Chapter 3

Shannon's Theory of Secrecy

3.1 Introduction to attack and security assumptions

After an introduction to some basic encryption schemes in the previous chapter we will in the sequel try to explain modern theory for the design of cryptographic primitives. The starting point is to give a more thorough treatment of possible attack scenarios. We have a symmetric encryption system in mind, according to the model described in the previous chapter. We list a number of possible attacks.

Ciphertext-only attack: Eve (the enemy) is assumed to have access to the ciphertext $\mathbf{c}$. The target for Eve would be to try to recover the secret key k, the plaintext $\mathbf{m}$, or possibly some partial information about the plaintext. This is the weakest form of an attack that we consider.

Known-plaintext attack: Eve knows the plaintext and the ciphertext. She tries to recover the secret key k. Alternatively, she might know a part of the plaintext together with the full ciphertext, and she tries to recover the unknown part of the plaintext (or just some partial information about it). It should be noted that a known-plaintext attack is a basic attack that all cryptographic primitives should be resistant against. For example, Eve should not be able to recover the key from a known plaintext and the corresponding ciphertext. It is very often the case in real applications that some data known to Eve is encrypted. So the known-plaintext attack is not unrealistic at all.

Chosen-plaintext attack: Not only does Eve know the plaintext but she is able to choose it herself. She gets the corresponding ciphertext. Her target is to recover the secret key k. As before, we can also consider an attack when she chooses a part of the plaintext and tries to recover some other unknown part of it. This attack scenario is of course less realistic compared to the attacks mentioned before. However, there are several applications when this is also a realistic attack. One such example is when Eve is attacking a primitive implemented in a protected device (e.g. smart card). She might have access to the device and can give arbitrary input and observe the output.

Chosen-ciphertext attack: Eve is assumed to have access to a decryption algorithm and can feed it with an arbitrary ciphertext, observing the corresponding plaintext. Again the prime target is to recover k.

We sometimes consider also *related-key attacks*. Here plaintexts have been encrypted by different keys that are almost identical (they may differ in only one bit). Eve tries to recover one of the keys. An important scenario in certain applications is the area of *side-channel attacks*. Here Eve has access to a device performing encryption or decryption and gets additional information from side channels. This could be how the device consumed power during its execution, the time it takes the devise to perform different operations, etc. Eve is attacking the implementation of an algorithm and not the algorithm itself. However, the possibilities of a secure implementation depend a lot on the algorithm itself.

The second topic to consider is what we mean by a secure primitive. As we have just described, the security is related to the attack scenario we consider. However, the security can be of different kind.

Unconditional security: A cryptographic primitive is said to be unconditionally secure if we can prove that it cannot be broken even if Eve have infinite computational resources. This is the strongest kind of security we can achieve. Note that under this assumption Eve is allowed to do an *exhaustive key search*, i.e., exhaustively try all possible keys.

Computational security: A cryptographic primitive is said to be computationally secure if we can prove that the best algorithm for breaking it requires at least T operations, where T is some large fixed number. It is a very rare event that a cryptosystem can be proved secure under this assumption.

Provable security: A cryptographic primitive is said to be provably secure if its security can be reduced to some well-studied problem. This means that breaking the primitive implies that we can solve the well-studied problem. For example, a primitive might have been proved secure provided that an integer n cannot be factored. The security have then been reduced to the factoring problem. As long as we cannot factor large integers, the primitive is secure. The way we reduce one computational problem to another is an interesting topic and we will come back to that later.

Heuristic security: If there is no known method of breaking the primitive but we cannot prove the security in any sense. Actually, most ciphers used today have a security of this kind.

After this introduction of attack scenarios and different kinds of security for a primitive, we move to the first study of cryptographic primitives. In 1949 Claude Shannon published his paper on secrecy systems entitled “Communication Theory of Secrecy Systems”. This was the first formal treatment of the secrecy problem in open litterature. For the first time a cipher could be proved to be secure.

It is important to note that Shannon considered only ciphertext-only attacks and considered only unconditional security, i.e., Eve is assumed to have infinite computing power. This combination is best studied through probability theory and information theory.

3.2 Information theory

First a few basics from probability theory. Consider an experiment or trial of some kind, resulting in some outcome. The set of all possible outcomes is called the *sample space* of the experiment, denoted Ω.

