Obfuscation $\Rightarrow$ (IND-CPA Security $\not\Rightarrow$ Circular Security)

Antonio Marcedone[1,⋆] and Claudio Orlandi[2]

[1] Scuola Superiore di Catania, University of Catania, Italy, amarcedone@cs.au.dk
[2] Aarhus University, Denmark, orlandi@cs.au.dk

Abstract *Circular security* is an important notion for public-key encryption schemes and is needed by several cryptographic protocols. In circular security the adversary is given an extra "hint" consisting of a *cycle* of encryption of secret keys i.e., $(E_{pk_1}(sk_2), \ldots, E_{pk_n}(sk_1))$. A natural question is whether every IND-CPA encryption scheme is also circular secure. It is trivial to see that this is not the case when $n = 1$. In 2010 a separation for $n = 2$ was shown by [ABBC10, GH10] under standard assumptions in bilinear groups.
In this paper we finally settle the question showing that for every n there exist an IND-CPA secure scheme which is not n-circular secure.
Our result relies on the recent progress in cryptographic obfuscation.

1 Introduction

Public-key encryption schemes allow anyone to take a plaintext and create a corresponding ciphertext that carries little or no information about the encrypted plaintext, in the eyes of everyone else but the owner of the secret key.

One might think that for an encryption scheme all plaintexts are equal, but it turns out that some plaintexts are more equal than others. In particular, secret-keys (or functions of them) are a very special kind of plaintexts.

But why would you want to encrypt a secret key? A prime example is fully-homomorphic encryption (FHE): At the heart of virtually every fully-homomorphic encryption scheme there is a technique called "bootstrapping" that requires users to publish, in their public key, an encryption of the secret key [Gen09]. For another example think of two cryptographers, Alice and Bob, who get married and decide they should not keep any secret from each other and therefore decide to share their secret keys with each other. To do so Alice sends an encryption of her secret key sk_A to Bob using his public key pk_B, while Bob sends an encryption of his secret key sk_B to Alice using her public key pk_A. This is not a far fetched example and there are applications where this is actually done, see [CL01].

Suppose now that the the evil eavesdropper Eve gets to see these encryptions of secret keys: is the encryption scheme still secure, or can Eve use this extra information to break its security?

Circular Security. In the FHE example, a secret key was encrypted under its own public key and we call this a 1-cycle i.e., Eve learns $E_{pk}(sk)$. When Alice and Bob both encrypt their secret keys under the other party's public key, we get a 2-cycle i.e., Eve learns $E_{pk_A}(sk_B)$ and $E_{pk_B}(sk_A)$. In general, we are interested in what happens when Eve learns the encryptions of n secret keys $(sk_1, \ldots, sk_n)$ under public keys $(pk_2, \ldots, pk_n, pk_1)$ respectively. If an encryption scheme is still secure when the adversary is given such a cycle of encryptions of secret keys, we say that the scheme is n-circular secure[3]. This notion was first

⋆ Work done while visiting Aarhus University.

[3] There are different ways of defining circular security. The interested reader can check [CGH12] and reference therein for a discussion on the definitions. In this paper we will show a scheme where the adversary (given a cycle of encryption of secret keys) can recover *all the secret keys*, thus breaking even the weakest notions of circular security. Therefore the actual definition used is irrelevant for us.

defined in [CL01, BRS02]. Since then it has been an open problem to understand the relationship between the standard definition of security for public key encryption schemes (namely *indistinguishability under chosen-plaintext-attack* or *IND-CPA* for short) and n-circular security.

IND-CPA Security $\not\Rightarrow$ 1-Circular Security. It is quite easy to show that IND-CPA security does not imply 1-circular security. Take any IND-CPA secure scheme (G, E, D) and construct (G, E', D) as follows: on input m, the modified encryption scheme $E'_{pk}(\cdot)$ first checks if $m \stackrel{?}{=} sk$.[4] If so, E' outputs m, else it outputs $E_{pk}(m)$. The modified scheme is still IND-CPA secure (as it behaves exactly like E for all $m \neq sk$), but since $E'_{pk}(sk) = sk$ it is clear that it would be a very bad idea to let Eve learn this value.

Pairing Assumptions $\Rightarrow$ (IND-CPA Security $\not\Rightarrow$ 2-Circular Security). Surprisingly, it was quite harder to show that there are IND-CPA schemes that are not 2-circular secure. The reason for this is that the secret keys are generated independently and therefore the encryption algorithm does not have a way of distinguishing a secret key from a message (in fact, every message could be a secret key). This problem has been open for about a decade until it was finally solved in 2010 by [ABBC10, GH10]. Both these results hold under the assumption that some problems are hard in bilinear groups. The counterexample is obtained by embedding some extra elements in the ciphertexts. These extra values do not help the adversary to break the IND-CPA game but, when combined together using a bilinear map, allow to effectively decrypt one of the two "circular" ciphertexts and recover a secret key.

