Liliana Salvador

Segurança Absoluta em Sistemas de Cifra de Chave Simétrica

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2005

Liliana Salvador

Segurança Absoluta em Sistemas de Cifra de Chave Simétrica

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2005

Liliana Salvador

Segurança Absoluta em Sistemas de Cifra de Chave Simétrica

Tese submetida à Faculdade de Ciências da Universidade do Porto para obtenção do grau de Mestre em Informática no Ramo Ciência de Computadores

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
2005

Agradecimentos

Começo por agradecer ao meu orientador, o Professor Luís Filipe Coelho Antunes, pela colaboração, orientação e sobretudo pelo seu constante apoio, incentivo e paciência que sempre me reservou. Os seus conselhos e rigor científico constituíram uma ajuda preciosa e fundamental na realização deste trabalho. Agradeço também o esforço desenvolvido na leitura e sugestões de revisão deste documento.

Ao Professor Armando Matos agradeço pela sua disponibilidade, contínua boa disposição e sugestões de revisão. Este trabalho sem a sua presença teria sido muito menos divertido.

Agradeço à Professora Sophie Laplante do grupo de Complexidade e Algoritmos da Universidade de Paris Sul, por me ter acolhido e orientado durante a primeira fase de desenvolvimento deste trabalho.

Agradeço ao Luís pelo seu incansável e constante apoio, por tudo o que me ensinou e por ter acreditado sempre em mim.

Agradeço também à Valquiria Leite, Ana Costa, Sílvia Rocha, David Pereira, Sónia Sousa, Jorge Coelho e Tânia Magalhães por sempre me terem acompanhado, animado e "compreendido"as minhas aleatorieadades durante a realização deste trabalho! :-)

Por último, mas não menos importante, um agradecimento especial à minha familia pelo apoio e incentivo que sempre me proporcionaram, compreendendo sempre as minhas ausências.

Resumo

Alguns sistemas criptográficos de chave simétrica têm *segurança absoluta*, isto é, mantêm-se seguros mesmo que o adversário tenha um poder computacional ilimitado. Nestes casos, o conhecimento da mensagem cifrada em nada ajuda o adversário a descobrir a chave previamente acordada pelos intervenientes legítimos da comunicação. As provas de segurança absoluta destes sistemas criptográficos são baseadas na entropia de Shannon que mede a quantidade de informação de um evento. No entanto, esta medida opera num meio probabilístico, não sendo possível determinar a quantidade de informação de um objecto em particular e incorporar uma noção de dificuldade computacional na sua fórmula. Deste modo, a complexidade de Kolmogorov, uma medida de informação efectiva, é usada para demonstrar a segurança absoluta de alguns sistemas criptográficos simétricos, com a vantagem de se poder medir a complexidade de objectos individuais. O uso das suas propriedades e variantes tem também como objectivo, num futuro próximo, a avaliação da segurança de sistemas criptográficos de chave pública.

Abstract

Some symmetric cryptosystems have *unconditional security*, that means that if an opponent with unlimited computational power tries to attack them, they remain secure. In these cases, the knowledge of the cipher message doesn't help the opponent to discover the key that was previously chosen by the legitime intervenients of the communication. The proofs of the unconditional security of these cryptosystems are based on Shannon's entropy that measures the quantity of information of an event. This measure operates in a probabilistic environment, doesn't measure the information of an individual object and is difficult to incorporate in its formula the computational dificulty factor. Therefore, Kolmogorov complexity, an effective measure of information, is going to be used to demonstrate the unconditional security of some symmetric cryptosystems, with the advantage that able us to measure the complexity of individual objects. With this measure, we can also use its properties and variants to try to measure the security of public cryptosystems.

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Este capítulo pretende fornecer uma visão global do trabalho desenvolvido. Após uma breve exposição do tema, é descrita a motivação que levou ao desenvolvimento deste trabalho, os seus objectivos e por último a estrutura da dissertação.

1.1 Tema

Este trabalho incide principalmente no papel da *complexidade de Kolmogorov* na *criptografia*, partindo das relações existentes entre complexidade de Kolmogorov e *teoria de informação*, e teoria de informação com a criptografia.

A complexidade de Kolmogorov estuda a quantidade de informação contida em objectos individuais, usualmente descritos por sequências binárias finitas. A complexidade de uma sequência binária é definida através do comprimento do menor programa que a produz.

A teoria da informação quantifica a informação de um evento em termos de probabilidade de cada uma das suas instâncias. Baseia-se na entropia de Shannon que exprime o número de dígitos binários necessários para especificar o resultado de um evento.

A criptografia tem como objectivo o desenvolvimento de sistemas criptográficos seguros. Para sistemas simétricos, esta segurança é demonstrada através do uso da teoria de informação.

Os pontos focados baseiam-se fortemente na *computabilidade* e na *teoria da probabilidade discreta*, sendo necessário algum conhecimento destas áreas para uma correcta compreensão deste trabalho.

1.2 Motivação

Existem várias noções de segurança de sistemas criptográficos. A maioria deles são baseados em suposições provenientes da teoria da complexidade, por exemplo $P \neq NP$ ou na factorização de inteiros com valor muito elevado que se pensa não poder ser efectuada em tempo polinomial. Contudo, existem sistemas criptográficos (simétricos) em que é possível demonstrar a *segurança absoluta* contra um adversário com um poder computacional ilimitado. As provas clássicas de segurança absoluta são baseadas na noção de entropia que mede a quantidade de informação em situações onde o poder computacional é ilimitado. No entanto, esta medida não fornece uma ferramenta de trabalho satisfatória para a análise de sistemas criptográficos de chave pública, sempre baseados em suposições criptográficas (imposição de poder computacional limitado ao adversário).

1.3 Objectivos

O objectivo deste trabalho é utilizar a complexidade de Kolmogorov, uma medida rigorosa da quantidade de informação numa sequência binária individual, como medida de segurança em criptografia. Pretende-se substituir a entropia de Shannon pela complexidade de Kolmogorov e demonstrar a segurança absoluta de alguns sistemas criptográficos simétricos, tais como o *one time pad*, *o segredo partilhado* e a *autenticação*.

1.4 Estrutura da Tese

Esta dissertação encontra-se organizada em 6 capítulos.

O segundo capítulo, **Teoria de Informação** descreve esta teoria e como esta se expressa na entropia. Define alguma notação que irá ser utilizada ao longo do trabalho e faz uma breve revisão sobre a teoria da probabilidade discreta.

O terceiro capítulo, **Complexidade de Kolmogorov** define a complexidade de Kolmogorov, focando algumas propriedades e variantes. Apresenta algumas noções sobre computabilidade que são essenciais para este capítulo e para os seguintes.

O quarto capítulo, **Segurança e Teoria de Informação** explora um pouco a criptografia, começando brevemente pela sua história e definindo os protocolos criptográficos *one time pad*, *segredo partilhado* e *autenticação*. Define segurança absoluta de um sistema simétrico com base na entropia e apresenta as provas de segurança dos protocolos criptográficos acima referidos que são o ponto de estudo deste trabalho.

O quinto capítulo, **Segurança de Instâncias Individuais** é iniciado com a análise

da relação entre a complexidade de Kolmogorov e a entropia de Shannon. O conceito de segurança absoluta é redefinido com base nesta complexidade, uma medida de informação efectiva. Por último, são apresentadas provas da segurança absoluta de instâncias individuais dos protocolos já mencionados no capítulo anterior.

O sexto capítulo, **Conclusão**, é uma breve reflexão relativa ao trabalho apresentado e trabalho futuro.

Capítulo 2

Teoria de Informação

A teoria da informação teve o seu ínicio em 1948 [27] e foi proposta por C. E. Shannon. Segundo o autor, esta teoria é utilizada para medir a quantidade de *informação* de um evento.

Neste capítulo, introduz-se a notação utilizada ao longo deste trabalho, alguns conceitos básicos sobre a teoria da probabilidade discreta [36], e algumas propriedades da teoria de informação.

2.1 Fundamentos Teóricos

Nesta secção vão ser apresentadas noções essenciais sobre sequências binárias [33] e [2], e códigos prefixos.

Sequência Binárias

Os símbolos são representados por $a, b, c, \ldots$, as palavras por $x, y, z, \ldots$ e os alfabetos por $\Sigma, \Gamma, \ldots$

Definição 2.2 *1. Um alfabeto é um conjunto finito não vazio de objectos representado por Σ. A cardinalidade de Σ é representada por $|\Sigma|$.*

2. Um símbolo é um elemento de um alfabeto.

3. Dado um alfabeto Σ. uma sequência ou palavra sobre Σ é uma sequência finita de símbolos pertencentes a Σ.

4. *Σ^n é o conjunto de todas as palavras pertencentes a Σ com comprimento n. O conjunto de todas as palavras pertencentes a Σ é representado por Σ^*.*

5. *O comprimento da palavra $x \in \Sigma^*$ é o número de símbolos em x, representado por $|x|$. A notação é a mesma que se usa para a cardinalidade, mas usualmente o contexto desambigua o seu uso. A palavra vazia é representada por ϵ.*

6. *Dadas duas palavras x e $y \in \Sigma^*$, a concatenação de x e y, representada por xy, é a palavra z que consiste nos símbolos de x, seguidos dos símbolos de y.*

Seja $A = \{0, 1\}$ o conjunto binário e A^* o conjunto de todas as sequências binárias:

$$A^* = \{\epsilon, 0, 1, 00, 01, 11, 000, \ldots\}$$

Com a representação binária usual, algumas sequências binárias não representam números naturais e cada número natural pode ser representado por mais do que uma sequência. Ao longo deste trabalho, será utilizada a representação binária diádica apresentada a seguir:

$$(\epsilon, 0), (0, 1), (1, 2), (00, 3), (01, 4), (10, 5), (11, 6), (000, 7), \ldots$$

Existe uma bijecção entre A^* e $\mathbb{N}$, em que é associado a cada sequência binária finita um número natural que corresponde ao seu índice na ordem lexicográfica. O número natural e a sua representação binária diádica sao considerados o mesmo objecto.

Seja x uma palavra pertencente a A. Se x for considerado um inteiro de acordo com a representação binária descrita acima, então $|x| = \lfloor \log(x + 1) \rfloor$ e, para $x \geq 2$,

$$\lfloor \log x \rfloor \leq |x| \leq \lceil \log x \rceil$$

Aqui, $\lceil x \rceil$ é o menor inteiro maior ou igual a x, $\lfloor x \rfloor$ é o maior inteiro menor ou igual que x e log corresponde ao logaritmo na base 2.

Definição 2.3 *Uma sequência x é um prefixo da sequência y se existir um $z \neq \epsilon$ tal que $y = xz$. Um conjunto A é livre de prefixos se nenhum elemento de A for prefixo de qualquer outro elemento do conjunto.*

Como se pode codificar qualquer objecto numa sequência binária, a partir deste ponto convencionamos que todo o objecto é uma sequência binária e como tal $\Sigma = \{0, 1\}^*$. As sequências binárias são denominadas simplesmente por sequências.

Códigos Prefixos

Considere-se uma comunicação segura entre um emissor A e um receptor B. A prentende enviar uma mensagem (sequência pertencente a Σ^*) a B de forma segura. Quando B receber a mensagem cifrada, ele pode decifrá-la e obter a mensagem original. Para tal, A e B precisam de acordar previamente numa codificação ou num método de descrição. Existe uma relação entre sequências (objectos) e as suas codificações. Esta relação é caracterizada pela função $d : \Sigma^* \to \Sigma^*$ em que $d(y) = x$ é interpretado como "y é uma codificação para o objecto original x". A recuperação de cada objecto original é representada por $e = d^{-1}$.

A função $d : \Sigma^* \to \Sigma^*$ define um código prefixo se o seu domínio for livre de prefixos. Um código prefixo pode ser obtido reservando um símbolo, por exemplo, o zero como marca de separação. Supondo que $x = x_1 x_2 \dots x_n$ é codificado como

$$\bar{x} = \underbrace{11\dots1}_{n} 0 x_1 x_2 \dots x_n$$

A esta codificação de x dá-se o nome de *versão auto-delimitada* de x. É sabido que o tamanho da sequência após o 0 é exactamente n e daqui resulta que $|\bar{x}| = 2n+1$. Assim o tamanho das codificações prefixas é o dobro do tamanho da sequência original. No entanto, é possível obter uma codificação mais simples definindo $x' = |\bar{x}|x$. Verifica-se que $|\bar{x}| = 2|x| + 1$ e o código d' com $d'(x') = x$ é um código prefixo satisfazendo $|\bar{x}'| = |x| + 2\log(|x|) + 1$, para todo o $x \in \Sigma^*$.

Desigualdade de Kraft

De acordo com a codificação anterior, verifica-se que o comprimento das codificações poderá ultrapassar o comprimento da sequência original. Para tal não suceder, L. G. Kraft propôs uma restrição no número de codificações de um dado tamanho.

Teorema 2.3.1 (Desigualdade de Kraft) *Seja $l_1, l_2, \dots$ uma sequência finita ou infinita de números naturais. Existe um código prefixo com esta sequência como comprimento das suas codificações binárias se e só se*

$$\sum_n 2^{-l_n} \leq 1$$

Dem. ($\Rightarrow$) Nesta demonstração é utilizada a correspondência biunívoca entre as sequências binárias finitas x e o intervalo $\Gamma_x = [0.x, 0.x + 2^{-|x|})$. É de observar que o comprimento do intervalo correspondente a x é $2^{-|x|}$. Um código prefixo corresponde a um conjunto de intervalos disjuntos no intervalo $[0, 1)$, o que prova que a desigualdade é válida para este tipo de códigos.

($\Leftarrow$) Supondo que $l_1, l_2, ...$ são dados tais que a desigualdade se verifica e assumindo que a sequência é não-decrescente, escolhem-se a partir do extremo inferior do intervalo $[0, 1)$, intervalos disjuntos adjacentes $I_1, I_2, ...$ de comprimentos $2^{-l_1}, 2^{-l_2}, ...$ Desta forma, para cada $n \geq 1$, o extremo inferior de I_n é $\Sigma_{i=1}^{n} 2^{-l_i}$. É de notar que o extremo superior de I_n coincide com o extremo inferior de I_{n+1}. Já que a sequência de l_i's é não decrescente, cada intervalo $I_n = \Gamma_x$ para qualquer sequência binária x de tamanho $|x| = l_n$. Assim, a sequência binária x correspondente a I_n é a n-ésima codificação. $\diamond$

Esta desigualdade também pode ser demonstrada recorrendo a árvores binárias. Na figura 2.1 está representada uma árvore de aridade d em que cada nó tem d filhos e cada ramo representa um símbolo de codificação. Os ramos que têm origem na raiz representam os d possíveis valores do primeiro símbolo de codificação. Desta forma, o caminho desde a raiz até às folhas fornece todos os símbolos da codificação e representa um código prefixo.

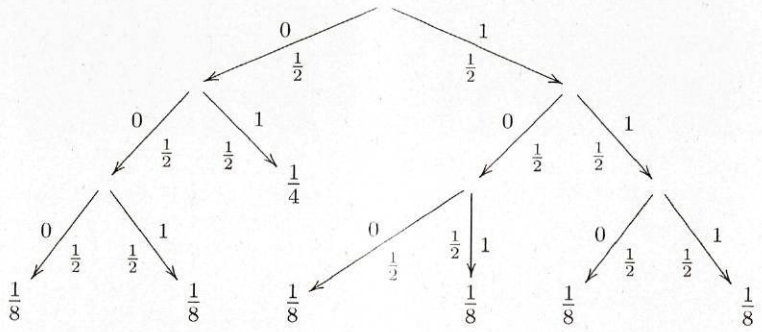

Figura 2.1: Desigualdade de Kraft

Cada codificação elimina os seus descendentes como possíveis codificações. Seja *lmax* o comprimento da mais longa codificação do conjunto de codificações. Consideram-se todos os nós da árvore no nível *lmax*. Um nó no nível l_i tem no máximo $d^{l_{max}-l_i}$ descendentes no nível *lmax*. Então, somando todas as codificações, tem-se

$$\sum d^{l_{max}-l_i} \leq d^{l_{max}}$$

ou

$$\sum d^{-l_i} \leq 1$$

que é a *desigualdade de Kraft.*

Se um código tiver codificações binárias $l_1, l_2, \ldots$ e for unicamente decifrável, então satisfaz a desigualdade de Kraft.

Teoria da Probabilidade Discreta

Um espaço de *probabilidade discreta* é um conjunto $\mathcal{S}$ finito ou enumerável (conjunto com um número infinito de elementos que podem ser postos em correspondência unívoca com os inteiros positivos). Este espaço é conhecido como *espaço amostral* e contém todos os resultados possíveis de uma experiência aleatória. Os elementos do espaço amostral $\mathcal{S}$ são denominados por *eventos elementares* e os subconjuntos $A, B, \ldots$ de $\mathcal{S}$ são conhecidos por *eventos*. Cada evento elementar pode ser visto como um possível resultado de um acontecimento. Em cada espaço amostral estão definidas funções de variável real que mapeiam cada elemento do espaço com um número real. A estas funções dá-se o nome de variáveis aleatórias e são usualmente representadas por $X, Y, Z, \ldots$. Os valores que elas podem tomar são representados por $x, y, z, \ldots$.

Um *processo estocástico* ou *processo aleatório* é uma colecção de variáveis aleatórias definidas num mesmo espaço de probabilidades.

Variáveis aleatórias unidimensionais

Seja X uma variável aleatória. A cada resultado possível x_i é associado um valor $p_X(x_i), i \geq 0$ representando a probabilidade $P(X = x_i)$ e satisfazendo as seguintes condições:

- $p_X(x_i) \geq 0$ para todo x
- $\sum_{x_i \in \mathcal{S}} p_X(x_i) = 1$

Por conveniência, a *distribuição de probabilidade* será representada por $p(x)$ em vez de $p_X(x)$.

Variáveis aleatórias n-dimensionais, $n > 1$

Sejam X e Y duas variáveis aleatórias. Então, o par (X, Y) é considerado uma variável aleatória bidimensional. Da mesma forma, o tuplo $(X_1, \ldots, X_n)$ é considerado uma variável aleatória n-dimensional.