Let Ω be finite,

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}.$$

The elements ω_i in Ω are called *elementary events*. An *event* E is simply any subset of Ω. We assume that the elementary events have an associated probability measure $P(\omega_i)$ such that $0 \leq P(\omega_i) \leq 1$, $\sum_{i=1}^{n} P(\omega_i) = 1$. The *probability* of an event E is given by

$$P(E) = \sum_{\omega \in E} P(\omega).$$

A discrete random variable X takes values from a finite set $\mathcal{X}$. It is a mapping from $X(\omega) : \Omega \longrightarrow \mathcal{X}$. It has a *probability distribution* $P(X)$ where the notation $P(X = x)$ means the probability that the random variable X takes the value $x \in \mathcal{X}$. The probability $P(X = x)$ is given as

$$P(X = x) = \sum_{\omega : X(\omega) = x} P(\omega).$$

We use the notation $P(x)$ rather than $P(X = x)$ for convenience. Obviously $P(x) \geq 0$ for all $x \in \mathcal{X}$ and

$$\sum_{x \in \mathcal{X}} P(x) = 1.$$

Let $E \subset \mathcal{X}$. Then $X \in E$ is an event and its probability is calculated as

$$P(X \in E) = \sum_{x \in E} P(x).$$

Example 3.1. *Let X denote the outcome when throwing a die. Then $\Omega = \mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ and $P(X = x) = 1/6$ for all $x \in \mathcal{X}$.*

Consider the event of throwing an even number, i.e., $E = \{2, 4, 6\}$. Its probability is $P(X \in E) = \sum_{x \in E} P(x) = 1/2$.

Note that a random variable does not need to take on numerical values. Considering the example above, we can define a new random variable Y taking values from $\mathcal{Y} = \{\text{odd}, \text{even}\}$. Still, $\Omega = \{1, 2, 3, 4, 5, 6\}$ and $Y(1) = \text{odd}$, $Y(2) = \text{even}$, ..., $Y(6) = \text{even}$ giving

$$P(Y = \text{odd}) = \sum_{\omega : Y(\omega) = \text{odd}} P(\omega) = 1/2,$$

etc.

A pair of random variables X, Y defined on the same sample space can be considered as a single random variable, say $Z = (X, Y)$. The random variable Z takes values in $\mathcal{X} \times \mathcal{Y}$, with $Z(\omega) = (X(\omega), Y(\omega))$. Usually, we want to keep the X and Y variables visible, so we write $P(X, Y)$, etc., without introducing a new random variable.

A similar reasoning can be made for events E and F. We can consider the joint event G that both E and F occur. Then G corresponds to the event $G = E \cap F$. The notation $P(E, F)$ is defined as $P(E \cap F)$.

Two events E and F are said to be *independent* if

$$P(E, F) = P(E)P(F).$$

Correspondingly, two random variables X, Y are said to be independent random variables if

$$P(X = x, Y = y) = P(X = x)P(Y = y), \forall x \in \mathcal{X}, y \in \mathcal{Y}.$$

Next we introduce conditional probabilities.

Definition 3.1. *The conditional probability $P(E|F)$ is defined as*

$$P(E|F) = \frac{P(E,F)}{P(F)},$$

assuming $P(F) \neq 0$.

For random variables we have a similar definition of $P(X|Y)$ as

$$P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}.$$

We will now introduce the concept of entropy, which is a measure of uncertainty of a random variable.

Definition 3.2. *The* entropy $H(X)$ *of a discrete random variable X is defined as*

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log P(x).$$

The log is to the base 2 and entropy is expressed in bits. Also, we use the convention that $0 \log 0 = 0$, which is easily justified since $x \log x \to 0$ as $x \to 0$.

Recall that the expectation $\mathbb{E}(F(X))$ of a function $F(X)$ is defined as

$$\mathbb{E}(F(X)) = \sum_{x \in \mathcal{X}} P(x)F(x).$$

Clearly, an alternative way of expressing the entropy is

$$H(X) = \mathbb{E}(-\log P(X)).$$

We can see that the actual values of X are not used in the calculation of the entropy, only its probability distribution. This means that the entropy of X can be calculated even if X does not take on numerical values.

The interpretation of entropy can be viewed as some kind of uncertainty about the outcome of the random variable. If X takes one value with probability 1 and other values with probability 0, then the entropy is 0 bits. There is no uncertainty since we know what value X will take.

