Lecture Notes for CS-578 Cryptography (SS2006) Prof. Michael Backes

1. Historical Encryption Schemes and the One-time Pad

Lecture 1 Saarland University

In this chapter we examine some historic schemes and show how to break them quite easily. We furthermore introduce some basic notation to set up the ground for future studies of more evolved encryption schemes. More information can be found in the beautiful book *The Codebreakers* by David Kahn, which contains a very nice overview of the history of cryptography up to World War II).

1.1 Preliminaries

Let us first fix some important principles of secure encryption. There have been (and still are) encryption schemes where the algorithm is kept secret by the company/inventor. This is called *security by obscurity* and should be avoided under any circumstances: While the inventor usually claims that the system is secure, the user has no (easy) way to verify this claim. The algorithm might be flawed, and an attacker might invest enough time to break the scheme; in fact reverse-engineering crypto algorithms with nowadays tools is usually not that hard. Also, the inventor might have built in some sort of *trap-door*, which enables him to read all encrypted messages.

Thus a common design criterion for every modern cryptosystem is that the algorithms used for encryption and decryption are publicly known, and that the security of the scheme only relies on the secrecy of a short secret called the key. This is called *Kerkhoff's Principle*.

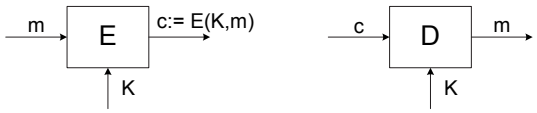

Figure 1.1: Basic Cipher

Let us conclude this brief overview by pointing out which parts an encryption system consists of; a depiction is given in Figure 1.1. For the next weeks we will deal with *symmetric encryption* only, i.e., both the *sender* (usually called Alice) and the receiver (usually called Bob) of a secret message share a common secret key K. There is an "efficient" algorithm E, called *encryption (algorithm)*, which takes the *key* and a *plaintext* and outputs a *ciphertext*, and there is an "efficient" algorithm D, called *decryption (algorithm)*, which takes the *key* and a *ciphertext* and outputs a plaintext. The notion of efficiency will be formally introduced later. We require that the decryption of a ciphertext yields the original message again. This property is called *correctness* of the encryption scheme.

Usually we denote symmetric keys by $K, K_1, K_2, \ldots$, messages by $m, m_1, m_2, \ldots$, and ciphertexts by $c, c_1, c_2, \ldots$. Sometimes we will denote plaintext messages with lower-case letters, i.e., `secret message`, and ciphertext messages with upper-case letters, i.e., `KAJFUG MSJFTOA`.

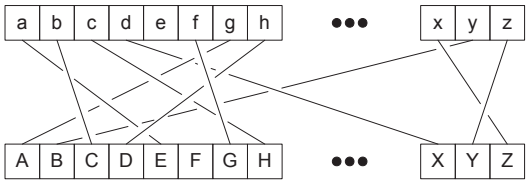

Figure 1.2: Substitution Cipher

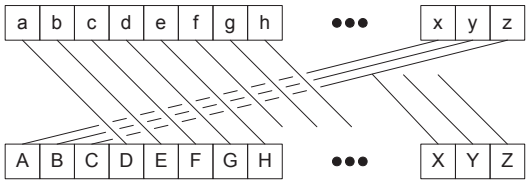

Figure 1.3: Shift-Cipher

1.2 Historic Ciphers

1.2.1 Substitution Cipher

The *substitution cipher* is the oldest and one of the most famous ciphers in history and literature, and it comes in several variants. For simplicity we assume a text to consist of letters only, removing any spaces and punctuation. We also assume that the substitution alphabet, i.e., the symbols the ciphertext consists of, are letters as well.

Then a key K of the substitution cipher is a permutation of the set $\{a, b, \ldots, z\}$, cf. Figure 1.2. A message $m = m_1||m_2||\ldots||m_q$ is encrypted by computing

$$c = K(m_1)||K(m_2)||\ldots||K(m_q),$$

where the m_i are single letters and where $||$ denotes concatenation of strings.

