Complexidade e criptografia...

— Matéria opcional —

Fevereiro 2006

Armando B. Matos

Índice

Algumas noções de teoria dos números

Eficiência: medida em termos do comprimento (número de bits ou de dígitos) $n = |x|$ dos números, não em termos de x. Em bits:

$$|x| = \lceil \log(x+1) \rceil$$

Nota. Há 2^{n-1} inteiros positivos de comprimento n (porquê?) e há $2^n - 1$ inteiros positivos de comprimento $\leq n$ (porquê?); assim, qualquer algoritmo que considere todos os inteiros de comprimento n ou $\leq n$ é necessariamente exponencial e portanto impraticável se n não for muito pequeno.

Seja n o comprimento do maior dos inteiros que fazem parte dos dados (são operandos) em cada um dos algoritmos seguintes.
Algoritmos para

- Soma: Algoritmo escolar: $O(n)$
- Multiplicação
  - Algoritmo escolar: $O(n^2)$
  - Algoritmo baseado na divisão recursiva dos inteiros em 2 partes: $O(n^{\log_2 3})$
  - Algoritmo baseado no FFT: $O(n(\log n)(\log\log n))$
- Exponenciação discreta: $x^y \mod z$: $O(n^3)$.
- Primalidade de x
  - Algoritmo que testa divisibilidade por $2,\ldots, x/2$: Exponencial
  - Algoritmo que testa divisibilidade por $2,\ldots, \lfloor\sqrt{x}\rfloor$, hiper-polinomial.
  - Algoritmos "aleatorizados" de Rabin-Miller (1980), polinomiais
    Nota sobre algoritmos "aleatoriados".
  - Agrawal *et al* publicaram em 2002 um algoritmo polinomial (determinístico) para o teste da primalidade. Este resultado torna obsoletas algumas suposições e resultados da complexidade da teoria dos inteiros.
- Factorização: não é conhecido nenhum algoritmo polinomial.

Exercício. Mostre que os dois primeiros algoritmos referidos para o teste da primalidade de x são hiper-polinomiais (em $n = |x|$).

Exercício. Mostre que é possível calcular $x^y \mod (z)$ em tempo $O(n^3)$. Implemente esse algoritmo numa linguagem imperativa como o C supondo que os operandos são suficientemente pequenos por forma a não ser necessário utilizar aritmética de precisão arbitrária.
Sugestão: Começe por calcular as sucessivas potências $x^2 \mod z$, $x^4 \mod z$, $x^8 \mod z$,...

Teoria dos números
Algumas definições e resultados I

Definição. (m, n): máximo divisor comum entre m e n.

Definição. $\Phi(n) = \{m \mid m < n, (m, n) = 1\}$, conjunto dos inteiros menores que n que são primos com n.
Exemplo: $\Phi(12) = \{1, 5, 7, 11\}$.

Definição. Função de Euler, $\phi(n) = |\Phi(n)|$.
Exemplo: $\phi(12) = 4$

Exercício. Mostre que n é primo sse $\phi(n) = n - 1$.

Exercício. Suponha que m e n são primos entre si. Mostre que $\phi(mn) = \phi(m)\phi(n)$.

Teorema (de Euler) Sejam a e m primos entre si. Então

$$a^{\phi(m)} = 1 \mod m$$

Exemplo Verifique o teorema de Euler para $m = 12$, $a = 1, 5, 7, 11,$.
Temos $\phi(12) = 4$ e, por ex., $5^4 = 625 = 52 \times 12 + 1$.

Teorema (Caso particular do Euler):
Se p é primo, então para todo o a, $1 \leq a < p$ é $a^{p-1} = 1 \mod p$.

Exemplo Verifique o teorema de Euler para $m = 7$, $a = 1, 2, \cdots, 6$.
Temos por exemplo $3^6 = 729 = 104 \times 7 + 1$.

Definição. $(A, +, \times)$ é anel: A adição "+" é um grupo comutativo. A multiplicação "×" é associativa e distributiva relativamente à adição.

Teorema. O conjunto $\{0, 1, \cdots, n-1\}$ juntamente com as operações de "+" e "×" módulo n formam um anel comutativo com elemento neutro para a multiplicação designado por anel dos resíduos de n. Se n for primo, cada elemento $a \neq 0$ tem um inverso multiplicativo a^{-1} tal que $aa^{-1} = 1$ (o conjunto $\{1, 2, \cdots, p-1\}$ é um grupo relativamente à multiplicação. A estrutura é um *domínio de integridade*).
Se n não é primo, no conjunto $\mathbb{Z}_n^\star$ dos inteiros positivos menores que n e relativamente primos com n (isto é, $\text{mdc}(i, n) = 1$ para qualquer $i \in \mathbb{Z}_n^\star$) cada elemento tem também um inverso módulo n.

Teoria dos números
O algoritmo de Euclides

O algoritmo de Euclides determina o máximo divisor comum entre dois inteiros e corresponde à recorrência

$$mdc(m,n) = \begin{cases} n & \text{se } m = 0 \\ mdc(n \bmod m,\ m) & \text{se } m \geq 1) \end{cases}$$

Exercício. Implemente o algoritmo de Euclides em linguagem C ou noutra linguagem imperativa.

Exercício. Siga o algoritmo de Euclides para os dados $m = 24$, $n = 18$.

Exercício. Justifique o funcionamento do algoritmo dado, isto é, mostre que termina sempre dando $(m,\ n)$ como resultado.