Obfuscation $\Rightarrow$ (IND-CPA Security $\not\Rightarrow$ Circular Security). In this paper, we show that IND-CPA security does not imply n-circular security *for any* n. More precisely, for every n, we can construct a scheme that is not n'-circular secure for every $n' < n$. We can show our result assuming that software obfuscation is possible, as defined by [BGI$^+$01, BGI$^+$12].

(False $\Rightarrow$ True)? One might now object that our theorem is trivial: the same paper that defined obfuscation also proved that this notion is *impossible to achieve!* However [BGI$^+$01, BGI$^+$12] "only" proved that there exist no single obfuscator that can obfuscate every circuit under the strongest possible notion of obfuscation – namely "virtual black box" (VBB) obfuscation – and during the last decade obfuscators for limited class of circuits have been shown, such as [CD08, Wee05, CRV10, HRSV11].

In a surprising turn of event – and thanks to the recent breakthrough on a candidate for multilinear maps [GGH13a] – the first candidate cryptographic obfuscation was presented in [GGH$^+$13b]. The obfuscation of [GGH$^+$13b] does not contradict the impossibility result of [BGI$^+$01, BGI$^+$12], as it achieves a weaker notion called *indistinguishability obfuscation* (iO). Yet, this arguably very weak notion of obfuscation allows for a long list of unexpected applications [SW13, HSW13, GGHR14, BZ13, PPS13, KNY14], and one could say that the result in [GGH$^+$13b] is "*an impractical obfuscation for all practical purposes*"[5].

Following this result, even a candidate VBB obfuscator has been proposed in [BR13]. This result overcomes the impossibility result of [BGI$^+$01, BGI$^+$12], by proving the security of the scheme in the *generic graded encoding scheme model*: this can be thought as the analogue of the *generic group model* for *discrete logarithm*, extended to the case of multilinear maps.

In Section 2 we show how to separate IND-CPA and circular security using VBB obfuscation, as this powerful tool allows for very simple and intuitive constructions. Then, in Section 3 we show the same result using the weaker (and therefore more realistic) assumption that an iO obfuscator exists.

[4] Note that it is always possible to check if $m = sk$ by, for example, encrypting a bunch of random messages using $E_{pk}(\cdot)$ and decrypting them using m i.e., the encryption algorithm checks if $D_m(E_{pk}(r)) = r$ for enough random values r. If the results are all correct, one can assume whp that $m = sk$.

[5] Cit. Yuval Ishai.

Relation to [KRW13]. In October 2013 Koppula, Ramchen and Waters [KRW13] posted on ePrint a similar result with a proof of security under indistinguishability obfuscation. On the same day, we posted a draft of our result which only showed a counterexample under the assumption of VBB obfuscation (Section 2). Subsequently, in February 2014 we updated our draft with Section 3, which is a simple application of the punctured programming technique from [SW13] to our construction of Section 2. Thus that addition achieves a counterexample based only on indistinguishability obfuscation. While recognizing that [KRW13] were first in showing the separation using iO only, we believe that our counterexample has some advantages.

Circular Security of Bit-Encryption. In the previous discussion on circular security, we made the implicit conjecture that the secret keys are element of the plain-text space (or how could it be possible to encrypt them?). It has been conjectured that every IND-CPA bit-encryption scheme (that is, an encryption scheme that can only encrypt messages in $\{0,1\}$) is also circular secure. Rothblum [Rot13] shows an IND-CPA bit encryption scheme which is not 1-circular secure assuming the existence of multilinear maps in which the SXDH assumption holds. Koppula, Ramchen and Waters [KRW13] give a different separation, based on the existence of iO obfuscation.

The good news. While our work provides strong evidence for the fact that not all IND-CPA secure public key encryption schemes achieve circular security, there are a number of encryption schemes that can be proven secure even under these attacks. We refer the interested reader to [BHHO08, CGH12, Hof13, BGK11, BG10] and references therein.

1.1 Technical Overview

The simplest way of constructing a public-key encryption scheme in a world where obfuscation exists is probably the following: a secret key is just a random string s and a public key is a circuit P that outputs 1 on input s and $\perp$ otherwise. We write $s \xrightarrow{P} 1$ for compactness. We can think of a plaintext m as a circuit $1 \to m$. Now to encrypt m under public key P one can construct a ciphertext C with the functionality $s \xrightarrow{C} m$ by composing the two circuits $s \xrightarrow{P} 1 \to m$. Correctness is trivial to check and security follows from the fact that the circuits are obfuscated and can therefore only be used as "black-boxes".

To break the circular security of the scheme, we add another circuit to the ciphertexts that "recognizes" circular encryptions without otherwise affecting the security of our scheme. Using the public key P we define a new circuit Q which takes as input a string y and a program B: Q evaluates $B(y)$ and checks if the result is equal to s using P and, if so, outputs s. In other words, Q only outputs s to someone that already knows it.

