2nd Chapter: Fundamentals of Cryptography

COMP 41280

Félix Balado

School of Computer Science
University College Dublin

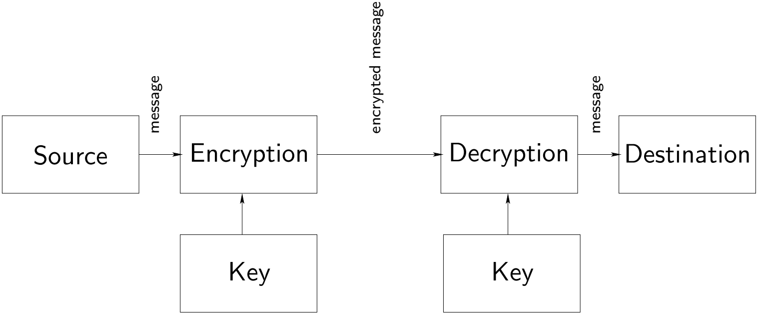
Outline of the Chapter

Information Security Requirements

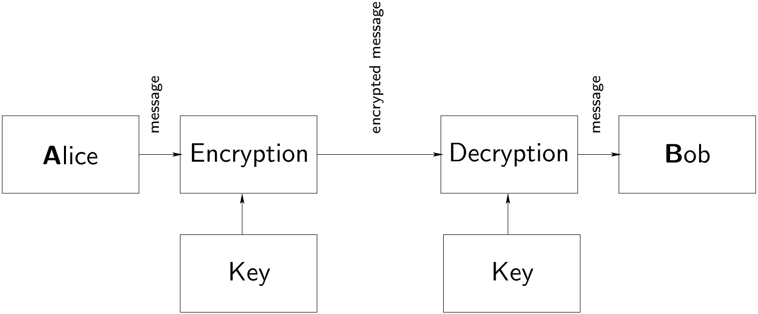
- *Privacy*:
  - protected information should not be accessible to unauthorised third parties
- *Authentication*:
  - the recipient of some information should be able to verify its origin and authorship
- *Integrity*:
  - the recipient of a message should be able to verify that it has not been forged or altered in transit
- *Non repudiation*:
  - a message authenticated by a trusted third party cannot be repudiated at a later stage by its sender

Cryptography can deal with aspects of all these requirements

Shannon's Model for Cryptography (1948)



Shannon's Model for Cryptography (1948)



Terminology and Basic Concepts

- Basic encryption/decryption operation flow:

$$\boxed{L \underset{T}{\rightarrow} C \underset{T'}{\rightarrow} L}$$

  - $T(\cdot,\cdot)$, $T'(\cdot,\cdot)$: families of invertible (injective) transformations which depend on a parameter K (encryption system, or cipher)
    - $C = T(L,K)$: encryption of L
    - $L = T'(C,K) = T'(T(L,K),K)$: decryption of C
  - L: cleartext or plaintext (unencrypted message, "in the clear")
  - C: ciphertext (encrypted or ciphered message)
  - K: encryption/decryption key

Terminology and Basic Concepts

- Basic encryption/decryption operation flow:

$$\boxed{L \underset{T}{\rightarrow} C \underset{T'}{\rightarrow} L}$$

  - $T(\cdot,\cdot)$, $T'(\cdot,\cdot)$: families of invertible (injective) transformations which depend on a parameter K (encryption system, or cipher)
    - $C = T(L, K)$: encryption of L
    - $L = T'(C, K) = T'(T(L, K), K)$: decryption of C
  - L: cleartext or plaintext (unencrypted message, "in the clear")
  - C: ciphertext (encrypted or ciphered message)
  - K: encryption/decryption key
  - in many cryptographic methods $T(\cdot, K)$ does not operate on a single symbol, but on a block of n symbols

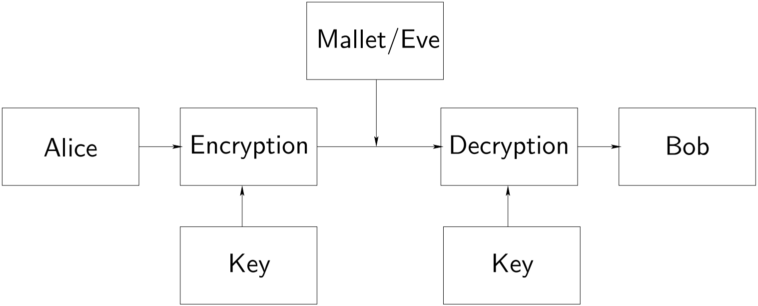
Terminology and Basic Concepts

- Basic encryption/decryption operation flow:

