Cifras Assimétricas

MICEI/MSDPA

José Carlos Bacelar Almeida
(jba@di.uminho.pt)

Motivação

- Problema:

  Numa comunidade de *n* agentes, o estabelecimento de canais seguros (utilizando cifras simétricas) requer a partilha de $\frac{n*(n-1)}{2}$ chaves

- O **pré-acordo de chaves** é um procedimento custoso (requer a utilização de canais seguros...) e pouco flexível (e.g. considere-se a inclusão de mais um agente na comunidade...).
- ... em "redes abertas" a *distribuição de chaves é um problema*: o pré-acordo não é, de forma alguma, uma solução satisfatória.

Motivação (cont.)

- Analogia com exemplos práticos sugere a possibilidades de alternativas viáveis...
- Exemplo:

  Admita-se que dispomos de uma cifra (simétrica) em que a operação de cifra é comutativa, i.e.

  $$E_{k1}(E_{k2}(X))=E_{k2}(E_{k1}(X))$$

  - Para *A* comunicar *M* com *B* pode:
    - *A* envia a *B* $E_{KA}(M)$ - em que *KA* é só conhecida por *A*.
    - *B* devolve a *A* $E_{KB}(E_{KA}(M))=E_{KA}(E_{KB}(M))$ - em que *KB* só é conhecida por *B*.
    - *A* decifra mensagem recebida e re-envia a *B* o resultado, i.e. $E_{KB}(M)$
    - *B* decifra mensagem *M*.

  ... ou seja, *A* e *B* comunicam de forma segura sem partilharem segredos... (a mensagem *M* circula sempre protegida com, pelo menos, uma operação de cifra)

Cifra Assimétrica (Criptografia de Chave Pública)

- Conceito introduzido por *Diffie & Hellman* em 1976.
- Conceito base:
  - Duas chaves distintas são utilizadas na operação de cifra *Kc* de decifragem *Kd*.

    $$E(Kd,E(Kc(M))=M$$

  - O conhecimento de uma chave não permite retirar informação sobre a outra.
- ...leva ao conceito de *função de sentido único com segredo...*
  - Cifra com uma das chaves deve ser uma função de sentido único - não deve ser computacionalmente viável inverter essa função.
  - Mas informação adicional (outra chave) permite calcular operação inversa...
  - *Teoria de números* tem-se revelado a principal fonte de problemas que se acredita satisfazerem critérios requeridos...

Utilização básica... (o conceito)

- Utilização:
  - Cada agente dispõe de um par de chaves *(Kc,Kd)*

  Cifra:

  - Chave pública: *Kc* Chave privada: *Kd*
  - Para *A* enviar mensagem *M* a *B:* envia $E(Kc^B,M)$ - note que Kc^B é publicamente conhecida...
  - *B* decifra a mensagem utilizando a sua chave privada: E(KdB,M)=M

  ***A* dispõe de garantias que só *B* pode extrair o conhecimento de *M* porque só ele dispõe do conhecimento da chave privada.**

  Assinatura:

  - Chave pública: Kd Chave privada: Kc
  - *A* envia o par $(M,E(Kc^A(M))$ a *B* -
  - *B* verifica a assinatura comparando resultado da decifragem do criptograma (com Kd^A que é pública) com mensagem recebida.

  ***B* (ou qualquer agente) dispõe de garantias que *M* foi realmente enviada por *A* porque só ele dispunha da chave privada.**

Utilização (na prática)

- As cifras assimétricas são tipicamente várias ordens de grandeza menos eficientes do que as simétricas (e.g. 1000x)...
- ...por isso, são normalmente utilizadas em conjunção com estas (e não alternativamente).
- Utilização típica:

  Envelope digital - utilizado para garantir confidencialidade na transmissão de uma mensagem

  - *A* gera uma **chave de sessão *K*** (para uma cifra simétrica)
  - *A* envia a *B* par com $E(Kc^B,K)$ e $E_K(M)$ - $E_K(-)$ é uma cifra simétrica
  - *B* decifra *K* e utiliza essa chave para decifrar *M*.

  Assinatura digital - utilizada para garantir integridade/autenticidade/não repúdio da mensagem.