Seja (X, Y) uma variável aleatória. A cada resultado possível (x_i, y_j) é associado um valor $p(x_i, y_j)$ que representa a *probabilidade conjunta* $P(X = x_i, Y = y_j)$ e satisfaz as seguintes condições:

- $p(x_i, y_j) \geq 0$ para todo $(x_i, y_j), i \geq 0, j \geq 0$
- $\sum_{x_i \in \mathcal{S}} \sum_{y_j \in \mathcal{S}} p(x_i, y_j) = 1$

Probabilidade Condicional

A *probabilidade condicional* de x dado y é definida por

$$p(x|y) = \frac{p(x,y)}{p(y)}$$

sempre que $p(y)$ for positivo.

Duas variáveis aleatórias X e Y são *independentes* se para todo o $x \in X$ e $y \in Y$

$$p(x,y) = p(x)p(y)$$

A *probabilidade marginal* pode ser obtida a partir da probabilidade conjunta através do somatório:

$$p(X = a_i) \equiv \sum_{y \in Y} p(X = a_i, y)$$

Valor Esperado

Seja X uma variável aleatória discreta com valores possíveis $x_1, x_2, \ldots, x_n$. Sejam $p(x_i)$ as probabilidades correspondentes a cada valor. Então, o *valor esperado* de X é definido da seguinte forma:

$$E(X) = \sum_{i=1}^{n} x_i p(x_i)$$

2.4 Entropia de Shannon

A unidade de medida utilizada na teoria de informação é o dígito binário. Segundo Shannon, esta teoria é utilizada para medir a quantidade de *informação* de um acontecimento. O autor atribuiu-lhe o nome de *entropia* que exprime o número de dígitos binários necessários para especificar o resultado de um acontecimento. Esta medida revela o grau de incerteza quando é selecionado um elemento de um espaço amostral. Assim, um valor grande de entropia significa que é necessária uma grande quantidade de informação para descrever a situação da qual se tem pouco conhecimento, enquanto que um valor nulo de entropia significa que o conhecimentoda situação é completo. O método de codificação dos acontecimentos é baseado na suposição de que as mensagens (cujo conteúdo é ignorado) são elementos de um espaço amostral e só as características desse espaço determinam a codificação e não as características dos resultados dos acontecimentos.

Seja X uma variável aleatória com distribuição de probabilidade p. A entropia de p corresponde à incerteza de um observador (que sabe que X é distribuído de acordo com p), antes de observar o resultado $X = x$. Imagine-se agora que o observador é o

receptor da mensagem que tem o valor de X. A partir desta dualidade, a entropia é definida como a média da quantidade de informação que o observador ganha depois de receber o valor do resultado x da variável aleatória X.

Sejam $\mathcal{S}$ e $\mathcal{Z}$ dois espaços amostrais com três e dois símbolos respectivamente (ver figura 2.2). Seja $\mathcal{M}$ o espaço amostral que contém $\mathcal{S}$ e $\mathcal{Z}$. A partir de $\mathcal{Z}$ não se sabe qual será o símbolo a ser escolhido e a esta incerteza também se dá o nome de entropia. Qual é a incerteza do espaço amostral M que contém os subespaços $\mathcal{S}$ e $\mathcal{Z}$?

Número de símbolos:

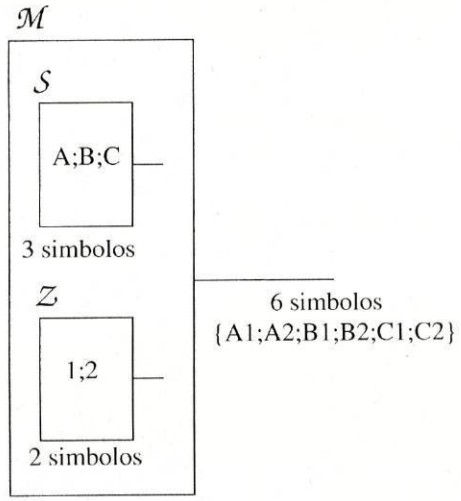

Figura 2.2: Incerteza usando o número de símbolos

Note-se que $\mathcal{M}$ tem seis símbolos, $(A_1; A_2; B_1; B_2; C_1; C_2)$, $\mathcal{S}$ tem três e $\mathcal{Z}$ tem dois símbolos, respectivamente. Se a entropia for medida como o tamanho do espaço amostral, esta medida não é aditiva, propriedade desejável numa medida de informação. Logo, torna-se necessário encontrar uma alternativa.

Aditividade com base nos logaritmos:

Uma vez que a função logarítmica tem algumas propriedades desejáveis numa medida de informação, utiliza-se o logaritmo do número de símbolos (ver figura 2.3).

Com o uso da função logarítmica, $\log(3) + \log(2) = \log(6)$, facilmente se verifica que esta nova medida é aditiva. Chega-se assim, empiricamente, a uma medida de incerteza de eventos equiprováveis. Deste modo, a entropia de uma variável aleatória X com resultados equiprováveis num espaço amostral finito $\mathcal{S}$ é dada por $H(X) = \log|\mathcal{S}|$. Escolhendo uma mensagem x pertencente $\mathcal{S}$, a entropia de X é removida atribuíndo $X = x$ e transmintindo informação $I = \log|\mathcal{S}|$ pela selecção de x. O inteiro $\lceil \log|\mathcal{S}| \rceil$ é interpretado como o número de dígitos necessários para ser transmitido numa comunicação entre um emissor e um receptor. Sendo esta medida de incerteza válida para eventos equiprováveis, questionou-se se esta poderia ser estendida para eventos não equiprováveis. A fórmula da incerteza associada ao evento X é

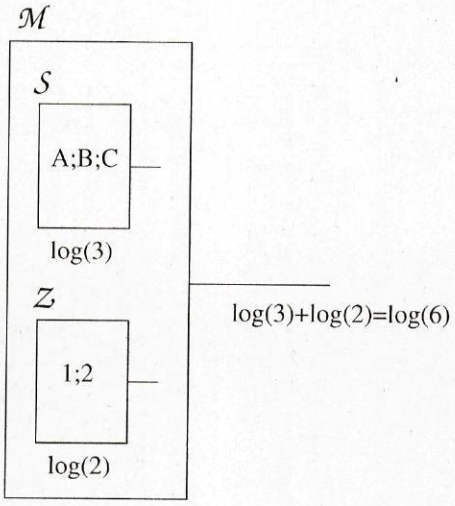

Figura 2.3: Incerteza usando o logaritmo do número de símbolos

apresentada da seguinte forma:

$$\log(X) = -\log(X^{-1}) = -\log\left(\frac{1}{X}\right)$$

Seja $p = \frac{1}{X}$, a surpresa da ocorrência do i^{esimo} símbolo, definida por analogia com $-\log(p)$, como $-\log(p_i)$.

É possível argumentar que esta medida de entropia foi obtida de forma empírica e que outras medidas mais adequadas possam existir. No entanto, supondo que se tem um conjunto de eventos possíveis com probabilidades de ocorrência $p_1, p_2, \ldots, p_n$, estas probabilidades são o único conhecimento que se tem acerca de qual evento irá ocorrer.

Pretende-se que a medida $H(p_1, p_2, \ldots, p_n)$ satisfaça as seguintes propriedades:

1. H contínua em p_i.
2. Se todos os p_i's forem iguais ($p_i = \frac{1}{n}$), então H deve ser uma função crescente e monótona em n.
3. $H(p_1, p_2, ..., p_n) = H(p_1 + p_2, ..., p_n) + (p_1 + p_2)H\left(\frac{p_1}{p_1+p_2}, \frac{p_2}{p_1+p_2}\right)$

A única medida H que satisfaz as três propriedades acima é da forma

$$H = K\sum_{i=1}^{N} p_i \log 1/p_i$$

em que K é uma constante.

Entropia Marginal

Os postulados anteriores levam-nos à definição de entropia:

Definição 2.5 *Seja X uma variável aleatória e $\mathcal{S}$ um espaço amostral, a entropia é a surpresa média de X. Shannon [27] propôs a seguinte fórmula para a definir:*

$$H(X) = -\sum_{x \in \mathcal{S}} p(x) \log p(x)$$

Além das três características anteriores, para $|\mathcal{S}| = n$, a função H tem outras propriedades que a tornam ainda mais atractiva como medida de informação.

- $H(p_1, \ldots, p_n)$ é uma função côncava em p_i.
- Para cada n, H atinge o seu único máximo para a distribuição uniforme $p_i = 1/n$.
- $H(p_1, \ldots, p_n)$ é zero se e só se algum p_i tiver valor um. O valor de entropia é zero se e só se não se ganhar nenhuma informação, isto é, se já se souber que o resultado é i.

Exemplo 2.5.1 *Seja*

$$X = \begin{cases} 1 & \textit{com probabilidade } p \\ 0 & \textit{com probabilidade } 1-p \end{cases}$$

Então,

$$H(X) = -p \log p - (1-p) \log(1-p)$$

A figura 2.4 ilustra o gráfico da função $H(X)$. A concavidade da entropia é igual a 0 quando $p = 0$ ou $p = 1$. Por outro lado, a incerteza é máxima quando $p = \frac{1}{2}$, que corresponde ao máximo valor de entropia ($H(X) = 1$).

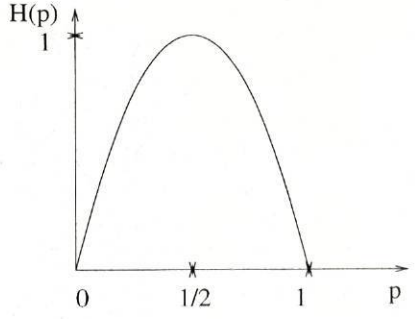

Figura 2.4: Entropia versus probabilidade

Uma mensagem é produzida por um processo estocástico que emite símbolos a_i com probabilidades p_i dadas. A entropia de um espaço amostral é definida como a eficiência com que cada uma destas mensagens pode ser transmitida.

A entropia também pode ser definida recorrendo à noção do valor esperado. Seja X uma variável aleatória com uma distribuição de probabilidade $p(x)$, então o valor esperado da variável aleatória $g(x)$ é definido como

$$E_p(g(x)) = \sum_{x \in S} g(x)p(x)$$

Tendo conhecimento sobre a distribuição de probabilidade, pode-se escrever $E(g(x))$ em vez de $E_p g(x)$. Se $g(x) = \log \frac{1}{p(x)}$, a entropia de X é interpretada como o valor esperado de $\log \frac{1}{p(x)}$ em que X é obtido de acordo com a distribuição de probabilidade $p(x)$. Então,

$$H(X) = E_p\left(\log \frac{1}{p(x)}\right)$$

Como o valor esperado de um valor positivo é sempre positivo e como a probabilidade de x varia entre 0 e 1, o logaritmo de $\frac{1}{p(x)}$ é sempre positivo. OAAssim, verifica-se que a entropia de uma variável aleatória X também é sempre positiva.

Entropia Conjunta

A definição de entropia pode ser estendida a um par de variáveis aleatórias. A partir deste ponto, incorre-se num abuso de notação relativamente à representação dos espaços amostrais.

Seja $p(x, y)$ a distribuição de probabilidade da ocorrência conjunta do evento $X = x$ e do evento $Y = y$. A *entropia conjunta* $H(X, Y)$ é definida como

$$H(X,Y) = -\sum_{x \in X}\sum_{y \in Y} p(x,y) \log p(x,y) = -E_p(\log p(x,y))$$

A entropia das variáveis aleatórias X e Y também pode ser definida à custa da probabilidade conjunta da seguinte forma:

$$\begin{aligned} H(X) &= -\sum_{x \in X}\sum_{y \in Y} p(x,y) \log \sum_{y \in Y} p(x,y) \\ H(Y) &= -\sum_{x \in X}\sum_{y \in Y} p(x,y) \log \sum_{x \in X} p(x,y) \end{aligned}$$

Das três equações anteriores, é possível concluir que

$$H(X,Y) \leq H(X) + H(Y) \tag{2.1}$$

com igualdade se e só se X e Y forem independentes.

Entropia Condicional

A *entropia condicional* de Y dado X permite analisar a informação que X tem acerca Y. Esta é definida como o valor esperado da entropia de Y para cada valor de X considerando a probabilidade desse valor.

Seja (X, Y) um par de variáveis aleatórias e $p(x, y)$ a distribuição de probabilidade conjunta, a entropia condicional $H(Y|X)$ é definida como

$$\begin{aligned} H(Y|X) &= \sum_{x\in X} p(x)H(Y|X=x) \\ &= -\sum_{x\in X} p(x)\sum_{y\in Y} p(y|x)\log p(y|x) \\ &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(y|x) \\ &= -E_p(\log p(Y|X)). \end{aligned}$$

e revela em média, a incerteza a que se está de Y para conhecermos X.

Recorrendo à definição da entropia condicional, a entropia conjunta pode ser redefinida como a entropia de X mais a entropia de Y dado X, que origina a *regra em cadeia para um par de variáveis aleatórias*

$$\begin{aligned} H(X,Y) &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y) \\ &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x)p(y|x) \\ &= -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x) - \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(y|x) \\ &= -\sum_{x\in X} p(x)\log p(x) - \sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(y|x) \\ &= H(X) + H(Y|X). \end{aligned}$$

Com base nas equações $H(X,Y) = H(X) + H(Y|X)$ e 2.1, obtém-se a seguinte desigualdade

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

que significa que o conhecimento de uma variável aleatória X não pode aumentar a incerteza de Y. De facto, a incerteza de Y vai diminuir a não ser que X e Y sejam independentes.

Note-se que a entropia condicional não é simétrica, $H(Y|X) \neq H(X|Y)$, ao contrário da entropia conjunta, $H(X,Y) = H(Y,X)$, obtendo-se

$$H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Com base na regra em cadeia para um par de variáveis aleatórias, se a variável aleatória Z for conhecida, obtém-se a seguinte igualdade

$$H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$$

o que permite deduzir a *regra da cadeia para uma sequência de n variáveis aleatórias*, $X_1, \ldots, X_n$, com distribuição de probabilidade $p(x_1, \ldots, x_n)$:

$$\begin{aligned}
H(X_1, X_2) &= H(X_1) + H(X_2|X_1) \\
H(X_1, X_2, X_3) &= H(X_1) + H(X_2, X_3|X_1) \\
&= H(X_1) + H(X_2|X_1) + H(X_3, X_2, X_1) \\
&\vdots \\
H(X_1, X_2, \ldots, X_n) &= H(X_1) + H(X_2|X_1) + \ldots + H(X_n|X_{n-1}, \ldots, X_1) \\
&= \sum_{i=1}^{n} H(X_i|X_{i-1}, \ldots, X_1) \\
&\leq \sum_{i=1}^{n} H(X_i)
\end{aligned}$$

com igualdade se e só se X_i forem independentes.

Das esquações anteriores, verifica-se que a entropia de uma sequência de variáveis aleatórias não ultrapassa a soma das suas entropias marginais, uma vez que, informação adicional não aumenta a incerteza de um conjunto de variáveis aleatórias.

Em resumo, as propriedades mais importantes da entropia são as seguintes:

- $H(X) \geq 0$ com igualdade se e só se $p_i = 1$ para um certo i.
- $H(X) \leq \log(|\mathcal{S})|$, com igualdade se e só se X for uniformemente distribuído sobre $\mathcal{S}$, ou seja se e só se $p_i = \frac{1}{|\mathcal{S}|}$ para todo o i.
- $H(X|Y) \leq H(X)$ com igualdade se e só se X e Y forem independentes.
- $H(X,Y) = H(X) + H(Y|X)$
- $H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$
- $H(X_1, X_2, ..., X_n) \leq \sum_{i=1}^{n} H(X_i)$, com igualdade se e só se as variáveis aleatórias X_i forem independentes.

2.6 Informação Mútua

A *informação mútua* $I(X;Y)$ é a redução da incerteza de X em relação ao conhecimento de Y e é definida da seguinte forma:

$$\begin{aligned} I(X;Y) &= \sum_{x\in X, y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\ &= \sum_{x\in X, y\in Y} p(x,y)\log\frac{p(x|y)}{p(x)} \\ &= -\sum_{x\in X, y\in Y} p(x,y)\log p(x) + \sum_{x\in X, y\in Y} p(x,y)\log p(x|y) \\ &= -\sum_{x\in X, y\in Y} p(x)\log p(x) - \left(-\sum_{x\in X, y\in Y} p(x,y)\log p(x|y)\right) \\ &= H(X) - H(X|Y). \end{aligned}$$

É de salientar que a informação mútua $I(X;Y)$ corresponde à intersecção da informação em X com a informação de Y.

$I(X;Y)$ é a quantidade de informação em X sobre Y e X contém tanta informação sobre Y como Y tem de X. $I(X;Y)$ é zero se e só se X e Y forem independentes. Uma vez que $H(X,Y) = H(X) + H(Y|X)$, tem-se

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

Note-se também que a informação mútua de uma variável aleatória sobre si mesma é a entropia da variável aleatória.

$$I(X;X) = H(X) - H(X|X) = H(X)$$

Teorema 2.6.1 (Informação Mútua)

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

Do teorema anterior, verifica-se que

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

de onde se retira uma propriedade muito útil e interessante da teoria de informação, que é a *simetria de informação*.

A *informação mútua condicional* entre X e Y, dada a variável aleatória Z, é definida como

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

e é igual a zero se e só se X e Y forem independentes.

Capítulo 3

Complexidade de Kolmogorov

Nos anos 60, Kolmogorov [13], Solomonoff e Chaitin [5] desenvolveram de forma independente uma medida de informação contida num objecto individual. Esta medida é baseada no menor programa (descrição) que produz o objecto, usualmente conhecida como complexidade de Kolmogorov ou complexidade algorítmica. Kolmogorov pretendia desenvolver a teoria da probabilidade e a teoria de informação. Solomonoff estava interessado na criação de um modelo de inferência. Chaitin desenvolvia o seu trabalho em complexidade de algoritmos e sequências infinitas. Percorrendo caminhos diferentes, os três chegaram ao mesmo resultado.

A complexidade de Kolmogorov estuda a quantidade de informação contida em objectos individuais, usualmente descritos por sequências binárias finitas. Para tornar a análise de complexidade de objectos individuais independente da forma como são codificados, Kolmogorov usou máquinas de Turing para representar programas que descrevem esses objectos. A existência de uma máquina de Turing universal garante que a complexidade de Kolmogorov seja um conceito objectivo, em que a complexidade não depende da máquina escolhida como referência para a medir, mas sim do objecto em causa.