If X takes on two possible values, both with probability 1/2, then the entropy is 1 bit. If X takes on four possible values, all with probability 1/4, then the entropy is 2 bit, and so on.

If we consider (X, Y) as one random variable we get

$$H(X,Y) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(x,y) \log P(x,y).$$

We need to define conditional entropy.

Definition 3.3. *The* conditional entropy $H(X|Y)$ *of a discrete random variable* X*, conditioned on another discrete random variable* Y *is defined as*

$$H(X|Y) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} P(x,y)\log P(x|y).$$

Note that this is equivalently expressed as $H(X|Y) = \mathbb{E}(-\log P(X|Y))$. If $P(y) \neq 0$ we can introduce the notation

$$H(X|Y=y) = -\sum_{x\in\mathcal{X}} P(x|y)\log P(x|y).$$

It is then straightforward to derive the expression

$$H(X|Y) = \sum_{y\in\mathcal{Y}} P(y)H(X|Y=y).$$

This may be a convenient way to calculate $H(X|Y)$.

The entropy function has several important properties.

Theorem 3.1. *If* X *is a random variable taking values in the set* $\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{X}|}\}$ *then*

$$0 \leq H(X) \leq \log|\mathcal{X}|.$$

Furthermore, $H(X) = 0$ *if and only if* $P(x) = 1$ *for some* $x \in \mathcal{X}$*; and* $H(X) = \log|\mathcal{X}|$ *if and only if* $P(x) = 1/|\mathcal{X}|$ *for all* $x \in \mathcal{X}$*.*

A similar property holds for the conditional entropy.

Theorem 3.2. *If* X *is a random variable taking values in the set* $\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{X}|}\}$ *then*

$$0 \leq H(X|Y) \leq \log|\mathcal{X}|.$$

Furthermore, $H(X|Y) = 0$ *if and only if for every* y $P(x|y) = 1$ *for some* $x \in \mathcal{X}$*; and* $H(X|Y) = \log|\mathcal{X}|$ *if and only if for every* y $P(x|y) = 1/|\mathcal{X}|$ *for all* $x \in \mathcal{X}$*.*

Also for $H(X|Y=y)$ similar inequalities hold.

Consider a number of random variables $X_1, X_2, \ldots X_n$. The following chain rule is often very useful.

Theorem 3.3.

$$H(X_1X_2\ldots X_n) = H(X_1) + H(X_2|X_1) + H(X_3|X_1X_2) + \cdots H(X_n|X_1X_2\ldots X_{n-1}).$$

In particular, $H(XY) = H(X) + H(Y|X) = H(Y) + H(X|Y)$.

Finally, the following inequality shows that the uncertainty of a random variable X can never increase by knowledge of the outcome of another random variable.

Theorem 3.4.

$$H(X|Y) \leq H(X)$$

with equality if and only if X *and* Y *are independent.*

The inequality leads to the fact that

$$H(XY) \leq H(X) + H(Y),$$

again with equality if and only if X and Y are independent.

Consider $H(X)$ in the case $|\mathcal{X}| = 2$. There are two possible values, one with probability p and the other with probability $1-p$. This case is so common that the entropy function has received a notation of its own,

$$h(p) = -p \log p - (1-p) \log(1-p).$$

We will now introduce the concept of mutual information. To start with, we define the *relative entropy* $D(P(X)||Q(X))$ between two probability distributions $P(X)$ and $Q(X)$ as

$$D(P(X)||Q(X)) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}.$$

This can equivalently be written as

$$D(P(X)||Q(X)) = \mathbb{E}_P(\log \frac{P(x)}{Q(x)}).$$

Note that the expectation is taken over $P(X)$ and in general $D(P(X)||Q(X)) \neq D(Q(X)||P(X))$. The relative entropy is a measure of the distance between two distributions. It can be thought of as the inefficiency of assuming distribution $Q(X)$ when the correct distribution is $P(X)$.

Definition 3.4. *The* mutual information $I(X;Y)$ *between random variables* X *and* Y *is defined as*

$$I(X;Y) = D(P(X,Y)||P(X)P(Y)),$$

or equivalently

$$I(X;Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}.$$

The mutual information $I(X;Y)$ measures the information (in bits) we receive about the random variable X when observing the outcome of the random variable Y. But it also describes the information we receive about the random variable Y when observing the outcome of the random variable X. Hence the name mutual information and the property $I(X;Y) = I(Y;X)$. Some very important properties for the mutual information are summarized in the following theorem.

