Segurança da Informação

Diffie-Hellman, RSA, DSA

Acordo de Chaves Diffie-Hellman

Problema do Logaritmo Discreto

- Dados os inteiros a, p e x, é "fácil" calcular $y = a^x \bmod p$.
- Dados a, p e y, é "difícil" calcular x.
- Com valores de k bits, são necessários cerca de k^3 passos para calcular y.
- Para calcular x, são necessários

$$2^{c\sqrt[3]{k \ln^2 k}}$$

passos (esforço *quase* exponencial).

Diffie-Hellman

- Primeiro algoritmo assimétrico conhecido publicamente (1976).
- Segurança relacionada com a dificuldade de calcular o logaritmo discreto.
- Objetivo: acordo de chave através de um canal potencialmente inseguro.
- Extensão: cifração (ElGamal).

Chaves Diffie-Hellman

- Parâmetros:
  - q: primo.
  - p: primo tal que p-1 seja múltiplo de q.
  - g: inteiro entre 2 e p–2 tal que $g^q \bmod p = 1$.
- Chave privada:
  - x: inteiro aleatório entre 1 e q–1.
- Chave pública:
  - $y = g^x \bmod p$.

Exemplo

- $q = 11$, $p = 23$, $g = 2$.
- Notar que $2^{11} = 1 \pmod{23}$.
- Chave privada: $x = 9$.
- Chave pública: $y = 2^9 \bmod 23 = 6$.

Protocolo Diffie-Hellman

- Alice (*A*) e Beto (*B*) desejam estabelecer comunicação segura.
- Possibilidade: usar um algoritmo simétrico para cifrar as mensagens trocadas entre *A* e *B*.
- Problema: *A* e *B* precisam compartilhar uma chave simétrica.

Protocolo Diffie-Hellman

- Pares de chaves:

  x_A, $y_A = g^{x_A} \bmod p$

  x_B, $y_B = g^{x_B} \bmod p$

- *A* obtém a chave pública de *B* e vice-versa a partir de uma base de chaves.

- *A* calcula $y_B^{x_A} \bmod p$
  $= (g^{x_B})^{x_A} \bmod p$
  $= g^{x_A x_B} \bmod p$

- *B* calcula $y_A^{x_B} \bmod p$
  $= (g^{x_A})^{x_B} \bmod p$
  $= g^{x_A x_B} \bmod p$

Exemplo

- $q = 11$, $p = 23$, $g = 2$.
- $x_A = 5$, $y_A = 2^5 \bmod 23 = 9$.
- $x_B = 8$, $y_B = 2^8 \bmod 23 = 3$.

- *A* calcula $y_B^{x_A} \bmod p$
  $= 3^5 \bmod 23$
  $= 13$
- *B* calcula $y_A^{x_B} \bmod p$
  $= 9^8 \bmod 23$
  $= 13$

- *A* e *B* usam a chave compartilhada $K = 13$ com um algoritmo simétrico.

Limitações

- A chave compartilhada é sempre a mesma (protocolos mais avançados possibilitam chaves voláteis de sessão).

- As operações envolvidas não sugerem um mecanismo óbvio para definir assinaturas digitais (uso exclusivo para negociação de chave de sessão).

Criptossistema RSA

Fatoração Inteira

- Dados dois primos p e q e um inteiro n, é fácil verificar se $n = pq$: trabalhando com valores de k bits, são necessários cerca de k^2 passos para calcular n.
- Dado apenas n, é difícil calcular p e q.
- Para calcular p e q, são necessários

$$2^{c\sqrt[3]{k \ln^2 k}}$$

passos (esforço *quase* exponencial).

Algoritmo RSA

- Rivest, Shamir, Adleman (1977).
- Primeiro sistema assimétrico completo (cifração e assinatura).
- Algoritmo assimétrico mais amplamente utilizado no mundo.
- Patenteado até 2000 (EUA, Canadá).

Chaves RSA

- Módulo:

  $n = pq$ (p e q: primos de tamanho similar),

  $\varphi(n) = (p-1)(q-1)$.

- Expoente público:

  e, inversível mod $\varphi(n)$,

  isto é, primo relativo a $\varphi(n)$.