Neste capítulo introduzem-se alguns fundamentos teóricos sobre computabilidade e ordens de grandeza, a definição de complexidade de Kolmogorov, bem como algumas propriedades e variantes. Para um estudo detalhado sobre computabilidade, aconselha-se a consulta de [3], [16], [33], [11] e [10], e para a complexidade de Kolmogorov as seguintes referências [17], [15] e [18].

3.1 Fundamentos Teóricos

A complexidade de Kolmogorov usa alguns modelos e métodos de computação disponibilizados pela teoria da computabilidade.

Esta teoria tem origem no trabalho de Church [7] e Turing [35]. Procura evidenciar limites nas computações e verificar a eficiência dos cálculos. O modelo computacional usado neste estudo é a máquina de Turing.

Máquinas de Turing

Antes da invenção dos computadores modernos em 1936, Alan Turing propôs um modelo teórico de uma máquina de cálculo com uma estrutura muito simples e com uma memória ilimitada. A figura 3.1 representa uma *máquina de Turing*. Esta é constituída por um controlo finito (conjunto finito de estados), uma fita dividida em células e uma cabeça de leitura/escrita que se pode deslocar ao longo da fita. Cada célula pode conter um símbolo. A máquina pode ler ou escrever símbolos na fita e pode encontrar-se num dado estado. Como a fita é infinita, usa-se um símbolo, o chamado símbolo branco, para delimitar a parte da fita que contém a informação relevante. Numa transição, dependendo do símbolo lido na fita pela cabeça e do estado do controlo finito, a máquina muda de estado; escreve um símbolo na célula que está debaixo da cabeça; move a cabeça para a esquerda ou para a direita.

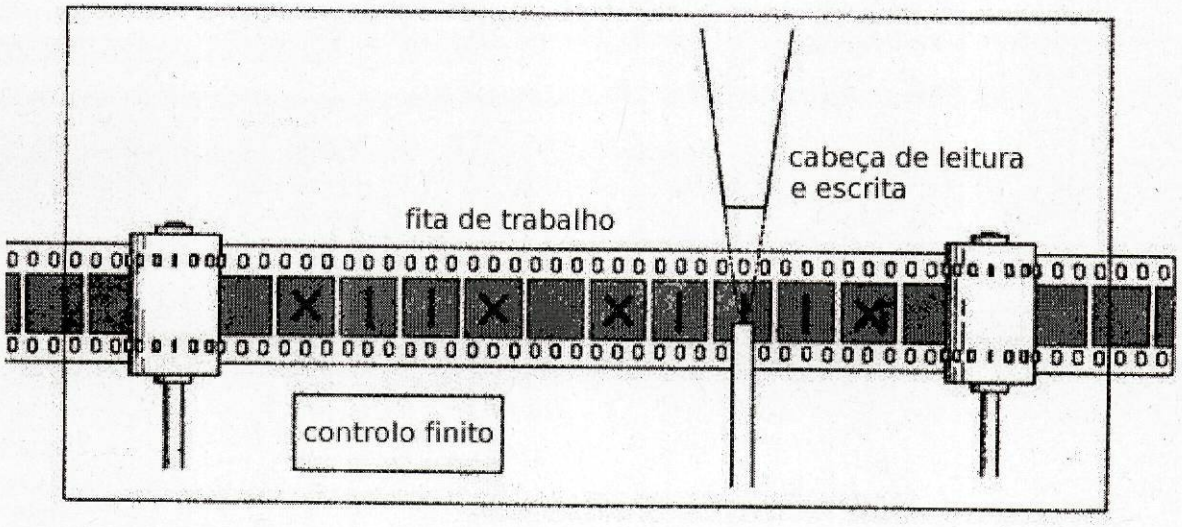

Figura 3.1: Máquina de Turing

Formalmente, uma máquina de Turing (MT) é um tuplo

$$M = (S, \Sigma, \Gamma, \delta, s_0, b, F)$$

onde

- S é um conjunto finito de estados
- Γ é o conjunto finito de símbolos da fita
- b é um símbolo de Γ, designado por branco
- Σ é um subconjunto de Γ que não inclui b, o conjunto dos símbolos de entrada
- δ é a função de transição, função parcial de $S \times \Gamma$ em $S \times \Gamma \times \{e, d\}$
- s_0 é o estado inicial
- $F \subseteq S$ é o conjunto de estados finais (de aceitação)

Os dados para a máquina de Turing são sempre uma sequência binária. A codificação em binário de um objecto O é representada por $\langle O \rangle$. A sequência x é colocada na fita de dados de entrada (um símbolo por célula) e a cabeça é posicionada na primeira célula de cada fita. É através da função de transição que a cabeça da máquina de Turing é movimentada para a célula seguinte, alterando o seu valor na fita de trabalho.

Dada uma máquina de Turing e um conjunto de dados de entrada, é efectuada uma determinada sucessão de operações que podem ou não terminar num número finito de passos.

Conjectura de Church Turing

Turing após ter estudado o modelo formal, máquina de Turing, demonstrou que qualquer processo que seja um procedimento efectivo pode ser simulado numa máquina de Turing.

A seguinte conjectura é a base de todos os desenvolvimentos não só em complexidade computacional como também em complexidade de Kolmogorov, λ-calculus e várias outras áreas em Ciência de Computadores.

Conjectura 3.1.1 (Conjectura de Church-Turing) *Todo o problema computável pode ser simulado numa máquina de Turing.*

Esta conjectura baseia-se no trabalho de Church [7], Turing [35] e Post que independentemente formularam ideias equivalentes. Church desenvolveu o lambda-calculus e conjecturou que qualquer função calculada por um algoritmo (colecção de instruções simples que vão efectuar alguma tarefa), pode ser calculada por este cálculo. Turing desenvolveu a máquina de Turing e conjecturou que todos os algoritmos podiam ser corridos nela. Post desenvolveu a máquina Post, com o objectivo de ser considerada

a máquina algorítmica universal. Assim, existe uma equivalência entre as noções de "procedimento efectivo" e"cálculo efectivo", o que significa que existe uma noção objectiva de computabilidade efectiva, independente de uma formalização em particular.

Linguagens

A linguagem aceite pela máquina de Turing M é o conjunto das palavras $x \in \Sigma^*$ aceites por M e é representada por

$$\mathcal{L}(M) = \{x \in \Sigma^* | M \text{ aceita } x\}$$

Definição 3.2 *Uma linguagem $L \subseteq \Sigma^*$ é* decidível *ou* recursiva *se e só se existir uma máquina de Turing M tal que $L = \mathcal{L}(M)$ e M pára para todos os dados.*

Ou seja, qualquer que seja $x \in \Sigma^*$, a máquina de Turing M pára quando, inicialmente, x é dada na fita, e o estado em que parou é final se e só se $x \in L$.

Definição 3.3 *Uma linguagem $L \subseteq \Sigma^*$ é* semi-decidível *ou* recursivamente enumerável *se e só se existir uma máquina de Turing M tal que $L = \mathcal{L}(\mathcal{M})$.*

Associada a qualquer linguagem $L \subseteq \Sigma^*$, a função (total) característica de L, $ch_L : \Sigma^* \to \{0,1\}^*$ é definida por:

$$ch_L(x) = \begin{cases} 1 & x \in L \\ 0 & x \notin L \end{cases}$$

Se uma linguagem L for decidida por uma máquina de Turing M, então também existe uma máquina de Turing M' que calcula a sua função característica (e vice-versa).

Se uma linguagem L for aceite por uma máquina de Turing M, então também existe uma máquina de Turing M' que calcula a sua função (parcial) semi-característica, s_L onde, s_L é definida por:

$$s_L(x) = \begin{cases} 1 & x \in L \\ \uparrow & x \notin L \end{cases}$$

isto é, M' pára com o valor 1 se M parar e M' não pára se M não parar ($s_L(x) \uparrow$ significa que a função não está definida para x).

Deste modo, verifica-se que se associa uma função parcial a cada máquina de Turing. Os dados da máquina de Turing são apresentados como n-tuplos de sequências binárias na forma de uma só sequência binária com versões auto-delimitadas dos x_i's. O inteiro representado por uma sequência binária retornado pela máquina de Turing quando

esta pára é o resultado da computação. Assim, cada máquina de Turing define uma função parcial de n-tuplos ($n \geq 1$) de inteiros em inteiros. Esta função é chamada de *recursiva parcial*. Se a máquina de Turing parar para todos os dados de entrada, então a função que está a ser calculada é definida para todos os argumentos e é denominada de *recursiva total* ou simplesmente *recursiva*.

A descrição de uma máquina de Turing M é então representada pelo n-tuplo $\langle M \rangle$. Para codificar os n-tuplos usa-se a seguinte bijecção entre $\mathbb{N}^n$ e $\mathbb{N}$:

$$\begin{aligned} p^n &: \mathbb{N}^n \to \mathbb{N} \\ p(x) &= x \\ p(x,y) &= \frac{(x+y)(x-y+1)}{2} + x \\ p^n(x_1,...,x_n) &= p(x_1, p^{n-1}(x_2,...,x_n)) \end{aligned}$$

Escreve-se $p^n(x_1,\ldots,x_n) = \langle x_1,\ldots,x_n \rangle$ e $(\langle x_1,\ldots,x_n \rangle)_i = x_i$.

Lema 3.3.1 *Para todo o n:*

- *p^n é bijectiva*
- *p^n é parcial recursiva*

A cada codificação da máquina de Turing M, $\langle M \rangle$, associa-se um número natural (índice), chamado de *número de Gödel* de M. Dado um número natural e, M_e representa máquina de Turing que lhe está associada.

Existe uma enumeração efectiva de máquinas de Turing $T_1, T_2, \ldots$ que determina uma enumeração efectiva de funções parciais recursivas $\phi_1, \phi_2, \ldots$ tal que ϕ_i é a função calculada por T_i, para todo o i.

Um resultado importante é o da existência de máquinas de Turing universais, U, que podem simular qualquer outra máquina de Turing com base no seu índice na enumeração considerada. Os dados de entrada da máquina universal U são da forma $1^i 0 p$ onde i é o índice da máquina de Turing que U está a simular, 0 é uma marca de separação e p o programa de entrada.

Indecidibilidade do problema da paragem

Existem várias questões em matemática que não podem ser respondidas por nenhum procedimento efectivo. Uma destas questões é saber se uma máquina de Turing com

um determinado dado pára - *problema da paragem*. Este problema tem um grande valor teórico pois é considerado como a linha de divisão entre o que pode ser feito numa máquina (dando memória e tempo ilimitado) e o que não pode ser feito. Um problema de decisão é *decidível* se a linguagem que o reconhece for decidida por uma máquina de Turing.

Teorema 3.3.2 *Seja M uma máquina de Turing e $w \in \Sigma^*$. A linguagem*

$$\mathcal{L} = \{\langle M, w\rangle | M \text{ é uma máquina de Turing e } M \text{ aceita } w\}$$

é indecidível.

Dem. Assume-se que $\mathcal{L}$ é decidível e obtém-se uma contradição. Sejam T e M máquinas de Turing e w uma sequência. Supõe-se que T decide $\mathcal{L}$. T com dados de entrada $\langle M, w\rangle$, pára e aceita se M aceitar w. Por outro lado, T pára e rejeita se M falhar e não aceitar w:

$$T(\langle M, w\rangle) = \begin{cases} \textit{aceita} & \text{se } M \text{ aceitar } w \\ \textit{rejeita} & \text{se } M \text{ não aceitar } w \end{cases}$$

Constrói-se uma nova máquina de Turing S com T como subrotina. S chama T que determina o que M faz quando recebe como dados a sua própria descrição ($\langle M\rangle$). Assim que S tiver determinado esta informação, faz exactamente o oposto. Isto é, rejeita se M aceitar e aceita se M não aceitar.

Inicialmente, assumiu-se que T decide $\mathcal{L}$. Então, usa-se T para construir S que, com dados $\langle M, w\rangle$, aceita quando M não aceitar w e rejeita quando M aceita w. O que contradiz a suposição acima. Logo $\mathcal{L}$ é indecidível. ◇

Ordens de Grandeza

Para exprimir e comparar o crescimento assimptótico de algumas funções em relação a outras, P. Bachman criou uma notação para lidar com este tipo de aproximações.

Sejam f, g funções de variável real, então:

- $f(x) = O(g(x))$ se existirem $c \in \mathbb{R}^+$, $x_0 \in \mathbb{N}$ tal que a desigualdade $|f(x)| \leq c|g(x)|$ verifica-se para qualquer $x \geq x_0$.
- $f(x) = o(g(x))$ se $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$
- $f(x) = \Theta(g(x))$ se $f(x) = O(g(x))$ e $f(x) = \Omega(g(x))$
- $f(x) = \Omega(g(x))$ se existe $c > 0$ tal que $|f(x)| \geq c|g(x)|$

3.4 Complexidade de Kolmogorov Clássica

A complexidade de Kolmogorov associa a quantidade de informação de um objecto individual ao comprimento da sua menor descrição. Algumas sequências binárias x, podem ser descritas usando um número de dígitos binários menor que o seu comprimento. Isto sucede quando são encontradas subsequências regulares e, neste caso, as sequências podem ser substituídas por descrições mais curtas. Estas sequências são compressíveis e são chamadas de *simples*. Pòr outro lado, as sequências que são irregulares, não podem ser comprimidas numa descrição de menor tamanho e são chamadas de *aleatórias*. Estas sequências têm uma complexidade de Kolmogorov muito próximo do seu comprimento em dígitos binários em que a sua menor descrição é a própria sequência binária.

Sem restrições ao método de descrição das sequências, estas deparam-se com um paradoxo análogo ao paradoxo de Berry:

O menor número não pode ser descrito em menos de quinze palavras.

Assim, nenhum número pode ser solução pois a própria definição de menor número já tem menos que quinze palavras. Contudo, impondo algumas restrições ao método de descrição (fazendo uso das máquinas de Turing), este paradoxo deixa de se colocar na complexidade de Kolmogorov. Mas, de todos os programas que produzem uma sequência binária, qual deles deve ser escolhido para representar a medida de complexidade? Dentro do espírito de Occam *"a explicação mais simples é a mais fiável"*, deve ser seleccionado o menor programa que produz a sequência binária como medida de complexidade.

Uma enumeração efectiva de máquinas de Turing $T_1, T_2, \ldots$, induz uma enumeração efectiva de funções parciais recursivas $\phi_1, \phi_2, \ldots$ tais que para todo o i, T_i calcula ϕ_i. Na literatura, a complexidade de Kolmogorov clássica é definida em termos de funções parciais recursivas, ou em termos de máquinas de Turing e é representada por C. Estas duas versões são, de facto, a mesma: $C_{T_i}(y) = C_{\phi_i}(y) = x$.

Definição 3.5 *Seja M uma máquina de Turing e $x, y, p \in \Sigma^*$. A complexidade de x dado y relativamente a M é definida por:*

$$C_M(x|y) = \begin{cases} min\{|p| : M(\langle p, y\rangle) = x\} & \\ \infty & \text{se não existe } p \text{ tal que } M(\langle p, y\rangle) = x \end{cases}$$

onde $\langle ., . \rangle$ é uma bijecção $\mathbb{N}^2$ em $\mathbb{N}$.

Ou seja, é o tamanho do menor programa para M que com dados y produz x. A informação de y pode ser usada pelo programa p para calcular x.

A complexidade de Kolmogorov incondicional é definida de modo análogo, em que o segundo argumento é a palavra vazia ε.

$$C_M(x) = C_M(x|\varepsilon)$$

Para efectuar comparações entre objectos, a medida de informação que os descreve deverá ser independente dos métodos de descrição utilizados. Dadas duas máquinas de Turing M_1 e M_2, a seguinte relação é válida para todo o $x \in \Sigma^*$

$$|C_{M_1}(x) - C_{M_2}(x)| \leq c_{M_1 M_2}$$

isto é, a complexidade de x em relação a duas máquinas diferentes difere de uma constante que não depende de x, mas apenas das máquinas referidas. Este resultado é a base da complexidade de Kolmogorov. Assim, o próximo teorema - *Teorema da Invariância*, demonstra que a complexidade de Kolmogorov é uma propriedade intrínseca aos objectos. Neste teorema é utilizada uma máquina de Turing universal U que simula qualquer outra máquina M. Recebe como dados a descrição da máquina M e um programa p da seguinte forma:

$$\langle i, p\rangle = \overbrace{111...1}^{i} 0p = 1^i 0p$$

existe um delimitador (o zero) que separa a descrição da máquina de Turing M e o programa p. Deste modo, a máquina de Turing universal U irá simular a execução do programa p em M_i, que é a i-ésima máquina de enumeração.

Teorema 3.5.1 (Teorema da Invariância) *Existe uma máquina de Turing universal U, que para toda a máquina de Turing M e sequência $x \in \Sigma^*$, existe uma constante c_M tal que*

$$C_U(x) \leq C_M(x) + c_M$$

Dem. Seja U uma máquina de Turing universal e seja $n = e(M)$ o índice da máquina M para uma dada enumeração e. Seja p o menor programa tal que $M(\langle p\rangle) = x$, $x \in \Sigma^*$. Pela definição de máquina de Turing universal, tem-se que $U(\langle 1^n 0p\rangle) = x$. Logo,

$$\begin{aligned} C_U(x) &\leq |1^n 0p| \\ &= |p| + n + 1 \\ &= C_M(x) + n + 1 \\ &= C_M(x) + c_M \end{aligned}$$

$\diamond$

O teorema da invariância também é válido para a complexidade condicional e pode ser expresso em termos de funções parciais recursivas.