$$\boxed{L \underset{T}{\rightarrow} C \underset{T'}{\rightarrow} L}$$

  - $T(\cdot,\cdot)$, $T'(\cdot,\cdot)$: families of invertible (injective) transformations which depend on a parameter K (encryption system, or cipher)
    - $C = T(L, K)$: encryption of L
    - $L = T'(C, K) = T'(T(L, K), K)$: decryption of C
  - L: cleartext or plaintext (unencrypted message, "in the clear")
  - C: ciphertext (encrypted or ciphered message)
  - K: encryption/decryption key
  - in many cryptographic methods $T(\cdot, K)$ does not operate on a single symbol, but on a block of n symbols
- The wrong decryption key should lead to a wrong plaintext
  - in symmetric-key cryptography (Shannon's model), if $K' \neq K$ then $T'(T(L, K), K') \neq L$

Privacy Threats



- Attacker's nicknames:
  - **Mal**let, **Mal**lory: malicious man in the middle
  - **Eve**: eavesdropper

Cryptanalysis (I)

- Kerckhoffs principle: it is always safer to assume that all the details of the cryptographic scheme are publicly known
  - only the key can remain secret
  - therefore
    - the encryption system (T, T') is public, static (not modifiable)
    - the key (K) is private, dynamic (modifiable)
  - security through obscurity (i.e., by assuming that Mallet does not know T, T') is a very bad idea in the long run
    - Shannon: "the enemy knows the system"

Cryptanalysis (II)

- Cryptanalysis: gathering information about the secret communication between Alice and Bob
    - point of view of Mallet
- An attack to a cryptosystem is an attempt to cryptanalyse it
    - examples (successful attacks):
        - recover cleartext L, either completely or partially
        - recover the key K (in which case future communications with that key will also be intercepted)

Cryptanalysis (II)

- Cryptanalysis: gathering information about the secret communication between Alice and Bob
  - point of view of Mallet
- An attack to a cryptosystem is an attempt to cryptanalyse it
  - examples (successful attacks):
    - recover cleartext L, either completely or partially
    - recover the key K (in which case future communications with that key will also be intercepted)
- Mallet might also carry out active attacks
  - example:
    - replace genuine encrypted message C by forged encrypted message C_F (without necessarily obtaining L or K)

Cryptanalysis (III)

- Cryptanalysis is as much a science as an art
  - sometimes Mallet can attempt a systematic strategy
  - at other times no specific methodology is known
- Mallet can exploit side information available to him:
  - examples: message language, message content, format, etc
  - how can Mallet exploit side information?
    - common sentences
    - frequency of letters, digrams, trigrams,...
    - format information (headers, footers, tags, etc)
- Without side information, the attacker may always attempt exhaustive analysis (brute force): try all keys in key space

Cryptanalysis (Toy Example)

- Try to decipher the following ciphertext, knowing that $T(\cdot, K)$ is a 4-character permutation of the input characters

$$C = \text{“RDFO ATFI ELOP”}$$

Cryptanalysis (Toy Example)

- Try to decipher the following ciphertext, knowing that $T(\cdot, K)$ is a 4-character permutation of the input characters

$$C = \text{“RDFO ATFI ELOP”}$$

  - easier to decipher if side information “message conveys car makes” is available to Mallet

Cryptanalysis (Toy Example)

- Try to decipher the following ciphertext, knowing that $T(\cdot, K)$ is a 4-character permutation of the input characters

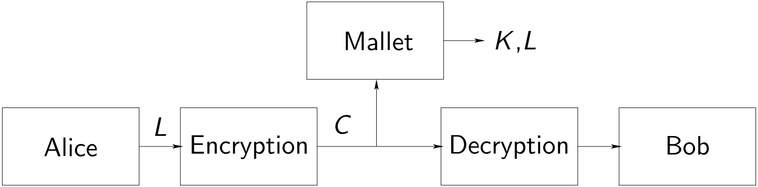
$$C = \text{“RDFO ATFI ELOP”}$$

  - easier to decipher if side information “message conveys car makes” is available to Mallet
  - in this toy example, brute force is not too difficult either, as there are only $4! = 24$ keys (per 4-character block)

- These tricks tend not to be useful with modern methods

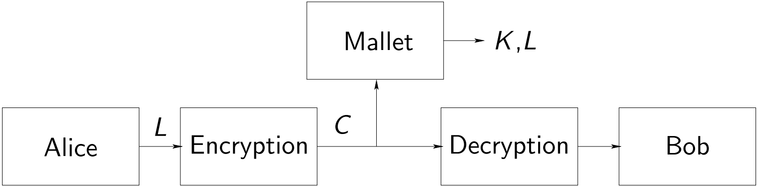
Cryptanalysis Models (I)

