1 Introduction

1.1 Information Theory and Cryptography

Information theory is a close cousin of probability theory. While probability allows us to model what it means to accurately know or estimate an unknown, information theory allows us to exactly capture the amount of uncertainty. The foremost exponent of this notion is Shannon's entropy $H(X)$ for a random variable X. Without knowing anything else about X, there is an "uncertainty" of $H(X)$ about X. When we know a correlated random variable Y, this uncertainty reduces to $H(X \mid Y)$, the conditional Shannon entropy of X given Y. Shannon defined this reduction in uncertainty $I(X \wedge Y) := H(X) - H(X \mid Y)$ as the measure of information revealed by Y about X. Over the years, Shannon theory has evolved to provide a comprehensive justification for these measures being appropriate measures of information. These measures of information are now gainfully and regularly applied across areas such as biology, control, economics, machine learning, and statistics.

Cryptography theory, the science of maintaining secrets and honesty in protocols, adopted these notions at the outset. Indeed, it was Shannon himself who wrote the first paper to mathematically model secure encryption, and he naturally adopted his notions of information in modeling security. Encryption requires us to send a message M over a public communication channel in such a manner that only the legitimate receiver, and no one else, gets to know M from the communication C. Shannon considered $I(M \wedge C)$ as the measure of information about M leaked by the communication.[1] Thus, at the outset information theory provided a way to measure secrecy in cryptography. The two theories were joined at birth!

But a major development took place in cryptography in the late 1970s. Diffie and Hellman invented an interesting key exchange scheme which was not information-theoretically secure, but was secure in practice. Specifically, their scheme relied on the fact that discrete exponentiation is easy to compute, but (computationally) very difficult to invert. This insight led to the quick development of many fascinating and practical cryptography protocols, all seemingly

[1] To be precise, Shannon's original paper did not consider partial information leakage and did not talk about $I(M \wedge C)$, but the notion was clear and was picked up in subsequent works.

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

difficult to break in practice but clearly not secure in the information-theoretic sense of Shannon. This was the birth of the field of computational cryptography.

In another remarkable insight, Goldwasser and Micali formulated the notion of semantic security for formally analyzing computational security. This new formulation related security of encryption to the ability to test certain hypotheses about messages by looking at the communication. Over the years this idea evolved, in particular to handle the challenge of formalizing security for an adversary that can deviate from the protocol. The modern framework defines security in terms of the difference in the ability of the adversary in an ideal (secure) system and a system under attack. If every adversarial behavior for a protocol can be "simulated" in the ideal system, the protocol is deemed to be secure.

This modern formulation is very flexible: it can be applied to both information-theoretic and computational settings. In the computational setting, the adversary is restricted to using polynomial-time algorithms; in the information-theoretic setting, there is no computational restriction on the adversary. It is a subtle point, but there is no strict hierarchy between the two notions. They should be viewed as two different assumption classes under which one can analyze the security of various cryptographic primitives.

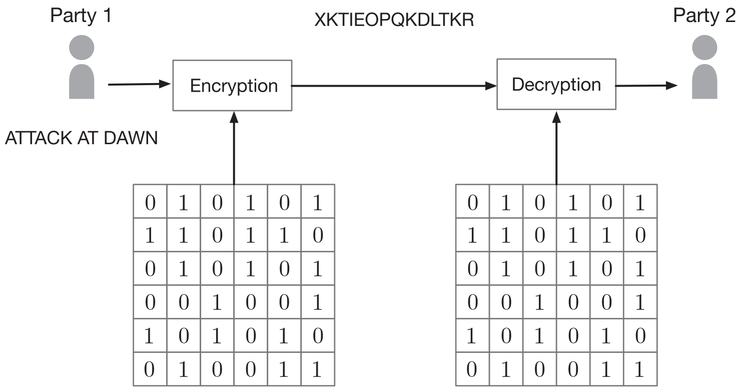
Specifically, computational cryptography assumes the availability of certain computational primitives such as one-way functions which are easy to compute but computationally hard to invert. Using such primitives, we design cryptographic protocols that remain secure as long as the adversary is computationally restricted to using polynomial-time algorithms. On the other hand, *information-theoretic cryptography* seeks to establish cryptographic protocols that are information-theoretically secure. Often this requires additional resources; for instance, encryption is possible only when the parties share secret keys and two-party secure computation requires the availability of nontrivial correlated observations (such as oblivious transfer).