Algoritmo de Euclides estendido
Determina m' e n' (que podem ser negativos) tais que $m'm + n'n = (m,n)$.

```
euc_par(m,n){
  if(m=0) return(m'=0, n'=1);
  r1, m1 = euc_par(n mod m, m);
  return(m'=m1-(n/m)r1, r1);
  /* n/m representa o quociente inteiro da divisão de n por m   */
```

Exercício. Siga o algoritmo para $m = 12$, $n = 18$.

Exercício. Sejam a e c primos entre si. Mostre que o algoritmo de Euclides estendido pode ser adaptado para calcular o inverso multiplicativo de a módulo c.
Sugestão. Seja $m = a$, $n = c$ e considere $m'm + c'c = 1$. Resulta $aa' = 1 \mod c$.

Teoria dos números, primalidade

O Teorema seguinte pode ser demonstrado com resultados da teoria dos números que referiremos de seguida.

Teorema Um número $p \geq 2$ é primo sse existir um número $2 \leq r < p$ tal que verifica simultaneamente as condições

1. $r^{p-1} = 1 \mod p$
2. Para todo o divisor primo q de $p-1$ é $r^{\frac{p-1}{q}} \neq 1 \mod p$

Definição Um inteiro r nas condições do Teorema anterior diz-se uma *raiz primitiva* de p.

Corolário (Teorema de Pratt) O problema da primalidade pertence a $\mathbf{NP} \cap \mathbf{co} - \mathbf{NP}$.
O problema da primalidade é obviamente $\mathbf{co} - \mathbf{NP}$ pois o seu complementar - "O número n é composto?" - tem testemunhos polinomiais: factores a e b tais que $n = ab$. Pelo teorema anterior o problema da primalidade é também **NP** pois para cada primo existe um testemunho (certificado) polinomial. A descoberta recente que já referimos (Agrawal *et al*, 2002) de que o problema da primalidade pertence a P torna este teorema ultrapassado (embora verdadeiro!).

Teorema (do "resto chinês") Seja $n = p_1 p_2 \cdots p_k$ um produto de primos distintos e seja $(r_1, r_2, \cdots, r_k)$ um tuplo de inteiros $1 \leq r_i < p_i$ para $i = 1, \ldots, k$. Então existe um único $r < n$ primo com n (isto é, pertencente a $\Phi(n)$) tal que, para $i = 1, \cdots, k$, é $r_i = r \mod p_i$.

Exemplo Seja $n = 70 = 2 \times 5 \times 7$ e consideremos o tuplo $(1, 2, 6)$. A solução (única) é 27 uma vez que $1 = 27 \mod 2$, $2 = 27 \mod 5$, $6 = 27 \mod 7$.

Teoria dos números
Primalidade e fatorização

Dizemos que um problema é ⟨fácil⟩ quando existe algoritmo polinomial que o resolve e difícil caso contrário. A presença de "..." indica que há alguma incerteza na afirmação (por exemplo, ... ⟨Difícil⟩...).

Problema de decisão Um inteiro n é primo?
⟨Fácil⟩. Resultado já referido de 2002.

Problema de decisão Um inteiro n é composto?
⟨Fácil⟩. Semelhante em complexidade ao anterior.

Problema Factorizar um inteiro não primo, isto é, encontrar a e b, $2 \leq a \leq b < n$ tal que $n = ab$.
... ⟨Difícil⟩....Parece ser um problema para o qual os melhores algoritmos são hiper-polinomiais.

Alguns problemas em anéis de resíduos

Problemas no anel dos resíduos de n, $\{0, 1, \cdots, n-1\}$.

Problema Exponenciação discreta
Dados a, b e m, determinar $a^b \mod m$.
⟨Fácil⟩. Como vimos é polinomial, $O(n^3)$.

Problema Logaritmo discreto.
Dado a, m e e, determinar b tal que $e = a^b \mod m$.
... ⟨Difícil⟩....

Problema Inverso multiplicativo, determinação de b tal que $ab = 1 \mod m$ supondo que a e m são primos entre si.
⟨Fácil⟩. Como vimos o algoritmo de Euclides estendido pode ser usado para resolver este problema pelo que existe solução polinomial.

Definição. Dizemos que a é um *resíduo quadrático* módulo n se existe $b \in \mathbb{Z}_n^\star$ tal que $b^2 = a \mod n$.

Problema do resíduo quadrático Dado um inteiros a e n saber se a é um resíduo quadrático módulo n.
⟨Fácil⟩ se n é primo, ... ⟨Difícil⟩... se n é composto.

Criptografia – princípios gerais

Alguns objectivos que se pretendem atingir em canais inseguros (A e B são, por exemplo, pessoas)

- *Transmissão de informação em canais inseguros*
  A transmite informação sigilosa (ex: mensagens pessoais, número do cartão de crédito), num canal inseguro como a internet.
- *Provas de "conhecimento zero"*
  A convence B que sabe algo de "valioso" (a solução de um problema difícil, a demonstração de um teorema...) sem nada revelar sobre essa solução.
- *Prova de identidade*
  A convence B da sua própria identidade - isto é, que se trata mesmo de A e não de um impostor (autenticação da assinatura).
- *Modelo de Yao*
  A e B, com um mínimo de comunicação, calculam ambos $f(x, y)$; inicialmente A conhece x, B conhece y e o algoritmo de cálculo de $f(\cdot, \cdot)$ é conhecido dos dois.
- *Compromisso*
  Num canal ou rede insegura A faz uma escolha privada, sendo capaz de validar essa escolha e incapaz de "fazer batota", isto é, dizer mais tarde que a escolha foi outra.