1. Ciphertext only

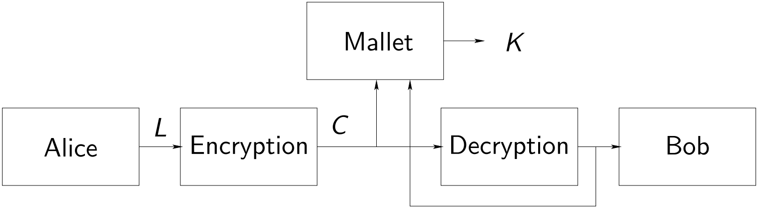


Cryptanalysis Models (I)

1. Ciphertext only

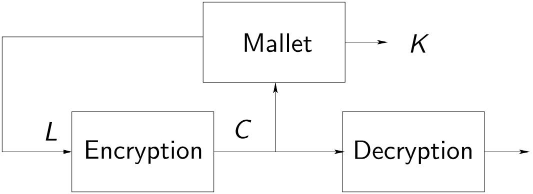


2. Known cleartext



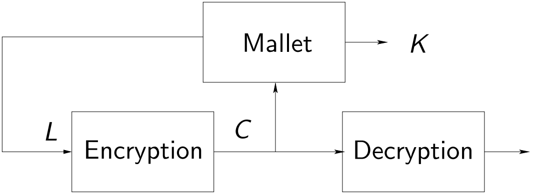
Cryptanalysis Models (II)

3. Chosen cleartext



Cryptanalysis Models (II)

3. Chosen cleartext



4. Chosen ciphertext

Outline of the Chapter

Probability and Cipher Security: Toy Example

Why does probability matter for the security of a cipher?

- Assume $\mathcal{L} = \{a, b\}$, i.e. only two plaintext symbols
  - pmf: $p(L = a) = \frac{1}{4}$ and $p(L = b) = \frac{3}{4}$
- Suppose $\mathcal{K} = \{k_1, k_2, k_3\}$, i.e. three possible keys
  - pmf: $p(K = k_1) = \frac{1}{2}$, $p(K = k_2) = p(K = k_3) = \frac{1}{4}$
- Now suppose that $\mathcal{C} = \{1, 2, 3, 4\}$, and that $T(L, K)$ is:

$T(L, K)$	a	b
k_1	1	2
k_2	2	3
k_3	3	4

Probability and Cipher Security: Toy Example

Why does probability matter for the security of a cipher?

- Assume $\mathcal{L} = \{a, b\}$, i.e. only two plaintext symbols
  - pmf: $p(L = a) = \frac{1}{4}$ and $p(L = b) = \frac{3}{4}$
- Suppose $\mathcal{K} = \{k_1, k_2, k_3\}$, i.e. three possible keys
  - pmf: $p(K = k_1) = \frac{1}{2}$, $p(K = k_2) = p(K = k_3) = \frac{1}{4}$
- Now suppose that $\mathcal{C} = \{1, 2, 3, 4\}$, and that $T(L, K)$ is:

$T(L,K)$	a	b
k_1	1	2
k_2	2	3
k_3	3	4

- With the information above anyone can compute $p(C) = \sum_{L \in \mathcal{L}, K \in \mathcal{K}} p(C|K, L) p(K, L)$
  - $p(C = 1) = \frac{1}{8}$, $p(C = 2) = \frac{7}{16}$, $p(C = 3) = \frac{1}{4}$, $p(C = 4) = \frac{3}{16}$

Probability and Cipher Security: Toy Example

- Using the Bayes' theorem we can compute $p(L|C) = \frac{p(C|L)p(L)}{p(C)}$

Probability and Cipher Security: Toy Example

- Using the Bayes' theorem we can compute $p(L|C) = \frac{p(C|L)p(L)}{p(C)}$
- Therefore:
  - $p(L = a|C = 1) = 1$
    - if the attacker observes $C = 1$ he **knows** $L = a$ and $K = k_1$
  - $p(L = a|C = 2) = \frac{1}{7}$
    - if the attacker observes $C = 2$ he **guesses** $L = b$ and $K = k_1$
  - $p(L = a|C = 3) = \frac{1}{4}$
    - if the attacker observes $C = 3$ he **guesses** $L = b$ and $K = k_2$
  - $p(L = a|C = 4) = 0$
    - if the attacker observes $C = 4$ he **knows** $L = b$ and $K = k_3$

$\rightarrow$ ciphertext can reveal information about cleartext to attacker

- this is what we would like to avoid

Encryption Security

There are two types of encryption methods as regards their vulnerability to cryptanalysis