As K is a permutation there exists the (unique) inverse permutation K^{-1}. This is used to decrypt a given ciphertext by the following rule: For $c = c_1||\ldots||c_q$, compute

$$m' = K^{-1}(c_1)||\ldots||K^{-1}(c_q).$$

It is easy to see that this scheme is *correct*, i.e., that decryption of an encryption (with the same key) yields the message again. Formally,

$$\begin{aligned} D(K, E(K, m)) &= D(K, K(m_1)||\ldots||K(m_q)) \\ &= K^{-1}(K(m_1))||\ldots||K^{-1}(K(m_q)) \\ &= m_1||\ldots||m_p \\ &= m. \end{aligned}$$

A small example based on the key K shown in Figure 1.2: if the plaintext `abc` is encrypted under this key, the ciphertext becomes `ECH`.

Cryptanalysis The conceptionally simplest attack on any encryption scheme is the so-called *brute-force attack*, where each possible key is used to decrypt a ciphertext and the resulting message is investigated. Thus the practicability of this attack relies precisely on the size of the key-space. For the substitution cipher the size of the key-space is $26! \approx 2^{86}$, so a brute-force attack on a substitution cipher is clearly impractical.

However, using statistical attacks, the substitution cipher can be broken quite easily. The main observation is that letters in average text do not occur with the same probability. In English text, for example, the letter "e" appears with probability approx. 12%, while "z" has probability below

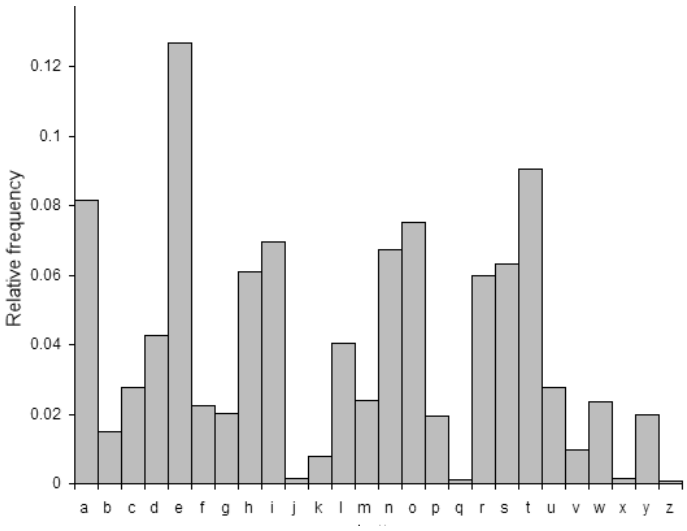

Figure 1.4: Letter frequencies in english Text

1%, cf. Figure 1.4. So a reasonable guess is that the most common letter in the ciphertext is the encryption of "e", or at least of one of the most common letters. Iterating this step will finally yield the plaintext. Of course letter frequencies depend on the writer of a text, on the topic, and so on, but in practice a text of a certain (not too small length) is easily breakable.

1.2.2 Shift-Ciphers and Caesar's Cipher

A special form of the substitution cipher is the shift-cipher. There the permutation K is chosen from a restricted set, namely by shifting the letters by a fixed number of positions in the alphabet. Given a number $\alpha_K \in \{0, \ldots, 25\}$, the permutation K maps each letter to its α_K'th successor, "wrapping" around after the Z and starting with A again if necessary (Figure 1.3). If one identifies $\{a, \ldots, z\}$ with $\{0, \ldots, 25\}$ then $K(\beta) = \beta + \alpha_K \bmod 26$. Julius Caesar used this cipher with a fixed value $\alpha_K = 3$, which is nowadays known as *Caesar's Cipher*. A variant is the so-called ROT-13 scheme, where $K_\alpha = 13$. This is sometimes used in Newsgroups and the Internet to prevent people from reading things by accident. For example, movie discussion boards "encrypt" details of the plot or the ending using ROT-13. This is not an encryption in a strict sense, however, the same mechanism is applied.

Cryptanalysis The size of the key-space is 26, rendering the Shift-cipher completely insecure. The plaintext can be recovered either by a brute-force attack testing each of the 26 keys and testing if the resulting plaintext makes any sense, or by frequency analysis as described above.