Muitas aplicações...

Criptografia, notação

Nomenclatura e notação relativas à transmissão de uma mensagem A para B:

$\boxed{w}$ - *Texto* a transmitir. Em inglês: "plaintext" ou "cleartext".

$\boxed{k}$ - *Chave*. Em inglês: "key".

$\boxed{c}$ - *Cifra* ou *texto encriptado*. Em inglês: "cryptotext" ou "ciphertext".

$\boxed{e}$ - Algoritmo para encriptar(= cifrar).

$\boxed{d}$ - Algoritmo para decifrar (= "descriptar").

Sistema de encriptação

Em geral:
- A codifica a mensagem à custa de uma chave e do algoritmo para encriptação.
- B descodifica a mensagem à custa de uma chave (da mesma ou de outra) e do algoritmo de decifrar.

Criptografia

Sistema de encriptação, em geral

- Espaço de texto (original), seja $\Sigma^\star$.
- Espaço de texto encriptado, seja $\Delta^\star$.
- Espaço das chaves, K
- Função de encriptação, $e:\ K \times \Sigma^\star \to \Delta^\star$.
- Função de decifração, $d:\ K \times \Delta^\star \to \Sigma^\star$.

As funções e e d têm de ser inversas no sentido seguinte

$$\forall k \in K,\ \forall w \in \Sigma^\star,\ \ d(k, e(k, w)) = w$$

Existem sistemas com um par de chaves, $(k_1,\ k_2)$, em que k_1 é a chave de encriptação e k_2 a chave de decifração.

$$\forall \text{ par de chaves } (k_1, k_2),\ \forall w \in \Sigma^\star,\ \ d(k_1, e(k_2, w)) = w$$

Por ordem de segurança os sistemas criptográficos podem ser

- **Inseguros:** A existência de um texto cifrado relativamente longo permite com alguma facilidade (frequência das letras...) conhecer a chave e/ou o algoritmo de decifração, decifrando-o.
- **Seguros no sentido da dificuldade computacional (teoria da Complexidade):** O conhecimento do texto cifrado pode dar alguma informação sobre o texto original (chave ou algoritmo de enencriptação) mas os melhores algoritmos para esse fim demorariam um tempo proibitivo (por exemplo, exponenciais). Diz-se que o problema da decifração é "intratável".
- **Seguros no sentido da teoria da Informação:** O conhecimento do texto cifrado não dá qualquer informação pelo que se torna, mesmo em princípio, impossível decifrá-lo.

Outra distinção importante:

- Chave privada
- Chave pública

Apenas aceitando a segurança no sentido da sentido da dificuldade computacional é possível a existência da (hoje tão utilizada!) criptografia de chave pública.

Métodos de chave privada

Alguns métodos de chave pequena

CÉSAR - $\Sigma = \Delta = \{\text{' '}, A, B, \ldots, Z\}$, $k \in \{0, 1, \ldots, 26\}$, e e h são morfismos, basta defini-los para as letras:

$$\begin{aligned} e(k, a) &= \text{Letra (ciclicamente) } k \text{ posições à frente de } a \\ d(k, a) &= e(27 - k, a) \end{aligned}$$

Exemplo.

$$e(3, \text{“MODELOS FORMAIS”}) = \text{“PRGHORVCIRUPDLV”}$$

Muito pouco seguro. Algumas linhas de texto encriptado podem ser suficientes para o decifrar.

Exercício. Como procederia para, dada uma cifra encriptada por este método, descobrir a mensagem original?

PERMUTAÇÂO DAS LETRAS - Outro sistema mono-alfabético; uma chave é uma qualquer permutação das letras do alfabeto. Há 27! chaves possíveis se incluirmos também o espaço " ". Testar as chaves todas é inviável (exercício: justifique numericamente esta afirmação).

Exercício. Como procederia para, dada uma cifra encriptada por este método, descobrir a mensagem original?

SENHA-CÉSAR - A chave é uma palavra secreta (senha) que define uma sequência de transformações das letras. Por exemplo, para a senha “CABE” a a sequência de transformações (em ciclo) é $(3, 1, 2, 5)$. Se o texto original é $w = a_1a_2a_3a_4a_5a_6 \ldots$ e cifra é

$$c = e(3, a_1)e(1, a_2)e(2, a_3)e(5, a_4)e(3, a_5)e(1, a_6) \ldots$$

Um pouco mais seguro mas não muito...

Exercício. Como procederia para, dada uma cifra encriptada por este método, descobrir a mensagem original?

"One time pad" – um método com chave grande

- |chave| = |mensagem|
- Método seguro em termos da teoria da Informação

Chave k, $|k| = |w|$, previamente comunicada por métodos mais seguros.

$$c = w \oplus k \text{ (ou exclusivo, bit a bit)}$$

A decifração é também o ou exclusivo:

$$w = c \oplus k = (w \oplus k) \oplus k$$

Vantagens e inconvenientes

1. Vantagem: Método muito seguro, seguro em termos da teoria da Informação; se a chave se mantiver secreta, a segurança é absoluta no sentido em que o conhecimento de c não dá qualquer informação (entropia 0) sobre as possíveis mensagens.
2. Inconveniente: Necessidade de canais mais seguros para transmitir previamente a chave.
3. Inconveniente: Grande tamanho da chave.