Theorem 3.5.

$$\begin{aligned}
I(X;Y) &= H(X) - H(X|Y), && (3.1)\\
I(X;Y) &= H(Y) - H(Y|X), && (3.2)\\
I(X;Y) &= H(X) + H(Y) - H(X,Y), && (3.3)\\
I(X;Y) &= I(Y;X), && (3.4)\\
I(X;X) &= H(X). && (3.5)
\end{aligned}$$

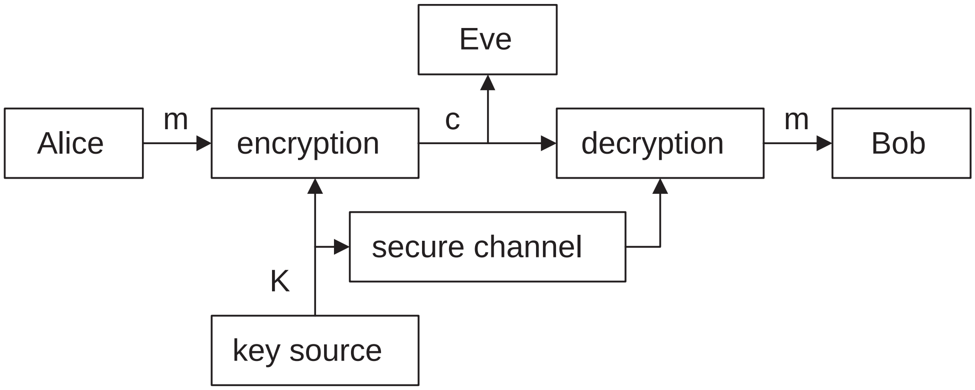

Figure 3.1: Shannons model of a secrecy system

Finally we define the *conditional mutual information* $I(X;Y|Z)$ in an analogue way,

$$I(X;Y|Z) = D(P(X,Y|Z)||P(X|Z)P(Y|Z)),$$

where

$$D(P(X|Z)||Q(X|Z)) = \sum_{z \in \mathcal{Z}} P(z) \sum_{x \in \mathcal{X}} P(x|z) \log \frac{P(x|z)}{Q(x|z)}$$

is the *conditional relative entropy*. Clearly,

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) = H(Y|Z) - H(Y|X,Z),$$

etc. We end by giving the inequality

Theorem 3.6.

$$I(X;Y) \geq 0,$$

with equality if and only if X and Y are independent.

3.3 Shannons theory of secrecy

Shannons model of a secrecy system is essentially the model we saw in Chapter 2. The model appears again in Figure 3.3. We consider a given set of encryption functions, one for each key k, mapping a sequence of plaintext letters $\mathbf{m} = m_1, m_2, \ldots, m_i \in \mathcal{M}$ to a sequence of ciphertext letters $\mathbf{c} = c_1, c_2, \ldots, c_i \in \mathcal{C}$. If not otherwise stated, we assume that the plaintext and ciphertext letters are from the same alphabet.

Eve has access to the ciphertext $\mathbf{c}$ and her task is to obtain some information about either the transmitted message (plaintext) or the key k used by Alice and Bob. For a length N sequence, let $\mathbf{M} = (M_1, M_2, \ldots, M_N)$ be the random variable corresponding to the plaintext Alice is sending, and let $\mathbf{C} = (c_1, c_2, \ldots, c_N)$ be the the random variable corresponding to the ciphertext Bob is receiving. Also, $K \in \mathcal{K}$ is a random variable representing the chosen key.

We introduce the *key entropy*

$$H(K) = -\sum_{k \in \mathcal{K}} P(k) \log P(k),$$

and the *message entropy*

$$H(\mathbf{M}) = -\sum_{\mathbf{m} \in \mathcal{M}^N} P(\mathbf{m}) \log P(\mathbf{m}).$$