This book is a comprehensive presentation of information-theoretically secure cryptographic primitives, with emphasis on formal security analysis.

1.2 Overview of Covered Topics

As mentioned in the previous section, a systematic study of cryptography with security analysis in mind was initiated in Shannon's landmark paper. The focus of Shannon's paper was enabling secure communication between legitimate parties over a public communication channel that may not be secure. This is an important problem which cryptography has solved since then, and has enabled secure banking and communication over the Internet. Among other things, Shannon's focus in the paper was the secret key encryption system described in Figure 1.1. In this system, Party 1 (sender) sends a secret message to Party 2 (receiver). To secure their message, the parties encrypt and decrypt the message using a shared secret key (a sequence of random bits). One of Shannon's contributions is

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press



0	1	0	1	0	1
1	1	0	1	1	0
0	1	0	1	0	1
0	0	1	0	0	1
1	0	1	0	1	0
0	1	0	0	1	1

0	1	0	1	0	1
1	1	0	1	1	0
0	1	0	1	0	1
0	0	1	0	0	1
1	0	1	0	1	0
0	1	0	0	1	1

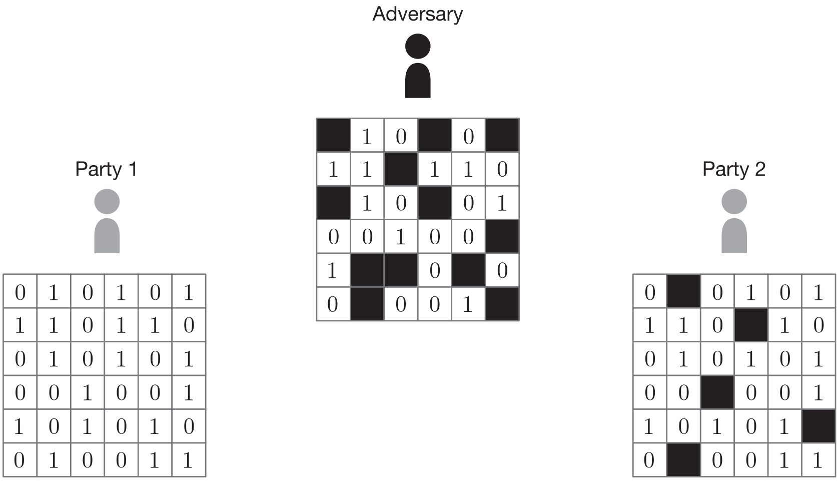
Figure 1.1 A description of secret key encryption. We depict a common anecdotal motivation where a General wants to command their officer to "attack at dawn" over an insecure channel.

to formally describe the notion of security for this secret key encryption system. In particular, Shannon defined the notion of *perfect secrecy* which ensures that an eavesdropper observing all the communication sent over the channel cannot glean any information about the message the sender wants to send to the receiver. The main theoretical result Shannon established was the following: in order to attain perfect secrecy, the length of the shared secret key must be as long as the length of the message. In Chapter 3, we will cover the definition of perfect secrecy and Shannon's result, as well as some other relevant concepts in secret key encryption. We also define notions of approximate secrecy as a relaxation to the perfect secrecy requirement, which lays the foundation for security analysis of modern cryptographic schemes.