Os inconvenientes referidos tornam o método pouco interessante para a transmissão frequente de mensagens (ex: internet). Tem interesse para fins militares....
O conhecimento do texto encriptado c não dá qualquer informação sobre o texto original:

$$\text{pr}\{w = w_0 \mid c = c_0\} = \text{pr}\{w = w_0\}$$

Porquê?
Independentemente de w, qualquer c tem a probabilidade 2^{-n} de ocorrer porque

$$\forall c\ \exists! k\ \ w \oplus k = c$$

Exercício. Considere a seguinte troca de mensagens que utiliza uma variante do "one-time pad" sem pré-transmissão de chave onde $|w| = |k_A| = |k_B|$ e "$\oplus$" é o ou exclusivo, bit a bit.

(1) A envia a mensagem w criptada com uma chave própria k_A: $k_A \oplus w$.

(2) B recebe a mensagem criptada e crifra-a com a sua própria chave, enviando-a de retorno a A: $k_B \oplus (k_A \oplus w)$.

(3) A cripta a mensagem recebida e envia-a para B: $k_A \oplus (k_B \oplus (k_A \oplus w))$.

(4) Finalmente B cripta a mensagem recebida e envia-a para A: $k_B \oplus (k_A \oplus (k_B \oplus (k_A \oplus w)))$

Perguntas:

1. Que mensagem recebe A no fim?
2. Mostre que "$\oplus$" (entre palavras de igual comprimento né uma operação associativa, comutativa, tem 0 (n bits iguais a 0) como elemento neutro e, para toda a palavra x é $x \oplus x = 0$.
3. Um "mau" que observasse o canal inseguro que informação poderia obter (w, k_A ou k_B) em função das mensagens interceptadas (de entre (1),..., (4)).
4. B fica a conhecer w?

(Exercício, resposta parcial. . .)
Suponhamos que o mau (Eve ou E) sabe como se obtiveram as mensagens enviadas mas não os valores de k_1, k_2 e w.

Com o conhecimento das mensagens passa a saber esses valores. Sejam m_1, m_2, m_3 e m_4 as mensagens. Dada a comutatividade, associatividade e idempotência de $\oplus$, temos

$$\begin{array}{lcll} m_4 & = & w & \rightarrow w \\ m_3 & = & k_B \oplus w & \rightarrow k_B \\ m_2 & = & k_B \oplus k_A \oplus w & \rightarrow k_A \end{array}$$

Chaves e teoria da Informação

Resultado da teoria da Informação:
A qualquer sistema criptográfico podemos associar um inteiro n, *distância unitária* tal que qualquer texto encriptado de comprimento $\geq n$ determina univocamente a chave usada.

$$n \approx \frac{\text{Incerteza inicial na chave}}{\text{Redundância por símbolo na linguagem do texto}}$$

A "incerteza inicial na chave" e a "redundância por símbolo" são medidas em termos de entropia. Grande redundância corresponde a pequena incerteza - pequena entropia.

Entropia e informação
No outro texto de criptografia disponível na página desta disciplina, `[4B]`, dá-se uma ideia da utilização da entropia para análise da segurança dos métodos criptográficos de chave privada. Em particular foram brevemente referidos nas aulas teóricas os seguintes temas.

- A informação necessária para seleccionar 1 elemento de um conjunto de n: $\log n$ bits. Dois conjuntos com m e n elementos: aditividade da informação.
- Transmissão de um conjunto de mensagens, probabilidades associadas. A informação (ou surpresa) que resulta de receber a mensagem particular i é (em bits): $\log(1/p_i)$.
- A entropia: (Shannon[1] 1948) surpresa média $H(p_1 \cdots p_n) = \Sigma_i p_i \log(1/p_i)$. Teorema do canal.
- Entropia condicional. Segurança absoluta de um sistema criptográfico:
  $\forall w \quad H(w|c) = H(w)$, isto é, o conhecimento da cifra nunca altera as probabilidades das mensagens.
- O "one-time pad" tem segurança absoluta (demonstração), o RSA não.

[1] Em Física o conceito de *entropia* é análogo mas muito anterior.

Métodos de chave pública

Existem 2 chaves, uma k_1 de encriptação e outra k_2 de decifração que são normalmente geradas em pares (k_1, k_2).

A chave e o algoritmo de encriptação são públicos. O algoritmo de decifração é também público, mas a chave correspondente k_2 é privada.

Como é que A manda uma mensagem w para B?

1.	B gera um par de chaves (k_1, k_2) e publica (através do canal inseguro, por exemplo, pela internet) a chave de encriptação k_1.
2.	A encripta w, $c = e(k_1, w)$ e envia a mensagem encriptada c para A pelo canal inseguro.
3.	B recebe e decifra c: $w = d(k_2, c)$.

Notemos que só B conhece k_2 e que esta chave nunca circula no canal inseguro!

Complexidade dos métodos de chave pública

O bom funcionamento deste método baseia-se em algumas propriedades da complexidade das funções.

- O cálculo de $c = e(k_1, w)$ é eficiente (polinomial).
- O cálculo de $w = d(k_2, c)$ é eficiente (polinomial).
- A determinação do único w tal que $c = e(k_1, w)$ pode ser teoricamente possível mas praticamente impossível (por exemplo se existirem apenas algoritmos de tempo exponencial).

Estes métodos são seguros em termos de complexidade mas não em termos da teoria da Informação, isto é, a sua segurança baseia-se apenas no tempo de execução proibitivo dos algoritmos de pesquisa.

