Chapter 1

Introduction

Cryptography is an ancient subject. As long as humans have used writing, we have attempted to obscure the meaning of our communications using encryption. These early ciphers were primitive, of course. Modern cryptography, as we now know it, only came into its own with the advent of the modern digital computer. It is no coincidence that two of the most important figures in the birth of computing, Alan Turing and Claude Shannon, each spent a portion of their careers thinking about encryption.

Today, cryptography is vastly more important than it has been at any time in the past. Encryption (and authentication) are critical to securing every network connection made by an application on your phone, every phone call you make, and every purchase you make in a store. The field of applied cryptography has also expanded to include many techniques beyond encryption, including modern zero-knowledge proofs that play an increasing role in cryptocurrencies, as well as technologies like multi-party computation and fully-homomorphic encryption. Cryptography has become so critical to the functioning of our digital society that events decade down the road – notably, the creation of efficient quantum computers – raise enough concern that we feel the need to respond to it today.

In the public world, the progress of modern cryptography (also known as the science of *cryptology*) has followed two parallel tracks. In one track, practitioners devise new algorithms and protocols, and deploy them to meet the various needs of industry. This engineering-focused track is the birthplace of protocols such as SSL/TLS and IPSec, as well as more recent systems like the Signal Protocol, which secures billions of private conversations. The second track is the path of academic and scientific progress: in this track we find the development of provable security, public key cryptography, and many of the advanced techniques that will secure the computer networks of tomorrow. While these two tracks have come into closer contact over the years, they often fail to meet each other. The result is the existence of deployed cryptographic systems that fall to known attacks, as well as research cryptography that addresses the wrong needs.

1.1 What this book is, and what it isn't

The purpose of this book is to help bridge the gaps between these two paths, and to teach a rigorous and *science-based* approach to cryptographic engineering. It is aimed at the practitioner who wishes to build modern cryptographic systems using the best tools and knowledge available to them. Most critically, this book is *not a textbook aimed at cryptography students.* (If you need one of those, you can find many recommendations in the final section of this book.) This book is aimed at cryptographic engineers and practitioners: it teaches you everything I think you need to know in order to build systems that are safe and efficient, ranging from simple encryption protocols to advanced technology such as zero-knowledge proof systems and FHE.

While this book is aimed at practitioners, this does not mean it is shallow. The purpose of this book is not to provide a cookbook or a reference book you can use to grab snippets of code. This would not last for long, anyway, because libraries and technologies change too quickly. It is intended to give you a deep and durable understanding of *how the technologies work* so you can think about them as a cryptographer would. Specific questions like "which library should I use" will be addressed in the appropriate sections, but they are not the focus of the work. The point of this book is to save you the pain and difficulty of reading decades worth of research papers and software libraries, so you can get to the point where you can actually do things.

More concretely, this book represents the culmination of everything that I've learned over the 25 years of my career in this field. It covers decades of history, including numerous vulnerabilities that continue to plague us. It will not simply cover the mathematics of cryptography, but the software techniques you will need to build it well. And most critically, it is my attempt to instill in the reader a *cryptographer's mindset*: more than just algorithms and techniques, this implies an opinionated and adversarial way to think about cryptographic systems.

1.2 What you'll need

This book is aimed at programmers and engineers, not mathematicians. This is because I am one of the former, and certainly not the latter. I don't expect you to come to this area with anything more than high-school algebra, or at best a bit of college-level linear algebra. It'll make your life easier (in some later chapters) if you have a bit of comfort with these things, and can reason about things like matrix multiplication and polynomials. But I won't assume you have any expertise in this. Where the mathematics is necessary, most will be of the sort you can reason about with pencil and paper, or at most a simple Python script.

You should also know as well that I write for myself as much as you. The best way to really understand a topic is to teach it. And the best way to teach it is to explain every single aspect rigorously. Just because I've been working in this area for years doesn't mean I don't have my own blind spots — areas where I "knew" how to do something, then found out my understanding was patchy. This book is as much my attempt to fill those in as it is my attempt to teach you.

1.3 Outline of this book

The remainder of this book follows a pretty simple outline. In the next chapter I'll begin by giving a short overview of the history of cryptography, from the earliest classical ciphers up to what we now refer to as the modern era. Next I'll start with the boring details of encryption, block ciphers and stream ciphers. The following chapters will introduce public-key cryptography and modern cryptographic protocols. From there I'll move on into more exotic techniques, such as two-party protocols, multi-party computation and zero-knowledge.

This book is loosely based on courses I've taught for years at Johns Hopkins, but it's intended to be more exhaustive. Wherever I can, I'll try to give you pointers to outside resources where you can learn more, or read the original works in their entirety. A final chapter gives a list of further reading, including cryptography texts and other engineering resources that you might find useful as you expand your knowledge.

With all of this said, welcome to the field – and good luck.