Teorema 3.5.2 *Existe uma função parcial recursiva universal f_0 tal que para toda a função parcial recursiva f_n e sequência $x \in \Sigma^*$, existe uma constante c_{f_n} tal que*

$$C_{f_0}(x) \leq C_{f_n}(x) + c_{f_n}$$

Dem. Seja f_0 a função calculada pela máquina de Turing universal U tal que U com dados $\langle n, p\rangle$ em que $n = e(M)$, simula M, com dados $\langle p\rangle$. Isto é, se M calcula a função parcial recursiva f_n, então, $f_0(\langle n, p\rangle) = f_n(\langle p\rangle)$. Então, para todo o n,

$$C_{f_0}(x) \leq C_{f_n}(x) + c_{f_n}$$

onde $c_{f_n} = n + 1$. ◇

Os teoremas anteriores indicam que a menos de uma constante, nenhuma máquina de Turing pode ser melhor que a máquina de Turing universal. A constante c_M (c_{f_n}), embora possa ser grande, é assimptoticamente desprezável, pois não depende de x. Desta forma, se nada for dito em contrário, a complexidade de uma sequência x, refere-se à máquina de Turing universal U uma vez que esta pode simular qualquer outra. O índice de U pode ser omitido na definição de $C_U(x)$ ficando apenas $C(x)$.

Assim, se conclui que a medida de complexidade é um atributo intrínseco do objecto, o que a torna um conceito objectivo e útil. Sem esta propriedade, a complexidade de Kolmogorov seria totalmente infrutífera.

Incompressibilidade

O teorema da invariância além de mostrar que a complexidade é um atributo intrínseco ao objecto, também é uma ferramenta importante na definição de majorantes de $C(x)$, $x \in \Sigma^*$:

$$C(x) \leq |x| + c$$

O tamanho do menor programa que produz x é menor ou igual, a menos de uma constante, que o tamanho de x. Para tal, basta definir uma máquina de Turing M que copia os dados para o resultado e assim, para todo o x, tem-se $C_M(x) = |x|$. O resultado segue pelo teorema da invariância. Também se verifica que

$$C(x|y) \leq C(x) + c$$

que significa que informação adicional não aumenta a complexidade de uma sequência. Para demonstrar a desigualdade acima, define-se uma máquina de Turing T tal que para quaisquer y e z, a máquina com dados $\langle z, y\rangle$ retorna x se e só se a máquina

universal de referência, U, calcular x com dados $\langle z, \epsilon \rangle$. Então, $C_T(x|y) = C(x)$ e pelo teorema da invariância, existe uma constante c tal que

$$\begin{aligned} C(x|y) &\leq C_T(x|y) + c \\ &= C(x) + c \end{aligned}$$

Definição 3.6 *Para toda a constante c, uma sequência binária x é* c-incompressível *se*

$$C(x) \geq |x| - c$$

Algumas sequências binárias podem ser bastante compressíveis, contudo, a maioria das sequências binárias são incompressíveis, como se verifica no próximo teorema.

Teorema 3.6.1 (Teorema da Incompressibilidade) *Seja c um inteiro positivo e* $y \in \Sigma^*$ *fixo. Qualquer conjunto finito A com cardinalidade m tem pelo menos* $m(1 - 2^{-c}) + 1$ *elementos x com* $C(x|y) \geq \log m - c$.

Dem. O número de programas de tamanho menor que $\log m - c$ é

$$\sum_{i=0}^{\log m - c - 1} 2^i = 2^{\log m - c} - 1$$

Então, existem pelo menos $m - m2^{-c} + 1$ elementos em A que não têm nenhum programa de comprimento menor que $\log m - c$. ◇

É possível determinar quantas sequências binárias de tamanho n são c-incompressíveis. Sabe-se que do total de 2^n sequências binárias de tamanho n tem-se $2^{n-c} - 1$ que são simples. O número de sequências binárias de tamanho n que são c-incompressíveis é $2^n - 2^{n-c} + 1$. A fracção de sequências binárias de comprimento n que são c-incompressíveis no total de 2^n sequências binárias é, para um n suficientemente grande, $1 - 2^{-c}$. Para $1 < c < n$ e para n suficientemente grande, $1 - 2^{-c}$ tende para 1. Logo, a maioria das sequências binárias são incompressíveis.

Exemplo 3.6.1 ***Existe um número infinito de primos.***
Supõe-se que existe um número finito de primos, seja $p_1, \ldots, p_k$ *a lista de todos os primos, para algum* $k \in \mathbb{N}$. *Seja* $m = p_1^{e_1}, \ldots, p_k^{e_k}$ *uma sequência aleatória de tamanho n que é descrita por* $\langle e_1, \ldots, e_k \rangle$. *Cada um dos expoentes é menor que o logarítmo de m, então podem ser descritos usando* $\log \log m$ *de símbolos. Deste modo*

$$|\langle e_1, \ldots, e_k \rangle| \leq 2k \log \log m$$

Como $m \leq 2^{n+1}$ *tem-se,*

$$|\langle e_1, \ldots, e_k \rangle| \leq 2k \log(n + 1)$$

então,

$$C(m) \leq 2k \log(n+1) + c.$$

Para um n grande, isto contradiz $C(m) \geq n$*, que segue a partir do facto de que m é aleatório.*

Algumas propriedades de C

Teorema 3.6.2 *A complexidade C não é subaditiva nos seus argumentos.*

Dem. Seja $\langle . \rangle : N \times N \rightarrow N$ uma bijecção recursiva sobre os números naturais que codificam x e y como $\langle x, y \rangle$. Define-se $C(x, y) = C(\langle x, y \rangle)$ como o comprimento do menor programa tal que U calcula x e y e um modo de as separar. Sejam p um programa mínimo que produz x e q um programa mínimo que produz y. Então, existe uma máquina de Turing T que simula os dois programas e que produz x seguido de y. No entanto, qualquer T terá que saber onde é que tem que dividir os seus dados para identificar p e q. Uma forma de fazer isto é usar os dados $|\bar{p}|pq$ ou $|\bar{q}|qp$. Desta forma, para todo o x, y, tem-se

$$C(x, y) \leq C(x) + C(y) + 2 \log(\min(C(x), C(y)))$$

Nesta desigualdade, o termo logarítmico não pode ser eliminado, a não ser que surja como informação adicional o tamanho do menor programa que gera x. Neste caso ficaria $C(x, y|C(x)) \leq C(x) + C(y) + O(1)$. ◇

Incomputabilidade de C

O problema da paragem não é computável uma vez que não é possível verificar quando é que uma máquina de Turing termina com um determinado conjunto de dados. Pelo mesmo motivo, a complexidade de Kolmogorov também não é computável.

Teorema 3.6.3 *A complexidade de Kolmogorov não é computável.*

Dem. Supõe-se que existe um programa ENCONTRAC que dado s como dados calcula $C(s)$ que é constituído por 10^9 dígitos binários. Constrói-se o programa UPSS da seguinte forma:

```
inicio:  gera proximo s
usa ENCONTRAC para calcular C(s)
se C(s)< 10^12 volta ao inicio
escreve s
pára
```

O programa UPSS é um pouco maior que o programa ENCONTRAC. Quando UPSS pára e escreve s, esta sequência binária tem $C(s) \geq 10^{12}$ (calculada por ENCONTRAC) que é muito maior do que UPSS. Isto é uma contradição porque, por definição, nenhum programa menor do que $C(s)$ pode gerar s. Pode-se concluir que ENCONTRAC não pode existir e, consequentemente, a complexidade de Kolmogorov não é computável. ◇

Considere-se a relação entre o problema da paragem e a complexidade de Kolmogorov: seja PARAGEM o programa que verifica se um dado programa pára ou não. Supondo que o programa PARAGEM existe, então também existe o programa ENCONTRAC. Mas como já se verificou, o programa ENCONTRAC não existe e assim, também não existe o programa PARAGEM.

Simetria de Informação em C

Kolmogorov [14] e Levin [1] independentemente provaram um dos resultados mais poderosos na complexidade de Kolmogorov: a *simetria de informação*. A simetria de informação surge a partir da informação algorítmica (mútua) entre dois objectos que quantifica a informação que um objecto tem acerca do outro.

Definição 3.7 *A informação algorítmica de y contida em x é*

$$I_C(x : y) = C(y) - C(y|x)$$

Uma vez que $C(y) \geq C(y|x) + O(1)$, tem-se $I_C(x : y) \geq 0$.

Seja $C(x, y) = C(\langle x, y \rangle)$. Ou seja, a menos de uma constante, $C(x, y)$ é o comprimento do menor programa em que U calcula x e y e um modo de os separar. $\langle x, y \rangle$ pode ser descrito usando uma descrição pequena de y, uma descrição pequena de x dado y e uma indicação de separação dos dois termos. Então por [18], tem-se $C(x, y) \leq C(y) + C(x|y) + 2|C(x)| + O(1)$, que originou o seguinte teorema:

Teorema 3.7.1 *Qualquer que seja a sequência x e y,*

$$C(x, y) = C(x) + C(y|x) + O(\log(C(x, y)))$$

Uma vez que $C(x, y) = C(y, x)$, a menos de um termo logarítmico, o seguinte corolário verifica-se:

Corolário 3.7.2 (Simetria de Informação) *Qualquer que seja a sequência x e y, a seguinte igualdade verifica-se a menos de um termo logarítmico*

$$C(x) - C(x|y) = C(y) - C(y|x)$$

Ou seja,

$$|I_C(x:y) - I_C(y:x)| = O(\log C(x,y))$$

Logo pode-se concluir que, a menos de um termo logarítmico, a informação de x contida y é igual à quantidade de informação de y contida em x.

3.8 Complexidade de Kolmogorov Livre de Prefixo

Solomonoff inicialmente introduziu a complexidade algorítmica com o objectivo de atribuir uma probabilidade prévia universal a cada sequência binária finita. Escolhendo a máquina de Turing de referência U (tal como no teorema da invariância), induziu uma distribuição de probabilidade p sobre $\mathbb{N}$ (equivalentemente $\{0,1\}^*$) definida por $p(x) = \sum 2^{-|q|}$, onde a máquina U pára com resultado x, para todos os dados q. Infelizmente, p não é uma distribuição de probabilidade, uma vez que a série $\sum_x p(x)$ diverge e, para cada x, $p(x) = \infty$. Mais tarde, redefiniram $p(x)$ considerando apenas o programa mais pequeno que calcula x: $p(x) = 2^{-C(x)}$. No entanto, $\sum_x p(x)$ continua a divergir e p continua a não ser uma distribuição de probabilidade. Esta divergência segue da divergência das séries harmónicas $\sum_x 1/x$.

Levin e mais tarde Chaitin salvaram a ideia de Solomonoff. Considerando apenas os programas livres de prefixo, criaram a *complexidade de Kolmogorov livre de prefixo*. Esta nova variante de complexidade de Kolmogorov representada por K usa *máquinas de Turing livres de prefixo* que aceitam linguagens A tal que para qualquer $x, y \in A$, se $x \neq y$, então x não é um prefixo de y e y não é um prefixo de x.

Uma máquina de Turing livre de prefixo é uma máquina com uma fita de dados, algumas fitas de trabalho e uma fita de resultados. A cabeça de leitura só pode ler da esquerda para a direita. Em cada passo, a máquina tem um dos seguintes passos de execução:

1. lê um bit dos dados e move a cabeça para a direita
2. retorna o resultado e pára
3. entra em ciclo infinito

Existe uma enumeração efectiva $T_1, T_2, \ldots$ de máquinas livres de prefixo que calculam exactamente as funções parciais recursivas livres de prefixo $\phi_1, \phi_2, \ldots$.

Tal como na complexidade de Kolmogorov clássica, a complexidade livre de prefixo é definida em termos de funções parciais recursivas, ou em termos de máquinas de Turing. Estas duas versões são, de facto, a mesma: $K_{T_i}(y) = K_{\phi_i}(y) = x$.

As máquinas de Turing livres de prefixo e nomeadamente as universais de referência U, só aceitam programas p tal que nenhum prefixo de p é um programa aceite por essas máquinas. Com o uso de codificações livres de prefixo, pode-se agora concatenar descrições sem ser necessário marcar onde uma descrição termina e a outra começa.

Teorema 3.8.1 *As funções C e K são assimptoticamente iguais. Para todos os $x, y \in \Sigma^*$ tem-se, a menos de uma constante*

$$\begin{aligned} C(x|y) &\leq K(x|y) \\ &\leq C(x|y) + 2\log C(x|y) \end{aligned}$$

Dem. $C(x)$ é menor que $K(x)$, uma vez que a definição de $C(x)$ não tem a informação necessária para tornar o programa livre de prefixo. O mesmo se verifica para a complexidade condicional $C(x|y) \leq K(x|y)$. Para se provar a segunda desigualdade, usa-se o facto de que, para todo o x e $y \in \Sigma^*$, a máquina de referência da complexidade C calcula x com dados $\langle y, p\rangle$ tal que $|p| = C(x|y)$. Sabe-se que $|\bar{p}|p$ é a codificação livre de prefixo para p, então, $K(x|y) \leq C(x|y) + 2|C(x|y)| + c$. ◇

Simetria de informação em K

Para esta medida de complexidade, a simetria de informação também se verifica a menos de um termo logarítmico. A informação algorítmica com base na complexidade livre de prefixo é representada por I.

$$I(x : y) - I(y : x) = O(\log(K(x, y)))$$

3.9 Distribuição Universal

A distribuição universal algorítmica foi originalmente estudada por R.J. Solomonoff em 1964, com o objectivo de prever os próximos símbolos de um prefixo finito numa sequência binária infinita.

A distribuição universal combina três princípios fundamentais para a formulação de hipóteses: princípio da indiferença, princípio de Occam e regra de Bayes.

- **Princípio da Indiferença:** Guardar todas as hipóteses que são consistentes com os factos.
- **Princípio de Occam:** De todas as hipóteses consistentes com os factos, escolher a mais simples (este princípio é equivalente à escolha do menor programa que produz uma dada sequência).
- **Regra de Bayes:** A probabilidade de uma hipótese ser verdadeira é proporcional à probabilidade prévia das hipóteses, multiplicada pela probabilidade dos dados observados, assumindo que a hipótese era verdadeira.

Seja $\mathcal{X} = \mathcal{N}$ ou equivalentemente a $\{0,1\}^*$. A função $f : \mathcal{X} \to [0,1]$ é uma função de densidade de probabilidade se $\sum_{x\in\mathcal{X}} f(x) = 1$. É considerada uma sub-função de densidade de probabilidade se $\sum_{x\in\mathcal{X}} f(x) \leq 1$.

As distribuições de probabilidade em conjunto finitos $\mathcal{X}$ são identificadas com as suas correspondentes funções de densidade de probabilidade que também são chamadas *medidas de probabilidade*. Uma sub-função de densidade de probabilidade é designada por *semi-medida* de probabilidade.

Levin mostrou que se pode enumerar de um modo efectivo todas as distribuições de probabilidade enumeráveis $p_1, p_2, \ldots$. Em particular, existe uma distribuição de probabilidade universal enumerável, conhecida por $\mathbf{m}$ tal que

$$\forall_{k\in\mathcal{K}}\exists_{c\in\mathcal{Q}}\forall_{x\in\mathcal{N}}[c\mathbf{m} \geq p_k(x)]$$

Isto é, $\mathbf{m}$ domina multiplicativamente cada p_k. Em [18], os autores demonstram que a função $\mathbf{m}$ não é recursiva e que é uma semi-medida de probabilidade.

Probabilidade prévia

Seja $T_1, T_2, \ldots$ uma enumeração efectiva padrão de máquinas de Turing livres de prefixo. Seja x uma sequência, $Q(x)$ é definida como a probabilidade de uma máquina de Turing, com dados aleatórios parar e retornar x como resultado. A probabilidade de uma máquina de Turing parar recebendo como dados o programa p é $2^{-|p|}$. Então, para cada máquina T,

$$Q_T(x) = \sum_{T(p)=x} 2^{-|p|}$$

Nota 3.10 *Neste contexto, é necessário o uso de máquinas livres de prefixo. Pela desigualdade de Kraft, a série acima converge* (≤ 1) *ao serem apenas considerados os programas que páram numa máquina de Turing livre de prefixo. Assim, o somatório das probabilidades prévias não excede a unidade, sendo considerada uma semi-medida de probabilidade.*

Teorema 3.10.1 *Qualquer que seja a máquina de Turing A e sequência $x \in \Sigma^*$,*

$$Q_U(x) \geq c' Q_A(x)$$

Dem. Seja p' um programa para a máquina de Turing livre de prefixo A que imprime x. Pelo teorema da invariância, existe um programa p para a máquina de Turing universal U de tamanho menor que $|p'| + c_A$. Então, tem-se

$$Q_U(x) = \sum_{p:U(p)=x} 2^{-|p|} \geq \sum_{p':A(p')=x} 2^{-|p'|-c_A} = c'_A Q_A(x)$$

◇

Probabilidade Algorítmica

Um objecto é simples se puder ser brevemente descrito. O tamanho da descrição do objecto depende do método utilizado para o descrever. A menor descrição efectiva auto-delimitada de um objecto x é quantificada por $K(x)$. Esta descrição origina uma noção invariante da probabilidade algorítmica que é definida por:

$$R(x) = 2^{-K(x)}$$

Afirmar que um objecto é mais simples que outro é o mesmo que dizer que esse objecto tem maior probabilidade algorítmica.

Teorema da Codificação

Este teorema estabelece uma relação estreita entre a semi-medida discreta universal $\mathbf{m}(x)$, a probabilidade prévia universal $Q_U(x)$ e a probabilidade algorítmica $R(x)$.

Teorema 3.10.2 (Teorema da Codificação) *Existe uma constante c tal que para qualquer sequência x,*

$$-\log \mathbf{m}(x) = -\log Q_U(x) = K(x)$$

Esta igualdade verifica-se a menos de uma constante aditiva c.

Dem. Por definição, tem-se $2^{-K(x)} \leq Q_U(x)$, para todo o x. $Q_U(x)$ é enumerável, então enumeram-se todos os programas para x na máquina de referência U.