Vemos que a existência de problemas inerentemente complexos é fundamental para o funcionamento dos métodos de chave pública.

Pretende-se que, para cada k_1, a função

$$e(k_1, \cdot) : \Sigma^\star \to \Delta^\star$$

seja "one-way", isto é, que seja eficiente de calcular no sentido directo mas intratável (por exemplo, exponencial) no sentido inverso.

Chave privada e chave pública, comparação

1. Chave privada:
   Há necessidade de enviar previamente a chave por um canal seguro para a outra pessoa.

2. Chave pública:
   A chave de encriptação é conhecida de todos, pode existir por exemplo, num ficheiro da internet. Não é necessário outro canal de comunicação. Nem sequer é necessário conhecer a outra pessoa.

Analogia

1. Chave privada:
   Envio um cofre. Enviei previamente uma cópia da chave (k) por meios seguros.

2. Chave pública:
   Disponibilizo cofres abertos para quem me quizer enviar mensagens, bastando pressionar a tampa (k_2) para ficarem trancados. Só eu disponho da chave (k_2) para abrir o cofre.

Funções facilmente computáveis num só sentido

A existência de protocolos seguros em termos da teoria da Complexidade é baseada na possível existência das chamadas funções "one-way".
Dizemos que $f : \Sigma^\star \to \Sigma^\star$ é "one-way" quando

1. f é injectiva.

2. $|x|$ e $|f(x)|$ estão polinomialmente relacionados, isto é, existe $k > 0$ tal que, para todo o $x \in \Sigma^\star$, é $|x|^{\frac{1}{k}} \leq |f(x)| \leq |x|^k$.

3. $f(x)$ pode ser calculada em tempo polinomial em termos de $|x|$.

4. $f^{-1}(y)$ não pode ser calculada em tempo polinomial em $|y|$. Nota. O valor de $f^{-1}(y)$ é definido como x se existe x tal que $f(x) = y$ ou "$\perp$" se não existe tal x.

Nota Pelas 2 primeiras condições existe um algoritmo que determina $f^{-1}(y)$; tal algoritmo pode consistir numa pesquisa exaustiva de x tal que $f(x) = y$. Mas, pela última condição, nenhum desses algoritmos é eficiente (polinomial).

Suponhamos por exemplo que definimos: "eficiente" ≡ "polinomial" e "intratável" ≡ "NP-completo".

- A encriptação $c = e(k_1, w)$ deve ser eficiente
- Dado c e k_1, determinar w tal que $c = e(k_1, w)$ deve ser intratável.

Conclusão: (com as nossas convenções) se P=NP, não existe criptografia de chave pública. Mas na prática não basta ser NP-completo: é necessário que, com grande probabilidade, as instâncias sejam difíceis de solucionar.

Na prática usa-se no método RSA como "intratável" um problema, o da factorização, que não é um problema de decisão e que não está aparentemente associado a algum problema NP-completos.

Funções facilmente computáveis num só sentido

Existem funções "one-way"?

Pensa-se que o produto de primos é uma função "one-way"

Multiplicação de primos e factorização de inteiros
Sejam p e q primos. O produto pq é computável em tempo polinomial mas não é conhecido nenhum algoritmo polinomial para factorizar um número em 2 primos. Mais concretamente

1. Existe um algoritmo A polinomial que calcula o produto de dois primos dados p e q. Além disso, sendo p e q primos, esse produto é uma função injectiva.
2. Pensa-se que não existe um algoritmo polinomial A que factoriza um inteiro x

$$F(x) = \begin{cases} (p,q) & \text{Se existem primos } p \text{ e } q \text{ tais que } x = pq \\ 0 & \text{no caso contrário} \end{cases}$$

Nota. A descoberta recente (2002) de que o problema da primalidade (problema de decisão "um inteiro dado é primo?") pertence à classe P permite dispensar a necessidade de certificados de primalidade.

O método RSA

Método devido a Rivest, Shamir e Adleman, baseado na teoria dos números.

Criação das chaves Sejam p e q primos grandes (centenas de dígitos). Calcula-se

$$n = pq, \;\; \phi(n) = (p-1)(q-1)$$

(ϕ é a função de Euler) Escolha-se $e \geq 2$ primo com $\phi(n)$ e determine-se d tal que

$$ed = 1 \mod \phi(n)$$

Como sabemos a determinação de d, inverso multiplicativo de e é eficiente.

Chave pública (n, e).
n: "módulo", e: "expoente".

Encriptação O texto original é dividido em blocos w representados por sequências de dígitos. O texto encriptado é

$$c = w^e \mod n$$

Descriptação É conhecido $c = w^e(\mod n)$ bem como d. Chama-se a d o expoente de descriptação.

Pelo teorema seguinte é

$$w = c^d \mod n$$

O valor d é a componente essencial da descriptação.

Teorema Seja $ed = 1 \mod \phi(n)$ e $c = w^e \mod n$ nas condições indicadas. Então $w = c^d$.

Dem. Existe j tal que $ed = j\phi(n) + 1$. Pelo teorema de Euler

$$w^{\phi(n)} = 1 \mod n$$

desde que nem p nem q dividam w. Nestas condições

$$c^d = w^{de} = w^{j\phi(n)+1} = (w^{\phi(n)})^j w = w \mod n$$

É fácil ver que esta igualdade continua verdadeira mesmo que p ou q dividam w (verificar!). Assim, temos sempre $c^d = w \mod n$.