- Expoente privado:

  d, inverso de e mod $\varphi(n)$, isto é,

  $e \cdot d \equiv 1 \pmod{\varphi(n)}$.

Exemplo

- $p = 7$, $q = 17$, $n = 119$.
- $\varphi(n) = (7-1)(17-1) = 96$.
- Expoente público:
  - $e = 5$.
- Expoente privado:
  - $d = 5^{-1} \pmod{96} = 77$.

- Em caso de dúvida:

  $5^{-1} \equiv 5^{\varphi(96)-1} \equiv 5^{31} \equiv 77 \pmod{96}$.

  ou diretamente: $5 \cdot 77 = 4 \cdot 96 + 1$.

Operação do RSA

- M: inteiro no intervalo entre 0 e n–1.
- Encriptação: $C = M^e \bmod n$.
- Decriptação: $M = C^d \bmod n$.
- Por que funciona?
  - $C^d \bmod n = (M^e \bmod n)^d \bmod n = M^{e \cdot d} \bmod n = M^{e \cdot d \bmod \varphi(n)} \bmod n = M^1 \bmod n = M$.

Exemplo

- $p = 7$, $q = 17$, $n = 119$, $e = 5$, $d = 77$.
- M = 19 (inteiro no intervalo 0 até 118).
- $C = M^e \bmod n = 19^5 \bmod 119 = 66$.
- $M = C^d \bmod n = 66^{77} \bmod 119 = 19$.
- Observação:

  $66^{77} \bmod 119 =$

  $(((66^7 \bmod 119)^{10} \bmod 119) \cdot (66^7 \bmod 119)) \bmod 119 =$

  $((59^{10} \bmod 119) \cdot 59) \bmod 119 = (81 \cdot 59) \bmod 119 = 19,$

  onde $66^7 \bmod 119 = 59$ e $59^{10} \bmod 119 = 81$.

Detalhes dos Parâmetros

- Normalmente o expoente *e* é pequeno em comparação com o expoente *d*.
- Restrição: $e > \log_2 n$.
- Valor mais popular: *e* = 65537 (17 bits, 2 iguais a 1). Vantagem: exponenciação rápida.
- Dados do TCR são armazenados com o expoente *d* para acelerar as operações aritméticas com a chave privada.

Assinaturas RSA

- Trocar o papel das chaves.
  - Remetente encripta com sua chave privada.
  - Destinatário decripta com a chave pública do remetente.

- Envelope criptográfico:
  - Remetente assina uma mensagem e cifra com a chave pública do destinatário, que inverte essas operações com as chaves complementares.

Cuidados Especiais

- Existem dezenas de vulnerabilidades potenciais na utilização do RSA.
- Não são vulnerabilidades intrínsecas do algoritmo!
- Implementações de produção devem tomar os cuidados necessários para evitar todas essas vulnerabilidades.

RSA e Fatoração Inteira

- O conhecimento dos primos p e q permite calcular o expoente d a partir de e.
- O conhecimento do expoente d permite fatorar o módulo $n = pq$.
- A equivalência entre a fatoração inteira e o cálculo de $C^d \bmod n$ sem envolver d, p, q era só uma conjectura até 2008.

Chaves RSA

- Tamanhos comuns de chaves RSA (e.g. ICP-Brasil, SPB):
  - 1024 bits [uso geral]
  - 2048 bits [autoridade certificadora]
  - Tamanho dobrado a partir de 2009.
- Tamanhos recomendados pelo governo da Alemanha (2004):
  - 2048 bits [uso geral]
  - 3072 bits [autoridade certificadora]

Assinaturas DSA

DSA

- Assinatura digital baseada no problema do logaritmo discreto.
- Peculiaridade: algoritmos de cifração e de assinatura baseados em logaritmo discreto são, via de regra, heterogêneos (fórmulas não relacionadas).
- RSA: cifração e assinatura.
- DSA: somente assinatura.