As suggested by Shannon's pessimistic result, one of the most important problems in cryptography is how to share a secret key among the legitimate parties. In most current technology, the secret key is exchanged by using so-called public key cryptography, which guarantees security against a computationally bounded adversary. However, in certain applications, it is desirable to have a method to share a secret key even against an adversary who has unlimited computational power. In fact, utilizing certain physical phenomena, methods to share a secret key have been proposed, such as quantum key distribution, or key generation using a wireless communication signal. In Chapter 10, we will cover the data processing part of those key agreement methods, termed the secret key agreement problem.[2] When a secret key is shared using a certain physical carrier, the secret key observed by the receiver is disturbed by noise; furthermore, a part of the key may be leaked to the adversary; see Figure. 1.2. Thus, we have to correct

[2] Technically speaking, in quantum key distribution, we need to consider the density operator instead of random variables so that we can take care of an adversary who may have quantum memory to store eavesdropped signals, which is beyond the scope of this book. However, apart from mathematical difficulties arising from analysis of the density operator, most of the cryptographic concepts necessary for quantum key distribution are covered in this book.

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press



Party 1

0	1	0	1	0	1
1	1	0	1	1	0
0	1	0	1	0	1
0	0	1	0	0	1
1	0	1	0	1	0
0	1	0	0	1	1

Adversary

■	1	0	■	0	■
1	1	■	1	1	0
■	1	0	■	0	1
0	0	1	0	0	■
1	■	■	0	■	0
0	■	0	0	1	■

Party 2

0	■	0	1	0	1
1	1	0	■	1	0
0	1	0	1	0	1
0	0	■	0	0	1
1	0	1	0	1	■
0	■	0	0	1	1

Figure 1.2 A description of the situation in secret key agreement.

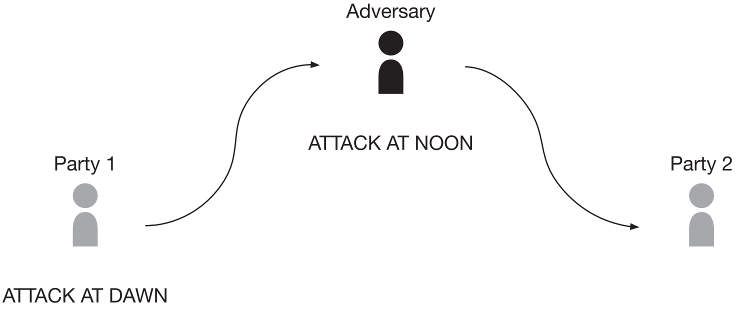


Figure 1.3 A description of authentication. An adversary intercepts the legitimate "attack at dawn" message and replaces it with a fake "attack at noon" message.

the discrepancy between the keys observed by the parties; also, we have to eliminate the information leaked to the adversary. The data processing handling the former problem is termed *information reconciliation*, and it will be covered in Chapter 6. On the other hand, the data processing handling the latter problem is termed *privacy amplification*, and it will be covered in Chapter 7. Furthermore, in Chapter 4, we will cover a cryptographic tool termed the universal hash family, which is used for information reconciliation and privacy amplification.

Along with secret key encryption, the second important problem for enabling secure communication over a public channel is that of *authentication*. In this problem, we would like to prevent an adversary from forging or substituting a transmitted message; see Figure. 1.3. This topic will be covered in Chapter 8 after the key tool, the *strong universal hash family*, is presented in Chapter 4. One of the major results in the authentication problem is that a secure authentication scheme can be realized by using a secret key of length that is of logarithmic order of the message length. This is in contrast to Shannon's result

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

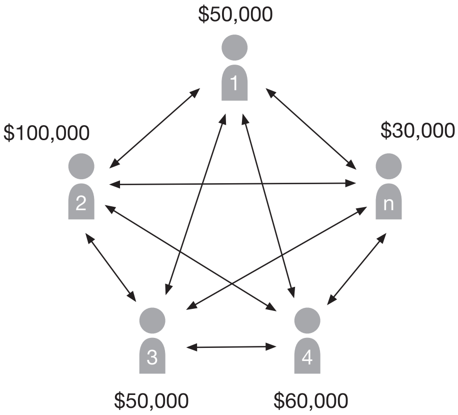


Figure 1.4 A description of multiparty secure computation.