Os fundamentos computacionais do método RSA são:

1.	A eficiência da exponenciação discreta, $c = w^e \mod n$, do inverso multiplicativo e da geração de primos aleatórios.
2.	A dificuldade do logaritmo discreto: dado c, e e n determinar w tal que $c = w^e \mod n$.

O método RSA

Porque é difícil o ataque?
Quem conhece a chave pública (n, e) e a cifra c (mas não conhece d) pretende encontrar w tal que

$$w^e \mod n = c \mod n$$

Este problema é o problema do logaritmo discreto para o qual, como vimos, todos os algoritmos conhecidos são hiper-polinomiais (e praticamente inutilizáveis).

Será possível descriptar mensagens c conhecendo apenas c e a chave pública (n, e)?

Como vimos, um método para o conseguir seria a existência de um algoritmo eficiente (polinomial) para factorizar n (não se conhece nenhum!):

1. Determina-se p e q tais que $n = pq$
2. Calcula-se $\phi(n) = (p-1)(q-1)$
3. Determina-se d tal que $de = 1 \mod \phi(n)$
4. Calcula-se a mensagem original $c^d \mod n$

Mas poderia haver outro método mais indirecto... Haverá?
Não se sabe. Sabe-se que no caso especial $e = 2$ tal método não pode existir.

Teorema de Rabin No método RSA com $e = 2$ factorizar n (supostamente um produto de 2 primos) e descriptar uma mensagem c são polinomialmente equivalentes.

Dem: No sentido $\rightarrow$ já vimos.
Suponhamos então que n é o produto de dois primos p e q (desconhecidos) e que

$$x^2 = c \mod n$$

tem pelo menos uma solução. Suponhamos que temos um método (oráculo) polinomial de determinar *uma solução* desta equação. Vamos ver que isso resultaria também num método polinomial (mas "aleatorizado") para factorizar n.

Se $\alpha^2 = c \mod n$, então também $n - \alpha$ satisfaz a igualdade. Na realidade, a equação $x^2 = c \mod n$ tem no máximo 4 soluções. São da forma

$$\alpha,\ \beta,\ n-\alpha,\ n-\beta$$

Ataques ao RSA

Equações com solução para $n = 3 \times 5$.

```
x2=  1 (mod  15):    1   4  11  14
x2=  4 (mod  15):    2   7   8  13
x2=  6 (mod  15):    6   9
x2=  9 (mod  15):    3  12
x2= 10 (mod  15):    5  10
```

Teorema de Rabin - continuação
Supondo que a equação tem solução admitimos que o oráculo dá com igual probabilidade uma qualquer das soluções. Procedemos assim

1. Escolhemos α aleatóriamente com igual probabilidade, entre 1 e $n-1$.
2. Damos ao oráculo os dados $c = \alpha^2$ e n. A equação tem obviamente uma solução (α).
3. Com probabilidade positiva o oráculo dá uma solução β com $\beta \neq \alpha$ e $\beta \neq n - \alpha$.
4. Se for este o caso (ver abaixo), um dos factores primos de n é (n, α) e o outro é (n, β). A factorização obtém-se assim pelo algoritmo de Euclides.
5. Se não for o caso, recomeça-se do início.

A probabilidade de sucesso converge para 1.

Exercício. Suponha que α e β são raízes da equação $x^2 = c \mod n$ e que n é o produto de 2 primos, e que $\beta \neq \alpha$ e $\beta \neq n - \alpha$. Mostre que um dos primos é o máximo divisor comum entre n e $\alpha + \beta$ e que o outro é o máximo divisor comum entre n e $\alpha - \beta$

Para o caso $e > 2$ não se conhece nenhum resultado importante.
Portanto, dentro do conhecimento actual e assumindo que o problema da factorização é intratável, podem existir algoritmos polinomiais de descriptação.

Criptografia baseada no problema "knapsack"

Problema de decisão "knapsack"
Instância. Inteiro x e conjunto de inteiros

$$S = \{a_1, a_2, \cdots, a_n\}$$

Pergunta. x é a soma de alguns dos a_i (sem repetição)? Em símbolos: existem $1 \leq i_1 \leq i_2 \leq \cdots i_k \leq n$ tais que $a_{i_1} + a_{i_2} + \cdots + a_{i_k} = x$?

Exemplo: $n = 88$, $S = \{7, 21, 23, 31, 47, 54, 71\}$
Trata-se de um problema **NP**-completo. Não existe portanto nenhum algoritmo conhecido que seja polinomial mesmo no pior caso. O problema mantém-se NP-completo quando $n = (\Sigma_{i=1}^{n} a_i)/2$ (problema da partição).
Alguns casos particulares do problema pertencem a **P**. Um caso trivial: instâncias em que $n > \Sigma_{i=1}^{n} a_i$. Outro caso:

Problema de decisão "knapsack" super-crescente
Tem a restrição: cada a_i é maior que a soma de todos os precedentes, isto é

$$a_k > \Sigma_{i=1}^{k-1} a_i \text{ para } j = 2, \cdots, n$$

Exercício. Mostre existe um algoritmo polinomial para o problema do "knapsack" super-crescente.
Sugestão. Mostre que a_n entra na soma sse $x \geq a_n$. Por um processo análogo determine se cada um dos elementos a_{n-1}, a_{n-2},..., a_1 está ou não incluído na soma.

Exercício. Discuta o problema do "knapsack" no caso em que $a_i = 2^{i-1}$ (para $i = 1, \cdots, n$).