  - *A* utiliza uma função de hash criptográfica para calcular *H=hash(M)*
  - *A* envia a *B* o par constituído por *M* e $E(Kc^A,H)$.
  - *B* determina valor de hash e compara-o com resultado da decifragem

(Assinatura digital ...

O principal contributo da criptografia assimétrica foi o de permitir a definição de um *análogo digital* do conceito de *assinatura de um documento*.

- Em geral, podemos identificar uma assinatura digital como um "suplemento" à mensagem que nos permite verificar:
  - **Integridade:** a mensagem não modificada após a assinatura
  - **Autenticidade:** a identidade do *assinante* pode ser confirmada
  - **Não repúdio:** é possível demonstrar a identidade do assinante
- Destas propriedades resulta que, se o (**A**)ssinante produzir uma assinatura $x=\mathrm{Sig}^{A}(M)$, o (**V**)erificador com o par (M,x):
  - pode verificar que a origem de M é A, i.e. $\mathrm{Ver}^{A}(M,x)=true$
  - não pode produzir $M' \neq M$ tal que $\mathrm{Ver}^{A}(M',x)=true$

Obs.1: na essência do conceito de assinatura digital está uma assimetria entre as capacidades do verificador e do assinante: o primeiro deve estar habilitado a verificar as assinaturas produzidas pelo segundo sem dispor da capacidade de, ele próprio, as produzir.

Obs.2: note que os MACs garantem os dois primeiros requisitos mas falham no último (não repúdio) - nesse caso o verificador dispõe de tanta informação como o assinante.

...)

- As cifras assimétricas permitem a realização de um "Esquemas de Assinaturas" (como vimos atrás)

Obs.: as assinaturas digitais são facilmente duplicáveis (um aspecto que as distingue das assinaturas correntes). Este facto determina um cuidado particular em certas aplicações dessas assinaturas (e.g. ordens para transações financeiras...).

- Por vezes, existe interesse em incluir no "esquema de assinaturas" funcionalidade/propriedades que estendem as já referidas...

  Assinaturas não repudiávies - em que o procedimento de verificação requer a intervenção do assinante...

  Assinaturas com recuperação de mensagem (ARM) - o mecanismo de verificação não requer a mensagem assinada (como no esquema referido, designado Assinatura com Apêndice de Mensagem (AAM)).

  ...(consultar *Stinson, cap.6*)

Man-in-the-middle

- A utilização de técnicas criptográficas assimétricas requer uma associação fidedigna entre pares de chaves e identidades dos agentes.
- Na ausência desse pressuposto, é possível a um intruso fazer-se passar por outro agente comprometendo a segurança da técnica: ataque vulgarmente designado por *man-in-the-middle.*
- Exemplo:

  Suponhamos que *A* deseja cifrar uma mensagem para *B*.
  - Ao pedido de *A* relativo à chave pública de *B, I* responde com a sua própria chave pública Kc^I.
  - *A* envia $E(Kc^I,M)$...
  - *I* intercepta essa mensagem, decifra-a, e torna-a a cifrar utilizando a verdadeira chave pública de *B*
  - *B* decifra mensagem...

  A e *B* supõe que *M* se matem secreta mas *I* decifrou mensagem sem problemas...
- O mesmo tipo de ataque funciona para a assinatura digital...

Certificação das chaves

Problema descrito mostra que nunca devemos utilizar cifras assimétricas sem uma *confiança* plena na associação entre pares de chaves e identidades dos agentes...

Obs.: Notar que o problema já está presente no esquema hipotético (com cifras simétricas) utilizado para motivar o conceito...