1. Insecure methods
2. Secure methods
   a) unconditionally secure: there is no method by means of which K can be found by an attacker
   b) computationally secure: there are methods that would allow an attacker to find K but they cannot be implemented in practice

Unconditionally Secure System (Perfect Cipher)

- Definition: a perfect cipher is such that an attacker can never have enough information to determine $T(\cdot, K)$ and $T'(\cdot, K)$, independently of available processing power and time
  - we then speak of perfect secrecy, or unconditional security

Unconditionally Secure System (Perfect Cipher)

- Definition: a perfect cipher is such that an attacker can never have enough information to determine $T(\cdot, K)$ and $T'(\cdot, K)$, independently of available processing power and time
  - we then speak of perfect secrecy, or unconditional security
- **Criterion** (Shannon): in an unconditionally secure system, or perfect cipher, C and L must be statistically independent:
  - $p(L|C) = p(L)$, or equivalently $p(L, C) = p(L)p(C)$
  - note: this is from the point of view of Mallet, unaware of K

Unconditionally Secure System (Perfect Cipher)

- Definition: a perfect cipher is such that an attacker can never have enough information to determine $T(\cdot, K)$ and $T'(\cdot, K)$, independently of available processing power and time
  - we then speak of perfect secrecy, or unconditional security
- **Criterion** (Shannon): in an unconditionally secure system, or perfect cipher, C and L must be statistically independent:
  - $p(L|C) = p(L)$, or equivalently $p(L, C) = p(L)p(C)$
  - note: this is from the point of view of Mallet, unaware of K
- Consequence for the mutual information: $I(L; C) = 0$
  - interpretation: on average, the ciphertext does not reveal any information about the plaintext
  - also $I(K; C) = 0$ (as $H(K|C) = H(K)$ with perfect security)

Unconditionally Secure System

- Unconditional security implies that

$$\boxed{H(C) \leq H(K)}$$

- Proof:
  - $H(C|L) = H(C)$, because of the independence assumption in an unconditionally secure system
  - furthermore:
    - $H(L|K) = H(L)$, because message and key are independent
    - $H(C|L, K) = 0$, because of Kerckhoffs principle

Unconditionally Secure System

- Unconditional security implies that

$$\boxed{H(C) \leq H(K)}$$

- Proof:
  - $H(C|L) = H(C)$, because of the independence assumption in an unconditionally secure system
  - furthermore:
    - $H(L|K) = H(L)$, because message and key are independent
    - $H(C|L,K) = 0$, because of Kerckhoffs principle
  - now, we can apply the chain rule for the entropy to $H(C,L,K)$ in two ways:
    - $H(C,L,K) = H(L,K) + \cancelto{0}{H(C|L,K)} = H(L) + H(K)$
    - $H(C,L,K) = H(C,L) + H(K|C,L) \geq H(C,L)$

Unconditionally Secure System (II)

- finally, using the two expressions above we have:

$$0 = H(C, L, K) - H(L) - H(K)$$

Unconditionally Secure System (II)

- finally, using the two expressions above we have:

$$\begin{aligned} 0 &= H(C, L, K) - H(L) - H(K) \\ &\geq H(C, L) - H(L) - H(K) \end{aligned}$$

Unconditionally Secure System (II)

- finally, using the two expressions above we have:

$$\begin{aligned} 0 &= H(C, L, K) - H(L) - H(K) \\ &\geq H(C, L) - H(L) - H(K) \\ &= H(C|L) - H(K) \end{aligned}$$

- as $H(C|L) = H(C)$ with unconditional security, then we have that $H(C) \leq H(K)$

$\square$

Interpretation: a low-entropy key will be bad for security

Perfect Ciphers and Key Space Size

- Related to the previous result, we have the following implication of perfect security:
  - in a perfect cipher, $\boxed{\#\text{keys} \geq \#\text{messages}}$ (that is, $|\mathcal{K}| \geq |\mathcal{L}|$)

Perfect Ciphers and Key Space Size

- Related to the previous result, we have the following implication of perfect security:
  - in a perfect cipher, $\boxed{\#\text{keys} \geq \#\text{messages}}$ (that is, $|\mathcal{K}| \geq |\mathcal{L}|$)
- <u>Proof</u>: (*by contradiction*)
  - assume that $\#\text{keys} < \#\text{messages}$ and take a cipher value C_0 which has nonzero probability, that is $p(C_0) > 0$
  - if there are less keys than messages, then for any key K we can always find a message L_0 such that $T'(C_0, K) \neq L_0$