Exercício. Mostre que para efeitos de eficiência (polinomial ou não polinomial) podemos supor que o conjunto S é dado como uma sequência ordenada por ordem crescente.

Encriptação da mensagem:

Chave pública Uma sequência $S = (a_1, a_2, \cdots, a_n)$.

Encriptação O texto w é divido em blocos de n bits cada um dos quais é visto como um vector coluna B de n bits. O texto encriptado é o inteiro $c = (a_1, a_2, \cdots, a_n)B$.

Exemplo: Chave pública (7, 21, 23, 31, 47, 54, 71).
Bloco de texto: 0110100. Fica

$$c = (7, 21, 23, 31, 47, 54, 71) \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = 91$$

Criptografia "knapsack" – Construção da chave pública

Informação e processamento privados

1. Gera-se uma instância do "knapsack" super-crescente, $A' = (a'_1, a'_2, \cdots, a'_n)$
2. Considera-se um $M > 2a'_n$, o *módulo*.
3. Gera-se um inteiro u, primo com n.
4. Calcula-se o inverso de u (módulo M), isto é u^{-1} tal que $uu^{-1} = 1 \mod M$

A informação privada é (A', u, u^{-1}, M).

Construção da chave pública Baralhar: Cada a'_i é multiplicado por u (módulo M).
Isto é, $A = (a_1, a_2, \cdots, a_n)$ com

$$a_i = ua'_i \mod M$$

Criptografia "knapsack" – Descriptação
A descriptação é baseada no seguinte resultado.

Teorema Sejam A', A, M e u como definidos atrás, c' um inteiro e $c = uc' \mod M$. Então
- (c', A') tem no máximo uma solução. - (c, A) tem no máximo uma solução. - Se (c, A) tem solução então (c', A') tem solução.

Dem. Resulta do exercício que (c', A') ou não tem solução ou tem uma só solução.
Seja X (vector de bits) uma possível solução de (c, A), isto é, $c = AX$. Vem

$$c' = u^{-1}c = u^{-1}AX = u^{-1}uA'X = A'X \mod M$$

Como $M > a'_n$ é $A'X < M$ e portanto $c' = A'X$. Assim, se (c, A) tiver solução, (c', A') também tem solução.

Pelo resultado anterior, se conhecermos (A', u, u^{-1}, M) e a mensagem criptada c podemos reconstituir B (bloco de n bits de w.

Descriptação:

1. Calcula-se $c' = u^{-1}c \mod M$.
2. Resolve-se o "knapsack" super-crescente (c', A'). A solução é B.

Exercício. Supondo que foi de facto encriptada uma mensagem em c, mostre que o algoritmo descrito acima permite obter de forma unívoca a mensagem original. Mostre que este algoritmo é polinomial.

Protocolos diversos

Protocolos – Transmissão de 1 bit

Transmissão de $b \in \{0, 1\}$ aproveitando o RSA.

- Directamente: $b^e \mod n = b$! Muito má ideia (**Porquê?**).
- Último bit de um x fixo: encriptar $2x+b$: má ideia se x for usado repetidamente (**Porquê?**).
- Usando um x aletório (com igual probabilidade), $0 \leq x < n$.
  Notas:
  - Teorema Obter o último bit de uma mensagem criptografada pelo método RSA é polinomialmente equivalente a obter toda a mensagem.
  - Por este processo podemos transmitir w como uma sequência de bits. Este método
    * Resulta em mensagens transmitidas mais longas mas...
    * É mais seguro pois resiste (i) A tentativas descobrir se w pertence a um pequeno conjunto de mensagens possíveis (ii) Ao aproveitamento da possível ocorrência de mensagens repetidas.

Protocolos – Assinatura digital - I

A envia w a B e convence-o que se trata mesmo de A - e não de um impostor que também conhece a chave pública!
A mensagem vai assinada.

Aplicações Muitas, por exemplo: Autenticar a entidade que envia chaves públicas/privadas.

Será possível?
É possível com o RSA, solução elegante: A envia w seguido de w descriptado, como se fosse uma mensagem encriptada.

Chamando E e D aos algoritmos de encriptação e de descriptação, temos o protocolo:

1. A calcula $z = (w, D(d_A, w))$ e envia z encriptado.
2. B recebe a mensagem e descripta, obtendo z.
3. B encripta a segunda parte de z (que é $D(d_A, w)$) como se fosse enviá-la para A (B conhece z e e_A mas não d_A nem w!):
$$E(e_A, D(d_A, w)) = D(d_A, E(e_A, w)) = w$$
Como sabemos, a igualdade $D(d_A, E(e_A, w)) = w$ verifica-se em qualquer sistema criptográfico. O resultado é w, igual à primeira parte!
4. Se é igual a w, B reconhece a assinatura de A porque a construção de z parece exigir o conhecimento de d_A. Se não é igual é porque se trata de um impostor.

Exercício. Mostre que o sistema RSA satisfaz
$$E(e_A, D(d_A, w)) = D(d_A, E(e_A, w))$$
(tal deve-se essencialmente ao facto da multiplicação ser comutativa). Um sistema criptográfico nestas condições diz-se *comutativo*.

Protocolos – Conhecimento 0

A convence B que sabe qualquer coisa - por exemplo, a solução de um problema difícil - sem dar a B a mínima informação sobre essa solução!
Por exemplo, A vai convencer B que conseguiu colorir um determinado grafo com 3 cores - problema NP-completo - sem dizer qual é essa coloração.

