Cryptography 101: From Theory to Practice

Chapter 8 – Pseudorandom Functions

Rolf Oppliger

February 24, 2022

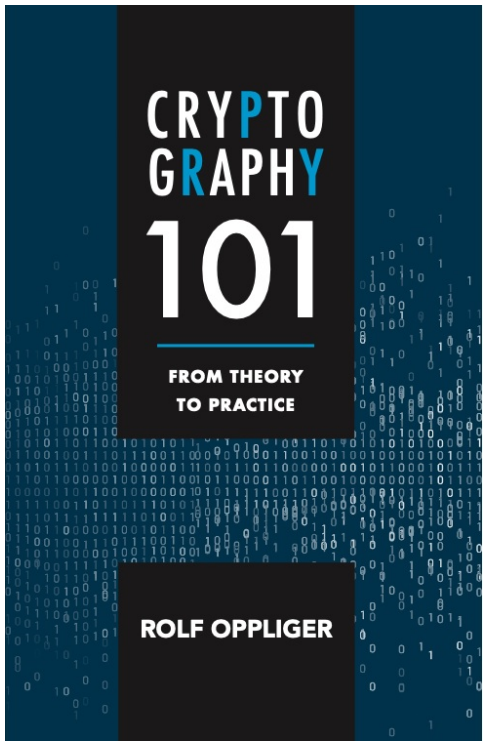
Terms of Use

whoami

rolf-oppliger.ch
rolf-oppliger.com

- Swiss National Cyber Security Centre NCSC (scientific employee)
- eSECURITY Technologies Rolf Oppliger (founder and owner)
- University of Zurich (adjunct professor)
- Artech House (author and series editor for information security and privacy)

Reference Book



ISBN 978-1-63081-846-3

`https://books.esecurity.ch/crypto101.html`

Challenge Me

Outline

8. Pseudorandom Functions

8. Pseudorandom Functions

8. Pseudorandom Functions

8.1 Introduction

- According to Definition 2.8, a **PRF** is a family $F : \mathcal{K} \times X \to Y$ of (efficiently computable) functions, where each $k \in \mathcal{K}$ determines a function $f_k : X \to Y$ that is indistinguishable from a random function, i.e., a function randomly chosen from $\mathrm{Funcs}[X, Y]$
- Because there is one function f_k for every $k \in \mathcal{K}$, there are "only" $|\mathcal{K}|$ functions in F, whereas there are $|Y|^{|X|}$ functions in $\mathrm{Funcs}[X, Y]$
- This means that one can use a small key to determine a particular function $f_k \in F$, but the function still behaves like a random function

8. Pseudorandom Functions

8.1 Introduction

- Similarly, a **PRP** is a family $P : \mathcal{K} \times X \to X$ of (efficiently computable) permutations, where each $p \in \mathcal{K}$ determines a permutation $p_k : X \to X$ that is indistinguishable from a random permutation, i.e., a permutation randomly chosen from $\mathrm{Perms}[X]$
- The logic of a PRP is essentially the same (but there are $|X|!$ permutations in $\mathrm{Perms}[X]$)
- PRFs and PRPs are omnipresent and heavily used in cryptography

8. Pseudorandom Functions

8.2 Security of a PRF

- Intuitively, a PRF is **secure** if an adversary (i.e., PPT algorithm A) cannot tell it apart from a random function
- Consider the security game, in which A can interact with $g : X \to Y$ to decide whether it is random (i.e., an element from $\mathrm{Funcs}[X, Y]$, meaning that $g \xleftarrow{r} \mathrm{Funs}[X, Y]$) or pseudorandom (i.e., an element from a PRF family $F : \mathcal{K} \times X \to Y$, meaning that $k \xleftarrow{r} \mathcal{K}$ and this key fixes a function f_k from F)

8. Pseudorandom Functions

8.2 Security of a PRF

- The PRF advantage of A with respect to F is defined as

$$\mathrm{Adv}_{\mathrm{PRF}}[A, F] = \left| \Pr_{g \xleftarrow{r} F}[A(g) = 1] - \Pr_{g \xleftarrow{r} \mathrm{Funcs}[X, Y]}[A(g) = 1] \right|$$

- To argue about the security of PRF F, one considers the PPT algorithm A with maximal PRF advantage
- The PRF advantage of F is defined as

$$\mathrm{Adv}_{\mathrm{PRF}}[F] = \max_{A}\{\mathrm{Adv}_{\mathrm{PRF}}[A, F]\}$$