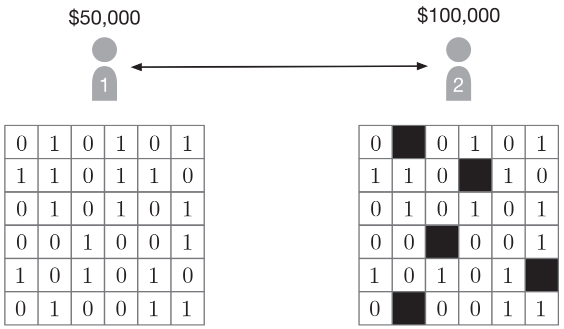
for secure encryption, and suggests that authentication has a much milder secret key requirement in comparison to encryption.

In the second part of the book, we move beyond secure message transmission to more modern problems in cryptography, the ones that are fueling the blockchain and web3.0 applications. Specifically, we present the concepts of secure computation in Chapters 12–19. In secure computation, two or more parties with their own private data seek to execute a program on their collective data and ascertain the outcome. However, the parties do not want to reveal their data to each other beyond the minimal revealed by the outcome. For an anecdotal example, the parties may represent employees who want to compute the average of their salaries without revealing their individual salaries to others; see Figure. 1.4. An important question in this problem is how many dishonest parties we can tolerate in order to realize secure computation. It turns out that, depending on the power of the adversary (whether the adversary can modify the protocol or not), we can realize secure computation when honest parties form a majority or a "supermajority."

For the two-party setting, one dishonest party means a dishonest majority, and we cannot realize secure computation for any nontrivial function. For instance, the parties cannot compare who is wealthier without leaking the value of their salaries to each other. However, if we assume that the parties have some additional resources at their disposal – for instance, they may have access to correlated observations – then it is possible to realize secure computation; see Figure 1.5. In contrast to the reliable communication system in which noise is always troublesome, it turns out that noise can be used as a resource to realize certain tasks in cryptography; this is covered in Chapter 13 and Chapter 14.

When parties in a peer-to-peer network do not trust each other, it is difficult to get consensus among the parties. This is one of the major problems that must be addressed when we consider more than two parties. An important example is the *broadcast* problem where one party wants to send the same message to multiple

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press



0	1	0	1	0	1
1	1	0	1	1	0
0	1	0	1	0	1
0	0	1	0	0	1
1	0	1	0	1	0
0	1	0	0	1	1

0	■	0	1	0	1
1	1	0	■	1	0
0	1	0	1	0	1
0	0	■	0	0	1
1	0	1	0	1	■
0	■	0	0	1	1

Figure 1.5 A description of two-party secure computation using noisy correlation.

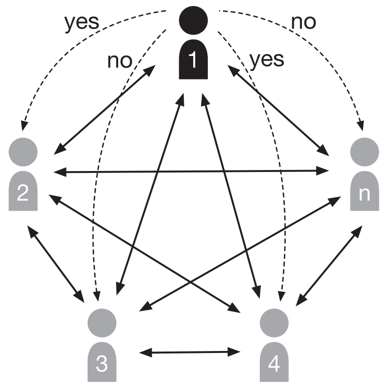


Figure 1.6 A description of broadcast.

parties. But the first party can cheat and send different messages to different parties; see Figure 1.6. In order to detect such an attack, other parties need additional communication to confirm that the message from Party 1 is consistent. Broadcast is a fundamental primitive for multiparty secure computation and is covered in Chapter 18. A key result of this problem is that, if honest parties in the network form a "supermajority," it is possible to realize a secure broadcast. The main secure function computation results are given in Chapters 12, ,17, and 19: two-party secure computing with passive adversary is in Chapter 12, with active adversary in Chapter 17, and multiparty secure computing in Chapter 19.

1.3 Overview of the Technical Framework

Our technical agenda in this book is two-fold. First, we want to lay foundations for a "resource theory" of cryptography, explaining how the availability of different kinds of correlation makes different cryptographic primitives feasible even under information-theoretic security. Second, we want to bring out through examples how various notions of information-theoretic security have evolved in cryptography, culminating in the composable security framework for an active adversary. We elaborate on each of these items below.

