

# On Necessary Padding with IO

Justin Holmgren\*

## Abstract

We show that the common proof technique of padding a circuit before IO obfuscation is sometimes necessary. That is, assuming indistinguishability obfuscation (IO) and one-way functions exist, we define samplers  $\text{Sam}_0$ , which outputs  $(\text{aux}_0, C_0)$ , and  $\text{Sam}_1$ , which outputs  $(\text{aux}_1, C_1)$  such that:

- The distributions  $(\text{aux}_0, \text{iO}(C_0))$  and  $(\text{aux}_1, \text{iO}(C_1))$  are perfectly distinguishable.
- For padding  $s = \text{poly}(\lambda)$ , the distributions  $(\text{aux}_0, \text{iO}(C_0\|0^s))$  and  $(\text{aux}_1, \text{iO}(C_1\|0^s))$  are computationally indistinguishable.

We note this refutes the recent “Superfluous Padding Assumption” of Brzuska and Mittelbach[BM15].

---

\*Email: holmgren@csail.mit.edu. This work was done while the author was visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant #CNS-1523467. This research was also supported by NSF Eager CNS1347364, NSF Frontier CNS1413920, the Simons Foundation (agreement dated June 5, 2012), Air Force Laboratory FA875011-20225, and Lincoln Lab PO7000261954.

# 1 Introduction

Proofs of security for cryptographic constructions using indistinguishability obfuscation (IO) typically show that the obfuscations of two circuits are indistinguishable if those circuits are artificially padded to a larger size. This is a consequence of the fact that IO guarantees indistinguishability of two obfuscated circuits only if the input circuits are of equivalent sizes. But because of the ubiquitous hybrid argument style of proof, the two circuits must typically be padded not only to the size of larger circuit, but rather to the size of the largest circuit in the hybrid argument.

At this point, one may wonder whether this padding is an artifact of our security proof, or whether it is really necessary. In this work, we show that in the presence of arbitrary auxiliary information, there are distributions of equally-sized pairs of circuits, whose obfuscations are indistinguishable only if sufficiently padded.

# 2 Preliminaries

We assume familiarity with puncturable pseudorandom functions [BW13, BGI14]. In particular, we will use the fact that if one-way functions exist, then there is a puncturable PRF family  $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda>0}$  in which each  $f \in \mathcal{F}_\lambda$  maps  $\{0, 1\}^\lambda$  to  $\{0, 1\}$ .

We also assume the existence of an indistinguishability obfuscator [GGH<sup>+</sup>13]. This is a p.p.t. Turing machine  $i\mathcal{O}$  such that:

- $i\mathcal{O}(C, 1^\lambda)$  outputs a circuit which is functionally equivalent to  $C$ .
- If  $C$  and  $C'$  are two circuits of the same size and same functionality, then the advantage of any p.p.t. adversary in distinguishing  $i\mathcal{O}(C, 1^\lambda)$  from  $i\mathcal{O}(C', 1^\lambda)$  is negligible in  $\lambda$ .

We will frequently omit the security parameter  $1^\lambda$  as an argument of  $i\mathcal{O}$ .

We will write  $C\|0^s$  to denote a *padded* version of the circuit  $C$ , which is of size  $|C| + s$ .

# 3 Techniques

We want to show a pair of distributions  $(\mathbf{aux}_0, C_0)$  and  $(\mathbf{aux}_1, C_1)$  such that:

- $(\mathbf{aux}_0, i\mathcal{O}(C_0))$  is distinguishable from  $(\mathbf{aux}_1, i\mathcal{O}(C_1))$
- For some padding  $p$ ,  $(\mathbf{aux}_0, i\mathcal{O}(C_0\|0^p))$  is indistinguishable from  $(\mathbf{aux}_1, i\mathcal{O}(C_1\|0^p))$ .

In our construction,  $C_0$  and  $C_1$  are defined simply as circuits which evaluate a (puncturable) PRF.  $\mathbf{aux}_0$  and  $\mathbf{aux}_1$  are (sufficiently padded) obfuscated circuits, each of which have this same PRF inside.  $\mathbf{aux}_b$  takes a “small” circuit as input,

and checks whether this circuit agrees with the PRF on a “large” set of test inputs. If it does, then  $\text{aux}_b$  outputs  $b$ . Otherwise,  $\text{aux}_b$  outputs 0.

“Small” and “large” are chosen so that the obfuscated  $C_0$  or  $C_1$  is small, but not large.

The first bullet is easy; the slightly tricky part is to show that when  $C_0$  and  $C_1$  are padded to be large (even before obfuscation is applied), then  $(\text{aux}_0, \text{iO}(C_0))$  is indistinguishable from  $(\text{aux}_1, \text{iO}(C_1))$ . We show this in a three-step hybrid argument, starting with  $(\text{aux}_b, \text{iO}(C_b))$  for arbitrary  $b$ .