When creating a ciphertext we append a circuit R to the ciphertext, where R is an obfuscation of Q with the first input fixed to m. Following the definition of Q, the circuit R provides the following functionality: On input a ciphertext C, the circuit R tries to decrypt C with m and, if the output is s, releases the secret key s. So, if Q_1 is the circuit made from public key P_1 and then its first input is fixed to s_2, the ciphertext will now contain a circuit $R_{1,2}$ that can recognize encryptions of s_1 under the key s_2. So, our new scheme is not 2-circular secure!

The next observation is that any two circuits $x \xrightarrow{A} y$ and $y \xrightarrow{B} z$ can be composed into a circuit $x \xrightarrow{C} z$. In particular, from a set of n encryptions $s_i \xrightarrow{C_i} s_{(i+1 \mod n)}$ for $i = 1, \ldots, n$ one can compute n circuits

$$s_{(i+1 \mod n)} \xrightarrow{C_i^*} s_i$$

Clearly the size of these circuits grows with n, but this is not a problem as long as we set the input size of Q to be big enough.

This concludes the intuitive description of our "attack". To see that the scheme is IND-CPA secure, consider an intermediate game where we replace the real public key P with a circuit that always outputs $\perp$. If this can be done in an indistinguishable way, then we are done: if P always outputs $\perp$, then also C, Q

output $\perp$ on any input, and therefore contain no information whatsoever about m (at least in the ideal world where the simulator only has oracle access to the circuits).

Now it is easy to see that if VBB obfuscators exist, we can replace P with a circuit that always output $\perp$ and the adversary will only distinguish if he queries the oracles on the secret key s. However, iO obfuscation does not allow to perform this replacement, as it only guarantees that obfuscations of circuits computing the same functions are indistinguishable. To fix that, we replace the public key P with a a string $p = \text{PRG}(s)$: this still allows the other circuits C, Q to check if their input is equal to the secret key by computing $p \stackrel{?}{=} \text{PRG}(x)$, and at the same time it allows us to replace p with an indistinguishable uniform random string (for which no secret key exists) in the hybrid game. Then, when p is a uniform random string, C and Q always output $\perp$ and we can therefore use iO obfuscation to argue that encryptions of m_0 are indistinguishable from encryptions of m_1.

1.2 Preliminaries

In this section, we state the notation and conventions used in the rest of the work. To keep the paper self contained, we will also recall some relevant definitions and theorems.

Notation and Conventions: We use lowercase letters s, x, y for strings in $\{0,1\}^n$. We use uppercase letters P, C, Q, R for "plaintext" circuits and $\overline{P}, \overline{C}, \overline{Q}, \overline{R}$ for obfuscated circuits. We call $\mathcal{P}$ the set of all polynomial-size circuits. We use the notation $P(x \in X) \in Y$ when we want to say that a circuit P takes input from $X \cup \{\perp\}$ and returns a value in $Y \cup \{\perp\}$. We write $P_{a \to b}$ for a circuit P that outputs b if $x = a$ and $\perp$ otherwise, and $P_{* \to \perp}$ for the circuit that outputs $\perp$ on any input. For all circuits we define $P(\perp) = \perp$.

If S is a set $s \leftarrow S$ is a uniformly random sample from S. If A is a randomized algorithm, $x \leftarrow A$ is the output of A on a uniformly random input tape.

Definition 1 (Pseudorandom generator). *We say that a function* $\text{PRG} : \{0,1\}^k \to \{0,1\}^y$ *(with* $y > k$*) is a secure pseudorandom generator if no PPT adversary* $\mathcal{A}$ *can distinguish between a random string* $c \leftarrow \{0,1\}^y$ *and the output of the* $\text{PRG}(s)$ *on a uniformly random point* $s \in \{0,1\}^k$*.*

Definition 2 (IND-CPA). *Let* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public key encryption scheme. Let us define the following experiment (parametrized by a bit b) between an adversary* $\mathcal{A}$ *and a challenger:*

$\mathbf{IND\text{-}CPA}^b_{\Pi}(\mathcal{A}, k)$:

1. *The challenger runs* $(sk, pk) \leftarrow \mathsf{Gen}(1^k)$ *and gives pk to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *outputs two messages* (m_0, m_1) *of the same length.*
3. *The challenger computes* $\mathsf{Enc}(pk, m_b)$ *and gives it to* $\mathcal{A}$
4. $\mathcal{A}$ *outputs a bit* b' *(if it aborts without giving any output, we just set* $b' \leftarrow 0$*). The challenger returns* b' *as the output of the game.*

We say that Π *is* secure against a chosen plaintext attack *if for any k and any PPT adversary* $\mathcal{A}$

$$\mathbf{Adv}(\mathcal{A}) \stackrel{\text{def}}{=} \left| \Pr\left[\mathbf{IND\text{-}CPA}^1_{\Pi}(\mathcal{A}, k) = 1\right] - \Pr\left[\mathbf{IND\text{-}CPA}^0_{\Pi}(\mathcal{A}, k) = 1\right] \right| \leq \text{negl}(k).$$