Clearly, we have

$$H(K) \leq \log |\mathcal{K}|,$$

and

$$H(\mathbf{M}) \leq \log |\mathcal{M}|^N.$$

The key entropy describes the uncertainty Eve faces regarding the unknown key a priori (i.e. without having observed the transmitted ciphertext). Similarly, the message entropy describes the uncertainty regarding the transmitted message. Now, Eve observes the transmissed ciphertext. The remaining uncertainty is described by the *key equivocation*

$$H(K|\mathbf{C}) = -\sum_{k \in \mathcal{K}, \mathbf{c} \in \mathcal{C}^N} P(k, \mathbf{c}) \log P(k|\mathbf{c}),$$

and the *message equivocation*

$$H(\mathbf{M}|\mathbf{C}) = -\sum_{\mathbf{m} \in \mathcal{M}^N, \mathbf{c} \in \mathcal{C}^N} P(\mathbf{m}, \mathbf{c}) \log P(\mathbf{m}|\mathbf{c}).$$

From Theorem 3.4 we have

$$H(K|\mathbf{C}) \leq H(K),$$

and

$$H(\mathbf{M}|\mathbf{C}) \leq H(\mathbf{M}).$$

Our uncertainty about the key and the message can never increase by observing the ciphertext.

A *nonprobabilistic* encryption scheme is an encryption function for which every plaintext message is mapped to a unique ciphertext under a fixed key. Most encryption functions are nonprobabilistic.

Theorem 3.7. *For a nonprobabilistic encryption scheme we have*

$$H(\mathbf{M}|\mathbf{C}) \leq H(K|\mathbf{C}).$$

Proof. The expression $H(K, \mathbf{M}|\mathbf{C})$ can written

$$H(K, \mathbf{M}|\mathbf{C}) = H(K|\mathbf{C}) + H(\mathbf{M}|\mathbf{C}, K).$$

When key and ciphertext are given, the plaintext is uniquely determined, since we consider only nonprobabilistic encryption schemes. So $H(\mathbf{M}|\mathbf{C}, K) = 0$.

Since $H(\mathbf{M}|\mathbf{C}) \leq H(K, \mathbf{M}|\mathbf{C})$ we get $H(\mathbf{M}|\mathbf{C}) \leq H(K|\mathbf{C})$. □

Let $|\mathcal{M}| = |\mathcal{C}| = L$. The maximum entropy of the considered alphabet $H_0 = \log L$ is sometimes called the *rate of the alphabet*.

The actual entropy of the message source per alphabet symbol is denoted by $H(M)$ and given by

$$H(M) = \frac{H(\mathbf{M})}{N},$$

for a length N message $\mathbf{M} = (M_1, M_2, \ldots, M_N)$.

There are two cases to consider. The first one is for memoryless source. Then

$$H(\mathbf{M}) = H(M_1) + H(M_2|M_1) + \cdots + H(M_N|M_1, \ldots, M_{N-1}),$$

and due to the memoryless property

$$H(\mathbf{M}) = H(M_1) + H(M_2) + \cdots + H(M_N) = NH(M_1).$$

Finally $H(M) = H(M_1)$, i.e., it is enough to find the uncertainty of a single message symbol.

The second case is when the source is not memoryless. Then we use the expression $H(M) = H_\infty$, where H_∞ is the entropy per letter defined by

$$H_\infty = \lim_{N\to\infty} \frac{H(M_1, M_2, \ldots, M_N)}{N}.$$

For English, the entropy per letter can be determined experimentally to be around $H_\infty = 1.5$.

Definition 3.5. *We define the* redundancy *of a source, denoted D, to be*

$$D = H_0 - H(M).$$

The redundancy is an important characterization of a source, since it describes how much a source can be compressed. For English, the redundancy is

$$D = \log 26 - 1.5 = 3.2 \text{ bits}.$$

This means that out of the 4.7 bits needed to represent a letter only 1.5 bits is necessary for unique representation. The remaining 3.2 bits can be removed. In more practical terms, a file of 1000 letters English text would require 4700 bits if represented in a basic form (or 5000 bits if each possible letter corresponds to a 5 bit pattern). The above arguments shows that 1500 bits is theoretically enough to represent the text (by clever and complex encoding). The remaining 3200 bits is the redundancy that “can be removed”.

One of the main results of Shannon was that he showed that due to the redundancy in a source, cryptosystems can be broken.