1. The PRF in  $\text{aux}_b$  and in  $C_b$  is punctured on the whole set of test inputs, and the corresponding test values are hard-coded. This is indistinguishable by  $\text{iO}$ .
2. These test values are replaced by truly random bits. This is indistinguishable by the puncturable PRF security.
3. Now there statistically does not exist any small circuit which agrees with all the test values. So  $\text{aux}_b$  is replaced by the obfuscated all zero function, which is indistinguishable by  $\text{iO}$ .

This last hybrid distribution on  $(\text{aux}_b, \text{iO}(C_b))$  is independent of the bit  $b$ .

## 4 Formal Proof

Let  $q$  be a polynomial such that  $q(\lambda)$  bounds the size of  $\text{iO}(f, 1^\lambda)$ , when  $f$  is sampled from the punctural PRF family  $\mathcal{F}_\lambda$ .

We now define a pair of algorithms  $(\text{Sam}_0, \text{Sam}_1)$ .  $\text{Sam}_b$  will output a pair  $(\text{aux}_b, C_b)$ . The algorithm  $\text{Sam}_b(1^\lambda)$  first samples a puncturable PRF  $f \leftarrow \mathcal{F}_\lambda$ .  $\text{Sam}_b$  first computes  $\text{aux}_b \leftarrow \text{iO}(A_{b,f})$ , where  $A_{b,f}$  is described in Algorithm 1, and  $\text{Sam}_b$  then computes  $C_b = B_f$ , where  $B_f$  is described in Algorithm 2.

**Input:** Circuit  $C$  of size  $q(\lambda)$   
**Data:** Bit  $b$ , PPRF  $f$

```

1 if  $C(i) = f(i)$  for all  $i \in \{0, \dots, q(\lambda) + \lambda\}$  then
2   | return  $b$ 
3 else
4   | return  $0$ 
5 end

```

**Algorithm 1:** Circuit  $A_{b,f}$

**Claim 1.** *The distributions  $(\text{aux}_0, \text{iO}(C_0))$  and  $(\text{aux}_1, \text{iO}(C_1))$  are perfectly distinguishable when  $(\text{aux}_b, C_b) \leftarrow \text{Sam}_b(1^\lambda)$ .*

*Proof.*  $\text{aux}_b$  computes the same function as  $A_{b,f}$  and  $C_b$  is  $B_f$ . The claim follows simply because  $A_{b,f}(\text{iO}(B_f)) = A_{b,f}(B_f) = b$ .  $\square$

|  |
|--|
| <b>Input:</b> $x \in \{0,1\}^\lambda$<br><b>Data:</b> PPRF $f$<br><b>1 return</b> $f(x)$ . |
|--|

**Algorithm 2:** Circuit  $B_f$

**Claim 2.** For some padding  $p_b$ , the distributions  $(\text{aux}_0, \text{iO}(C_0 \| 0^{p_b}))$  and  $(\text{aux}_1, \text{iO}(C_1 \| 0^{p_b}))$  are computationally indistinguishable when  $(\text{aux}_b, C_b) \leftarrow \text{Sam}_b(1^\lambda)$ .

*Proof.* Let  $p_b$  be padding so that  $|B_f \| 0^{p_b}| = |B_f^1|$ , where  $B_f^1$  is described in Algorithm 4. We define three indistinguishable hybrid distributions  $H_b^1$  through  $H_b^3$  such that:

- $H_b^1$  is indistinguishable from  $(\text{aux}_b, \text{iO}(C_b \| 0^{p_b}))$  when  $(\text{aux}_b, C_b) \leftarrow \text{Sam}_b(1^\lambda)$ .
- $H_b^3$  is independent of  $b$ .

**Hybrid  $H_b^1$ :** Hybrid  $H_b^1$  is sampled by first sampling a PPRF  $f \leftarrow \mathcal{F}_\lambda$ , and puncturing it on the set  $\{0, \dots, q(\lambda) + \lambda\}$  to obtain the punctured PRF  $f'$ . Let  $y_i = f(i)$  for  $i \in \{0, \dots, q(\lambda) + \lambda\}$ .