Definition 3 (Virtual Black-Box Obfuscator [BGI+01, BGI+12]). *Let* $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ *be a family of polynomial-size circuits, where* $\mathcal{C}_n$ *is a set of boolean circuits operating on inputs of length n. And let* $\mathcal{O}$ *be a PPT algorithm, which takes as input an input a length* $n \in \mathbb{N}$*, a circuit* $C \in \mathcal{C}_n$*, a security parameter* $k \in \mathbb{N}$*, and outputs a boolean circuit* $\mathcal{O}(C)$ *(not necessarily in* $\mathcal{C}$*).*

$\mathcal{O}$ *is a (black-box) obfuscator for the circuit family* $\mathcal{C}$ *if it satisfies:*

Preserving Functionality: *For every* $n \in \mathbb{N}$*, every* $C \in \mathcal{C}$ *and every* $\boldsymbol{x} \in \{0,1\}^n$*, with all but* $\text{negl}(k)$ *probability over the coins of* $\mathcal{O}$*:*

$$\left(\mathcal{O}(C, 1^n, 1^k)\right)(\boldsymbol{x}) = C(\boldsymbol{x})$$

Polynomial Slowdown: *For every $n, k \in \mathbb{N}$ and $C \in \mathcal{C}$, the circuit $\mathcal{O}(C, 1^n, 1^k)$ is of size at most* $\text{poly}(|C|, n, k)$.

Virtual Black-Box: *For every (non-uniform) polynomial size adversary $\mathcal{A}$, there exists a (non-uniform) polynomial size simulator $\mathcal{S}$, such that for every $n \in \mathbb{N}$ and for every $C \in \mathcal{C}$:*

$$\left| \Pr_{\mathcal{O},\mathcal{A}} \left[\mathcal{A}\left(\mathcal{O}(C, 1^n, 1^k)\right) = 1 \right] - \Pr_{\mathcal{S}} \left[\mathcal{S}^C(1^{|C|}, 1^n, 1^k) = 1 \right] \right| \leq \text{negl}(k)$$

Definition 4 (Indistinguishability Obfuscation [GGH$^+$13b]). *Given a circuit class $\{\mathcal{C}_k\}$, a (uniform) PPT machine $\mathcal{O}$ is called an indistinguishability obfuscator (iO) for $\{\mathcal{C}_k\}$ if it satisfies:*

Preserving Functionality: *For every $k \in \mathbb{N}$ and $C \in \mathcal{C}_k$,*

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

Indistinguishability: *For any (non necessarily uniform) distinguisher $\mathcal{D}$, all security parameters k and all couples $C_0, C_1 \in \mathcal{C}_k$ such that $C_0(x) = C_1(x)$ for all inputs x, we have that*

$$\left| \Pr[D(\mathcal{O}(k, C_0) = 1] - \Pr[D(\mathcal{O}(k, C_1)) = 1] \right| \leq \text{negl}(k)$$

Recently candidate obfuscators for every circuits have been presented: [BR13] shows that VBB obfuscation is possible under appropriate assumptions in a "generic group model" while [GGH$^+$13b] shows that iO obfuscation is possible under strong (but falsifiable) assumptions.

2 Separating IND-CPA Security from Circular Security

2.1 PKE from Obfuscation

We start by constructing a very simple IND-CPA public-key encryption scheme $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}$ based on obfuscation, and show some of its interesting property. In the next subsection, we will modify it in order to render it insecure under n-circular security attacks.

Key Generation: The algorithm $\mathsf{Gen}(1^k)$ chooses a random secret key $s \leftarrow \{0,1\}^k$. The public key is an obfuscated circuit $\overline{P} \leftarrow \mathcal{O}(P)$ where P is defined as follows:

def $P(x \in \{0,1\}^k) \in \{0,1\}$:

1. if $(x \stackrel{?}{=} s)$ output 1; else output $\perp$.

Encryption: The algorithm $\mathsf{Enc}(\overline{P}, m)$ on input a public key $\overline{P} \in \mathcal{P}$ and a message $m \in \{0,1\}^k$ outputs an obfuscated circuit $\overline{C} \leftarrow \mathcal{O}(C)$ where C is defined as follows:

def $C(x \in \{0,1\}^k) \in \{0,1\}^k$:

1. if $(\overline{P}(x) \stackrel{?}{=} 1)$ output m; else output $\perp$.

Decryption: The algorithm $\mathsf{Dec}(s, \overline{C})$ on input a secret key $s \in \{0,1\}^k$ and a ciphertext $\overline{C} \in \mathcal{P}$ outputs $m' = \overline{C}(s)$.

It is easy to check that if $(s, \overline{P}) \leftarrow \mathsf{Gen}$, then:

$$\mathsf{Dec}(s, \mathsf{Enc}(\overline{P}, m)) = m$$

Theorem 1. *If $\mathcal{O}$ is a VBB obfuscator for $\mathcal{P}$ according to Definition 3, then the scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *described above is IND-CPA secure according to Definition 2.*

To see that the scheme is IND-CPA secure, notice that thanks to the VBB property one can replace the public key $\overline{P}$ with an obfuscated version of $P_{*\to\perp}$ without the adversary noticing. Then, for every m, $\mathsf{Enc}(pk, m) = P_{*\to\perp}$, so in the ideal world (where the simulator only has oracle access to them) the ciphertexts contain no information at all about the messages. A formal argument follows.