Perfect Ciphers and Key Space Size

- Related to the previous result, we have the following implication of perfect security:
  - in a perfect cipher, $\boxed{\#\text{keys} \geq \#\text{messages}}$ (that is, $|\mathcal{K}| \geq |\mathcal{L}|$)
- Proof: (*by contradiction*)
  - assume that $\#$keys $<$ $\#$messages and take a cipher value C_0 which has nonzero probability, that is $p(C_0) > 0$
  - if there are less keys than messages, then for any key K we can always find a message L_0 such that $T'(C_0, K) \neq L_0$
  - for this L_0 we have that $p(C_0|L_0) = 0$
    - then this cannot be a perfect cipher, because by definition it would require $p(C_0|L_0) = p(C_0)$ (contradiction)

□

Computational Security

- Definition: a system is computationally secure when
  1. the ciphertext contains sufficient information to decipher a unique cleartext solution
  2. but the best practical attack is technologically limited in terms of processing power
- Thus its is not guaranteed that Mallet will succeed in a reasonable amount of time

Computational Security

- Definition: a system is computationally secure when
  1. the ciphertext contains sufficient information to decipher a unique cleartext solution
  2. but the best practical attack is technologically limited in terms of processing power
- Thus its is not guaranteed that Mallet will succeed in a reasonable amount of time
- In this type of real life systems, security is verified by stress-testing
- Example: DES (Data Encryption Standard)
  - cryptosystem with a key length of 56 bits (2^{56} different keys)
  - assume that Mallet knows some ciphertext-plaintext pairs: then it can always try all keys (brute force)
  - 1 key every 5 μs $\rightarrow$ 11,000 years to try all keys

Computational Security

- Definition: a system is computationally secure when
  1. the ciphertext contains sufficient information to decipher a unique cleartext solution
  2. but the best practical attack is technologically limited in terms of processing power
- Thus its is not guaranteed that Mallet will succeed in a reasonable amount of time
- In this type of real life systems, security is verified by stress-testing
- Example: DES (Data Encryption Standard)
  - cryptosystem with a key length of 56 bits (2^{56} different keys)
  - assume that Mallet knows some ciphertext-plaintext pairs: then it can always try all keys (brute force)
  - 1 key every 5 μs $\rightarrow$ 11,000 years to try all keys
  - however with a million parallel processors: 4 days

Computational Security: Confusion and Diffusion

- Shannon proposed that a good practical encryption method (i.e. computationally secure) should implement two features:
  - confusion: the relationship between C and K should be as complex as possible
  - diffusion: the statistical properties of L should be spread (diffused) across C
- These assumptions typically refer to blocks of symbols
  - block of n plaintext symbols $\leftrightarrow$ block of n ciphertext symbols
- We will see how these features are implemented when we study practical methods

Unconditional Security and the Unicity Distance

- Shannon also proposed to measure the secrecy of a practical cipher in terms of key equivocation: $H(K|C)$
  - if $H(K|C) = 0$ there is no uncertainty about the key when a ciphertext is known
    - then the cryptographic method is, in principle, breakable
    - (if Mallet has enough computational resources)

Unconditional Security and the Unicity Distance

- Shannon also proposed to measure the secrecy of a practical cipher in terms of key equivocation: $H(K|C)$
  - if $H(K|C) = 0$ there is no uncertainty about the key when a ciphertext is known
    - then the cryptographic method is, in principle, breakable
    - (if Mallet has enough computational resources)
- Typically, as the block length n increases the key equivocation decreases
  - **definition** (Shannon): the unicity distance (N_0) of a cipher is the smallest block length n such that $H(K|C) = 0$
  - equivalently, N_0 is the minimum amount of ciphertext symbols needed by an attacker to unequivocally determine the key

Unconditional Security and Random Ciphers

- The direct computation of the unicity distance N_0 of a given cipher $T(L, K)$ is usually difficult
  - $H(K|C)$ requires $p(K, C)$ (which can be hard to model)

Unconditional Security and Random Ciphers

- The direct computation of the unicity distance N_0 of a given cipher $T(L, K)$ is usually difficult
  - $H(K|C)$ requires $p(K, C)$ (which can be hard to model)
- However we can approximately compute the unicity distance just by assuming that we are dealing with a random cipher
  - **definition** (Shannon): $T(L, K)$ is a random cipher if the choice of K by Alice is made uniformly at random among all possibilities in $\mathcal{K}$

Unicity Distance Analysis

Assumptions that we will make in our unicity distance analysis:

- The cryptographic method is seen as a random cipher
  - the key K corresponds to a block of n symbols: $C^n = T(L^n, K)$
- Attacker:
  - ciphertext only attack
  - exhaustive cryptanalysis with unlimited processing power and time: Mallet can try all keys (brute force)
  - when attempting decryption, the attacker can distinguish nonsense blocks of information from meaningful ones (i.e. Mallet knows the plaintext language)