Pelo facto de $\mathbf{m}(x)$ ser universal na classe das semi-medidas enumeráveis discretas, obtém-se $Q_U(x) = O(\mathbf{m}(x))$. Falta demonstrar que $\mathbf{m}(x) = O(2^{-K(x)})$, que é equivalente a demonstrar que $K(x) \leq -\log \mathbf{m}(x) + O(1)$. Seja E um código prefixo tal que $|E(x)| \leq -\log P(x) + 2$. Codifica-se cada sequência de naturais x como $E(x) = p$, satisfazendo a desigualdade

$$|p| \leq -\log m(x) + O(1)$$

e usando a máquina livre de prefixo A tal que $A(p) = x$. Então, $K_A(x) \leq |p|$, logo, pelo teorema da invariância, $K(x) \leq K_A(x) + O(1)$. Assim, a menos de constantes aditivas, obtém-se o resultado pretendido

$$-\log Q_U(x) = -\log \mathbf{m}(x) = K(x)$$

◇

O teorema da codificação demonstra que a semi-medida enumerável discreta universal $\mathbf{m}$ e a distribuição de probabilidade prévia universal Q_U são iguais a menos de uma constante multiplicativa. Assim, a distribuição de probabilidade $\mathbf{m}$ pode ser considerada como a probabilidade prévia de objectos finitos na ausência de qualquer conhecimento acerca deles. As três formalizações distintas apresentadas acima, definem a mesma noção de probabilidade universal.

3.11 Complexidade de Kolmogorov mínima

A complexidade de Kolmogorov clássica C e a livre de prefixo K não são aditivas. No entanto, existe uma variante, a complexidade de Kolmogorov *mínima* que possui esta característica.

Teorema 3.11.1 *Sejam x, y sequências finitas. Então, a menos de uma constante aditiva,*

$$K(x,y) = K(x) + K(y|x, K(x))$$

É de notar que $K(x,y) = K(x, K(x), y) + O(1)$ e $K(x, K(x)) = K(\langle x, K(x)\rangle)$, então substituindo no teorema anterior, obtém-se o seguinte corolário:

Corolário 3.11.2 *A complexidade K é aditiva da seguinte forma:*

$$K(\langle x, K(x)\rangle, y) = K(\langle x, K(x)\rangle) + K(y|\langle x, K(x)\rangle) + O(1)$$

A medida de informação é simétrica se a informação em x sobre y for igual (a menos de constantes aditivas) à informação de y sobre x. Reescrevendo $K(x,y)$ segue-se do teorema anterior que

$$K(y) - K(y|x, K(x)) = K(x) - K(x|y, K(y)) + O(1)$$

Teorema 3.11.3 *A simetria de informação para a complexidade K é obtida do seguinte modo:*

$$I(\langle x, K(x)\rangle : y) = I(\langle y, K(y)\rangle : x) + O(1)$$

No entanto, não se pode substituir $\langle x, K(x)\rangle$ por $K(x)$ e $\langle y, K(y)\rangle$ por $K(y)$ no teorema anterior. Dado $\langle x, K(x)\rangle$, podem-se enumerar todos os programas mínimos para x. O primeiro encontrado é conhecido por x^*. A partir deste programa pode-se calcular $\langle x, K(x)\rangle$. Então, x^* e $\langle x, K(x)\rangle$ contêm a mesma informação apesar de as sequências binárias serem diferentes. Assim, pode-se substiuir x^* por $\langle x, K(x)\rangle$. A complexidade de um programa mínimo x^*, $K(x^*)$, é considerada, segundo Chaitin [6] como a *complexidade de Kolmogorov mínima* e representa-se por $K_c(x)$.

Definição 3.12 *Sejam $x, y \in \Sigma^*$, a complexidade de Kolmogorov mínima de x dado y é defininda como*

$$K_c(x|y) = K(x|y^*)$$

É fácil verificar que para qualquer sequência binária x,

$$K_c(x) = K(x) \text{ e } K_c(x,y) = K(x,y)$$

Substituindo no corolário 3.11.2, $\langle x, K(x)\rangle$ pelo programa mínimo x^* obtém-se

$$K_c(x,y) = K_c(x) - K_c(y|x) + O(1),$$

isto é, K_c é aditiva.

Definição 3.13 *Sejam $x, y \in \Sigma^*$, a informação algorítmica entre x e y é definida como*

$$I(x;y) = K_c(y) - K_c(y|x)$$

Com os resultados anteriores, obtém-se o seguinte teorema:

Teorema 3.13.1 (Simetria de Informação) *Sejam $x, y \in \Sigma^*$,*

$$I(x:y) = K_c(x) + K_c(y) - K_c(x,y) + O(1) = I(y:x) + O(1)$$

3.14 Complexidade de Kolmogorov com Recursos Limitados

A complexidade de Kolmogorov mede a quantidade de informação de uma sequência binária. Como já se verificou, esta medida não é computável. No entanto, pode-se aproximar por excesso, e a forma mais simples de o fazer é impondo limites no tempo de execução dos programas, quanto maior o tempo melhor será a aproximação. Ao ser considerado um limite no tempo, o momento em que a máquina de Turing irá parar passa a ser conhecido e o problema da paragem fica resolvido, premiando a complexidade de Kolmogorov com computabilidade.

Seja T uma máquina de Turing com múltiplas fitas de trabalho, uma fita de dados e uma fita de resultados. Considere-se a seguinte enumeração efectiva $\phi_1, \phi_2, \ldots$ de funções parciais recursivas tal que T_ϕ é a máquina que calcula ϕ.

Para $x, y \in \Sigma^*$ e $|x| = n$, define-se $t(n)$ como a função que representa o número de passos efectuados no limite de tempo t. Se $T_\phi(y) = x$ em $t(n)$ passos (tempo), então esta informação também é representada por $T^t_\phi(y) = x$ ou $\phi^t(y) = x$. Aqui também existe a bijecção entre os números naturais e o conjunto finito de sequência binárias.

Definição 3.15 *Sejam $x, y \in \Sigma^*$ e T uma máquina de Turing com múltiplas fitas. A complexidade de Kolmogorov com limites no tempo C^t_T de x condicionada a y é definida por*

$$C^t_T(x|y) = \min\{|p| : T(\langle p, y\rangle) = x \text{ em } t(n) \text{ passos}\}$$

e por $C^t_T(x|y) = \infty$ se não houver p.

A complexidade incondicional de x com limites no tempo é definida por

$$C_T^t(x) = C_T^t(x|\epsilon)$$

Note-se que se $t(n)$ for uma função total recursiva, a função C_T^t também é total recursiva.

O teorema da invariância é válido para esta medida de complexidade. No entanto, ao adicionar recursos limitados à complexidade, as propriedades do teorema da invariância tornam-se consideravelmente mais fracas.

Teorema 3.15.1 (Teorema da Invariância) *Existe uma máquina de Turing universal U tal que para qualquer outra máquina de Turing M, existe uma constante c tal que*

$$C_U^{ct\log t,c}(x|y) \leq C_M^t(x|y) + c$$

para todo o x e y. A constante c depende de M mas não de x e y.

A prova deste teorema é semelhante aos outros teoremas da invariância e pode ser encontrada em [18].

Capítulo 4

Segurança Absoluta e Teoria de Informação

Segundo Kahn [12], a sociedade chegou a um ponto em que, provavelmente influenciada pelo aumento da literacia, fez crescer de uma forma espontânea a criptografia. O modo de vida do ser humano começou a necessitar cada vez mais de privacidade dentro do meio em que vive. Juntamente com o crescimento desta necessidade surgiu a vontade de descobrir os segredos de outrém - criptoanálise. Daqui o aparecimento da criptologia que constroi novas formas de codificar informação e de quebrar a privacidade dos sistemas existentes. Com este interesse de descobrir informação escondida, a *segurança* torna-se num conceito predominante e incrivelmente necessário. O principal objectivo deste capítulo é demonstrar a *segurança absoluta* com base na teoria de informação, [21] de alguns protocolos criptográficos de chave simétrica, nomeadamente, o *one time pad*, o *segredo partilhado* e a *autenticação*.

Para uma breve mas rigorosa introdução sobre criptologia, consultar [19]. Para um conhecimento mais aprofundado sobre os temas acima referidos, consultar [12] para a história da criptografia, [9] e [24] para os protocolos criptográficos de chave simétrica e [34] para as demonstrações de segurança absoluta. Para uma visão breve e geral sobre a criptografia de hoje e o que será amanhã consultar [22].

4.1 Fundamentos Teóricos

O principal objectivo da criptografia é permitir que um emissor consiga comunicar com um receptor de um modo seguro sem que nenhum adversário consiga saber o que está a ser transmitido. A criptografia é usada no desenho de sistemas criptográficos, enquanto que a criptoanálise é usada para a quebra de sistemas. A criptologia (palavra de origem grega que significa oculto) é o termo que engloba a criptografia e a criptoanálise e é o escolhido para representar a àrea da comunicação em que

o segredo é o factor fundamental. A criptoanálise teve um papel predominante nas duas guerras mundiais. A criptografia até à segunda guerra mundial foi encarada mais como uma arte do que como uma ciência, tendo sido usada para fins militares e incidia exclusivamente na codificação de informação. Foi devido à definição de entropia e de segurança absoluta de um sistema criptográfico proposto por Shannon [27],[28], que a criptografia passou de arte a ciência. Shannon não só provou a segurança de alguns protocolos criptográficos já existentes, como também estabeleceu limites de segurança no tamanho da chave a ser comunicada. No entanto, a grande explosão em criptografia deu-se com a descoberta dos sistemas criptográficos de chave pública devido a Diffie e Hellman [8]. Os autores demonstraram que uma comunicação pode ser segura sem a transmissão prévia da chave secreta entre o emissor e o receptor. Estes sistemas de chave pública continuam inquebráveis até aos dias de hoje, embora não possuam nenhuma prova formal de segurança.

Segundo Maurer [22], a criptografia transformou-se tanto numa ciência matemática fascinante como também na chave tecnológica para a sociedade de informação em expansão, com uma forte relação entre a teoria e a prática.

Sistemas Criptográficos de Chave Simétrica

Um *sistema criptográfico* é um protocolo (sequência específica de acções que acompanham uma determinda tarefa, em que dois ou mais intervenientes executam cooperativamente) que permite que duas entidades (um emissor e um receptor) comuniquem entre eles de um modo seguro. Consiste na aplicação de algoritmos à informação (mensagem original) que é enviada de uma entidade à outra.

Formalmente um sistema criptográfico é definido da seguinte forma:

Definição 4.2 *Um sistema criptográfico de chave simétrica é um tuplo de cinco elementos* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$*, onde* $\mathcal{P}$ *é o conjunto das mensagens,* $\mathcal{C}$ *é o conjunto das cifras,* $\mathcal{K}$ *é o conjunto das chaves,* $e : \mathcal{P} \times \mathcal{K} \to \mathcal{C}$ *é o algoritmo de cifra,* $d : \mathcal{C} \times \mathcal{K} \to \mathcal{P}$ *é o algoritmo de decifra e* $d(e(m,k),k) = m$.

Os sistemas criptográficos de chave simétrica baseiam-se na troca prévia, através de um canal seguro, de uma chave entre o emissor (Alice) e o receptor (Bob) (ver figura 4.1). Esta chave é utilizada para codificar a mensagem original, que Alice pretende transmitir, dando origem à mensagem cifrada. Esta mensagem é enviada a Bob através de um canal inseguro e ele usando a chave préviamente acordada, consegue decifrá-la recuperando a mensagem original. Se o adversário (Oscar) conseguir interceptar a mensagem cifrada, não a conseguirá decifrar pois não tem em sua posse a chave acordada. Neste caso, terá que usar a criptoanálise e efectar um ataque ao sistema.

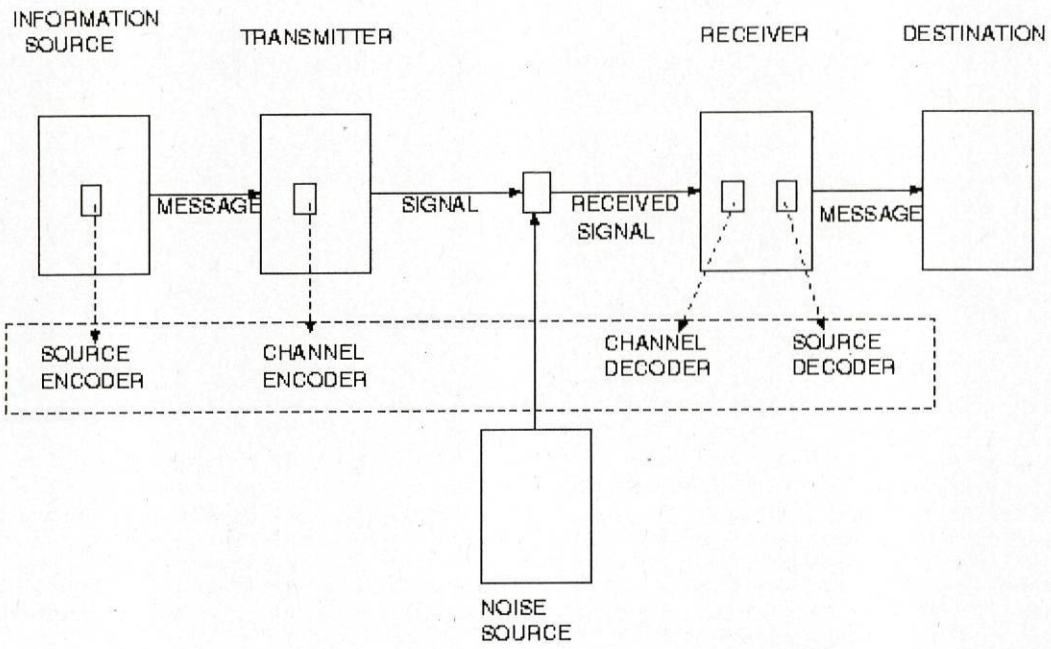

Figura 4.1: "A Mathematical Theory of Communication" C. E. Shannon

Segurança

"History has taught us: never understimate the amount of money, time, and effort someone will expend to thwart a security system. It's always better to assume the worst. Assume your adversaries are better than they are. Assume science and technology will soon be able to do things they cannot yet. Give yourself a margin for error. Give yourself more security than you need today. When the unexpected happens, you'll be glad you did"

Bruce Schneier "Why Cryptography is harder than it looks" (Counterpane Systems, 1997)

Tal como Schneier afirma em [25], deverá ser sempre assumido na criação de sistemas criptográficos e na elaboração de provas de segurança que o adversário conhece o sistema criptográfico em causa - *Princípio de Kerckhoff.* Se o adversário não tiver qualquer conhecimento sobre o sistema, a tarefa de descobrir a mensagem original torna-se mais difícil. No entanto, não se pretende que a base de segurança de um sistema criptográfico seja a falta de conhecimento do adversário sobre o sistema, mas sim criar um sistema criptográfico seguro, mesmo que os adversários tenham um total conhecimento sobre este.

Segundo Maurer [22], a expansão das infrastruturas da *informação* tem um impacto dramático na economia e na sociedade em geral. A informação torna-se num dos recursos mais importantes no dia-a-dia, mas contém propriedades indesejáveis: pode ser apagada sem deixar rasto, copiada sem qualquer custo e o seu valor pode ser

mudado rapidamente dependendo do contexto e do tempo. Pretende-se proteger este recurso, em particular, quando é considerada o aumento da sua exposição a todo o tipo de riscos e ataques, tais como: acesso ilegítimo, modificação e inserção de mais informação. Existem três requisitos básicos de segurança da informação:

1. escolha da informação, isto é, o conteúdo da mensagem (confidencialidade), a identidade de um utilizador (anonimato), ou a existência de informação ("streganography")

2. autenticação da informação

3. controlo do acesso à informação

Um dos problemas fundamentais da segurança de informação é a distinção entre informação boa e má, por exemplo, entre boas ou más aplicações informáticas, anexos de correio electrónico, ou entre bom ou mau tráfego na rede. Este problema de distinção (decidir se se trata de boa ou má informação) pode não ter uma solução clara. Assim, a formalização da segurança de informação tornou-se num campo muito importante e extremamente necessário. Este é um dos expoentes máximos de investigação em criptografia. Tal como descrito em [22] e [34], a segurança de um sistema criptográfico depende de várias suposições e como referido acima por Schneier, deverá ser sempre assumido o pior cenário, em que o adversário tem um conhecimento global do sistema criptográfico em causa e que poderá ter um poder computacional ilimitado.

Existem dois tipos de segurança, a *segurança computacional* e a *segurança incondicional ou absoluta.*

Segurança Computacional

Mede o poder computacional necessário para quebrar um sistema criptográfico.

Definição 4.3 *Um sistema criptográfico é* ***computacionalmente seguro*** *se não puder ser quebrado com os recursos existentes (actuais ou futuros).*

A segurança destes sistemas assenta na não exequibilidade da computação de os quebrar. Uma vez que não existe nenhuma prova da dificuldade computacional de um problema, a segurança computacional assenta sobre suposições computacionalmente intractáveis. Reduzir este número de suposições e requerimentos de modo a atingir um certo nível de segurança, é um dos principais objectivos da investigação em criptografia.

Segurança Incondicional ou Absoluta

Representa a segurança de um sistema criptográfico quando não existe nenhum limite no poder computacional do adversário.

Definição 4.4 *Um sistema criptográfico é considerado* ***incondicionalmente seguro*** *se não puder ser quebrado por um adversário com poder computacional ilimitado.*

Neste tipo de sistemas, o adversário não consegue obter qualquer informação sobre a mensagem original mesmo observando a mensagem cifrada.

Este tipo de segurança é o objecto de estudo deste trabalho.

Supõe-se que existe uma distribuição de probabilidade no espaço das mensagens originais, $\mathcal{P}$. A probabilidade prévia de que a mensagem original x ocorra é representada por $p_{\mathcal{P}}(x)$. Também se assume que a chave k seja escolhida (pela Alice e pelo Bob) usando uma distribuição de probabilidade fixa (normalmente, a chave é escolhida usando a distribuição uniforme, e deste modo, todas as chaves são equiprováveis). Cada chave irá ser usada uma só vez. A probabilidade da chave k ser escolhida é representada por $p_{\mathcal{K}}(k)$. Note-se que a chave é escolhida antes da Alice saber qual a mensagem original. Então assume-se que a seleção da chave k e da mensagem original x são eventos independentes.