1.2.3 Vigenere Cipher

The Vigenere Cipher is a generalization of the Shift-Cipher. While the latter relies on a single permutation K to shift every letter by α_K positions, the Vigenere cipher uses n such permutations $K_0, \ldots, K_{n-1}$ to shift different letters by a different number $\alpha_{K_0}, \ldots, \alpha_{K_{n-1}}$ of positions. For $m =$

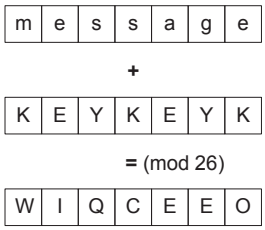

Figure 1.5: Vigenere Cipher

$m_0||\ldots||m_q$:

$$c_i := K_{i \bmod n}(m_i)$$

Decryption is done by shifting in the opposite direction:

$$m_i := K^{-1}_{i \bmod n}(c_i).$$

A small example is given in Figure 1.5.

Cryptanalysis The size of the keyspace is $26^n \approx 2^{4.7n}$, which becomes infeasible for brute-force attacks for moderate values of n. But again, statistical tests can be applied to break the scheme quite easily.

1.2.4 Rotor Machines and the Enigma

The Enigma was originally a commercial encryption machine that was later adopted by the military. A large number of variations of the Enigma machine exist; in the following we describe a generic model close to that used in the early days of World War II.

The machine comprises the following main elements. The *keyboard* contains letters from A to Z. In a first step, the letters from the keyboard are substituted using a *plug board*, which corresponds to the substitution cipher described above.

In the next step, the electric signal passes three *rotors*, each of which applies a fixed substitution to the letters, cf. Figure 1.6. If these rotors were static, this step would yield nothing than a substitution cipher again. However, after encrypting one letter, the rightmost rotor advances one step. After one full turn the middle rotor advances as well, and after one full turn of the middle rotor the leftmost rotor advances as well. Thus each letter is treated with a different substitution.

After passing the three rotors it enters the *reflector*. It permutes the letters again, but instead of passing it further it reflects the signal back to the same rotor, thus passing all three rotors in opposite direction, passing the plug-board one more time, and then reaching the *lamps*.

The reflector was invented to simplify decryption: The same setting can be used to decrypt the message again, as the signal passes the rotors in exactly the opposite direction. However, this is one of the main weaknesses of the Enigma, and one that helped to break it during World War II. Note that with the reflector, a letter is never encrypted to the same letter, as the three rotors perform a substitution and the reflector cannot "reflect" the signal back the same path through the rotors, as no electric circuit would be established then.

The key for the encryption consisted mainly of the permutation on the plug-board, the initial position of the rotors, and the order the rotors were placed into the machine.

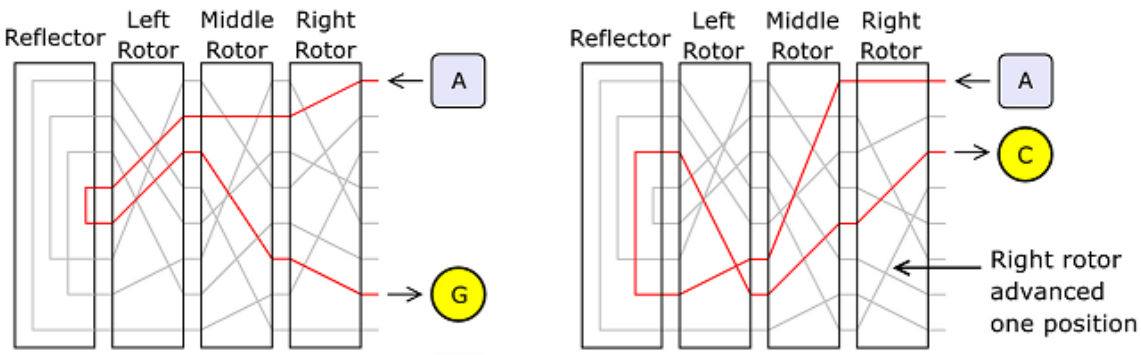

Figure 1.6: Functional description of the Enigma

Nowadays the Enigma can be easily broken by statistical analysis. The additional hardness was that cryptanalysts at that time neither knew the exact function of the Enigma or the wiring of the rotors. However, laziness of users facilitated this job, for example re-using the same key, using very weak keys (such as `AAA`), etc.

1.3 Classification of Attacks

In the previous section we implicitly assumed that the attacker intends to recover the plaintext given a ciphertext. While this is certainly a reasonable goal, there are other goals that are potentially easier to achieve, but should arguably be avoided as well. Apart from the goals an adversary pursues, there are different possibilities an attacker might exploit for attacking an encryption scheme, e.g., ciphertext-only attacks, or attacks where one assumes that certain plaintexts/ciphertext pairs are already known to the adversary. We will informally discuss the most common of these attack goals in the sequel.