1.3.1 A Resource Theory for Cryptography

Right at the outset, Shannon established a result showing that information-theoretically secure encryption requires the legitimate parties to share a secret key that is as large as the length of the messages they want to transmit. For authenticating a message (showing that the message is indeed sent by a legitimate user), parties need to share secret keys of logarithmic length in the message. Thus, if one has secret shared randomness, secret keys, then both authentication and encryption are possible. In fact, this secret shared randomness can itself be extracted from any nontrivial correlated observations (of appropriate amount) that the parties share. So, we can update the observation above and claim that any correlated observations which can give secret keys of appropriate size are a sufficient resource for both encryption and authentication, two primitives which alone account for most cryptographic applications used by industry today.

Interestingly, this shared secret key resource is not sufficient for two-party secure computing (beyond that for trivial functions). We will see that we need a more interesting correlation for secure computing, where either party has some part of the correlation left to itself. A prototypical example is oblivious transfer where one party gets two random bits (K_0, K_1) and the second gets a random bit B and K_B. In this correlation, both parties have some randomness that is not available to the other, and yet there is reasonable correlation between their observations. For multiparty secure computing, we actually do not need any additional resources beyond shared secret keys if a sufficiently large fraction of parties is honest. But the availability of primitives such as digital signatures, Byzantine agreement, and secure broadcast allow more efficient implementation. Indeed, recent progress in blockchains uses these primitives to enable very complex secure multiparty consensus.

An important point to note here is that enabling different cryptographic primitives requires different amounts of resources. For example, to extract a secret key of length equal to the message (which is needed for secure encryption), we need a sufficient amount of a specific type of correlation. The understanding of such tight results is not very mature, but there is a lot of clarity in some settings. For instance, we know exactly how long a secret key can be extracted using independent copies of given correlated random variables. However, such results are not well understood for general secure computing problems. For instance, we do not know what is the most efficient manner of using oblivious transfer to securely compute a given function.

We provide tools to tackle such questions as well. There are two parts to such results. The first part is a scheme which efficiently converts one type of resource into another. As an example, the *leftover hash lemma* shows that a random hash of length ℓ applied to a random variable X will produce an output that is

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

uniform and independent of another random variable Y as long as ℓ is less than roughly the conditional minimum entropy of X given Y. The second part is the so-called converse result in information theory, an impossibility result showing that nothing better will be possible. Most of our converse results in this book rely on a very general result relating the cryptographic task to the difficulty of a statistical problem involving testing the correlation in the resource.

When implementing secure computing of a function using simpler primitives, we view the function using its Boolean circuit. The problem then reduces to computing each binary gate, say NAND, securely, but without revealing intermediate outputs. For this latter requirement, we need a scheme called *secret sharing* which allows multiple parties to get parts of a secret in such a manner that they can reconstruct it only when they come together. It turns out that this secret sharing can be implemented without any additional resources. Thus, the only additional resource needed is a secure implementation of gates such as NAND.

There is subtlety when handling an active adversary. The protocol above requires each party to complete some computations locally, and an active adversary can modify these computations to its advantage. To overcome this difficulty, we introduce a very interesting primitive called *zero-knowledge proofs*, which allow one party to establish that it has indeed completed the desired calculation without revealing the input or output. Almost the same setup extends to multiple parties as well. In fact, as mentioned earlier, if there are sufficiently many honest parties, we can compute any function securely. At a high level, this works out because now there are multiple honest players holding shares of inputs or outputs, and they can have a majority over dishonest players. One extra primitive needed here is *verifiable secret sharing*, which we build using interesting polynomial constructions over finite fields.

1.3.2 Formal Security Definitions

We have already outlined earlier in the chapter how initial security definitions were formulated using information theory. If we want to ensure that a communication C does not reveal any information about a message M, we will require that the mutual information $I(M \wedge C)$ is small. Similarly, with some more care, we can capture the low leakage requirements for secure computing using different mutual information quantities. Note that here we already assume a distribution on M, or on inputs in secure computing. This distribution need not be fixed, and we can consider the worst-case distribution. It is an interesting philosophical exercise to consider what this distribution represents, but we will simply take this distribution as the prior knowledge the adversary has about the unknown.