Proof. We prove the the theorem by an hybrid argument. Let us define the following games:

Game 0: this is the same as $\mathbf{IND\text{-}CPA}^0(\mathcal{A}, k)$.
Game 1: this is the same as the previous one, but in step 1 we set the public key pk to be an obfuscation (of proper size) of $P_{*\to\perp}$.
Game 2: this is the same as the previous one, but in step 3 instead of an encryption of m_1 we give $\mathcal{A}$ an obfuscation of $P_{*\to\perp}$.
Game 3: this is the same as Game 4, but in step 1 we set the public key $pk \leftarrow \mathcal{O}(P_{*\to\perp})$.
Game 4: this is the same as $\mathbf{IND\text{-}CPA}^1(\mathcal{A}, k)$.

Proving that no adversary can distinguish between two consecutive games with more than negligible probability implies the security of our scheme.
We first prove that Game 0 and Game 1 (and similarly Games 4 and 3) are indistinguishable assuming the VBB property of $\mathcal{O}$. Assume by contradiction that there exists an adversary $\mathcal{A}$ such that $|\text{Game0}(\mathcal{A}, k) - \text{Game1}(\mathcal{A}, k)|$ is greater than any negligible function of k. Then we can build an adversary $\mathcal{A}'$ against the VBB property of $\mathcal{O}$ for the class of circuits $\mathcal{P}_k = \{P_{s\to 1} | s \in \{0,1\}^k\} \cup \{P_{*\to\perp}\}$ as follows. $\mathcal{A}'$ gets in input a circuit $pk \leftarrow \mathcal{P}_k$, and runs $\mathcal{A}$ simulating the **IND-CPA** game against it. Its goal is to distinguish whether $pk = P_{*\to\perp}$ (and output 1) or not (and output 0); it works as follows:

$\mathcal{A}'(pk, k)$:
1. Runs $\mathcal{A}$ giving it pk as the public key.
2. $\mathcal{A}$ outputs two messages (m_0, m_1) of the same length.
3. $\mathcal{A}'$ computes $\mathsf{Enc}(pk, m_0)$ and gives it to $\mathcal{A}$
4. When $\mathcal{A}$ outputs a bit b', $\mathcal{A}'$ outputs 1 if $b' = 0$ and 0 otherwise.

It is easy to see that, from $\mathcal{A}$'s point of view, this game is exactly like Game 1 when $pk = P_{*\to\perp}$, and exactly like Game 0 in the other case. Therefore (by contradiction) $\mathcal{A}'$ can distinguish between $P_{*\to\perp}$ and any other circuit in $\mathcal{P}_k$ with more than negligible advantage. However, no simulator (in the ideal world) can do this given only oracle access to pk, as this would imply querying the oracle for pk on input the only *random* point x such that $pk(x) \neq 1$, which can only happen with probability $\frac{1}{2^k}$.
As a final step, we prove that no adversary can distinguish between Games 1 and 2 (2 and 3 respectively) with more than negligible probability. The distribution of the view of $\mathcal{A}$ is identical in both games up to step 3, where it receives a direct obfuscation of $P_{*\to\perp}$ in Game 2, and an encryption $\mathsf{Enc}(pk, m_0)$ in Game 1. However, since we are using an obfuscation of $P_{*\to\perp}$ as a public key in both games, the ciphertexts given to the adversary are both functionally equivalent[6] to (obfuscations of) $P_{*\to\perp}$. Therefore, by the security property of the obfuscator (as in the ideal world we are giving the same oracle to the simulator in both cases), $\mathcal{A}$ cannot distinguish between the two distributions and therefore between the two games. □

2.2 Properties of Our Scheme

The scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defined in the previous section has an interesting property, namely that it is possible to combine ciphertexts together in order to achieve some flavour of *proxy re-encryption*, namely it is possible to delegate to someone the power to transform ciphertexts encrypted under a public key $\overline{P_1}$ into ciphertexts encrypted under a different public key $\overline{P_2}$ without having to release the corresponding secret key s_1. To see how this is possible, think of a proxy who is given two public keys $(\overline{P_1}, \overline{P_2})$ and

$$\overline{C_{1\to 2}} = \mathsf{Enc}(\overline{P_2}, s_1)$$

[6] This means that they have the same input/output behaviour on all inputs. We also note that by this property this part of the proof also works if we assume indistinguishability obfuscation instead of the VBB one.

(i.e., an encryption of secret key 2 using public key 1). It will be convenient now to say that a circuit $\overline{C}$ (not necessarily an output of Enc) is an encryption of m under key i if $\mathsf{Dec}(s_i, \overline{C}) = m$.