Redundancy

- The unicity distance analysis involves the concept of redundancy of the language in which the cleartext is written
  - <u>definition</u>: the redundancy of L, where $L \in \mathcal{L}$ and $|\mathcal{L}| = s$, is

$$R = \log_2 s - H(L) \text{ (in bits/symbol)}$$

  - in English, the alphabet size is $s = 26$; empirically, $H(L) \approx 1.5$ bits/letter, and then $R \approx \log_2 26 - 1.5 = 3.2$ bits/letter

Redundancy

- The unicity distance analysis involves the concept of redundancy of the language in which the cleartext is written
  - definition: the redundancy of L, where $L \in \mathcal{L}$ and $|\mathcal{L}| = s$, is

$$R = \log_2 s - H(L) \text{ (in bits/symbol)}$$

  - in English, the alphabet size is $s = 26$; empirically, $H(L) \approx 1.5$ bits/letter, and then $R \approx \log_2 26 - 1.5 = 3.2$ bits/letter
- Intuitively: redundancy is the information we can remove while still preserving the message (what we do in compression)
  - ad-hoc example: removing redundancy from an English text while (approximately) preserving message content

    "*Th cncpt f rdndncy n nglsh s ntrstng*"

Redundancy

- The unicity distance analysis involves the concept of redundancy of the language in which the cleartext is written
  - definition: the redundancy of L, where $L \in \mathcal{L}$ and $|\mathcal{L}| = s$, is

$$R = \log_2 s - H(L) \text{ (in bits/symbol)}$$

  - in English, the alphabet size is $s = 26$; empirically, $H(L) \approx 1.5$ bits/letter, and then $R \approx \log_2 26 - 1.5 = 3.2$ bits/letter
- Intuitively: redundancy is the information we can remove while still preserving the message (what we do in compression)
  - ad-hoc example: removing redundancy from an English text while (approximately) preserving message content

    "*Th cncpt f rdndncy n nglsh s ntrstng*"

    "*The concept of redundancy in English is interesting*"
  - this would not be possible if $R = 0$ in English

Meaningful Messages and Language Entropy

- There are s^n possible n-symbol block messages, of which
  1. $2^{nH(L)}$ are meaningful (on average), and have uniform probability $2^{-nH(L)}$
     - consequence of *asymptotic equipartition property* (large n):

$$-\frac{1}{n}\log_2 p(L_1, \cdots, L_n) \to H(L)$$

Meaningful Messages and Language Entropy

- There are s^n possible n-symbol block messages, of which
  1. $2^{nH(L)}$ are meaningful (on average), and have uniform probability $2^{-nH(L)}$
     - consequence of *asymptotic equipartition property* (large n):

$$-\frac{1}{n}\log_2 p(L_1, \cdots, L_n) \to H(L)$$

  2. the rest, $s^n - 2^{nH(L)}$ messages, are meaningless (or nonsense), and have negligible (≈ 0) probability

Meaningful Messages and Language Entropy

- There are s^n possible n-symbol block messages, of which
  1. $2^{nH(L)}$ are meaningful (on average), and have uniform probability $2^{-nH(L)}$
     - consequence of *asymptotic equipartition property* (large n):

$$-\frac{1}{n}\log_2 p(L_1, \cdots, L_n) \to H(L)$$

  2. the rest, $s^n - 2^{nH(L)}$ messages, are meaningless (or nonsense), and have negligible (≈ 0) probability

$$s^n \left\{ \begin{array}{l} 2^{nH(L)} \left\{ \begin{array}{c} \overbrace{\text{theenemydoesknowthesystem}}^{n} \\ \vdots \\ \text{wheredidtheylearnthatsong} \end{array} \right. \\ \hline s^n - 2^{nH(L)} \left\{ \begin{array}{c} \text{evioeqhrstmcgyevocmdzauvx} \\ \vdots \\ \text{rkvojyxoqpfimgwfgegliylfy} \end{array} \right. \end{array} \right.$$

False Keys

- Definition: a key $\hat{K}$ is false or spurious if, given $C^n = T(L^n, K)$, $\hat{L}^n = T'(C^n, \hat{K})$ is meaningful, but $\hat{K} \neq K$ and $\hat{L}^n \neq L^n$

False Keys

- Definition: a key $\hat{K}$ is false or spurious if, given $C^n = T(L^n, K)$, $\hat{L}^n = T'(C^n, \hat{K})$ is meaningful, but $\hat{K} \neq K$ and $\hat{L}^n \neq L^n$
- Let $f(C^n)$ be the number of false keys associated to a ciphertext, then the average number of false keys is

$$E(f(C^n)) = (\#\text{ keys} - 1) \times \frac{\#\text{ meaningful blocks}}{\#\text{ blocks}}$$