1.3.1 Adversarial Goals

The following goals of an attacker are usually discussed:

- *Total break:* The attacker recovers the key that was used to encrypt ciphertexts. This will of course enable him to decrypt any ciphertext it gets.
- *Universal Break:* The attacker finds an alternative method to decrypt *any* ciphertext, without necessarily recovering the secret key.
- *Partial Break:* The attacker finds an alternative method to decrypt *some* distinguished ciphertexts.
- *Partial Information:* The attacker finds a method to compute *partial information* about the plaintexts given some ciphertext, e.g., individual bits, checksum, ...

Note that these attacks are listed in decreasing strength, in the sense that if an attacker can *universally break* an encryption scheme, then he can also *partially break* the scheme and compute

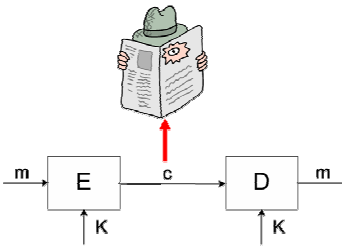

Figure 1.7: Known Cipher-text Attack

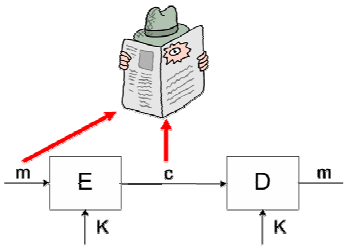

Figure 1.8: Known Plain-text Attack

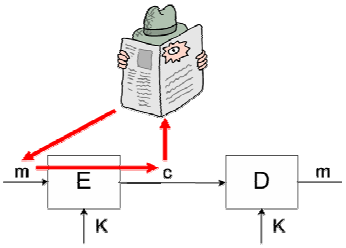

Figure 1.9: Chosen Plain-text Attack

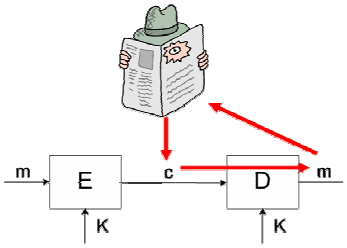

Figure 1.10: Chosen Cipher-text Attack

partial information, and so on. Note that in research, only the goal of preventing the adversary from obtaining any partial information is considered.

1.3.2 Adversarial Capabilities

The following adversarial capabilities are usually discussed:

- *Ciphertext-Only:* The adversary only sees one ciphertext (or a sequence of ciphertexts) encrypted with the (same) key, and he does not know the corresponding plaintext. This is illustrated in Figure 1.7.

- *Known-plaintext:* The adversary gets a sequence of paintext/ciphertext pairs encrypted using the same (unknown) key, and wants to attack an additional ciphertext whose corresponding plaintext is not known to him, cf. Figure 1.8.

- *Chosen Plaintext:* The adversary may choose a sequence of plaintext himself and get their encryptions using the same fixed key. After that, he wants to attack an additional ciphertext whose corresponding plaintext is not known to him, cf. Figure 1.9.

- *Chosen Ciphertext:* Similar to Chosen Plaintext, but the attacker may additionally choose ciphertexts and get their decryption under the considered key, cf. Figure 1.10. (This is the scenario typically considered in current research.)

1.4 Formal Definition of Ciphers

Before studying ciphers in more detail we start by giving a formal definition of symmetric encryption schemes.

Definition 1.1 (Symmetric Encryption Scheme) *A* symmetric encryption scheme *over* $(\mathcal{M}, \mathcal{C}, \mathcal{K})$ *is a tuple* (E, D) *where* E, D *are efficiently computable algorithms with* $\mathsf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$, $\mathsf{D} : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$, *such that for all* $K \in \mathcal{K}$ *and for all* $m \in \mathcal{M}$*:*