Instead of using mutual information quantities, an alternative approach (which is perhaps more popular in cryptography) is to define security operationally. For instance, indistinguishable security requires that for any two messages of the adversary's choice, by looking at the communication the probability of error for

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

the adversary to find out which message was sent remains close to 1/2 (which corresponds to a random guess). These operational notions can be shown to be roughly equivalent to notions using information quantities. It is important to note that this equivalence is valid for the information-theoretic security setting, but for the computational setting there is no closed form information quantity available. An advantage of having these closed form quantities is that when analyzing security for specific protocols, these quantities can be broken down into smaller components using so-called "chain rules," allowing simpler analysis. Of course, even the operational notions can be broken down into such smaller components using so-called "hybrid arguments." We will see all this in the book.

Both the notions of security mentioned above are great for curtailing leaked information, namely information revealed beyond what is expected. This suffices for handling a passive adversary who does not deviate from the protocol, but may try to get information it is not entitled to. However, these notions are not sufficient for handling an active adversary who may not even follow the protocol. A major achievement of cryptography theory is to have a sound method for analyzing security in the presence of an active adversary. To formalize security in this setting, the first thing to realize is that we must moderate our expectations: some attacks are unavoidable. For instance, if a party modifies its input to the protocol, there is nothing much that we can do about it. To formalize this, we can at the outset think of an ideal protocol to which parties give inputs and receive outputs as expected, allowing for the unavoidable (admissible) attacks. Security can then be defined as how different can information extracted by an adversary be when using an ideal protocol versus the protocol that we are analyzing.

This formulation is profound, but tedious to use in practice. In fact, most proofs in the literature omit many details, and perhaps writing all the details will make them very long. It is a fascinating story that the most basic result in two-party secure computation, namely the fact that the availability of oblivious transfer allows one to compute any function securely, has been proved several times over a span of roughly 30 years, and each time for a more powerful adversary. We believe this is because these arguments are so delicate. In this book, we make an attempt to collect many of these examples in one place, slowly building concepts to handle security analysis for an active adversary.

1.4 Possible Course Plans

With a few exceptions, the dependency of chapters in this book is depicted in Figure 1.7. Even though later chapters tend to depend on earlier chapters, dependencies are partial. For instance, the information-theoretic tools provided in Chapters 5–7 are not necessary to understand most parts of Chapters 15–19; knowledge of the secret key agreement in Chapter 10 is only necessary to derive

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

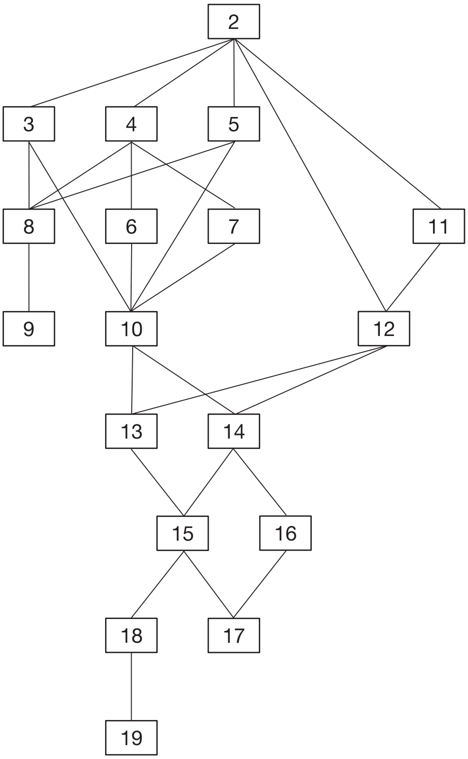


Figure 1.7 Dependence graph of the book; the numbers refer to the corresponding chapters.

the impossibility results of oblivious transfer and bit commitment in Chapter 13 and Chapter 14. Keeping in mind these partial dependencies, we suggest the following three possible course plans.

Basic Cryptography

Starting with Chapter 2, this course will cover two basic problems of cryptography: encryption (Chapter 3) and authentication (Chapter 8). Also, as preparatory tools for authentication, we will cover the basic concepts of universal hash family (Chapter 4) and hypothesis testing (Chapter 5, but Sections 5.5 and 5.6 may be skipped). Then, we will introduce computationally secure encryption and authentication (Chapter 9), in order to highlight how information-theoretic cryptography is related to computationally secure cryptography. If time permits, secret sharing (Chapter 11) may be covered as well since it does not require any other prerequisites from earlier chapters.