- Evidentemente que tal garantia pode ser conseguida por uma *pré-distribuição* de chaves (mas então não estamos longe do problema inicial...)
- Solução alternativa consiste em utilizar os próprios mecanismos disponibilizados pela técnica (assinatura digital) para estabelecer a confiança entre as associações par-de-chaves/identidades.
  - Todos os agentes dispõe da chave pública de um agente *fidedigno* - **Autoridade de certificação (CA)**. Essa chave pública deve ser obtida por via de um canal seguro...
  - A *CA* **garante** (assinando digitalmente) a associação entre *chave-pública/agente* - o que designamos por **certificado de chave pública**. É **responsabilidade** da *CA* a correcção da associação estabelecida.
  - Um qualquer agente pode verificar a assinatura de um certificado (atestando assim a validade da associação pretendida)

Aritmética modular

- Operações modulares (mod *n*)

  Adição - $x+y \bmod n$

  Subtração - $x-y \bmod n = x + (-y) \bmod n$

  Todos os valores $0 \le x < n$ dispõe de inversa aditiva, logo Z_n é um grupo (abeliano)

  Multiplicação - $x*y \bmod n$

  Divisão - $x/y \bmod n = x*(x-1) \bmod n$

  Para que $0 \le x < n$ disponha de inversa multiplicativa é necessário que $gcd(x,n)=1$ - isto é, x seja **primo relativo a** n. Podemo-nos então referir ao sub-grupo Z^*_n de Z_n .(os elementos de Z^*_n são os elementos de Z_n primos relativos a n)

- O corpo finito *GF(p)* [*p* primo]

  Quando *p* é primo, todos os valores de *Zp* com excepção do zero dispõe de inversa multiplicativa. Dessa forma ficamos perante a estrutura algébrica de um corpo finito, designada por **corpo de Galois GF(p)**.

Aritmética modular (cont.)

Property	Property
Associativity	$a + (b + c) \bmod n = (a + b) + c \bmod n$
	$a \times (b \times c) \bmod n = (a \times b) \times c \bmod n$
Commutativity	$a + b \bmod n = b + a \bmod n$
	$a \times b \bmod n = b \times a \bmod n$
Distributivity	$a \times (b + c) \bmod n = (a \times b) + (a \times c) \bmod n$
Existence of identities	$a + 0 \bmod n = 0 + a \bmod n = a \bmod n$
	$a \times 1 \bmod n = 1 \times a \bmod n = a \bmod n$
Existence of inverses	$a + (-a) \bmod n = 0$
	$a \times (a^{-1}) \bmod n = 1$ if $GCD(a, n) = 1$
Reducibility	$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
	$(a \times b) \bmod n = ((a \bmod n) \times (b \bmod n)) \bmod n$

Alguns resultados de teoria dos números

- Função *totient* $\varphi(n)$ de *Euler:* em Z_n, o conjunto de valores $0 \leq x < n$ são designados por **resíduos**. Aos resíduos que não dispõe de factores em comum com n dizemos tratarem-se de **resíduos reduzidos**. A função *totient de Euler* $\varphi(n)$ é definida como o número de resíduos reduzidos de n.
  - Se p *é* primo, então $\varphi(p)=p-1$
  - Se $n=p*q$ com p,q primos, então $\varphi(n)=(p-1)(q-1)$
- Teorema (pequeno) de *Fermat*: (p primo, $0<a<n$)

$$a^{p-1} \bmod p \equiv 1$$

...ou na versão generalizada de *Euler,* ($gcd(a,n)=1$)

$$a^{\varphi(n)} \bmod n \equiv 1$$

Alguns resultados da teoria dos números (cont.)

- Teorema Chinês dos Restos:

  Podemos simplificar as operações modulares em n *se conhecermos o resultado dessas operações nos factores primos de* n

$$p, q \text{ primos}; \quad p < q \quad ; \quad q * u \equiv 1 \bmod p$$

$$a \equiv x \bmod p$$

$$b \equiv x \bmod q$$

$$\text{Se } a \geq b \bmod p: \quad x = (((a - (b \bmod p)) * u) \bmod p) * q + b$$

$$\text{Se } a < b \bmod p: \quad x = (((a + p - (b \bmod p)) * u) \bmod p) * q + b$$

Corpos finitos $GF(p^n)$

- A estrutura de corpo de $GF(p)$ pode ser generalizada para polinómios com resíduos (módulo p) como coeficientes.
- Exemplo: $GF(2^n)$ corresponde ao corpo de polinómios de grau n com coeficientes binários. Obs.: estes polinómios podem ser representados de forma compacta como uma palavra em binário (os coeficientes)

  Adição - adição de polinómios (Xor das representações)

  Multiplicação - multiplicação de polinómios módulo um polinómio primitivo de grau n (pode ser realizado por via de Xor e deslocamentos das representações...)