False Keys

- Definition: a key $\hat{K}$ is false or spurious if, given $C^n = T(L^n, K)$, $\hat{L}^n = T'(C^n, \hat{K})$ is meaningful, but $\hat{K} \neq K$ and $\hat{L}^n \neq L^n$
- Let $f(C^n)$ be the number of false keys associated to a ciphertext, then the average number of false keys is

$$E(f(C^n)) \stackrel{(*)}{=} (\#\text{ keys} - 1) \times \frac{\#\text{ meaningful blocks}}{\#\text{ blocks}}$$

- Notes:
  - $^{(*)}$ random cipher assumption

False Keys

- Definition: a key $\hat{K}$ is false or spurious if, given $C^n = T(L^n, K)$, $\hat{L}^n = T'(C^n, \hat{K})$ is meaningful, but $\hat{K} \neq K$ and $\hat{L}^n \neq L^n$
- Let $f(C^n)$ be the number of false keys associated to a ciphertext, then the average number of false keys is

$$
\begin{aligned}
E(f(C^n)) &\stackrel{(*)}{=} (\#\text{ keys} - 1) \times \frac{\#\text{ meaningful blocks}}{\#\text{ blocks}} \\
&= (|\mathcal{K}| - 1) \times \frac{2^{nH(L)}}{2^{n \log_2 s}} \\
&= (|\mathcal{K}| - 1) \times 2^{-nR} \\
&= 2^{\log_2 |\mathcal{K}| - nR} - \epsilon
\end{aligned}
$$

- Notes:
  - $^{(*)}$ random cipher assumption
  - $\#$ blocks: $s^n = (2^{\log_2 s})^n = 2^{n \log_2 s}$

Unicity Distance

- With this result, we define the unicity distance of a cipher as

$$\boxed{N_0 = \frac{\log_2 |\mathcal{K}|}{R}}$$

  - N_0 is the solution to $\log_2 |\mathcal{K}| - nR = 0$ (exponent in $E(f(C^n))$), that is, a block size n such that $E(f(C^n)) < 1$

Unicity Distance

- With this result, we define the unicity distance of a cipher as

$$\boxed{N_0 = \frac{\log_2 |\mathcal{K}|}{R}}$$

  - N_0 is the solution to $\log_2 |\mathcal{K}| - nR = 0$ (exponent in $E(f(C^n))$), that is, a block size n such that $E(f(C^n)) < 1$
- Therefore:
  - if $n < N_0 \rightarrow$ <u>unconditionally secure system</u>
    - on average, there are false keys
    - not enough information for the attacker to uniquely determine the key (and hence the cleartext)
  - if $n \geq N_0 \rightarrow$ insecure system
    - on average, there are no false keys
    - enough information, or equivalently $H(K|C) = 0$ (null equivocation)

$\rightarrow$ In a good cipher N_0 should be as large as possible

Outline of the Chapter

Encryption and Source Coding

- We have mentioned that some cryptanalyses look for most frequent patterns in C (using histograms; more later)
  - these attacks can be thwarted if the symbols in C have uniform probabilities and no correlations among them (diffusion)

Encryption and Source Coding

- We have mentioned that some cryptanalyses look for most frequent patterns in C (using histograms; more later)
  - these attacks can be thwarted if the symbols in C have uniform probabilities and no correlations among them (diffusion)
- Source coding approximates this scenario by eliminating redundancy from the cleartext
  - it is a good idea to compress (that is, to source code) the cleartext before encrypting it
  - decreasing redundancy (R) increases the unicity distance (N_0)
  - even if the attacker is aware of compression, this increases the computational complexity of attacks

Encryption and Source Coding

- We have mentioned that some cryptanalyses look for most frequent patterns in C (using histograms; more later)
  - these attacks can be thwarted if the symbols in C have uniform probabilities and no correlations among them (diffusion)
- Source coding approximates this scenario by eliminating redundancy from the cleartext
  - it is a good idea to compress (that is, to source code) the cleartext before encrypting it
  - decreasing redundancy (R) increases the unicity distance (N_0)
  - even if the attacker is aware of compression, this increases the computational complexity of attacks
- In connection with this, we may also wonder whether a cipher could also possibly decrease the redundancy of the cleartext. . .

