Introduction

Encryption is one of the firsts and most fundamental primitives of modern cryptography. An encryption scheme allows anyone to take a message and create a corresponding "ciphertext" (using an "encryption key") that carries little or no information about the message in the eyes of everyone but the owner of a designated secret "decryption key".

We distinguish two important variations: in "Secret Key" (or "Symmetric") Encryption (SKE from now on), the encryption and decryption keys are equal and have to be kept secret, while in "Public Key" Encryption (PKE) the encryption key can be freely distributed and advertised to everyone (and it is therefore called a "public key"), while the decryption key ("secret key") must be kept confidential. One can think of a PKE scheme as the digital analogous of putting a message (usually referred to as the "plaintext") into a box with a ring that can be closed by a lock: Alice can now buy a single key (secret decryption key) and a bunch of locks (public encryption key) and send the open locks to all the people she knows (or leave them available at the post office). When Bob wants to send her a message, he can just take one of Alice's locks and use it to lock the box. Now Alice is the only one who can open the box (decrypt the message), and the extra open locks cannot be used to open it.

The standard definition of security for PKE schemes, namely IND-CPA security, is phrased as an interactive game between a malicious adversary (usually called Eve) and a challenger (Charlie): very informally, Charlie gives Eve a bunch of open locks (a public key), then Eve gives Charlie two pieces of paper with two messages on them. Charlie puts one of them (without Eve watching) into a box, seals it with one of the locks (encrypts it) and gives the locked box back to Eve, who has to try and say which of the two messages is in the box. The encryption scheme is secure if she cannot succeed with probability greater than random guessing.

While this definition (and the strengthened IND-CCA version) provides sufficient guarantees in a large variety of use cases, there are particular situations where it might not be sufficient for security.
Consider, for example, the case where Alice wants to encrypt all the contents of her computer, and she also keeps a copy of the encryption key as a file on the computer. So the message that she is encrypting (i.e. the contents of the whole disk) will contain the very same decryption key that can be used to recover the data (i.e. we are putting the key itself inside the locked box).
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 [10].

In both examples, it is not hard to notice that IND-CPA secure schemes do not give any guarantee, as the interactive game described above does not say anything about what happens when we put a key inside a box: for example, the shape of the key might not even fit into

the box (after all, our boxes are "designed" for messages, not keys). This would be very problematic, as trying to fit a key in a box of the wrong shape might break the box, which would reveal all the secrets it contains.

Circular Security. In the first 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. Roughly speaking, 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. A natural question is whether this seemingly stronger notion is indeed harder to achieve, i.e. whether every IND-CPA encryption scheme is also circular secure. It has long been known that this is not the case when $n = 1$. In 2010 a separation for $n = 2$ was shown by [1, 16] under standard assumptions in bilinear groups.

Original Contributions. In this work we finally settle the question showing that for every n there exists an IND-CPA secure scheme which is not n-circular secure. Our results rely on cryptographic program obfuscation.

More specifically, Chapter 1 provides a more formal overview of the basic definitions of security for PKE schemes, as well as a description of the tools that are used to achieve the results of the later chapters. It also includes a brief technical overview of how these results are obtained and mentions some important related results.

Chapter 2 provides the first of our results. Namely, for any $n \in$

$\mathbb{N}$, we present an encryption scheme which is IND-CPA secure but not $n-$circular secure. This construction assumes the existence of a Virtual Black Box (VBB) Obfuscator.

Chapter 3 leverages the punctured programs technique from Sahai and Waters [25] to modify the construction of Chapter 2 and achieve the same result under the weaker notion of Indistinguishability Obfuscation.

This dissertation is based on work conducted in 2013 and 2014, mostly while visiting the cryptography research group at Aarhus University. The results presented here were obtained in collaboration with Professor Claudio Orlandi, and published in [23]. Concurrently and independently, similar results were obtained by Koppula, Ramchen and Waters [21]. In particular, they were the firsts to obtain the separation from Indistinguishability Obfuscation. See both the papers for a more detailed comparison [23, 21].