  Obs.: Um polinómio primitivo é um polinómio que não pode ser expresso como o produto de outros dois polinómios (o equivalente aos números primos...)

Algoritmos para o cálculo de algumas operações modulares...

- **Adição; multiplicação; resíduo (módulo) -** adaptações dos algoritmos usuais...
- **Exponenciação -** (square and multiply)...
- **GCD -** o famoso algoritmo de Euclides...
- **Inversa multiplicativa -** generalização do algoritmo de Euclides...
- **Primalidade** (testar se um número é primo) - existem testes probabilisticos que nos permitem obter garantias (tão boas quanto necessárias) que um número é primo...

Alguns problemas (tidos por) intratáveis...

- Factorização de um inteiro:

  Dado um inteiro n determinar a sua factorização em números primos. Ou seja, determinar números primos $p_1,...,p_i$ tal que

  $$p_1 \times ... \times p_i = n$$

- Logaritmo discreto:

  Dado a, b e n, determinar x tal que

  $$a^x \bmod n = b$$

- Raiz quadrada discreta:

  Dado y e n, determinar x tal que

  $$x^2 \bmod n = y$$

RSA

- Algoritmo que realiza o conceito de criptografia de chave pública introduzido por *Diffie & Hellman.*
- Desenvolvida por *Ron Rivest, Adi Shamir & Leonard Adleman* - 1977/8.
- Baseada no problema da *factorização* de inteiros.

RSA - descrição

- Inicialização (produção do par de chaves)
  - Geram-se dois números primos grandes *p,q* (faz-se $n=p*q$, logo $\varphi(n)=(p-1)*(q-1)$)
  - Considera-se um valor *e* tal que seja primo relativo a $\varphi(n)$ (i.e. $gcd(e,\varphi(n))=1$).
  - Calcula-se *d* como a inversa de *e* no grupo multiplicativo $Z^*_{\varphi(n)}$, i.e. $e*d=1 \bmod \varphi(n)$.

  Chave para cifrar: *(n,e)* **Chave para decifrar:** *(n,d)*
- Utilização (operações de cifra e decifragem)
  - Ambas as operações são a exponenciação modular.

Cifra do texto limpo *x* ($0 \le x < n$) com chave *(n,e)*:

$$x^e \bmod n$$

Decifragem do criptograma *y* ($0 \le y < n$) com chave (n,d)

$$y^d \bmod n$$

RSA - correcção

- E*(Kd,*E*(Kc,M))=M*, i.e. $\left(x^e\right)^d \equiv x \pmod n$

porque, $\left(x^e\right)^d \bmod n \equiv x^{e*d} \bmod n \equiv x^{k*\varphi(n)+1} \bmod n$

Se $\gcd(x,p)=1$, o teorema de Fermat diz-nos que

$$x^{p-1} \bmod p \equiv 1 \Rightarrow x^{(p-1)*(q-1)*k} \bmod p \equiv 1$$

Assim,

$$x^{k*\varphi(n)+1} \bmod p \equiv x$$

para qualquer *x* (verifica-se trivialmente quando $x \mid p$).

Raciocinando da mesma forma para *q* obtemos:

$$x^{k*\varphi(n)+1} \bmod q \equiv x$$

e estas duas equações permitem-nos concluir (Teorema Chinês dos Restos):

$$x^{k*\varphi(n)+1} \bmod n \equiv x$$

RSA - segurança

- Derivar chave privada da chave pública:

  É possível definir um algoritmo (probabilístico) que permite calcular a factorização de *n* assumindo que dispomos de um oráculo para derivar a chave privada RSA da pública. Ou seja, os problemas são demonstrados equivalentes...

- Extrair mensagem do criptograma:

  "Acredita-se" que não é possível derivar a mensagem do criptograma sem se conhecer a chave privada...

  ...Ver questão da segurança dos bits adiante...

RSA - propriedades algébricas

- Comutatividade

$$\left(x^{e_1} \bmod n_1\right)^{e_2} \bmod n_2 \equiv \left(x^{e_2} \bmod n_2\right)^{e_1} \bmod n_1$$

- Propriedade multiplicativa

$$(x_1 \times x_2)^e \bmod n \equiv \left(x_1^{\,e} \bmod n\right) \times \left(x_2^{\,e} \bmod n\right)$$

- etc.