As duas distribuições de probabilidade em $\mathcal{P}$ e em $\mathcal{K}$ induzem uma distribuição de probabilidade em $\mathcal{C}$. De facto, não é difícil calcular a probabilidade $p_{\mathcal{C}}(y)$ em que y é a mensagem cifrada que é transmitida. Para uma chave $k \in \mathcal{K}$, tem-se

$$M(k) = \{e_k(x) : x \in \mathcal{P}\}$$

que representa o conjunto de todas as mensagens cifradas. Então, para todo o $y \in \mathcal{C}$, tem-se

$$p_{\mathcal{C}}(y) = \sum_{\{k:y \in M(k)\}} p_{\mathcal{K}}(k) p_{\mathcal{P}}(d_k(y))$$

Para cada $y \in \mathcal{C}$ e $x \in \mathcal{P}$, pode-se calcular a probabilidade condicional $p_{\mathcal{C}}(y|x)$ (probabilidade de y ser a mensagem cifrada conhecendo a mensagem original x) da seguinte forma:

$$p_{\mathcal{C}}(y|x) = \sum_{\{k:x=d_k(y)\}} p_{\mathcal{K}}(k)$$

Usando o teorema de Bayes, pode-se calcular a probabilidade condicional $p_{\mathcal{P}}(x|y)$ (probabilidade de x ser a mensagem original, conhecendo a respectiva mensagem cifrada):

$$p_{\mathcal{P}}(x|y) = \frac{p_{\mathcal{P}}(x) \sum_{\{k:x=d_k(y)\}} p_{\mathcal{K}}(k)}{\sum_{\{k:y \in M(k)\}} p_{\mathcal{K}}(k) p_{\mathcal{P}}(d_k(y))}$$

Do teorema de Bayes, verifica-se que a condição de que $p_{\mathcal{P}}(x|y) = p_{\mathcal{P}}(x)$, para todo o $x \in \mathcal{P}, y \in \mathcal{C}$ é equivalente a $p_{\mathcal{C}}(y|x) = p_{\mathcal{C}}(y)$, para todo o $x \in \mathcal{P}$, $y \in \mathcal{C}$. Assume-se que $p_{\mathcal{C}}(y) > 0$, para todo o $y \in \mathcal{C}$ (se $p_{\mathcal{C}} = 0$, então a mensagem cifrada nunca chega a ser utilzada e pode ser apagada do conjunto $\mathcal{C}$). Fixa-se $x \in \mathcal{P}$. Para cada $y \in \mathcal{C}$, terá que haver pelo menos uma chave k tal que $e_k(x) = y$. Segue-se que $|\mathcal{K}| \geq |\mathcal{C}|$. Em qualquer sistema criptográfico tem-se $|\mathcal{C}| \geq |\mathcal{P}|$ uma vez que cada regra de cifra é uma função injectiva.

4.5 Segurança Absoluta Clássica

A segurança absoluta de um sistema criptográfico também pode ser definida recorrendo à entropia (ver definição 2.4) como definido por Shannon em [28].

Sejam X, Y variáveis aleatórias com distribuições de probabilidade $p(x)$ e $p(y)$.

Teorema 4.5.1 *Seja $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ um sistema criptográfico de chave privada, P e C distribuições de probabilidade em $\mathcal{P}$ e $\mathcal{C}$, respectivamente, então*

$$H(P|C) = H(P) - I(P;C)$$

Dem. Pelo teorema da simetria de informação, sabe-se que $H(P) - H(P|C) = H(C) - H(C|P)$. Então

$$\begin{aligned} H(P|C) &= H(P) - (H(C) - H(C|P)) \\ &= H(P) - I(P;C) \end{aligned}$$

◇

Definição 4.6 *Um sistema criptográfico de chave privada tem* segurança absoluta *se a mensagem cifrada (C) não revelar nenhuma informação acerca da mensagem (P), i.e., se*

$$I(C;P) = 0$$

Assim, a segurança absoluta definida com base na entropia significa que a quantidade de informação que se tem sobre a mensagem original não é alterada com o conhecimento da mensagem cifrada.

Teorema 4.6.1 *Seja $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ um sistema criptográfico de chave privada, P, C e K distribuições de probabilidade em $\mathcal{P}$, $\mathcal{C}$ e $\mathcal{K}$, respectivamente. Se K e P forem independentes, então*

$$H(C) \geq H(P)$$

Dem.

$$\begin{aligned} H(C) &\geq H(C|K) \\ &= H(C,K) - H(K) \\ &= H(P,K) - H(K) \text{ (A função de cifra é bijectiva)} \\ &= H(P) + H(K|P) - H(K) \\ &= H(P) + H(K) - H(K) \text{ (por independência)} \\ &= H(P) \end{aligned}$$

◇

Definição 4.7 *Seja $(\mathcal{P}, \mathcal{C}, \mathcal{K}; e, d)$ um sistema criptográfico de chave privada, P, C e K distribuições em $\mathcal{P}$, $\mathcal{C}$ e $\mathcal{K}$ respectivamente. $H(K|C)$ é chamada a* equivocação da chave *e mede o conhecimento que se tem acerca da chave através do texto cifrado.*

Teorema 4.7.1 *Seja $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ um sistema criptográfico de chave privada, P, C e K distribuições em $\mathcal{P}$, $\mathcal{C}$ e $\mathcal{K}$ respectivamente. Se K e P forem independentes, então*

$$H(K|C) = H(K) + H(P) - H(C)$$

Dem.

$$\begin{aligned} H(K|C) &= H(K,C) - H(C) \text{ (a partir de } K \text{ e } C \text{ podemos deduzir } P) \\ &= H(K,P,C) - H(C) \\ &= H(C|K,P) + H(K,P) - H(C) \text{ (a partir de } K \text{ e } P \text{ podemos deduzir } C) \\ &= H(K,P) - H(C) \\ &= H(K) + H(P) - H(C) \text{ (por independência)} \end{aligned}$$

◇

4.8 Protocolos Criptográficos Seguros

Existem alguns protocolos criptográficos de chave simétrica que apresentam segurança absoluta, desses estudaremos o *one time pad*, o *segredo partilhado* e a *autenticação*.

4.8.1 One Time Pad

Este protocolo foi descoberto por Gilbert Vernam em 1917 e durante aproximadamente trinta anos foi considerado inquebrável mesmo sem se conhecer a prova da sua segurança.

Definição 4.9 (one time pad) *Seja $n \geq 1$ e sejam $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_2)^n$ os conjuntos das mensagens originais, dos textos cifrados e das chaves, respectivamente. Seja μ a distribuição uniforme usada na escolha da chave $k \in \mathcal{K}$. O protocolo* one time pad *de Vernam é um sistema criptográfico de chave privada em que a chave $k \in_\mu \mathcal{K}$ é independente da mensagem $m \in \mathcal{P}$. A chave é usada para cifrar m usando o algoritmo de cifra $e(m,k)$ que consiste no ou-exclusivo de k e m. O algoritmo de decifra é definido do mesmo modo, mas usando c e k em vez de m e k.*

Teorema 4.9.1 *O protocolo one time pad tem* ***segurança absoluta.***

Dem.
Dada a mensagem p e a cifra c, a chave k é unicamente determinada, isto é,

$$H(K|P,C) = 0$$

Por definição, sabe-se que $H(K) = n$ e $I(P;K) = 0$, logo

$$I(K,C|P) = H(K|P) - H(K|C,P) = n$$

Então, como $H(C) \leq \log|\mathcal{C}| = n$ tem-se

$$I(P;C) = 0$$

◇

O protocolo one-time tem algumas desvantagens: o facto de que $|\mathcal{K}| \geq |\mathcal{P}|$ significa que, são necessárias chaves com pelo menos n dígitos binários para cifrar n dígitos binários da mensagem original. Este facto não seria um grande problema se a chave pudesse ser utilizada para cifrar várias mensagens. No entanto, este acto iria pôr em causa a própria segurança do sistema. Os sistemas incondicionalmente seguros dependem do facto de que a chave só é utilizada uma única vez. Assim, uma nova chave precisa de ser gerada e comunicada através de um canal seguro para cada mensagem que irá ser enviada. A utilização deste protocolo gera alguns problemas na gestão de chaves, o que o limita em aplicações comerciais. Contudo, é bastante utilizado em aplicações militares e diplomáticas, onde a segurança incondicional é imprescendível.

4.9.1 Segredo Partilhado

Os esquemas de segredo partilhado foram inventados independentemente por Blakley [4] e Shamir [26]. Baseiam-se na partilha de um segredo por vários participantes e na possibilidade do segredo ser recuperado por diferentes sub-grupos pertencentes ao grupo original. Vamos considerar o *esquema de segredo partilhado de Shamir.*

Definição 4.10 (Segredo partilhado de Shamir) *Sejam t, w inteiros positivos tal que $t \leq w$ e seja $\mathcal{K} = \mathbb{Z}_p$ o conjunto de todas as possíveis chaves. O* esquema de

segredo partilhado-(t, w) *de Shamir é um método de partilhar a chave* $k \in \mathcal{K}$ *entre um conjunto de* w *participantes (representado por* $\mathcal{P} = \{P_i, 1 \leq i \leq w\}$*), de forma que qualquer número de* t *participantes possam calcular o valor de* k*, mas nenhum grupo de* $t-1$ *participantes o possam fazer.*

Seja $D \notin \mathcal{P}$ o distribuidor que é um participante especial que escolhe o valor da chave k. Quando D começa a distribuir a chave k pelo conjunto de participantes, ele secretamente dá a cada um uma informação parcial chamada de *parte*. Seja $\mathcal{S} = \mathbb{Z}_p$ o conjunto de todas as possíveis partes. Um subconjunto de participantes $\mathcal{B} \subseteq \mathcal{P}$, com base nas suas partes e irá tentar calcular a chave.

Se $|\mathcal{B}| \geq t$, então eles deverão ser capazes de calcular a chave.
Se $|\mathcal{B}| < t$, então eles não podem calcular a chave.

Definição 4.11 *O esquema de segredo partilhado-(t,w) de Shamir em* $\mathbb{Z}_p$*, com* $p \geq w+1$ *é constituído por duas fases:*

- ***Fase inicial***
  D publicamente escolhe w elementos distintos, diferentes de zero de $\mathbb{Z}_p$*, representados por* x_i*,* $1 \leq i \leq w$*. Para* $1 \leq i \leq w$*, D atribui o valor de* x_i *a* P_i*.*

- ***Distribuição das partes***
  Supondo que D pretende partilhar a chave $k \in \mathbb{Z}_p$*. D secreta e independentemente escolhe de uma forma aleatória* $t-1$ *elementos de* $\mathbb{Z}_p$*,* $a_1, \ldots, a_{t-1}$ *e constrói um polinómio aleatório* $a(x) \in \mathbb{Z}_p[x]$ *de grau no máximo de* $t-1$*, onde*

$$a(x) = k + \sum_{j=1}^{t-1} a_j x^j \bmod p$$

- *Para* $1 \leq i \leq w$*, D calcula secretamente a parte* $y_i = a(x_i)$ *e distribui-a ao participante* P_i*.*

Para $1 \leq i \leq w$, todo o participante P_i obtém um ponto (x_i, y_i) neste polinómio onde todos os coeficientes $a_0, \ldots, a_{t-1}$ são elementos desconhecidos de $\mathbb{Z}_p$ e $a_0 = k$ é a chave. O conjunto $\mathcal{B} \subseteq \mathcal{P}, \mathcal{B} = \{P_{i_j}, 1 \leq i_j \leq w, 1 \leq j \leq t\}$ pode reconstruir a chave por meios de interpolação polinomial como a fórmula de Lagrange que é uma fórmula explícita de recuperar $a(x)$ dados t pontos $(x_{i_1}, y_{i_1}), \ldots, (x_{i_t}, y_{i_t})$ no polinómio. A fórmula de Lagrange é a seguinte:

$$a(x) = \sum_{j=1}^{t} y_{i_j} \prod_{1 \leq k \leq t, k \neq j} \frac{x - x_{i_k}}{x_{i_j} - x_{i_k}}$$

Para reconstruir o segredo, é suficiente que os participantes calculem o termo constante $a(0) = k$.

Deste modo,

$$k = \sum_{j=1}^{t} b_j y_{i_j}$$

onde $b_j = \prod_{1 \leq k \leq t, k \neq j} \frac{x_{i_k}}{x_{i_k} - x_{i_j}}$ e esses valores são públicos.

No lema seguinte será assumido que as $t-1$ partes são fixas e que dada a última parte, a função f (bijectiva) retorna a chave k.

Lema 4.11.1 *Seja $\mathcal{B} = \{P_{i_1}, \ldots, P_{i_{t-1}}\}$ e os valores $y_{i_1}, \ldots, y_{i_{t-1}}$ do grupo de participantes fixos, então existe a seguinte função bijectiva $f : \mathbb{Z}_p \to \mathbb{Z}_p$ tal que, para todos os últimos jogadores i_t e dada a última parte como argumento, retorna como resultado o valor da chave k:*

$$f_{x_{i_t}}(y_{i_t}) = \sum_{j=1}^{t-1} b_j y_{i_j} + b_t y_{i_t} = k \tag{4.1}$$

E a sua inversa é $f^{-1}_{x_{i_t}} : \mathbb{Z}_p \to \mathbb{Z}_p$:

$$f^{-1}_{x_{i_t}}(k) = k - \sum_{j=1}^{t-1} b_j y_{i_j} \times (b_t)^{-1} \tag{4.2}$$

Dem. Provado pela fórmula de interpolação de Lagrange e pelo facto de que todos os $b_t \neq 0 \in \mathbb{Z}_p$ têm um inverso multiplicativo $b_t^{-1} \in \mathbb{Z}_p$. ◇

Definição 4.12 *Seja $\mathcal{B}$ o conjunto dos participantes que pretendem reconstruir a chave. Seja a_0 a chave e seja $\mathcal{S} = \{y_{i_j} : 1 \leq i_j \leq w, 1 \leq j \leq t\}$ o conjunto de todas as partes. Com base na teoria da informação, um esquema de segredo partilhado tem* ***segurança absoluta****:*

- *Se $|\mathcal{B}| \geq t$ então $H(a_0 | y_{i_1}, \ldots, y_{i_t}) = 0$*
- *Se $|\mathcal{B}| < t$ então $H(a_0 | y_{i_1}, \ldots, y_{i_{t-1}}) = H(a_0)$*

Teorema 4.12.1 *O esquema de segredo partilhado-(t, w) de Shamir tem* segurança absoluta.

Dem. Seja a_0 a chave secreta,

$$\begin{aligned} H(a_0|y_{i_1},\ldots,y_{i_t}) &= -\sum_{k\in\mathcal{K}} p(k|y_{i_1},\ldots,y_{i_t})\log(p(k|y_{i_1},\ldots,y_{i_t})) \\ &= -p(a_0|y_{i_1},\ldots,y_{i_t})\log(p(a_0|y_{i_1},\ldots,y_{i_t})) \\ &\quad - \sum_{k\in\mathcal{K}\setminus\{a_0\}} p(k|y_{i_1},\ldots,y_{i_t})\log(p(k|y_{i_1},\ldots,y_{i_t})) \\ &= -1\times 0-(|\mathcal{K}|-1)\times(0\times 0) \\ &= 0 \end{aligned}$$

Para $a_0\in\mathcal{K}, a_0=\sum_{j=1}^{t} b_j y_{i_j}$ onde $b_j=\prod_{1\le k\le t, k\neq j}\frac{x_{i_k}}{x_{i_k}-x_{i_j}}$ e os y_{i_j}'s sao independentes, tem-se:

$$p(a_0|y_{i_1},\ldots,y_{i_{t-1}}) = p(a_0)$$

$$\begin{aligned} p(a_0|y_{i_1},\ldots,y_{i_{t-1}}) &= p\left(a_0=\sum_{j=1}^{t} b_j y_{i_j}\,\middle|\,\sum_{j=1}^{t-1} b_j y_{i_j}\right) \\ &= p\left(f(y_{i_t})=\sum_{j=1}^{t} b_j y_{i_j}\,\middle|\,\sum_{j=1}^{t-1} b_j y_{i_j}\right) \\ &= p(f(y_{i_t})) \\ &= p(a_0) \end{aligned}$$

$$\begin{aligned} H(a_0|y_{i_1},\ldots,y_{i_{t-1}}) &= -\sum_{a_0\in\mathcal{K}} p(a_0|y_{i_1},\ldots,y_{i_{t-1}})\log p(a_0|y_{i_1},\ldots,y_{i_{t-1}}) \\ &= -\sum_{a_0\in\mathcal{K}} p(a_0)\log p(a_0) \\ &= H(a_0) \end{aligned}$$

◇

4.12.1 Código de Autenticação

Um *código de autenticação* fornece um método que assegura a integridade de uma mensagem. Neste trabalho consideramos um modelo de autenticação descrito por Simmons [29], [30], [31] e [32] em que participam três intervenientes: o emissor (Alice), o receptor (Bob) e o adversário (Oscar). O código de autenticação garante ao receptor da mensagem que esta foi enviada por um emissor legítimo. Esta característica é válida mesmo na presença de um adversário com poder computacional ilimitado e

com total conhecimento do sistema (à excepção da chave). Um adversário pode interceptar as mensagens enviadas pela Alice e enviar mensagens fraudulentas ao Bob. Em autenticação é utilizada uma chave simétrica partilhada pela Alice e pelo Bob.

Definição 4.13 (Código de Autenticação) *Um* código de autenticação *é um tuplo* $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \epsilon)$ *que verifica as seguintes condições:*

- $\mathcal{S}$ *é um conjunto finito de possíveis estados-fonte.*
- $\mathcal{A}$ *é um conjunto finito de possíveis etiquetas (estados-fonte cifrados).*
- $\mathcal{K}$*, o espaço de chaves, é um conjunto finito de possíveis chaves.*
- *Para cada* $k \in \mathcal{K}$*, existe uma regra de autenticação* $e_{\mathcal{K}} : \mathcal{S} \rightarrow \mathcal{A}$.

O conjunto das mensagens é representado por $\mathcal{M}$ e é definido como $\mathcal{M} = \mathcal{S} \times \mathcal{A}$.

Nota 4.14 *O* estado-fonte *num código de autenticação corresponde à mensagem original. Uma* etiqueta de autenticação *corresponde ao estado-fonte cifrado com a chave pré-definida. A* mensagem *que o emissor envia ao receptor corresponde ao par constituído pelo estado-fonte e pela sua respectiva etiqueta de autenticação.*

Existem dois tipos de ataques que um adversário pode efectuar:

- *Personificação*: Oscar escolhe uma nova mensagem (s, a), coloca-a no canal com a intenção que Bob a aceite como sendo a mensagem verdadeira. A probabilidade de sucesso deste tipo de ataque é representada por P_i.
- *Substituição*: Oscar intercepta uma mensagem (s, a) no canal e substitui-a por outra (s', a') com a intenção que Bob a aceite como autêntica. A probabilidade de sucesso deste tipo de ataque é representada por P_s.