Theorem 3.8.

$$H(K|\mathbf{C}) \geq H(K) - ND.$$

Proof. Since $H(K, \mathbf{M}, \mathbf{C}) = H(K, \mathbf{M}) = H(K, \mathbf{C})$ (unique encryption/decryption) and

$$H(K, \mathbf{M}) = H(K) + H(\mathbf{M})$$

we get on one hand

$$H(K, \mathbf{M}) = H(K) + NH(M),$$

and on the other hand

$$H(K, \mathbf{C}) = H(\mathbf{C}) + H(K|\mathbf{C}) \leq NH_0 + H(K|\mathbf{C}).$$

Combining these expressions gives

$$H(K|\mathbf{C}) \geq NH(M) + H(K) - NH_0 = H(K) - ND.$$

□

From the theorem we see that when the length of the encrypted message is more than $H(K)/D$ the right hand side becomes 0, i.e., the uncertainty about the key might be close to zero. This leads to the following definition.

Definition 3.6. *The* unicity distance *denoted N_0 is defined as*

$$N_0 = H(K)/D.$$

Let us comment shortly on this definition. If the message length is longer than N_0 the inequality in Theorem 3.8 becomes $H(K|\mathbf{C}) \geq 0$, i.e., the uncertainty about the key for Eve *could* be very small, but it does not necessarily have to be so.

If, on the other hand, the message length is shorter than N_0, this means that Eve faces some uncertainty about the key, i.e, there are many key values that could have generated the observed ciphertext. But this does not mean that there is an uncertainty about the message. It could be that all possible keys would decrypt to the same message.

Even though it is hard to fully describe the meaning of *unicity distance*, from a practical point of view it gives a rough borderline between the case when there are several possible solutions and the case when there is only one possible solution for the key or the message.

Example 3.2. *Consider a simple substitution cipher encrypting an English source. We have $H(K) = \log 26! = 88.4$ bit. Using $D = 3.2$ we get $N_0 = 88.4/3.2 \approx 28$ letters. The interpretation is that if we know more than 28 letters there is probably a unique value for the key giving a meaningful message. All other key values give messages that are not meaningful.*

Example 3.3. *A Vernam cipher has key size $H(K) = NH_0$. Since $D < H_0$ we get $N_0 > N$, i.e., the unicity distance is larger than the message length.*

Having demonstrated the fact that redundancy helps Eve in her cryptanalytic attempts, we come to two conclusions regarding services provided by a communication system.

- Compression should be done before encryption and it improves the security of a cryptosystem.

- Channel coding (adding parity bits for correction/detection of errors) should be done after encryption.

So the correct order of services is first to compress the source, then to encrypt the resulting data, and finally to perform channel coding.

We can strengthen the notion of an unbreakable system as follows.

Definition 3.7. *A cryptosystem is said to have* perfect secrecy *if*

$$I(\mathbf{M}, \mathbf{C}) = 0.$$

In a system with perfect secrecy, the plaintext and the ciphertext are independent. When Eve observes the ciphertext, she obtains no information at all about the plaintext ($H(\mathbf{M}|\mathbf{C}) = H(\mathbf{M})$). This is in some sense the strongest notion of security we can hope for. Unfortunately, a cryptosystem with perfect secrecy has a severe drawback, making it useless for practical purposes.

Theorem 3.9. *For a cryptosystem with perfect secrecy we have*

$$H(\mathbf{M}) \leq H(K).$$

Proof. We know that

$$H(\mathbf{M}|\mathbf{C}) \leq H(K|\mathbf{C}) \leq H(K).$$

But the definition of perfect secrecy implies $H(\mathbf{M}|\mathbf{C}) = H(\mathbf{M})$, so the theorem follows. □

In a system with perfect secrecy we need the key size to be at least as large the size of the plaintext (after compression). As most applications consider plaintexts of large sizes (like encrypting a file), this would lead to huge key sizes which is practically impossible to handle.

We end this chapter with the following result.

Theorem 3.10. *The Vernam cipher has perfect secrecy.*

Proof. We prove the result for the binary alphabet. For the Vernam cipher we have

$$\mathbf{C} = \mathbf{M} + \mathbf{K},$$

where $\mathbf{M} = (M_1, M_2, \ldots, M_N)$ and $\mathbf{K} = (K_1, K_2, \ldots, K_N)$, $M_i, K_i \in \mathbb{F}_2, i = 1, 2, \ldots, N$. Whereas $\mathbf{M}$ may have any distribution, $\mathbf{K}$ is uniformly distributed. Thus

$$I(\mathbf{M}; \mathbf{C}) = \sum_{\mathbf{m}} \sum_{\mathbf{c}} P(\mathbf{m}, \mathbf{c}) \log \frac{P(\mathbf{c}|\mathbf{m})}{P(\mathbf{c})} =$$

$$= \sum_{\mathbf{m}} \sum_{\mathbf{c}} P(\mathbf{m}, \mathbf{c}) \log \frac{1/2^N}{1/2^N} = 0.$$

□

Recall our discussion in the introduction of the chapter. The Vernam cipher is proved secure in the strongest possible sense, but this is only valid for the cipertext-only assumption.