$$\mathsf{D}(K, \mathsf{E}(K, m)) = m.$$

$\diamond$

Here

- E is called the *encryption* function,
- D is called the *decryption* function,
- $\mathcal{M}$ is a set called the *message space*. It contains all possible plaintexts, with typical examples being $\{a, \ldots, z\}^*$ (all words over the alphabet $a, \ldots, z$ of finite length), or $\{0,1\}^*$ (bitstrings of finite length).
- $\mathcal{C}$ is a finite set called the *ciphertext space*. It contains all encryptions that might occur in the scheme.
- $\mathcal{K}$ is a finite set called the *keyspace*. For symmetric encryption keys are drawn uniformly random from $\mathcal{K}$. We have seen for example $\mathcal{K} = \mathbb{N}_{26}$, $\mathcal{K} = (\mathbb{N}_{26})^*$, $\mathcal{K} = \mathsf{Perm}(\{a, \ldots, z\}) = \{p : \{a, \ldots, z\} \to \{a, \ldots, z\} \mid p \text{ is bijective}\}$

1.5 The One-time Pad

1.5.1 Definition of the One-time Pad

The One-time Pad, also called Vernam-cipher, was invented in 1917. It was the first scheme for which a security proof was given, namely by Claude Shannon in 1949. (Before that, not even definitions of security existed.) It may be defined over a variety of message spaces, for example $\{a, \ldots, z\}$ or $\{0,1\}$. We define it over bitstrings.

Cryptosystem 1 (One-time Pad) *Let* $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$ *be bitstrings of a fixed length* n*. Both encryption and decryption are defined by the xor of the key with the plaintext or ciphertext, respectively. Formally:*

$$\begin{aligned} \mathsf{E}(K, m) &:= K \oplus m \\ \mathsf{D}(K, c) &:= K \oplus c \end{aligned}$$

It is important that only one message may be encrypted with each key, thus the name of the scheme.

It is easy to see that the One-time Pad is indeed a correct cipher: Let $K \in \mathcal{K}$ and $m \in \mathcal{M}$. Then

$$\begin{aligned} \mathsf{D}(K, \mathsf{E}(K, m)) &= \mathsf{D}(K, K \oplus m) \\ &= K \oplus (K \oplus m) \\ &= (K \oplus K) \oplus m \\ &= 0 \oplus m \\ &= m \end{aligned}$$

1.5.2 Some Basic Probability Theory Notation

Recall that (M, D) is a *discrete probability space* iff M is a finite or countable set and $D : M \to [0, 1]$ such that $\sum_{m \in M} D(m) = 1$ and $P_D(E) = \sum_{m \in E} D(m)$, writing $P(E)$ if D is clear from the context. For sets containing one element we abbreviate $P_D(\{m\})$ by $P_D(m)$.

A *probabilistic function* $A : M \to N$ assigns each input a distribution on the output, i.e., it constitutes a deterministic function $M \to \mathsf{Dist}(N)$ where Dist is the set of all distribution on N. Probabilistic assignment $n \leftarrow A(m)$ then means executing the probabilistic algorithm A with input m and assigning the value to n. For a finite set X, we write $x \leftarrow_{\mathcal{R}} X$ to denote uniform random drawing of an element from X and assigning it to x.

Let *pred* be a predicate on a certain domain M. Then one defines the event that *pred* is fulfilled as $E_{pred} := \{m \in M : pred(m)\}$, and the *probability that pred is fulfilled* given a probabilistic assignment D is defined as

$$P(pred(x); x \leftarrow D) := P_D(E_{pred}). \tag{1.1}$$

Given probabilistic algorithms $A : M \to N$ and $B : M \times N \to O$, we define their *sequential execution* $(A, B) : M \to N \times O$ by

$$P_{(A,B)(m)}(n, o) := P_{A(m)}(n) P_{B(m,n)}(o). \tag{1.2}$$

Using Equations 1.1 and 1.2 one obtains the following general formula. For all $m \in M$

$$P(pred(m, n, o);\ n \leftarrow A(m), o \leftarrow B(m, n)) = \sum_{n,o:\ pred(m,n,o)} P_{A(m)}(n) P_{B(m,n)}(o).$$

Note that one can define a more general version involving i algorithms A_i as well as version with several external messages $m_1, \ldots, m_l$.

1.5.3 Perfect Secrecy of the One-time Pad

The first definition of security for encryption schemes was the definition of *perfect secrecy* given by Claude Shannon in 1949. Intuitively it says, that any message is encrypted to a specific ciphertext with the same probability. This essentially means that, given a ciphertext, *no* adversary gains *any* information which plaintext was encrypted.