A segurança de um código de autenticação depende dos valores das probabilidades de sucesso dos ataques acima descritos. Quanto mais pequenas forem as probabilidades, mais seguros serão os códigos de autenticação. É necessário obter limites nestas probabilidades.

Os códigos de autenticação para serem seguros deverão possuir as seguintes propriedades:

1. As probabilidades de ataque P_i e P_s deverão ter o menor valor possível para o nível de segurança pretendido ser atingido.

2. O número de estados-fonte deverá ser suficientemente elevado para ser possível comunicar a informação pretendida, anexando uma etiqueta de autenticação a cada estado.

3. O tamanho da chave deverá ser minimizado, uma vez que a chave irá ser comunicada através de um canal seguro (a chave deverá ser alterada após ter sido comunicada).

Uma ferramenta muito importante para este protocolo é a *desigualdade de Jensen.*

Uma função f diz-se convexa no intervalo $[a, b]$ se e só se, para todo os $x_1, x_2 \in [a, b]$ e $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1 - 1\lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

É estritamente convexa se a desigualdade anterior for estrita, sempre que $0 < \lambda < 1$. Por outro lado, a função g é estritamente côncava se e só se $-g$ for estritamente convexa.

Teorema 4.14.1 (Desigualdade de Jensen) *Para uma função convexa f e uma variável aleatória X*

$$E(f(X)) \geq f(E(X))$$

Dem. Demonstração para distribuições discretas com base na indução no número de pontos de distribuição. Para dois pontos de distribuição, verifica-se a seguinte desigualdade

$$p_1 f(x_1) + p_2 f(x_2) \geq f(p_1 x_1 + p_2 x_2)$$

que segue directamente da definição de função convexa.

Supõe-se que esta desigualdade verifica-se para distribuições com $k - 1$ pontos. Então para $p'_i = \frac{p_i}{1-p_k}$, $i = 1, 2, \ldots, k-1$, tem-se (note-se que $\sum_{i \in I} p'_i = 1$)

$$\begin{aligned}
\sum_{i=1}^{k} p_i f(x_i) &= p_k f(x_k) + (1 - p_k) \sum_{i=1}^{k-1} p'_i f(x_i) \\
&\geq p_k f(x_k) + (1 - p_k) f\left(\sum_{i=1}^{k-1} p'_i (x_i) \right) \\
&\geq f\left(p_k (x_k) + (1 - p_k) \sum_{i=1}^{k-1} p'_i (x_i) \right) \\
&= f\left(\sum_{i=1}^{k} p_i x^i \right)
\end{aligned}$$

A primeira desigualdade vem pela hipótese de indução e a segunda por definição de convexidade de uma função.

◇

Ataque de Personificação

A probabilidade, para $s \in \mathcal{S}$ e $a \in \mathcal{A}$ e uma chave $k_0 \in \mathcal{K}$ previamente escolhida pelos intervenientes, de Bob aceitar a mensagem (s, a) como autêntica é definida como:

$$\begin{aligned} payoff(s,a) &= prob(a = e_{k_0}(s)) \\ &= \sum_{k \in \mathcal{K}} p_{\mathcal{K}}(k) \end{aligned}$$

De modo a maximizar a sua oportunidade de sucesso, Oscar deverá escolher (s, a) de forma a maximizar $payoff(s, a)$, então

$$P_i = \max\{payoff(s,a) : s \in \mathcal{S}, a \in \mathcal{A}\}$$

Note-se que P_i não depende da distribuição de probabilidade de $p_{\mathcal{S}}$.

Teorema 4.14.2 *Seja $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \epsilon)$ um código de autenticação. Então*

$$\log P_i \geq H(K|M) - H(K)$$

Dem. A partir da definição de P_i, tem-se

$$P_i = \max\{payoff(s,a) : s \in \mathcal{S}, a \in \mathcal{A}\}$$

Uma vez que o máximo de $payoff(s, a)$ é maior que o seu valor esperado, então

$$P_i \geq \sum_{s \in \mathcal{S}, a \in \mathcal{A}} p_{\mathcal{M}}(s,a) payoff(s,a)$$

Pela desigualdade de Jensen,

$$\begin{aligned} \log P_i &\geq \log \sum_{s \in \mathcal{S}, a \in \mathcal{A}} p_{\mathcal{M}}(s,a) payoff(s,a) \\ &\geq \sum_{s \in \mathcal{S}, a \in \mathcal{A}} p_{\mathcal{M}}(s,a) \log payoff(s,a) \end{aligned}$$

Como $p_{\mathcal{M}}(s, a) = p_{\mathcal{S}}(s) \times payoff(s, a)$, tem-se

$$\log P_i \geq \sum_{s \in \mathcal{S}, a \in \mathcal{A}} p_{\mathcal{S}}(s) payoff(s,a) \log payoff(s,a)$$

Note-se que $payoff(s,a) = p_{\mathcal{A}}(a|s)$ (ou seja, é igual à probabilidade de a ser a etiqueta de autenticação, sabendo-se que s é o estado-fonte correspondente). Então,

$$\begin{aligned} \log P_i &\geq \sum_{s \in \mathcal{S}, a \in \mathcal{A}} p_{\mathcal{S}}(s) p_{\mathcal{A}}(a|s) \log p_{\mathcal{A}}(a|s) \\ &= -H(A|S) \end{aligned}$$

por definição de entropia condicional. O resultado é obtido demonstrando que

$$-H(A|S) = H(K|M) - H(K)$$

Por um lado tem-se,

$$H(K,A,S) = H(A|K,S) + H(A|S) + H(S)$$

Por outro,

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

onde é usado o facto de que $H(A|K,S) = 0$ (uma vez que a chave e o estado-fonte determinam unicamente a etiqueta de autenticação). A seguinte igualdade $H(K,S) = H(K) + H(S)$ verifica-se, uma vez que o estado-fonte e a chave são eventos independentes.

Igualando as duas expressões para $H(K,A,S)$ obtém-se

$$-H(A|S) = H(K|A,S) - H(K)$$

Mas a mensagem $m = (s,a)$ é definida como sendo um par constituído por um estado-fonte e uma etiqueta de autenticação (ou seja, $\mathcal{M} = \mathcal{S} \times \mathcal{A}$). Então, $H(K|A,S) = H(K|M)$. ◇

Corolário 4.14.3 *Seja $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \epsilon)$ um código de autenticação. Um código de autenticação tem* ***segurança absoluta*** *para um ataque de personificação se e só se*

$$P_i = 2^{H(K|M) - H(K)}$$

Ataque de Substituição

Num ataque de substituição, Oscar observa uma mensagem num canal de comunicação enviada por Alice e substitui-a por outra, na expectativa que Bob a aceite como válida.

A probabilidade p_m representa a probabilidade de Oscar conseguir enganar Bob com a substituição da mensagem m observada no canal por m'.

$$p_m = p(k|m)$$

Oscar pretende descobrir qual a chave utilizada, tendo um conhecimento prévio da mensagem que foi transmitida no canal. Assume-se que Oscar pretende maximizar a sua oportunidade de enganar Bob, então, tenta calcular $\max_k p(k|m)$.

Para calcular a probabilidade de um ataque de substituição, representado por P_s, calcula-se o valor esperado da quantidade p_m em relação à probabilidade de observar cada mensagem m no canal, $P_{\mathcal{M}}(m)$. A probabilidade P_s é definida da seguinte forma:

$$P_s = \sum_{m \in \mathcal{M}} p_{\mathcal{M}}(m) \max_k p(k|m)$$

Teorema 4.14.4 *Seja $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \epsilon)$ um código de autenticação. Então,*

$$\log P_s \geq H(K|M)$$

Dem. Segundo Maurer em [23] e [20], tem-se

$$\begin{aligned}
H(K|M) &= \sum_{k \in \mathcal{K}} p(k)(-\sum_{m \in \mathcal{M}} p(m|k) \log p(k|m)) \\
&\geq \min_k(-\sum_{m \in \mathcal{M}} p(m|k) \log p(k|m)) \\
&\geq \min_k(-\sum_{m \in \mathcal{M}} p(m) \log p(k|m)) \\
&= \sum_{m \in \mathcal{M}} p(m) \min_k(-\log p(k|m)) \\
&= \sum_{m \in \mathcal{M}} p(m)(-\log \max_k p(k|m)) \\
&\geq -\log \sum_{m \in \mathcal{M}} p(m) \max_k p(k|m)
\end{aligned}$$

Então,

$$\sum_{m \in \mathcal{M}} p(m) \max_k p(k|m) \geq 2^{-H(K|M)}$$

A primeira desigualdade segue do facto de que o menor valor que ocorre na média, $\min_k(-\log p(k))$, é majorado pela média. A segunda, pelo facto de que informação adicional não aumenta a incerteza e a última segue pela desigualdade de Jensen. $\diamond$

Corolário 4.14.5 *Seja $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \varepsilon)$ um código de autenticação. Um código de autenticação tem* **segurança absoluta** *relativamente a um ataque de substituição se e só se*

$$P_s = 2^{-H(K|M)}$$

Este capítulo introduz a noção de segurança absoluta com base na complexidade de Kolmogorov, explorando a estreita relação entre esta e a teoria de informação.

A estreita relação entre estas duas medidas abre um novo caminho de investigação, o uso da complexidade de Kolmogorov em criptografia.

Na próxima secção será descrito como estas duas medidas se relacionam. Na secção seguinte, apresentam-se provas da segurança absoluta com base na complexidade de Kolmogorov dos protocolos: one time pad, segredo partilhado e autenticação.

4.15 Complexidade de Kolmogorov e Entropia

A entropia de Shannon mede o número de dígitos binários necessários para especificar a incerteza de um resultado a partir de um espaço amostral de mensagens. A complexidade de Kolmogorov mede o número mínimo de dígitos binários necessários para descrever uma mensagem individual. Embora estas medidas de informação, conceptualmente, sejam muito distintas, em [18] mostra-se que a menos de uma constante aditiva, a entropia coincide com o valor esperado da complexidade de Kolmogorov. Esta relação surpreendente entre estas duas medidas de informação é a base deste trabalho.

Durante a comunicação entre um emissor e um receptor, quanto menor for o número de dígitos binários a serem transmitidos melhor. Shannon descobriu que o mínimo valor esperado do comprimento de uma codificação relativamente à probabilidade da sua sequência original (representado por L) é muito semelhante ao valor de entropia do conjunto das sequências originais.

Teorema 4.15.1 (Teorema da codificação sem ruído) *Seja $p(x)$ a probabilidade da sequência original x e $|x|$ o comprimento da sua codificação. $L = -\sum_x p(x)|x|$, se $H(p) = -\sum_x p(x) \log p(x)$, então,*

$$H(p) \leq L \leq H(p) + 1$$

Dem.

- $H(p) \leq L$

Uma vez que $\sum_x p(x) = 1$ e $\sum_x (2^{-|x|}/\sum_x 2^{-|x|}) = 1$, pela concavidade da função logaritmo, tem-se

$$-\sum_x p(x)\log p(x) \leq -\sum_x p(x)\log\frac{2^{-|x|}}{\sum_x 2^{-|x|}}$$

o que implica

$$-\sum_x p(x)\log p(x) \leq \sum_x p(x)|x| + (\sum_x p(x))\log\sum_x 2^{-|x|}$$

Uma vez que $\sum_x p(x) = 1$, $L = \sum_x p(x)|x|$ e $H(p) = -\sum_x p(x)\log p(x)$, pode ser reescrito como

$$H(p) \leq L + \log\sum_x 2^{-|x|} \tag{4.3}$$

Como a codificação de x é um código prefixo, pela desigualdade de Kraft, $\sum_x 2^{-|x|} \leq 1$. Então, pela desigualdade 4.3 verifica-se $H(p) \leq L$.

- $L \leq H(p) + 1$

  Seja $|x| = \lceil -\log p(x) \rceil$, então, $1 \geq \sum_x p(x) \leq \sum_x 2^{-|x|}$. Pela desigualdade de Kraft, existe um código prefixo com codificações de tamanho $|x_1|, |x_2|, \ldots$, então

$$L \leq \sum_x p(x)|x| \leq \sum_x p(x)(-\log p(x) + 1) = H(p) + 1$$

  e a segunda desigualdade está demonstrada.

◇

A entropia de Shannon tal como a complexidade de Kolmogorov, representa o valor mínimo do comprimento das codificações.

Teorema 4.15.2 *Seja p uma distribuição de probabilidade recursiva, então*

$$0 \leq \left(\sum_x p(x)K(x) - H(p)\right) \leq c_p$$

onde c_p é uma constante dependente apenas de p.

Dem. Pelo teorema anterior tem-se, $H(p) \leq \sum_x p(x)|x|$. Considere-se o menor programa auto-delimitado para x, $K(x)$, então

$$H(p) \leq \sum_x p(x)K(x)$$

A complexidade de Kolmogorov livre de prefixos $K(x)$ induz a distribuição universal $\mathbf{m}(x) = 2^{-K(x)}$ e o seguinte resultado

$$p(x) \leq 2^{K(p)}\mathbf{m}(x)$$

verifica-se. Substituindo na equação acima $-\log \mathbf{m}(x)$ por $K(x) + O(1)$ (ver teorema 4.15.1), obtém-se

$$-\log p(x) \geq K(x) - K(p) + O(1)$$

A constante c_p representa a quantidade $c_p = K(p) + O(1)$ e segue-se

$$\sum_x p(x)K(x) \leq H(p) + c_p$$

◇

Assim se conclui que a entropia $H(p) = -\sum_x p(x)\log p(x)$ da distribuição p é igual ao valor esperado da complexidade da complexidade de Kolmogorov $\sum_x p(x)K(x)$ em relação à probabilidade $p(.)$ Esta igualdade é bastante precisa pois difere apenas de uma constante aditiva que depende somente de p.

Outra prova desta relação é apresentada a seguir e usa o facto de quase todos as sequências serem incompressíveis.

Dem. Seja $p(x) = 2^{-n}$ a distribuição uniforme de probabilidade nos resultados de comprimento n. Quase todas as sequências são incompressíveis, ou seja, existem $2^n(1 - 2^{-c+1})$ sequências x_i's que têm $K(x) \geq n - c$. O seguinte cálculo mostra que a entropia $H(X) = -\sum_{|x|=n} p(x)\log p(x)$ é assimptóticamente igual ao valor esperado da complexidade de Kolmogorov ($E = \sum_{|x|=n} p(x)K(x)$) de uma palavra de comprimento n.

$$\frac{n}{n + O(1)} < \frac{H(X)}{\sum_{|x|=n} p(x)K(x)} < \frac{n}{(1 - 2^{-c+1})(n - c)}$$

Substituíndo $c = \log n$ obtém-se

$$\lim_{n\to\infty} \frac{H(X)}{\sum_{|x|=n} p(x)K(x)} = 1$$

◇

4.16 Segurança Absoluta Efectiva

Como se verificou no capítulo 3, a complexidade de Kolmogorov mínima goza de uma propriedade muito interessante e muito semelhante à da entropia que é a *simetria de informação*. No contexto da complexidade de Kolmogorov mínima, esta propriedade

permite uma determinação rigorosa da quantidade de informação que um objecto individual tem acerca de outro, e vice-versa. Nos próximos teoremas e definições será usada a complexidade de Kolmogorov mínima.

Teorema 4.16.1 *Seja* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ *um sistema criptográfico de chave privada onde* $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_2)^n$ *para* $n \geq 1$ *e sejam* $m, c, k \in \mathcal{P}, \mathcal{C}, \mathcal{K}$ *respectivamente, então*

$$K_c(m|c) = K_c(m) - I(m : c)$$

Dem. Por simetria de informação, tem-se

$$K_c(m) - K_c(m|c) = K_c(c) - K_c(c|m)$$

Então,

$$\begin{aligned} K_c(m|c) &= K_c(m) - (K_c(c) - K_c(c|m)) \\ &= K_c(m) - I(m : c) \end{aligned}$$

◇

Definição 4.17 *Seja* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ *um sistema criptográfico de chave privada onde* $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_2)^n$ *para* $n \geq 1$ *e sejam* $m, c, k \in \mathcal{P}, \mathcal{C}, \mathcal{K}$ *respectivamente. Uma instância* m, k *de um sistema criptográfico de chave privada é* d**-seguro** *se*

$$K_c(m|e(m, k)) \geq K_c(m) - d$$

Definição 4.18 *Seja* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ *um sistema criptográfico de chave privada onde* $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_2)^n$ *para* $n \geq 1$ *e sejam* $m, c, k \in \mathcal{P}, \mathcal{C}, \mathcal{K}$ *respectivamente. Uma instância* m, k *de um sistema criptográfico de chave privada tem* ***segurança absoluta*** *se e só se o texto cifrado* c *não revelar nenhuma informação sobre a mensagem* m*, isto é, se*

$$d = K_c(m) - K_c(m|c) = I(m : c) = 0$$

A constante d é considerada a penalização, isto é, a distância a que estamos do uso ideal do protocolo.

A definição de segurança absoluta com base na complexidade de Kolmogorov aqui introduzida é muito parecida com a definição de segurança absoluta com base na entropia de Shannon. Tal como esta, revela que a quantidade de informação que se tem sobre a mensagem original não é alterada com o conhecimento da mensagem cifrada. Ou seja, um sistema criptográfico tem segurança absoluta se a mensagem original e a correspondente mensagem cifrada forem completamente independentes uma da outra.