8. Pseudorandom Functions

8.2 Security of a PRF

- PRF F is secure, if $\mathrm{Adv}_{\mathrm{PRF}}[F]$ is negligible, i.e., for every polynomial p, there exists a $n_0 \in \mathbb{N}$ such that for all $n > n_0$

$$\mathrm{Adv}_{\mathrm{PRF}}[F] \leq \frac{1}{p(n)}$$

- The bottom line is that for a secure PRF F, there is no PPT algorithm that can distinguish an element from F from a truly random function
- This means that F behaves like a random function and can be used in place of it (mainly in security proofs)

8. Pseudorandom Functions

8.3 Relationship between PRGs and PRFs

- PRGs and PRFs are closely related to each other in the sense that one can construct one from the other
- Construct a PRG G from a PRF F:
  - Randomly select a key $k \in \mathcal{K}$ and iteratively apply f_k to an incrementing counter

$$G(k) = (f_k(i))_{i \geq 0} = f_k(0), f_k(1), f_k(2), f_k(3), \ldots$$

  - Note that f_k is pseudorandom and not "only" one-way
  - If F is a secure PRF, then G is a cryptographically secure PRG
  - The efficiency of G depends on the efficiency of F

8. Pseudorandom Functions

8.3 Relationship between PRGs and PRFs

- Construct a PRF F from a PRG G:
  - Let $G(s)$ be a PRG for $s \in \{0,1\}^n$ with stretch function $l(n) = 2n$
  - $G_0(s)$ refers to the first n bits of $G(s)$, whereas $G_1(s)$ refers to the last n bits of $G(s)$
  - $X = Y = \{0,1\}^n$, and $x = \sigma_n \cdots \sigma_2 \sigma_1$ is the bitwise representation of x
  - A PRG-based PRF $F : X \to Y$ can be defined as

$$f_s(x) = f_s(\sigma_n \cdots \sigma_2 \sigma_1) = G_{\sigma_n}(\cdots G_{\sigma_2}(G_{\sigma_1}(s)) \cdots)$$

  - The definition is simple, but the construction is not very intuitive (and too inefficient to be used in the field)

8. Pseudorandom Functions

8.3 Relationship between PRGs and PRFs

- Toy example
  - For $n = 2$, one can use a PRG G that is defined as follows:

$$\begin{aligned} G(00) &= 1001 \\ G(01) &= 0011 \\ G(10) &= 1110 \\ G(11) &= 0100 \end{aligned}$$

  - For $s = 10$ and $x = 01$ (i.e., $\sigma_2 = 0$ and $\sigma_1 = 1$), $f_s(x) = f_s(\sigma_2\sigma_1) = f_{10}(01) = G_0(G_1(10)) = 11$
  - To compute this value, one first computes $G_1(10) = 10$ (i.e., the last two bits of $G(10) = 11\underline{10}$) and then $G_0(10) = 11$ (i.e., the first two bits of $G(10) = \underline{11}10$)

8. Pseudorandom Functions

8.4 Random Oracle Model

- The **random oracle methodology** was proposed by Mihir Bellare and Philip Rogaway in the early 1990s
- The goal was to provide "a bridge between cryptographic theory and cryptographic practice"
- The methodology is widely used to design cryptographic systems (mostly protocols)
- The resulting systems are provably secure in the **random oracle model** (as opposed to the **standard model**)

8. Pseudorandom Functions

8.4 Random Oracle Model

- The random oracle methodology consists of three steps:
  - Design an ideal system in which all parties — including the adversary — have access to a random function
  - Formally prove the security of this ideal system
  - Replace the random function with a PRF and provide all parties with a specification of it
- As a result, one obtains an implementation of the ideal system in the real world
- A formal analysis in the random oracle model is not a security proof (because of the ideality assumption), but it provides useful evidence for the security of the system

8. Pseudorandom Functions

8.4 Random Oracle Model

- Unfortunately, it has been shown that random oracles cannot be implemented cryptographically
- In particular, it has been shown that an (artificially crafted) DSS exists that is secure in the random oracle model but gets totally insecure when the random oracle is implemented with a (family of) cryptographic hash function(s)
- In theory, the random oracle model is discussed controversially
- In practice, no protocol proven secure in the random oracle model has been broken so far (when used with a standard cryptographic hash function, like SHA-1)

8. Pseudorandom Functions

8.5 Final Remarks

- PRFs and PRGs are closely related
- It is possible to construct a PRG if one has a PRF, and — vice versa — to construct a PRF if one has a PRG
- The respective constructions are conceptually simple and straightforward, but they are purely theoretical and not meant to be used in the field
- In many situations, proving the security in the random oracle model (instead of the standard model) is the best one can do
- The literature is full of such "proofs"

Questions and Answers

Thank you for your attention