Definition 1.2 (Perfect Secrecy) *Let* (E, D) *be an encryption scheme on* $(\mathcal{M}, \mathcal{C}, \mathcal{K})$*. The encryption scheme provides* perfect secrecy *if and only if, for all* $m_0, m_1 \in \mathcal{M}$ *and for all* $c \in \mathcal{C}$*, the following holds:*

$$\mathsf{Pr}\left[c = c'; K \leftarrow_{\mathcal{R}} \mathcal{K}, c' \leftarrow \mathsf{E}(K, m_0)\right] = \mathsf{Pr}\left[c = c'; K \leftarrow_{\mathcal{R}} \mathcal{K}, c' \leftarrow \mathsf{E}(K, m_1)\right]$$

◇

Different equivalent variants of this definition exist based on statistical independence and entropy.

Shannon also proved that the One-time Pad satisfies this notion of perfect secrecy. Even if the proof is rather simple, it constituted a major step in crypto history since it was the first actual proof of security of a cryptographic scheme.

Proposition 1.1 *The One-time Pad provides perfect secrecy.*

Proof. Let $m_0, m_1 \in \mathcal{M}$ and $c \in \mathcal{C}$ be arbitrary and let $U_{\mathcal{K}}$ denote the uniform distribution on $\mathcal{K}$. For $i \in \{0,1\}$ it holds that

$$\begin{aligned}
\mathsf{Pr}\left[c = c'; K \leftarrow_{\mathcal{R}} \mathcal{K}, c' \leftarrow \mathsf{E}(K, m_i)\right] &= \sum_{K,c':\ c=c'} P_{U_{\mathcal{K}}}(K) \cdot P_{E(K,m_i)}(c') \\
&= \sum_{K} P_{U_{\mathcal{K}}}(K) \cdot P_{E(K,m_i)}(c) \\
&= \sum_{K} 1/|\mathcal{K}| \cdot \begin{cases} 1 & \text{if } c = K \oplus m_i \\ 0 & \text{else} \end{cases} \\
&= 1/|\mathcal{K}|
\end{aligned}$$

This holds for both i, so the claim follows. ∎

The One-time Pad has the disadvantage that the key is as long as the message and may not be reused. Thus deploying this scheme necessarily requires to securely transmit a large amount of keys. While this is done in very sensitive environments (the "Red Telephone" linking the White House with the Kremlin during the Cold War was encrypted with the One-time Pad) it is impractical for essentially any application. So the question naturally arises if one can securely encrypt with shorter keys. Unfortunately, Shannon also proved that this is not possible given the definition of perfect secrecy.

Proposition 1.2 (Optimality of the One-time Pad) *Let* (E, D) *be a cipher over* $(\mathcal{M}, \mathcal{C}, \mathcal{K})$*. If the cipher provides perfect secrecy, then* $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof. Let $m_1 \in \mathcal{M}$, $K_1 \leftarrow \mathcal{K}$, and $c \leftarrow \mathsf{E}(K_1, m_1)$. Assume for contradiction that $|\mathcal{K}| < |\mathcal{M}|$. Let $\mathcal{M}_c := \{m \in \mathcal{M} \mid \exists K \in \mathcal{K} \colon \mathsf{D}(K, c) = m\}$ denote the set of plaintexts that the ciphertext c could be decrypted to using some key K. Since decryption is deterministic and since $|\mathcal{K}| < |\mathcal{M}|$, we have that $|\mathcal{M}_c| < |\mathcal{M}|$. Thus there exist some $m_2 \in \mathcal{M} \setminus \mathcal{M}_c$, and this m_2 satisfies that $\forall K \in \mathcal{K} : \mathsf{D}(K, c) \neq m_2$ by definition of $\mathcal{M}_c$.

For the plaintexts m_1, m_2 and the ciphertext c one obtains

$$\begin{aligned}
\mathsf{Pr}\left[c = c'; K \leftarrow_{\mathcal{R}} \mathcal{K}, c' \leftarrow \mathsf{E}(K, m_1)\right] &> 0 \\
&= \mathsf{Pr}\left[c = c'; K \leftarrow_{\mathcal{R}} \mathcal{K}, c' \leftarrow \mathsf{E}(K, m_2)\right].
\end{aligned}$$

This contradicts the assumption that the cipher provides perfect secrecy, and thus the assumption $|\mathcal{K}| < |\mathcal{M}|$ was not correct. ∎