RSA - *protocol failures*

Módulo comum:

Se o mesmo módulo *n* é partilhado por uma comunidade, é possível recuperar a mensagem original se esta for enviada (cifrada) para vários agentes...

Expoente comum:

Se o mesmo expoente *e* for partilhado por uma comunidade, é possível recuperar a mensagem original a partir de *e*(e+1)/2* criptogramas (cifrados com diferentes chaves públicas)

Assinar após cifrar:

Torna possível ao receptor da mensagem "alterar" o seu par de chaves e argumentar que a mensagem assinada foi uma por si escolhida.

Baixa entropia da mensagem:

Qualquer algoritmo determinístico de chave pública disponibiliza ao intruso um oráculo para cifrar. Se a entropia da mensagem for baixa, é possível tabelar todos os possíveis pares mensagem/criptograma.

RSA - *bit security*

- Acredita-se que não é possível recuperar uma mensagem cifrada com o RSA do respectivo criptograma...
- Mas existirá forma de recuperar informação parcial (e.g. um bit) de uma mensagem cifrada com o RSA?

Será possível recuperar o bit menos significativo de uma mensagem cifrada com o RSA sem fazer uso da chave privada?

- É possível demonstrar que, assumindo a intratabilidade da factorização, assim é. De facto, é possível demonstrar a segurança simultânea de log2 L bits (*L* é o número de bits de *n*).
- Desta forma, é possível definir variantes "aleatórias" do RSA "*computacionalmente seguras*".

El-Gamal

- Algoritmo introduzido em 1984 por *T. El Gamal.*
- Baseado no problema do *logaritmo discreto.*
- Variantes para funcionar como cifra ou como assinatura...

El Gamal (cifra) - descrição

- Inicialização
  - escolher um primo p e dois inteiros, g e x, tal que $a<p$ e $x<p$
  - calcular $y = g^x \bmod p$
  - [chave privada, chave pública] = [x, (y, g, p)]
- Cifra de uma mensagem M
  - escolher (aleatoriamente) um inteiro k, $0<k<p-1$
    - tal que k não foi já utilizado e $gcd(k, p-1)=1$
  - calcular $a = g^k \bmod p$ e $b = M*y^k \bmod p$
  - criptograma: (a, b)
- Decifragem...
    - ...dada a chave pública (y,g,p), e o criptograma (a,b)
    - M = $b/a^x \bmod p$

El Gamal (assinatura) - descrição

- Inicialização
  - escolher um primo p e dois inteiros, g e x, tal que $g<p$ e $x<p$
  - calcular $y = g^x \bmod p$
  - [chave privada, chave pública] = [x, (y, g, p)]
- Assinatura de uma mensagem m
  - escolher (aleatóriamente) um inteiro k, $0<k<p-1$
    - tal que k não foi já utilizado e $gcd(k, p-1)=1$
  - calcular $r = g^k \bmod p$ e $s = k^{-1}*(m-x*r) \bmod (p-1)$
    - k^{-1} é a inversa multiplicativa de k mod $(p-1)$
  - assinatura: (r, s)
- Verificação da assinatura
    - ...dada a chave pública (y,g,p), e a assinatura (r,s) da mensagem m
    - calcular $y^r r^s \bmod p$
    - verificar se $y^r r^s \equiv g^m \bmod p$
      - Se SIM, a assinatura é válida!

El Gamal (assinatura) - correcção e segurança

- Correcção:

$$y^r r^s = (g^x)^r (g^k)^s \equiv g^{xr+ks} \bmod p \quad \Leftarrow \text{Definição de y e r}$$

$$\equiv g^{xr+k(k^{-1}(m-xr))} \bmod p \quad \Leftarrow \text{Definição de s}$$

$$\equiv g^{xr+(m-xr)} \bmod p \quad \Leftarrow \text{Definição de } k^{-1} \text{ e}$$

$$\equiv g^m \bmod p$$

$$g^x \equiv g^{x'} \bmod p$$

$$\text{Se } x \equiv x' \bmod (p-1)$$

$$\text{e } p \text{ é primo}$$

- Segurança:

Descobrir a chave privada da pública corresponde exactamente ao problema do logaritmo discreto!