Teorema 4.18.1 *Seja $(\mathcal{P}, \mathcal{C}, \mathcal{K}, e, d)$ um sistema criptográfico de chave privada onde $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_2)^n$ para $n \geq 1$ e sejam $m, c, k \in \mathcal{P}, \mathcal{C}, \mathcal{K}$, respectivamente. Se k e m forem completamente independentes $(I(k : m) = 0)$, então*

$$K_c(c) \geq K_c(m)$$

Dem.

$$\begin{aligned} K_c(c) &\geq K_c(c|k) \\ &= K_c(c,k) - K_c(k) \\ &= K_c(m,k) - K_c(k) \text{ (a função de cifra é bijectiva)} \\ &= K_c(m) + K_c(k|m) - K_c(k) \\ &= K_c(m) + K_c(k) - K_c(k) \text{ (por independência)} \\ &= K_c(m) \end{aligned}$$

◇

Teorema 4.18.2 *Seja $\mathcal{P}, \mathcal{C}, \mathcal{K}$, $e, d)$ um sistema criptográfico de chave privada onde $\mathcal{P} = \mathcal{C} = \mathcal{K} = (Z_2)^n$ para $n \geq 1$ e sejam $m, c, k \in \mathcal{P}, \mathcal{C}, \mathcal{K}$, respectivamente. Se k e m forem completamente independentes, a* equivocação da chave *é medida por*

$$K_c(k|c) = K_c(k) + K_c(m) - K_c(c)$$

Dem.

$$\begin{aligned} K_c(k|c) &= K_c(k,c) - K_c(c) \text{ (a partir de } k \text{ e } c \text{ pode-se deduzir } m) \\ &= K_c(k,m,c) - K_c(c) \\ &= K_c(c|k,m) - K_c(k,m) - K_c(c) \text{ (a partir de } k \text{ e } m \text{ pode-se deduzir } c) \\ &= K_c(k,m) - K_c(c) \\ &= K_c(k) + K_c(m) - K_c(c) \text{ (por independência)} \end{aligned}$$

◇

4.19 Protocolos Criptográficos Seguros

A definição de segurança absoluta com base na complexidade de Kolmogorov de protocolos criptográficos, permite agora fazer uma medição precisa e individual dos objectos envolvidos numa comunicação. Com base nesta medida de instâncias individuais, vão ser apresentadas as provas de segurança absoluta dos protocolos one-time-pad, segredo partilhado e autenticação.

4.19.1 One Time Pad

Teorema 4.19.1 *O protocolo one time pad de Vernam tem* **segurança absoluta.**

Dem. Sejam $m, k \in \Sigma^n$ e $e(m,k) = m \oplus k$ tal que:

- k é aleatório, $K_c(k) = n$.
- k é independente de m, $I(m;k) = 0$, que significa que, $K_c(k) = K_c(k|m)$.

Por simetria de informação, tem-se

$$K_c(m) - K_c(m|(m \oplus k)) = K_c(m \oplus k) - K_c(m \oplus k|m)$$

Então,

$$\begin{aligned} K_c(m|(m \oplus k)) &= K_c(m) - K_c(m \oplus k) + K_c(m \oplus k|m) \\ &\underset{K_c(m\oplus k|m)=K_c(k|m)}{=} K_c(m) - [K_c(m \oplus k) - K_c(k|m)] \\ &\underset{|m\oplus k|=n}{\geq} K_c(m) - [n - K_c(k|m)] \\ &= K_c(m) - [n - n] \\ &= K_c(m) \end{aligned}$$

◇

4.19.2 Segredo Partilhado

Definição 4.20 *Seja $\mathcal{B}$ um conjunto de participantes que pretende reconstruir a chave. Seja a_0 a chave e seja $\mathcal{S} = \{y_{i_j} : 1 \leq i_j \leq w, 1 \leq j \leq t\}$ o conjunto de todas as partes. Um esquema de segredo partilhado tem* **segurança absoluta** *com base na complexidade de Kolmogorov:*

1. *Se $|\mathcal{B}| \geq t$ então $K(a_0|y_{i_1}, \ldots, y_{i_t}) \leq O(1)$*
2. *Se $|\mathcal{B}| < t$ então $K(a_0|y_{i_1}, \ldots, y_{i_{t-1}}) \geq K(a_0) - d$*

Considere-se a sequência binária z onde $z = a_0.a_1 \ldots .a_{t-1}$, e o tamanho de cada a_i para $1 \leq i_j \leq w$ e $1 \leq j \leq t$ que é $\log p$. Então $|z| = t \log p$.

Teorema 4.20.1 *Com base na complexidade de Kolmogorov, o esquema de segredo partilhado (t,w) de Shamir é d-***seguro** *se $K(z|x_{i_1}, \ldots, x_{i_t}) \geq |z| - d$.*

Dem.

1. Existe um programa $\mathtt{p}_1$ que recebendo como dados de entrada todas as partes do segredo, calcula a fórmula de interpolação de Lagrange $a(0)$ e retorna a chave como resultado:

   Dados de entrada: As partes $y_{i_1}, \ldots, y_{i_t}$ e os valores públicos $x_{i_1}, \ldots, x_{i_t}$ de cada participante
   Calcular a fórmula de Lagrange $a(0)$
   Resultado: a chave a_0

   Então, $K(a_0|y_{i_1}, \ldots, y_{i_t}) \leq O(1)$.

2. Pretende-se provar que

   se $K(z|x_{i_1}, \ldots, x_{i_t}) \geq |z| - d$ então $K(a_0|y_{i_1}, \ldots, y_{i_{t-1}}) \geq K(a_0) - d = \log p - d$

   Considere-se um programa mínimo p_2 que calcula $p_2(y_{i_1}, \ldots, y_{i_{t-1}}|x_{i_1}, \ldots, x_{i_t}) = a_0$, onde $|p_2| = l$.

   Então existe um algoritmo que calcula z definido da seguinte forma:

   Dados de entrada: as partes $y_{i_1}, \ldots, y_{i_{t-1}}$
   Em memória: os valores públicos $x_{i_1}, \ldots, x_{i_t}$,
   Simular $p_2(y_{i_1}, \ldots, y_{i_{t-1}}) = a_0$
   Calcular os coeficientes de $a(x)$ usando a fórmula de Lagrange
   Imprimir $a_0, a_1, \ldots, a_{t-1}$

   Foi assumido no início da prova que $K(z|x_{i_1}, \ldots, x_{i_t}) \geq |z| - d$. Então,

   $$t \log p - d \leq k(z|x_{i_1}, \ldots, x_{i_t}) \leq (t-1) \log p + l$$

   e obtém-se $l \geq \log p - d$. Deste modo, pode-se concluir que

   $$K(a_0|y_{i_1}, \ldots, y_{i_{t-1}}) \geq \log p - d = K(a_0) - d$$

◇

Teorema 4.20.2 *Se z for incompressível, o esquema de segredo partilhado–(t, w) de Shamir, tem* **segurança absoluta.**

Dem. Demonstrado pelos teoremas anteriores e pelo facto de que se z for incompressível, então $d = 0$. ◇

4.20.1 Código de Autenticação

Sejam P_i e P_s as probabilidades de ataque definidas no capítulo anterior e considere-se a seguinte notação: O símbolo $=^+$ representa a igualdade a menos de uma constante aditiva e os símbolos $\leq^*$ e $\geq^+$ representam, respectivamente, as desigualdades a menos de uma constante multiplicativa e aditiva.

Ataque de Personificação

Teorema 4.20.3 *Seja $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \epsilon)$ um código de autenticação. Então*

$$\log P_i \geq K(k|m) - K(k)$$

Dem. Sejam s' e a', o estado e a etiqueta de autenticação que maximizam a hipótese do Bob aceitar a mensagem como sendo autêntica.

$$\begin{aligned} P_i &= \max\{payoff(s,a), s \in \mathcal{S}, a \in \mathcal{A}\} \\ &= payoff(s', a') \\ &= p(a'|s') \\ &\leq^* 2^{-K(a'|s')} \end{aligned}$$

A prova fica completa mostrando que $-K(a'|s') = K(k|m) - K(k)$.

Por um lado, tem-se

$$K(a,k,s) =^+ K(k|a,s) + k(a|s) + K(s)$$

Por outro,

$$\begin{aligned} K(a,k,s) &=^+ K(a|k,s) + K(k|s) + K(s) \\ &=^+ K(k) + K(s) \end{aligned}$$

onde é usado o facto de que $K(a|k,s) \leq O(1)$ uma vez que a chave e o estado-fonte determinam unicamente a etiqueta de autenticação. A chave e o estado-fonte são independentes, então $I(k:s) = 0$.

Igualando as duas expressões que representam $K(k,a,s)$, obtém-se

$$\begin{aligned} K(a|s) &=^+ K(k) - K(k|a,s) \\ &=^+ K(k) - K(k|m) \end{aligned}$$

Então,

$$\begin{aligned} P_i &\leq^* 2^{K(k)-K(k|m)} \\ \log P_i &\geq^+ K(k|m) - K(k) \end{aligned}$$

$\diamond$

Corolário 4.20.4 *Seja $(\mathcal{S}, \mathcal{A}, \mathcal{K}, \epsilon)$ um código de autenticação. Um código de autenticação tem* **segurança absoluta** *para um ataque de personificação se e só se*

$$P_i = 2^{K(k|m)-K(k)}$$

Capítulo 5

Conclusão

Este último capítulo apresenta uma síntese do trabalho apresentado nesta dissertação, referindo as conclusões obtidas, as contribuições e as perspectivas de desenvolvimento futuro.

5.1 Síntese

Neste trabalho concentramo-nos vivamente em três áreas de investigação: a teoria de informação, a complexidade de Kolmogorov e a criptografia. Vamos fazer um breve resumo daquilo que foi apresentado e proposto ao longo desta dissertação.

A teoria da informação baseia-se na *entropia*, que mede a quantidade de informação necessária para especificar o resultado de um acontecimento probabilístico. Um valor elevado de entropia significa uma maior incerteza relativamente a um dado resultado. Para Shannon, a quantidade de informação de uma sequência depende probabilisticamente do contexto em que está inserida. Salientamos uma propriedade muito interessante da entropia que é a *simetria de informação*.

A complexidade de Kolmogorov, por nós designada *complexidade de instâncias individuais*, mede a quantidade de informação contida num objecto individual. Esta é a chamada *complexidade de Kolmogorov clássica*. A complexidade que só explora programas livres de prefixos, é conhecida como *complexidade de Kolmogorov livre de prefixos*, e a que considera o primeiro menor programa calculado por uma máquina de Turing, é denominada por *complexidade de Kolmogorov mínima*. Através do *teorema da invariância*, verificou-se que a complexidade de Kolmogorov é uma medida efectiva independente do método de formalização, isto é, um atributo intrínseco ao próprio objecto. Permite determinar a compressibilidade de sequências e a partir daí calcular a quantidade de informação do objecto. Existe uma relação directa entre quantidade de informação e probilidades. Verificou-se que através da distribuição universal, uma semi-medida discreta de probabilidade que domina multiplicativamente

qualquer outra, os objectos simples (com menor quantidade de informação), têm maior probabilidade algorítmica. Referimo-nos ainda a outra variante da complexidade: a *complexidade de Kolmogorov com limites nos recursos*. Esta medida é computável uma vez que existe um limite no tempo ou no espaço.

Os protocolos simétricos e de chave pública foram definidos e foram também apresentadas as noções de segurança computacional e absoluta. Incidimos o nosso trabalho nesta última, que se exprime pelo facto de a mensagem cifrada não revelar qualquer informação acerca da mensagem original, isto é, a informação mútua entre as duas mensagens é nula. Apresentamos as demonstrações da segurança absoluta clássica (com base na entropia) dos protocolos criptográficos simétricos *one time pad*, *segredo partilhado* e *autenticação*. Na demonstração do protocolo *segredo partilhado*, não consideramos as várias combinações de subgrupos que se podiam ter formado para juntar as suas partes e calcular a chave secreta. Incidimos na probabilidade de obter a chave, verificando apenas se havia um número de elementos suficiente para recuperar o segredo.

O ponto determinante para o início deste trabalho consistiu na relação existente entre a entropia e complexidade de Kolmogorov. Pelo *teorema da codificação sem ruído* e pela própria definição de complexidade de Kolmogorov, constata-se que ambas representam o valor mínimo do comprimento das codificações. Além deste facto e sob algumas restrições na distribuição de probabilidade das sequências, o valor esperado da complexidade de Kolmogorov é igual à entropia. Foi com base neste resultado e na semelhança de propriedades, que verificámos que a entropia de Shannon pode ser substituída pela complexidade de Kolmogorov nas definições de segurança absoluta de alguns sistemas criptográficos de chave simétrica. Esta substituição tem a vantagem de disponibilizar provas de segurança para instâncias individuais de um modo efectivo. A prova do *one time pad* baseia-se na simetria de informação (complexidade de Kolmogorov mínima), tal como na demonstração clássica. Criou-se um algoritmo para demonstrar a segurança absoluta do *segredo partilhado* e usou-se a distribuição universal na demonstração de segurança absoluta de um *código de autenticação* contra um ataque de personificação.

5.2 Contribuições

O objectivo deste trabalho que era estabelecer uma nova definição de segurança absoluta de protocolos criptográficos de chave simétrica, com base em instâncias individuais, foi atingido. Baseamo-nos nas relações já existentes da teoria de informação com a complexidade de Kolmogorov e com a criptografia e criamos uma terceira ligação entre a complexidade de Kolmogorov e a criptografia. Mas qual o papel da complexidade de Kolmogorov na criptografia e quais são as suas vantagens? O papel da complexidade de Kolmogorov na criptografia é o mesmo do da entropia, com a vantagem de verificar a segurança absoluta de um objecto individual, como alternativa

a um cenário probabilístico. A complexidade de Kolmogorov na criptografia leva-nos a voar um pouco mais alto e pensar ser possível medir a segurança de sistemas criptográficos de chave pública.

5.3 Futuro e Complexidade de Kolmogorov

O estudo da segurança absoluta de instâncias individuais está ainda no seu início. Existem pelo menos duas direcções que se podem tomar para trabalho futuro. Uma é prosseguir com a análise de segurança em protocolos criptográficos simétricos usando a complexidade de Kolmogorov (*código de autenticação* perante um ataque de substituição, *"bit-commitment"*,...). A outra é providenciar uma ferramenta de trabalho em sistemas criptográficos de chave pública, onde o poder computacional desempenha um papel importante. Se é difícil incorporar a noção de dificuldade computacional na fórmula de entropia, o mesmo já não se passa na complexidade de Kolmogorov, bastando para tal impor um limite de tempo na execução dos programas. Acredita-se que, com base na complexidade de Kolmogorov com limite nos recursos, se possa medir a segurança deste tipo de sistemas criptográficos.

Referências

[1] L. A. Levin A. K. Zvonkin. The complexity of finite objects and the algorithmic concepts of information and randomness. *Russian Math. Surveys*, 25(6):83–124, 1970.

[2] Luís Filipe Antunes. *Useful Information.* PhD thesis, University of Porto, 2002.

[3] C. H. Bennett. Logical depth and physical complexity. In R. Herken, editor, *The Universal Turing Machine: A Half-Century Survey*, pages 227–257. Oxford University Press, 1988.

[4] G. R. Blakley. Safeguarding cryptographic keys. *AFIPS Conference Proceedings*, 48:313–317, 1979.

[5] Gregory J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13(4):145–149, 1966.

[6] Gregory J. Chaitin. A theory of program size formally identical to information theory. *ACM 22*, pages 329–340, 1975.

[7] A. Church. An unsolvable problem of elementary number theory. *American journal of Mathematics*, 58:345–363, 1936.

[8] W. Diffie and M. Helman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[9] Oded Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, New York, NY, USA, 2000.

[10] Steven Homer and Alan L. Selman. *Computability and complexity theory.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[11] John E. Hopcroft and Jeffrey D. Ullman. *Introduction To Automata Theory, Languages, And Computation.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[12] David Kahn. *The Codebreaker.* Mcmillan, New York, 1967.

[13] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

[14] A. N. Kolmogorov. Combinatorial foundations of information theory and the calculus of probabilities. *Russian Mathematical Surveys*, 38(4):29–40, 1983.

[15] A. N. Kolmogorov and V. A. Uspenskii. Algorithms and randomness. *SIAM J. Theory Probab. Appl.*, 32:389–412, 1987.

[16] M. Koppel. Structure. In R. Herken, editor, *The Universal Turing Machine: A Half-Century Survey*, pages 435–452. Oxford University Press, 1988.

[17] L. A. Levin. On the notion of a random sequence. *Soviet Math. Dokl.*, 14:1413–1416, 1973.

[18] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications (2nd ed.)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[19] James L. Massey. An introduction to contemporary cryptology. *Proceedings of the IEEE*, 76(5):533–549, 1998.

[20] Ueli Maurer. A unified and generalized treatment of authentication theory. In *Proc. 13th Symp. on Theoretical Aspects of Computer Science (STACS'96)*, volume 1046 of *Lecture Notes in Computer Science*, pages 387–398. Springer-Verlag, 1996.

[21] Ueli Maurer. Information-theoretic cryptography. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 47–64. Springer-Verlag, August 1999.

[22] Ueli Maurer. Cryptography 2000 ± 10. In R. Wilhelm, editor, *Informatics — 10 Years Back, 10 Years Ahead*, volume 2000 of *Lecture Notes in Computer Science*, pages 63–85. Springer-Verlag, 2001.

[23] Ueli M. Maurer. Authentication theory and hypothesis testing. *IEEE Transaction on Information Theory*, 46(4):1350–1356, 2000.

[24] Bruce Schneier. *Applied Cryptography*. John Wiley and Sons, Inc, New York, NY, USA, 1996.

[25] Bruce Schneier. Why cryptography is harder than it looks. *Counterpane Systems*, 1997.

[26] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[27] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423; 623–656, 1948.

[28] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[29] G. J. Simmons. A game theory model of digital message authentication. *Congressus Numerantium*, 34:413–424, 1982.

[30] G. J. Simmons. Message authentication: a game on hypergraphs. *Congressus Numerantium*, 45:161–192, 1984.

[31] G. J. Simmons. Authentication theory/coding theory. *Advances in Criptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science, Springer-Verlag*, 196:411–432, 1985.

[32] Gustavus J Simmons. Authentication theory/coding theory. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 411–431, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[33] Michael Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 1996.

[34] Douglas R. Stinson. *Cryptography: theory and practice.* CRC Press, Boca Raton, Florida, 1995.

[35] A. Turing. On computable numbers with an application to the entscheidugsproblem. *Proceedings of the London Mathematical Society*, 42:230–365, 1936.

[36] R. W. Yeung. *A First Course in Information Theory.* Kluwer, New York, 2002.