$H_b^1$  then consists of  $(\text{iO}(A_{b,f}^1), \text{iO}(B_f^1))$ . The circuit  $A_{b,f}^1$  is described in Algorithm 3, with the values  $y_i$  hard-coded, and is padded to be as large as  $A_{b,f}$ . The circuit  $B_f^1$  is described in Algorithm 4, with the PPRF  $f'$  and the values  $y_i$  hard-coded.

|   |
|---|
| <b>Input:</b> Circuit $C$ of size $q(\lambda)$<br><b>Data:</b> Bit $b$ , values $y_i$<br><b>1 if</b> $C(i) = y_i$ for all $i \in \{0, \dots, q(\lambda) + \lambda\}$ <b>then</b><br><b>2   return</b> $b$<br><b>3 else</b><br><b>4   return</b> $0$<br><b>5 end</b> |
|---|

**Algorithm 3:** Circuit  $A_{b,f}^1$

|  |
|--|
| <b>Input:</b> $x \in \{0,1\}^\lambda$<br><b>Data:</b> Punctured PPRF $f'$ , values $y_i$ for $i \in \{0, \dots, q(\lambda) + \lambda\}$<br><b>1 if</b> $x \in \{0, \dots, q(\lambda) + \lambda\}$ <b>then</b><br><b>2   return</b> $y_x$ ;<br><b>3 else</b><br><b>4   Return</b> $f'(x)$ ;<br><b>5 end</b> |
|--|

**Algorithm 4:** Circuit  $B_f^1$

**Hybrid  $H_b^2$ :** Hybrid  $H_b^2$  is sampled identically to  $H_b^1$ , but each  $y_i$  is sampled uniformly at random.

**Hybrid  $H_b^3$ :** In hybrid  $H_b^3$ , the circuit  $A_{b,f}^1$  is replaced with the constant zero function, appropriately padded.

**Claim 3.**  $H_b^1 \approx \text{Sam}_b(1^\lambda)$ .

*Proof.* This follows from the security of  $i\mathcal{O}$ : the obfuscated circuits have the same functionality and size in both  $H_b^1$  and  $\text{Sam}_b(1^\lambda)$ .  $\square$

**Claim 4.** *Hybrid  $H_b^2 \approx H_b^1$ .*

*Proof.* This follows from the pseudorandomness of the punctured PRF  $f'$  at the (selectively) punctured set  $\{0, \dots, q(\lambda) + \lambda\}$ .  $\square$

**Claim 5.** *Hybrid  $H_b^3 \approx H_b^2$ .*

*Proof.* This follows from the security of  $i\mathcal{O}$ . A simple counting argument implies that with high probability (at least  $1 - 2^{-\lambda}$ ), there is no circuit  $C$  of size  $q(\lambda)$  such that  $C(i) = y_i$  for all  $i \in \{0, \dots, q(\lambda) + \lambda\}$ . Thus the circuit  $A_{b,f}^1$  with truly random  $y_i$ 's is functionally equivalent to the constant zero circuit with high probability.  $\square$

This completes the proof of Claim 2.  $\square$

## 5 Variants of the Superfluous Padding Assumption

One interesting restriction on  $\text{Sam}_0$  and  $\text{Sam}_1$ , proposed by [BM15] as a possible weaker assumption, requires that the marginal distribution of  $\text{aux}_0$  be the same as the marginal distribution of  $\text{aux}_b$ . While this does not hold for our counterexample, it can be easily modified to have this property. Rather than having  $\text{aux}_b$  output the bit  $b$ ,  $\text{aux}_b$  outputs a random string  $r$ . On input  $r$ ,  $C_b$  outputs  $b$ .

The proof techniques above, when applied to this modified construction, show how to move to a hybrid where  $\text{aux}_b$  is independent of  $r$ . We can then apply a standard injective PRG trick to make  $C_b$  independent of  $r$  and of  $b$ .

## 6 Implication About Double Obfuscation

The necessity of superfluous padding implies a surprising result. If  $i\mathcal{O}$  increases the size of circuits, then there are efficiently sampleable distributions on  $(\text{aux}_0, C_0)$  and  $(\text{aux}_1, C_1)$  such that  $(\text{aux}_0, i\mathcal{O}^k(C_0))$  is indistinguishable from  $(\text{aux}_1, i\mathcal{O}^k(C_1))$  for some integer  $k > 1$ , but  $(\text{aux}_0, i\mathcal{O}(C_0))$  is perfectly distinguishable from  $(\text{aux}_1, i\mathcal{O}(C_1))$ . This follows from our construction of  $\text{Sam}_0$

and  $\text{Sam}_1$  because the inner  $k - 1$  obfuscations are functionally equivalent to padding.

## References

- [BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *Public-Key Cryptography PKC 2014*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519. Springer Berlin Heidelberg, 2014.
- [BM15] Christina Brzuska and Arno Mittelbach. Universal computational extractors and the superfluous padding assumption for indistinguishability obfuscation. *Cryptology ePrint Archive*, Report 2015/581, 2015. <http://eprint.iacr.org/>.
- [BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazuo Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *Lecture Notes in Computer Science*, pages 280–300. Springer Berlin Heidelberg, 2013.
- [GGH<sup>+</sup>13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49, 2013.