Digital Signature Algorithm (DSA)

- Algoritmo de assinatura incluído no standard *Digital Signature Standard (DSS)* - 1991.
- Desenvolvido pela NSA para a NIST (baseado no *El-Gamal*).
- Desenhado para dispor de um procedimento de assinatura muito eficiente - e.g. muito mais eficiente do que o RSA... (em contrapartida, a verificação é muito mais pesada).
- É, por isso, particularmente adaptada para ser executada em ambientes com recursos limitados (e.g. *smartcards*).
- Desenhado para funcionar unicamente como assinatura (mas é possível desenvolver esquemas que permitem utilizar as rotinas de assinatura/verificação DSA para cifrar mensagens...)

DSA - descrição

- Inicialização
  - p é um primo de L bit ($512 \leq L \leq 1024$; L múltiplo de 64)
  - q é um factor primo de $(p-1)$ de 160 bit
  - $g=h^{(p-1)/q} \bmod p$, onde $h<p-1$; $g>1$
  - $y = g^x \bmod p$, em que $x<q$
  - [chave privada, chave pública] = [x, (y, g, p,q)]
- Assinatura de uma mensagem m (utiliza função de hash H)
  - escolher (aleatoriamente) um inteiro k, $0<k<q$
    - tal que k não foi já utilizado e $gcd(k, p-1)=1$
  - calcular $r = (g^k \bmod p) \bmod q$ e $s = k^{-1}*(H(m)+x*r) \bmod q$
  - assinatura: (r, s)
- Verificação da assinatura
    - ...dada a chave pública (y,g,p,q), e a assinatura (r,s) da mensagem m
    - calcular $w=s^{-1} \bmod q$; $u1=(H(m)*w) \bmod q$; $u2=(r*w) \bmod q$
    - verificar se $(g^{u1}y^{u2} \bmod p) \bmod q = r$

Acordo de chaves: esquema *Diffie & Helmman*

- Objectivo:

Estabelecer um segredo partilhado entre duas partes sem que este seja possível derivar das mensagens trocadas.

- Descrição:
  - p um primo e $g<p$
  - A gera um inteiro $1<x<p$ e envia a B "$g^x \bmod p$"
  - B gera um inteiro $1<y<p$ e envia a A "$g^y \bmod p$"
  - Segredo partilhado: $g^{xy} \bmod p = (g^y)^x \bmod p = (g^x)^y \bmod p$
- Problema:

  Vulnerável ao ataque *man-in-the-middle!!!*

Criptografia em Curvas Elípticas (ECC)

- O problema do "logaritmo discreto" pode ser expresso em qualquer corpo finito (e.g. $GF(p)$ ou $GF(p^n)$)
- ...em particular, podemos exprimir a exponenciação no grupo cíclico determinado por uma *curva elíptica* sobre o corpo considerado.
- Permite representações compactas para níveis de segurança pretendidos (e.g. 163 bit para níveis de segurança análogos as 1024 bit em RSA)
- ... e realizações eficientes das operações pretendidas...

- Argumenta-se, por isso, ser particularmente adequado para dispositivos com recursos limitados (e.g. *smartcards*)

Outras técnicas...

- Partilha de segredos

  Dispor de esquemas que permitam a partilha de um segredo por uma comunidade de *N* agentes onde sejam necessários (pelo menos) *p* agentes para o recuperar.

- Provas de conhecimento zero

  Exibir evidencia sobre o "conhecimento" de um dado objecto sem revelar qualquer informação sobre ele.

- Dinheiro electrónico; votação electrónica; ...

  Construir uma abstracção electrónica sobre "o dinheiro" ou "votação electrónica": Deve resolver questões como o *anonimato*; a *não duplicação*; etc.

- ...

Referências

- Cryptography: theory and practice – D. Stinson. [cap. 4,5 (6,8,9,11,13)]
- Applied Cryptography – *Bruce Schneier*. [cap. 19,20 (21,22,23)]