# Preface

Symmetric cryptography deals with:

1. the construction of efficient pseudo random functions (PRF), which are the building blocks of symmetric cryptography, and
2. symmetric cryptographic protocols, which are strategies to utilize the building blocks to solve some of our our day-to-day problems.

This book does not concern itself with the building blocks themselves; several well studied PRFs in the form of block ciphers and hash functions already exist. The focus of this book is instead on the (often under appreciated) range and utility of protocols and constructions that utilize symmetric PRFs.

Lack of widespread appreciation of the scope of symmetric cryptography has led to the unwarranted use of more expensive asymmetric cryptography in situations where symmetric cryptography is adequate. Perhaps, it is the sheer elegance of asymmetric primitives that instills in us the desire to honor them—by utilizing them even in situations where symmetric cryptography is adequate. This is one situation that this book aims to rectify.

The specific topics addressed in this book include:

1. various key distribution strategies for unicast, broadcast, and multicast security associations, and
2. strategies for constructing compact and efficient digests of dynamic databases.

A unified treatment of seemingly unrelated protocols is made possible by the fact that only three basic strategies, viz., hash chains, hash trees, and the surprising uniqueness of random subsets, are reused in a variety of different ways in different protocols.

Ultimately, the utility a cryptographic algorithm stems from the ability to leverage well-deserved assumptions regarding the properties of such algorithms; that we can virtually guarantee the existence of specific relationships between various inputs and outputs of the algorithm; for example, that the preimage of a cryptographic hash was chosen *before* the image was computed, and not vice-versa. Cryptographic algorithms are building blocks for the construction of application-specific cryptographic protocols, to enable enforcement of application-specific requirements, between various (application-specific) inputs and outputs.

By themselves, cryptographic protocols (unfortunately) do not provide the necessary (application-specific) *context* to the inputs and outputs. It is up to security protocols that utilize cryptographic protocols to do so. Consequently, practical security protocols will always need to make some *additional* noncryptographic assumptions regarding the environment in which the cryptographic protocol is executed, and the privacy of keys employed by the algorithms.

Almost every security issue we face in our day-to-day lives stems from the simple fact that many such noncryptographic assumptions turn out to be unjustified. For example, while the secure socket layer (SSL) is perfectly safe as a cryptographic protocol, when used as a security protocol, many vulnerabilities can crop up—like the recent Heart-bleed vulnerability, or the fact that SSL as a security protocol relies on the integrity of the public key infrastructure (PKI), which in turn relies on unverifiable assumptions regarding the integrity of PKI certificate authorities.

Perhaps the only practical recourse is to invest in an infrastructure to realize sufficiently trustworthy hardware modules. Such modules should be capable of guaranteeing a safe environment in which a wide variety of cryptographic protocols—necessary for a wide range of applications—can run unmolested. Only the well-deserved trust in the assumed properties of cryptographic algorithms, and the integrity of such hardware modules, can then be bootstrapped to realize security protocols—without the need to make unjustifiable assumptions like the integrity of software and hardware components in general purpose computers or the integrity of personnel/organizations with access to sensitive data processed in the computers. The versatility and low resource requirement for protocols based on symmetric PRFs make them very well suited for such an approach. Simple fixed functionality involving only PRF and logical operations, executed within the confines of deliberately resource limited modules, can be more readily verified to be free of malicious functionality. Almost every security protocol outlined in this book pays extra attention to additional constraints that may be imposed due to the fact that the security protocols will need to be executed inside a trustworthy (and severely resource limited) boundary.

Chapter 1 is a brief review of well-known properties of symmetric PRFs like hash functions and block ciphers. Chapter 2 outlines some useful constructions using PRFs that are reused throughout this book.

Chapter 3 presents key predistribution schemes for pairwise authentication—strategies that are traditionally considered as *nonscalable*. Two such schemes, the modified Leighton–Micali scheme (MLS) and the identity tickets (IT) schemes are, however, shown to be "scalable enough" for most practical applications. Chapter 4 outlines a strategy for employing such schemes in conjunction with trustworthy hardware modules with trivial functionality to secure the domain name system (DNS). This approach is compared with the current security protocol, DNSSEC, for securing DNS.

Chapters 5 and 6 present various scalable key predistribution schemes. Chapter 5 outlines many of the advantages of probabilistic schemes over deterministic schemes. Chapter 6 outlines three scalable schemes realized as extensions of nonscalable

schemes discussed in Chap. 3. Chapter 7 highlights special considerations for protecting the integrity of secrets inside resource limited tamper-responsive boundaries. Such considerations are taken into account to reevaluate the strengths of various key distribution schemes, and the overhead associated with each approach.

Chapter 8 reviews strategies for multicast security associations like one-to-many associations (or broadcast security) and group security associations facilitated using broadcast encryption. While most broadcast encryption schemes employ a tree-like structure, "flat" schemes based on probabilistic key distribution have some compelling advantages. The utility of such schemes for practical deployments of publish–subscribe systems is also discussed in this chapter.

Chapter 9 presents a useful authenticated data structure, the ordered Merkle tree (OMT), and it's utility in assuring the integrity of a wide variety of dynamic databases maintained by untrusted entities. Two variations of the OMT, viz., the index ordered Merkle tree (IOMT), and the domain ordered Merkle tree (DOMT), are discussed. Simple algorithms intended to be executed by a trusted resource limited "verifier," to assure the integrity of a database maintained by an untrusted "prover," are presented.

Chapter 10 discusses a new *credential transaction model* as a specification of application-specific security protocols. For any system with a desired set of assurances, the strategy is to identity different roles for participants in the system, and a set of "permitted credential transactions" for each role. The permitted credential transactions are chosen to guarantee that no desired assurance is violated. Thus, as long as we can assure the integrity of credential transactions, we can assure the integrity of the entire system (that all desired assurances are met).

The credential transaction model permits the design of a universal trusted base—as a hypothetical specification for trusted *credential management modules* (CMM). Irrespective of the specific nature of the system, CMMs are entrusted with the task of assuring the integrity of credential transactions. Only assumptions regarding the integrity of PRFs, and the integrity of simple algorithms executed inside CMMs to verify the integrity of credential transactions, are bootstrapped by the security protocol (the transaction model) to realize all desired assurances. Such an approach eliminates the need for unjustifiable trust in complex hardware/software components, and personnel with the ability to influence the operation of such computers. The core functional components of CMMs include functionality described in Chap. 7 for unicast security associations, and functionality for maintaining OMTs, described in Chap. 9.

Starkville, MS
April 2014

Mahalingam Ramkumar

http://www.springer.com/978-3-319-07583-9

Symmetric Cryptographic Protocols

Ramkumar, M.

2014, XVII, 234 p. 21 illus., 1 illus. in color., Hardcover
ISBN: 978-3-319-07583-9