Os objectivos fundamentais são:

1. A pretende convencer B mas não consegue fazer batota.
2. B tem apenas "capacidades" polinomiais.
3. B fica convencido mas...
4. não obtém qualquer informação sobre a solução.

Os ingredientes básicos da implementação são:

- Uso de aleatórios
- Interacção - troca de mensagens

Conhecimento 0 - colorir grafos com 3 cores

O problema 3-COL (problema NP-completo)
Instância: Um grafo não dirigido, $G = (V, E)$.
Pergunta: G pode ser colorido com 3 cores, isto é, existe uma função $f : V \to \{00, 01, 10\}$ tal que $\overline{(i,j)} \in E$ implica $f(i) \neq f(j)$?

Conhecimento 0 - o protocolo. Os passos seguintes (1. a 6.) repetem-se no máximo (quando há convencimento) $k|E|$ vezes onde k é um parâmetro de precisão (erro probabilístico).

1. Processamento interno de A
   Gera π, permutação aleatória das 3 cores.
   Para cada $i \in V$:
   - Gera chaves (p_i, q_i, d_i, e_i)
   - Gera aleatórios $0 \leq x_i, x'_i \leq \frac{p_i q_i}{2}$
   - Encripta os 2 bits da cor: $y_i = (2x_i + b_i)^{e_i} \mathrm{mod} p_i q_i, \;\; y'_i = (2x'_i + b'_i)^{e_i} \mathrm{mod} p_i q_i$.
2. A envia a B os $(e_i, p_i, q_i, y_i, y'_i)$ para $1 \leq i \leq |V|$
3. B escolhe um ramo aleatório $(i, j) \in E$ e envia-o a A.
4. A envia as chaves de descriptação d_i, d_j
5. B calcula a cor permutada de i: $b_i = (y_i^{d_i} \mathrm{mod} p_i q_i) \mathrm{mod} 2, \;\; b'_i = ({y'}_i^{d'_i} \mathrm{mod} p_i q_i) \mathrm{mod} 2$.
   Analogamente para j.
6. B não fica convencido se as cores permutadas dos 2 vértices forem iguais.

Conhecimento 0 - colorir grafos com 3 cores – Verificação

1. Se A dispõe da solução, consegue convencer B.
2. Se A não dispõe da solução o grafo encriptado e colorido que enviou no passo 2. a B tem pelo menos um ramo com extremidades coloridas com a mesma cor - e por isso com a mesma cor permutada. Assim, B tem, em cada passo, probabilidade $\geq \frac{1}{|E|}$ de detectar a "marosca". Ao fim de $k|E|$ passos independentes, se A não dispõe de solução, B detecta esse facto com probabilidade
$$\geq 1 - e^{-k}$$
que é arbitrariamente próxima de 1.

Notemos o seguinte

- A envia previamente o grafo colorido a B. É este que escolhe o par de vértices.
- Cada passo 1.-6. é independente dos outros. As chaves, aleatórios, etc. são gerados em cada passos.
- B só recebe como informação:
    1. Um grafo colorido encriptado que, para ele é puro lixo.
    2. Um par de cores permutadas, $\pi(i)$ e $\pi(j)$ que não lhe servem de nada para reconstituir a coloração do grafo.

  Assim, a informação transmitida a B é de facto nula.

Protocolos – Poker pela internet

Os métodos criptográficos usados em protocolos permitem obter resultados aparentemente impossíveis. Por exemplo, será possível o poker pela internet, entre 2 jogadores em locais diferentes (e sem nenhum deles estar a vigiar o outro)?

Objectivo, versão com 3 cartas, $a < b < c$

1. Uma carta para A, outra para B, diferentes.
2. Qualquer carta igualmente provável.
3. A conhece a sua carta mas não a de B (nem pode, em tempo polinomial, sabê-la); "vice-versa"
4. A anuncia a sua carta a B e B verifica que não houve batota.
5. B anuncia a sua carta a A e A verifica que não houve batota.
6. A maior das cartas ganha o jogo.

Estes objectivos generalizam-se facilmente para qualquer número de cartas...

Será possível? É!!!

Protocolo um pouco complexo

- ⋆ Os 2 jogadores geram um primo grande, p, do conhecimento de ambos. As chaves públicas são e_A e e_B e as secretas d_A e d_B.
- ⋆ Suponhamos que A dá as cartas: encripta-as e manda-as a B por ordem aleatória, a^{e_A}, b^{e_A} e c^{e_A} módulo p.
- ⋆ B escolhe uma das mensagens (que para ele são "lixo", como se as cartas estivessem viradas para baixo) e retorna-a para A.
- ⋆ A descodifica-a e guarda-a como a sua carta - foi B que a "escolheu" mas de uma forma aleatória.
- ⋆ B encripta as 2 mensagens (cartas) resultantes, seja a e c com a sua chave, obtendo $a^{e_A e_B}$ e $c^{e_A e_B}$ módulo p, e manda-as, por ordem aleatória, para A (que não as conhece, pois estão encriptadas por B).
- ⋆ A escolhe um destes "lixos", seja $a^{e_A e_B}$ (módulo p), descodifica-a com a sua chave d_A e manda o resultado para B, como a sua carta

$$a^{e_A e_B d_A} = a^{e_B} \mod p$$

- ⋆ B descripta-a, obtendo a.

Estes objectivos generalizam-se facilmente para qualquer número de cartas...

Exercício. Verifique se os objectivos são (ou parecem ter sido...) sido atingidos.