Then the proxy, using $\overline{C_1}$ s.t. $\mathsf{Dec}(s_1, \overline{C_1}) = m$ and $\overline{C_{1\to 2}}$ s.t. $\mathsf{Dec}(s_2, \overline{C_{1\to 2}}) = s_1$, can compute an encryption of m under key $\overline{P_2}$ by creating an obfuscated circuit $\overline{C_2} \leftarrow \mathcal{O}(C_2)$ where C_2 is defined as follows:

def $C_2(x \in \{0,1\}^k) \in \{0,1\}^k$
1. Output $\overline{C_1}(\overline{C_{1\to 2}}(x))$;

It is now easy to check that $\overline{C_2}(s_2) = m$ and that, due to the property of the VBB obfuscator $\mathcal{O}$, nothing else can be computed from $\overline{C_2}$.

2-cycle from n-cycle: Using this property, we can go from a cycle of n encryptions to $n-1$ cycles of length 2. Namely, let $\overline{C}_{i\to(i+1)} = \mathsf{Enc}(\overline{P_i}, s_{i+1})$ for all $i \in \{1, \ldots, n\}$ (where all additions are modulo n). Then one can create circuits

$$C^*_{(i+1)\to i} = \overline{C}_{(i+1)\to(i+2)} \circ \ldots \circ \overline{C}_{(i-1)\to i}$$

Note that in this case we are not even interested in re-obfuscating the concatenation of the circuits (like in the proxy re-encryption application) and the circuit $C^*_{(i+1)\to i}$ is a "functional ciphertext" in the sense that it is a circuit which decrypts to s_i on input s_{i+1}. The only difference between C^* and "regular" ciphertext is that the size of C^* grows with n. Given an obfuscator $\mathcal{O}$, it is possible to find an upper bound $\beta_n = \text{poly}(k, n)$ s.t. the size of $C^*_{(i+1)\to i}$ is less than β_n.

2.3 A PKE that is not n-Circular Secure

In this section, we add a new element to the ciphertexts to make the scheme from the previous section insecure under circular attacks. Let $\mathcal{B}$ be the set of circuits of size at most β_n defined above, then we define the following circuit:

def $Q(y \in \{0,1\}^k, B \in \mathcal{B}) \in \{0,1\}^k$:
1. If $(P(B(y)) = 1)$, output $B(y)$; else output $\perp$.

Key Generation: The keys s and $\overline{P}$ are defined as above.
Encryption: An encryption of m now is a pair $(\overline{C}, \overline{R})$ where $\overline{C}$ is defined as above and $\overline{R}(\cdot) \leftarrow \mathcal{O}(Q(m, \cdot))$.
Decryption: Unchanged.

Circular (in)Security of Our Scheme: In our new scheme an encryption contains a circuit R which "remembers" the message m and then, on any circuit B, it tests whether $B(m)$ is equal to the secret key and, if so, it outputs it.

It is easy to see that this new scheme is not 1-circular secure, as $\overline{R}(\overline{C}) = Q(s, \mathsf{Enc}(\overline{P}, s)) = s$. The scheme is also insecure under 2-circular attacks: let s_1, s_2 be two secret keys and $\overline{P}_1, \overline{P}_2$ their respective public keys. The output of $\mathsf{Enc}(\overline{P}_1, s_2)$ is $(\overline{C}_1, \overline{R}_{1,2})$. That is, $\overline{R}_{1,2}$ is a circuit that accepts as input any circuit C of size at most β_n, and if $C(s_2) = s_1$ it outputs s_1.

Therefore if the adversary is also given an encryption $(\overline{C}_2, \overline{R}_{2,1}) \leftarrow \mathsf{Enc}(\overline{P}_2, s_1)$, he can invoke $R_{1,2}(\overline{C}_1)$ and $R_{2,1}(\overline{C}_2)$ to recover s_1, s_2 respectively. As described in the previous section, from any longer cycle of size up to n one can compute a functionally working encryption of s_1 under key 2 i.e., a circuit that on input s_2 outputs s_1, that can be fed as well to R to recover the secret key. Therefore the attack generalizes to n-circularity (as long as the concatenation of n ciphertexts has length less than β_n).

IND-CPA Security of Our Scheme: The modified scheme is still IND-CPA secure: Unless one knows an encryption of the secret key, R cannot be exploited. More formally, we prove the following:

Theorem 2. *If $\mathcal{O}$ is a VBB obfuscator for $\mathcal{P}$ according to Definition 3, then the modified scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *described in this section is IND-CPA secure.*

Proof. The proof is very similar to the one of the corresponding Theorem 1, the main difference being that in this case we need to "substitute" the two parts of the challenge ciphertext separately (and thus we need two more hybrid games):