Encryption Systems and Compression

- **Theorem**: the entropy of the encrypted text can never be smaller than the entropy of the cleartext
- Proof:
  - first see that since $C = T(L, K)$ and $L = T'(C, K)$, then $H(C|L, K) = 0$ and $H(L|C, K) = 0$ (Kerckhoffs' principle)

Encryption Systems and Compression

- **Theorem**: the entropy of the encrypted text can never be smaller than the entropy of the cleartext
- Proof:
  - first see that since $C = T(L, K)$ and $L = T'(C, K)$, then $H(C|L, K) = 0$ and $H(L|C, K) = 0$ (Kerckhoffs' principle)
  - now, $H(C, L, K)$ can be written in two different ways using the chain rule of the entropy

$$\begin{aligned} H(C, L, K) &= H(C|L, K) + H(L, K) \\ &= H(L|C, K) + H(C, K) \end{aligned}$$

Encryption Systems and Compression

- **Theorem**: the entropy of the encrypted text can never be smaller than the entropy of the cleartext
- Proof:
  - first see that since $C = T(L, K)$ and $L = T'(C, K)$, then $H(C|L, K) = 0$ and $H(L|C, K) = 0$ (Kerckhoffs' principle)
  - now, $H(C, L, K)$ can be written in two different ways using the chain rule of the entropy

$$\begin{aligned} H(C, L, K) &= \cancelto{0}{H(C|L, K)} + H(L, K) \\ &= \cancelto{0}{H(L|C, K)} + H(C, K) \end{aligned}$$

Encryption Systems and Compression

- **Theorem**: the entropy of the encrypted text can never be smaller than the entropy of the cleartext
- Proof:
  - first see that since $C = T(L,K)$ and $L = T'(C,K)$, then $H(C|L,K) = 0$ and $H(L|C,K) = 0$ (Kerckhoffs' principle)
  - now, $H(C,L,K)$ can be written in two different ways using the chain rule of the entropy

$$\begin{aligned} H(C,L,K) &= \cancelto{0}{H(C|L,K)} + H(L,K) \\ &= \cancelto{0}{H(L|C,K)} + H(C,K) \end{aligned}$$

  - applying the chain rule of the entropy again:

$$\begin{aligned} H(L,K) &= H(L|K) + H(K) \\ H(C,K) &= H(C|K) + H(K) \end{aligned}$$

Encryption Systems and Compression (II)

- Therefore: $H(L|K) = H(C|K)$, that is, when the key is known the entropies of cleartext and ciphertext coincide
- Moreover $H(L|K) = H(L)$, because the key is chosen independently of the cleartext

Encryption Systems and Compression (II)

- Therefore: $H(L|K) = H(C|K)$, that is, when the key is known the entropies of cleartext and ciphertext coincide
- Moreover $H(L|K) = H(L)$, because the key is chosen independently of the cleartext
- Since $H(C|K) \leq H(C)$ (because conditioning cannot increase entropy) then

$$\boxed{H(L) \leq H(C)}$$

Encryption Systems and Compression (II)

- Therefore: $H(L|K) = H(C|K)$, that is, when the key is known the entropies of cleartext and ciphertext coincide
- Moreover $H(L|K) = H(L)$, because the key is chosen independently of the cleartext
- Since $H(C|K) \leq H(C)$ (because conditioning cannot increase entropy) then

$$\boxed{H(L) \leq H(C)}$$

  - interpretation: the shortest representation of the ciphertext will always need *at least* the same amount of bits/symbol as the shortest representation of the cleartext
  - if $H(C) = H(L)$ then the cipher is called nonexpansive

Encryption and Channel Coding

- Source coding can be critical if errors happen (such as during transmission through a communications channel):
  1. a wrong symbol may preclude decoding
  2. a missing symbol may preclude decoding (desynchronisation)

  - consider a bit error in a binary prefix source code...

Encryption and Channel Coding

- Source coding can be critical if errors happen (such as during transmission through a communications channel):
  1. a wrong symbol may preclude decoding
  2. a missing symbol may preclude decoding (desynchronisation)

  - consider a bit error in a binary prefix source code...

- <u>Without redundancy, errors are hard to tackle</u>
- Solution: channel coding (error correction coding), which reintroduces redundancy to detect/correct errors
  - simplest example of error detection: parity check
  - simplest example of error correction: repetition

Encryption and Channel Coding (II)

- All modern communications standards include more or less sophisticated channel coding (error correction)
- The information is organised in frames (blocks), and headers/footers are inserted with error correction redundancy

Encryption and Channel Coding (II)

- All modern communications standards include more or less sophisticated channel coding (error correction)
- The information is organised in frames (blocks), and headers/footers are inserted with error correction redundancy
- If we apply error correction before encryption, the attacker's job would be eased: higher plaintext redundancy
  - furthermore, channel errors will affect decryption, and Bob typically will not be able to recover original plaintext
- therefore channel coding headers have to be inserted after encryption