3.4 Exercises

Exercise 3.1. *Consider the sample space as the outcome when throwing a die. Then* $\Omega = \{1, 2, 3, 4, 5, 6\}$. *Consider three different random variables* $X, Y; Z$ *defined on* Ω, *where* X *is the received number,* Y *takes on two possible values, either "EVEN" or "ODD", depending on whether an even or an odd number is received. Finally,* Z *also takes on two possible values, either "LOW" or "HIGH", where "LOW" is obtained if the received number is at most 3.*

a) *Calculate*

1. H(X)
2. H(Y)
3. H(Z)
4. H(XY)
5. H(XZ)
6. H(YZ)
7. H(XYZ)

b) *Calculate*

1. I(X;Y)
2. I(X;Z)
3. I(Y;Z)
4. I(X;YZ)

Exercise 3.2. *Assume that* X_1, X_2, X_3 *are three random variables such that* $P(X_1 = x_1, X_2 = x_2, X_3 = x_3) = 1/4$ *if* $(x_1, x_2, x_3) \in \{(000), (011), (101), (110)\}$, *and zero otherwise. Calculate*

a) $H(X_1)$
b) $H(X_1X_2)$
c) $H(X_2|X_1)$
d) $H(X_1X_2X_3)$
e) $H(X_3|X_1X_2)$
f) $H(X_3)$
g) $I(X_1; X_3)$
h) $I(X_1X_2; X_3)$

Exercise 3.3. *Assume that* X_1, X_2, X_3 *are three random variables such that* $P(X_1 = x_1, X_2 = x_2, X_3 = x_3) = 1/5$ *if* $(x_1, x_2, x_3) \in \{(000), (001), (010), (100), (111)\}$, *and zero otherwise. Calculate*

a) $H(X_1)$
b) $H(X_2)$
c) $H(X_3)$
d) $H(X_2|X_1)$
e) $H(X_1X_2)$
f) $H(X_3|X_1X_2)$
g) $H(X_1X_2X_3)$
h) $H(X_2|X_1 = 0)$
i) $H(X_2|X_1 = 1)$

Exercise 3.4. *Let* $X \in \{rain, sunshine\}$ *be a produced weather forecast and let* $Y \in \{rain, sunshine\}$ *be the actual weather. The joint distribution is* $P(X = rain, Y = rain) = 1/4$,$P(X = rain, Y = sunshine) = 1/2$, $P(X = sunshine, Y = rain) = 0$,$P(X = sunshine, Y = sunshine) = 1/4$.

Examining this situation, you can see that the forecast is only correct with probability 0.5. *By just always predicting sunshine you can increase the probability of a correct weather forecast to* 0.75. *Use information theoretical arguments to decide whether the latter is a better forecast.*

Exercise 3.5. *Continuing the previous exercise, assume instead that* $P(X = rain, Y = sunshine) = 1/2$ *and* $P(X = sunshine, Y = rain) = 1/2$, *and zero otherwise. What can you say about the forecast in this case?*

Exercise 3.6. *Assume that Bob has two coins, one genuine with heads and tail; and one with heads on both sides. He randomly select one of the coins and tells us the outcome of two coin tosses. Determine how much information we get about which coin he selected from the outcome of the coin tosses.*

Exercise 3.7. *Prove that*

$$I(\mathbf{M}; \mathbf{C}) \geq H(\mathbf{M}) - H(K),$$

i.e., a cryptosystem with small key size will leak a lot of information about the plaintext.

Exercise 3.8. *Determine the unicity distance* N_0 *as a function of* n *for a Vigenére cipher with period* n *encrypting English text. What can you say about the security if we encrypt twice using two Vigenére encryptions with different keys but the same period?*

Exercise 3.9. *Determine the unicity distance* N_0 *for a Playfair cipher encrypting English text. In a Playfair cipher, the alphabet size is 25 and the key is an arbitrary permutation of the alphabet.*