Cryptography from Correlated Randomness

This is an advanced course which will start in the same manner as the previous basic course, but will focus on the role of correlated randomness in en-

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

abling cryptographic primitives. In addition to the topics above, we will cover secret key agreement (Chapter 10), oblivious transfer (Chapter 13), and bit commitment (Chapter 14). As preparatory tools for these problems, we will cover the universal hash family (Chapter 4), hypothesis testing (Chapter 5), information reconciliation (Chapter 6), and random number generation (Chapter 9). In order to motivate the oblivious transfer problem, we can cover some material from Chapter 12. This course will be suitable for graduate students working in cryptography, information theory, or quantum information theory. It highlights the main theme of this book: the use of information-theoretic tools in cryptography.

Secure Computation

This course will cover basic results on secure computation. Starting with Chapter 2, we will first cover secret sharing (Chapter 11), which is a basic tool for secure computation. Then, we will proceed to the two-party secure computation problem (Chapter 12). We will move then to the oblivious transfer (Chapter 13) and the bit commitment (Chapter 14) problems, but will not cover constructions as that would require prerequisites from earlier chapters. After that, we will cover some selected topics from Chapters 15–19. There are two possible paths. In the first one, we will highlight the completeness of oblivious transfer (Chapter 17) after covering the notions of active adversary, composability (Chapter 15), and zero-knowledge proofs (Chapter 16). In the second one, we will highlight the honest majority/supermajority threshold of multiparty secure computation (Chapter 19). This requires as prerequisites the notions of active adversary and composability (Chapter 15, if the active adversary is discussed) and broadcast (Chapter 18, which may be omitted if we assume existence of the broadcast). Selection of topics depends on the preference of the instructor; however, multiparty secure computation (Chapter 19) may be easier to digest compared to the completeness of oblivious transfer (Chapter 17).

Each chapter is supplemented by several problems. Some of these are just exercises to confirm results provided in the chapter or to fill omitted steps of proofs. Others are meant to be pointers to interesting results that are beyond the scope of the book. For some selected problems (mainly for those that are used in later chapters), answers are provided in the Appendix.

1.5 References and Additional Reading

The topic of cryptography is as classic as it is popular. There are already several excellent textbooks that provide a review of different aspects of this vast area. We review some of them below. For a historical perspective of cryptography and information theory, see [310, 313].

Most existing textbooks on cryptography are based on computational complexity theory. A thorough exposition can be found in the two volume textbook

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press

by Goldreich [141, 142]. A more introductory but rigorous treatment can be found in the textbook by Katz and Lindell [187]. In addition to these general textbooks, there are textbooks specializing in specific topics, such as two-party secure computation by Hazay and Lindell [166] or digital signatures by Katz [185].

Another important topic of cryptography is algebraic construction and practical implementation of cryptographic primitives. Textbooks covering these topics include those by Blahut [34] and Stinson [318]. In particular, the former is written by an information theorist with engineers in mind.

Some topics of information-theoretic cryptography, such as encryption or the secret key agreement, are treated in information theory textbooks. For instance, see the textbooks by Csiszár and Körner [88] and El Gamal and Kim [131]. A popular application of information theoretic-cryptography is physical layer security; a good resource on this topic is the textbook by Bloch and Barros [37]. In a different flavor, the book by Cramer, Damgård, and Nielsen contains a thorough treatment of information-theoretically secure multiparty computation and secret sharing [77].

Recent development of information-theoretic cryptography is closely tied to quantum information science. The classic textbook by Nielsen and Chuang broadly covers quantum computation, quantum cryptography, and quantum information theory [261]. For a more thorough treatment of quantum information theory, see the textbooks by Hayashi [161] and Wilde [348]. These books also treat problems related to converting one type of randomness to another.

https://doi.org/10.1017/9781108670203.002 Published online by Cambridge University Press