_{\mathsf{priv}}^i$ submits as its second challenge function C (as defined above). It receives a corresponding signing key $\mathsf{sk_C} \leftarrow \mathsf{FS.KeyGen}(\mathsf{msk}, \mathsf{P_C})$.
- $\mathcal{A}_{\mathsf{priv}}^i$ now simulates interaction with $\mathcal{B}$ as follows. For the first $i - 1$ post-challenge token queries T made by $\mathcal{B}$, $\mathcal{A}_{\mathsf{priv}}^i$ generates a signature using key $\mathsf{sk_C}$, i.e., $\sigma_T \leftarrow \mathsf{FS.Sign}(\mathsf{sk_C}, T)$. For $\mathcal{B}$'s i-th post-challenge query, $\mathcal{A}_{\mathsf{priv}}^i$ submits the pair of preimages (T, T) to the function privacy challenger (note that $1(T) = C(T) = T$) since, being $\mathcal{B}$ a valid adversary, it only asks queries T such that

$\mathsf{Checker}(T, m_0, m_1) = 1$), and receives a signature σ_T generated either using key $\mathsf{sk_1}$ or key $\mathsf{sk_C}$. $\mathcal{A}^i_{\mathsf{priv}}$ generates the remaining post-challenge queries of $\mathcal{B}$ using key $\mathsf{sk_1}$.

– Eventually $\mathcal{B}$ outputs a bit b'. If $b' = b$ is a correct guess, then $\mathcal{A}^i_{\mathsf{priv}}$ outputs function 1; otherwise, it outputs function C.

Note that if the function privacy challenger selected the function 1, then $\mathcal{A}^i_{\mathsf{priv}}$ perfectly simulates hybrid $H^{\mathcal{B}}_{i-1}$, otherwise it perfectly simulates hybrid $H^{\mathcal{B}}_i$. Thus, the advantage of $\mathcal{A}^i_{\mathsf{priv}}$ is exactly the difference in guessing the bit b in the two hybrids, $H^{\mathcal{B}}_i$ and $H^{\mathcal{B}}_{i-1}$ and the claim follows from the function privacy property.

Next, we define the following distribution $\mathcal{D}$ depending on $\mathcal{B}$.

– $\mathcal{D}(1^\lambda)$ gets $r \overset{R}{\leftarrow} \{0,1\}^\lambda$, samples a key pair $(\mathsf{vk}, \mathsf{msk} \leftarrow \mathsf{FS.Setup}(1^\lambda)$ and generates the signing key for the all-accepting function 1 by $\mathsf{sk_1} \leftarrow \mathsf{FS.KeyGen}(\mathsf{msk}, 1)$.
– Using $\mathsf{sk_1}$ and vk, $\mathcal{D}$ simulates the action of $\mathcal{B}$ in experiment $H^{\mathcal{B}}_q$ up to the point in which $\mathcal{B}$ outputs a pair of challenge messages m_1, m_1. Denote by $\mathsf{view}_{\mathcal{B}}$ the current view of $\mathcal{B}$ up to this point of the simulation.
– $\mathcal{D}$ generates a signing key $\mathsf{sk_C}$ for the function C as defined above and machines $M_{m_0||r}$ and $M_{m_1||r}$ as defined above (recall that as usual we omit to specify the subscript relative to vk).
– $\mathcal{D}$ outputs the tuple $(M_{m_0||r}, M_{m_1||r}, z = (\mathsf{view}_{\mathcal{B}}, \mathsf{sk_C}))$.

We now can construct an adversary $\mathcal{A}(1^\lambda, M', M_0, M_1, z)$ against the security of $\mathsf{e}\mathcal{O}$.

– $\mathcal{A}$ takes as input the security parameter 1^λ, an obfuscation M' of machine M_b for randomly chosen bit b, two machines M_0 and M_1, and auxiliary input $z = (\mathsf{view}_{\mathcal{B}}, \mathsf{sk_C})$.
– Using $\mathsf{view}_{\mathcal{B}}$, $\mathcal{A}$ returns $\mathcal{B}$ to the state of execution as in the corresponding earlier simulation during the $\mathcal{D}$ sampling process.
– Simulate the challenge ciphertex to $\mathcal{B}$ as M'. For each subsequent token query M made by $\mathcal{B}$, $\mathcal{A}$ answers it by producing a signature on M using $\mathsf{sk_C}$.
– Eventually, $\mathcal{B}$ outputs a bit b' for the challenge ciphertext that $\mathcal{A}$ returns as its own guess.

Note that the interaction with the adversary $\mathcal{B}$ in sampling from $\mathcal{D}$ is precisely a simulation in hybrid $H^{\mathcal{B}}_q$ up to the point in which $\mathcal{B}$ outputs the challenge messages, and the interaction with $\mathcal{B}$ made by $\mathcal{A}$ is precisely

a simulation of the remaining steps in hybrid $H_q^{\mathcal{B}}$. We are assuming that the advantage of $\mathcal{B}$ in hybrid H_q^i is $\geq 2a(\lambda)$ for some non-negligible function $a(\lambda)$. This implies that there is a polynomial $p(\lambda)$ such that for an infinite set S of values λ, it holds that the advantage of $\mathcal{B}$ in hybrid H_q^i for parameter λ is greater than $1/2p(\lambda)$. Thus, by an averaging argument, for all $\lambda \in S$, $\mathcal{A}$'s advantage (with respect to λ) in guessing the bit b on which it is challenged upon is greater than $1/p$ with probability greater than $1/p$ over the output of $\mathcal{D}$. By the security of eO this implies a corresponding PPT extractor E and polynomial $q(\lambda)$ and negligible function $\mathsf{negl}(\lambda)$ such for all $\lambda \in S$, with probability $1 - \mathsf{negl}(\lambda)$ over the output (M_0, M_1, z) of $\mathcal{D}$, it holds that:

if $\Pr\left[b \xleftarrow{R} \{0,1\}; M' \leftarrow \mathsf{eO}(1^\lambda, M_b) : \mathcal{A}(1^\lambda, M', M_0, M_1, z) = b \right] \geq \frac{1}{2} + \frac{1}{p(\lambda)}$, then $\Pr\left[w \leftarrow \mathsf{E}(1^\lambda, M_0, M_1, z) : M_0(w) \neq M_1(w) \right] \geq 1/q(\lambda)$. This implies that for an infinite number of values λ, with probability $\geq 1/p(\lambda) - \mathsf{negl}(\lambda)$ over the output (M_0, M_1, z) of $\mathcal{D}$, it holds that

$\Pr\left[w \leftarrow \mathsf{E}(1^\lambda, M_0, M_1, z) : M_0(w) \neq M_1(w) \right] \geq 1/q(\lambda)$.

We now show that such PPT extractor can not exist.

Claim 10 There can not exist a PPT extractor as above.

Suppose toward a contradiction that there exists such extractor that outputs a signature σ_A for some machine A, and a second input m_2 that distinguishes $M_{m_0 || r}$ from $M_{m_1 || r}$. We note that any signature output by the extractor must be a valid signature for a machine A for which the adversary asked a query. This follows from the unforgeability of FS. From this fact, and from the fact that the checker approved the triple (A, m_0, m_1), it follows that m_0 and m_1 are collision-resistant compatible with $\{A\}$. Therefore, this adversary can be used to break the collision-resistance compatibility with respect to m_0 and m_1 and $\{A\}$, contradicting the hypothesis.

It is trivial to see that the claim on input-specific run time holds if the the scheme is used with Turing machines of input-specific run time and that the claim on the succinctness follows easily from our construction and the succinctness of FS. This concludes the proof.