Game 0: this is the same as $\mathbf{IND\text{-}CPA}^0(\mathcal{A}, k)$.
Game 1: this is the same as the previous one, but in step 1 we set the public key $pk \leftarrow \mathcal{O}(P_{*\to\perp})$.
Game 1.5: this is the same as the previous one, but in step 3 instead of giving $\mathcal{A}$ a complete encryption $(\overline{C}, \overline{R})$ of m_0 we substitute $\overline{R}$ with an obfuscation (of proper size) of $P_{*\to\perp}$ and give him $(\overline{C}, \mathcal{O}(P_{*\to\perp}))$.
Game 2: this is the same as the previous one, but in step 3 instead of an encryption $(\overline{C}, \overline{R})$ of m_0 we give $\mathcal{A}$ two obfuscations $(\mathcal{O}(P_{*\to\perp}), \mathcal{O}(P_{*\to\perp}))$ (of proper size).
Game 2.5: this is the same as Game 3, but in step 3 instead of giving $\mathcal{A}$ a complete encryption $(\overline{C}, \overline{R})$ of m_1, we substitute $\overline{R}$ with an obfuscation (of proper size) of $P_{*\to\perp}$ and give him $(\overline{C}, \mathcal{O}(P_{*\to\perp}))$.
Game 3: this is the same as Game 4, but in step 1 we set the public key pk to be an obfuscation (of proper size) of $P_{*\to\perp}$.
Game 4: this is the same as $\mathbf{IND\text{-}CPA}^1(\mathcal{A}, k)$.

An adversary that distinguishes Game 1 and Game 1.5 (resp. Game 3 and game 2.5) can be used to break the indistinguishability between two different obfuscations of $P_{*\to\perp}$: in Game 1, the circuit R always outputs $\perp$ as $pk = P_{*\to\perp}$, while in Game 1.5 $P_{*\to\perp}$ is obfuscated directly. Indistinguishability between the other games follows from the same arguments as Theorem 1. □

3 Separation from Indistinguishability Obfuscation

To prove that our simple encryption scheme is IND-CPA secure, we had to argue that an obfuscation of a real public key $P_{s\to 1}$ is indistinguishable from an obfuscation of $P_{*\to\perp}$. However indistinguishability obfuscation only guarantees that an adversary cannot tell the difference between the obfuscation of two circuits computing the same function.

To fix this, change our simple scheme in the following way: let $\mathrm{PRG} : \{0,1\}^k \to \{0,1\}^{2k}$ be a pseudo-random generator, then we compute the public key pk as $pk = \mathrm{PRG}(s)$. Note that in the simple public key encryption scheme we only used the obfuscated program P to check if we had the right secret key. This can be done using the new public key as well, by evaluating the PRG. At the same time, this will allow us to replace a real public key with an (indistinguishable) uniformly random string for which (with very high probability) no secret key exists, and therefore all ciphertexts will be functionally equivalent to $P_{*\to\perp}$.

3.1 The Technical Details

Key Generation: The algorithm $\mathsf{Gen}(1^k)$ chooses a random secret key $s \leftarrow \{0,1\}^k$ and computes a string $p = \mathrm{PRG}(s) \in \{0,1\}^{2k}$.

Encryption: The algorithm $\mathsf{Enc}(p, m)$ on input a public key $p \in \{0,1\}^{2k}$ and a message $m \in \{0,1\}^k$ outputs an obfuscated circuit $\overline{C} \leftarrow \mathcal{O}(C)$ and $\overline{R}(\cdot) \leftarrow \mathcal{O}(Q(m, \cdot))$. where C, R are defined as follows:
def $C(x \in \{0,1\}^k) \in \{0,1\}^k$:
1. if $(\mathrm{PRG}(x) \stackrel{?}{=} p)$ output m; else output $\perp$.

def $Q(y \in \{0,1\}^k, B \in \mathcal{B}) \in \{0,1\}^k$:

1. If $(\mathrm{PRG}(B(y)) = p)$, output $B(y)$; else output $\perp$.

Decryption: The algorithm $\mathsf{Dec}(s, \overline{C}, \overline{R})$ on input a secret key $s \in \{0,1\}^k$ and a ciphertext $(\overline{C}, \overline{R}) \in \mathcal{P}$ outputs $m' = \overline{C}(s)$ (and ignores $\overline{R}$).

It can be easily verified that this scheme is also not circular secure, and we can argue that it is still IND-CPA secure.

Theorem 3. *If $\mathcal{O}$ be a iO obfuscator and* PRG *a secure pseudorandom generator, then* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *described in this section is IND-CPA secure.*

Proof. This proof is very similar to the one of Theorem 2. The only difference is that we use a uniformly random string (instead of an obfuscation of $P_{*\to\perp}$) as a fake public key. The hybrids are as follows:

Game 0: this is the same as $\mathbf{IND\text{-}CPA}^0(\mathcal{A}, k)$.
Game 1: this is the same as the previous one, but in step 1 we set the public key $pk \leftarrow \{0,1\}^{2k}$.
Game 1.5: this is the same as the previous one, but in step 3 instead of giving $\mathcal{A}$ a complete encryption $(\overline{C}, \overline{R})$ of m_0 we substitute $\overline{R}$ with an obfuscation (of proper size) of $P_{*\to\perp}$ and give him $(\overline{C}, \mathcal{O}(P_{*\to\perp}))$.
Game 2: this is the same as the previous one, but in step 3 instead of an encryption $(\overline{C}, \overline{R})$ of m_0 we give $\mathcal{A}$ two obfuscations $(\mathcal{O}(P_{*\to\perp}), \mathcal{O}(P_{*\to\perp}))$ (of proper size).
Game 2.5: this is the same as Game 3, but in step 3 instead of giving $\mathcal{A}$ a complete encryption $(\overline{C}, \overline{R})$ of m_1, we substitute $\overline{R}$ with an obfuscation (of proper size) of $P_{*\to\perp}$ and give him $(\overline{C}, \mathcal{O}(P_{*\to\perp}))$.
Game 3: this is the same as Game 4, but in step 1 we set the public key $pk \leftarrow \{0,1\}^{2k}$.
Game 4: this is the same as $\mathbf{IND\text{-}CPA}^1(\mathcal{A}, k)$.

An adversary $\mathcal{A}$ distinguishing between Games 0 and 1 could be used to build an adversary $\mathcal{A}'$ against the security of the PRG as follows:

$\mathcal{A}'(pk \in \{0,1\}^{2k}, 1^k)$:
1. Runs $\mathcal{A}$ giving it pk as the public key.
2. $\mathcal{A}$ outputs two messages (m_0, m_1) of the same length.
3. $\mathcal{A}'$ computes $\mathsf{Enc}(pk, m_0)$ and gives it to $\mathcal{A}$
4. When $\mathcal{A}$ outputs a bit b', $\mathcal{A}'$ outputs 1 if $b' = 0$ and 0 otherwise.

From $\mathcal{A}$'s point of view, this game is exactly the same as Game 0 in the case where pk is computed as $pk \leftarrow \mathrm{PRG}(s)$ from a uniformly random seed $s \leftarrow \{0,1\}^k$, while it is exactly like Game 1 if pk is uniformly random.
Note that, in the case where $pk \leftarrow \{0,1\}^{2k}$ is uniformly random, with all but negligible probability there exists no s such that $\mathrm{PRG}(s) = pk$, and therefore the circuits $\overline{C}, \overline{R}$ contained in the ciphertexts of all other hybrids are always functionally equivalent to $P_{*\to\perp}$. Therefore those hybrids are indistinguishable under the assumption that $\mathcal{O}$ is an indistinguishability obfuscator. □

Acknowledgements: The authors would like to thank Amit Sahai for the mantra "When you cannot solve a problem, try obfuscation" and Matthew Green for helpful comments.

References

[ABBC10] Tolga Acar, Mira Belenkiy, Mihir Bellare, and David Cash. Cryptographic agility and its relation to circular encryption. In *EUROCRYPT*, pages 403–422, 2010.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, pages 1–20, 2010.

[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 *CRYPTO*, pages 1–18, 2001.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BGK11] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *TCC*, pages 201–218, 2011.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *CRYPTO*, pages 108–125, 2008.

[BR13] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. Cryptology ePrint Archive, Report 2013/563, 2013. `http://eprint.iacr.org/`.

[BRS02] John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography*, pages 62–75, 2002.

[BZ13] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2013/642, 2013. `http://eprint.iacr.org/`.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In *EUROCRYPT*, pages 489–508, 2008.

[CGH12] David Cash, Matthew Green, and Susan Hohenberger. New definitions and separations for circular security. In *Public Key Cryptography*, pages 540–557, 2012.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT*, pages 93–118, 2001.

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In *TCC*, pages 72–89, 2010.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. Cryptology ePrint Archive, Report 2013/451, 2013. `http://eprint.iacr.org/`.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure mpc from indistinguishability obfuscation. In *TCC*, pages 74–94, 2014.

[GH10] Matthew Green and Susan Hohenberger. Cpa and cca-secure encryption systems that are not 2-circular secure. *IACR Cryptology ePrint Archive*, 2010:144, 2010.

[Hof13] Dennis Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *EUROCRYPT*, pages 520–536, 2013.

[HRSV11] Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. *J. Cryptology*, 24(4):694–719, 2011.

[HSW13] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2013/509, 2013. `http://eprint.iacr.org/`.

[KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for np from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2014/213, 2014. `http://eprint.iacr.org/`.

[KRW13] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. Cryptology ePrint Archive, Report 2013/683, 2013. `http://eprint.iacr.org/`.

[PPS13] Omkant Pandey, Manoj Prabhakaran, and Amit Sahai. Obfuscation-based non-black-box simulation and four message concurrent zero knowledge for np. Cryptology ePrint Archive, Report 2013/754, 2013. `http://eprint.iacr.org/`.

[Rot13] Ron Rothblum. On the circular security of bit-encryption. In *TCC*, pages 579–598, 2013.

[SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *IACR Cryptology ePrint Archive*, 2013:454, 2013.

[Wee05] Hoeteck Wee. On obfuscating point functions. In *STOC*, pages 523–532, 2005.