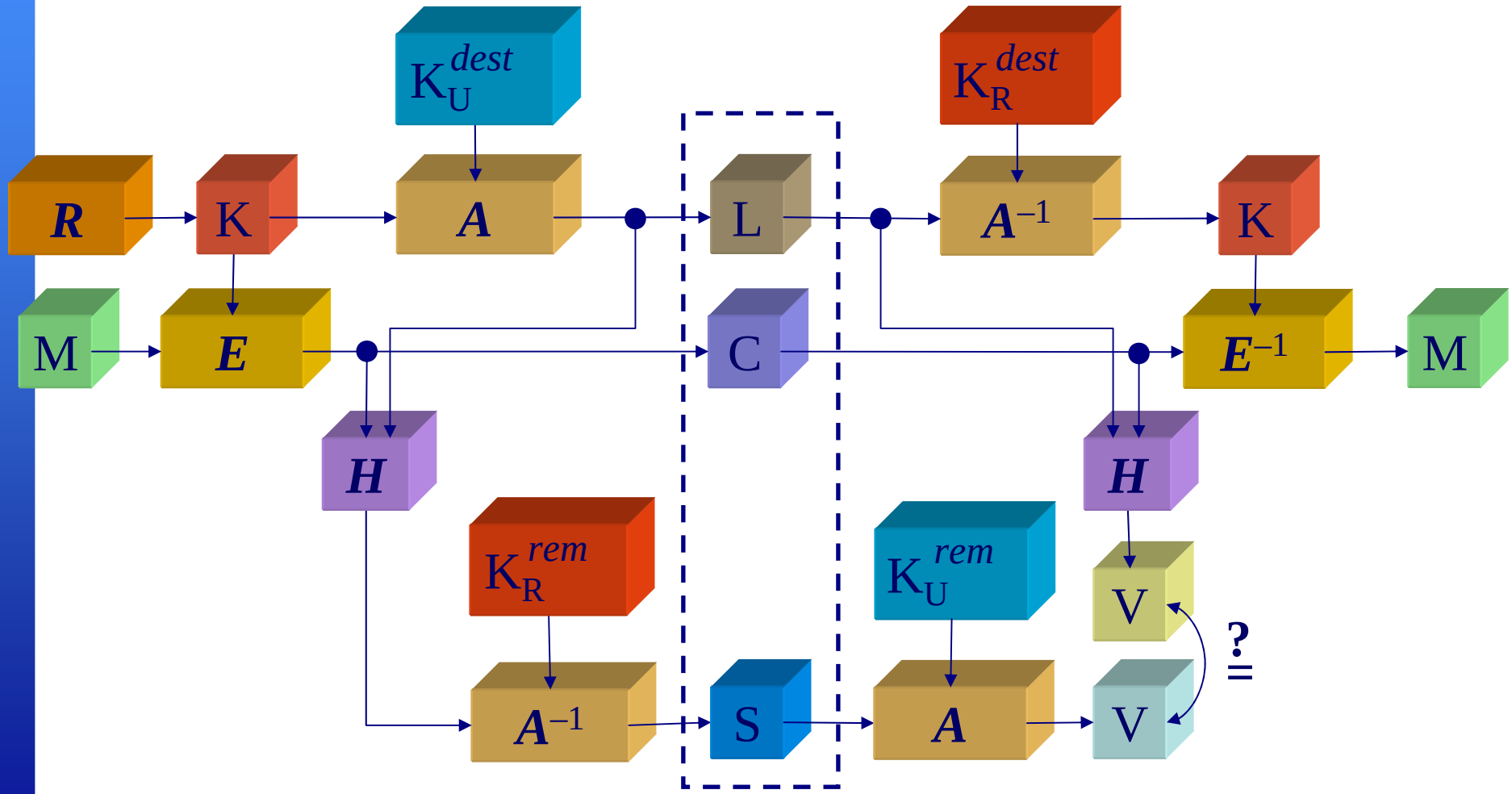
Algoritmo DSA

- Parâmetros:
  - q: primo.
  - p: primo tal que $p-1$ seja múltiplo de q.
  - g: inteiro entre 2 e $p-2$ tal que $g^q \bmod p = 1$.
- Chave privada:
  - x: inteiro aleatório entre 1 e $q-1$.
- Chave pública:
  - $y = g^x \bmod p$.

Algoritmo DSA

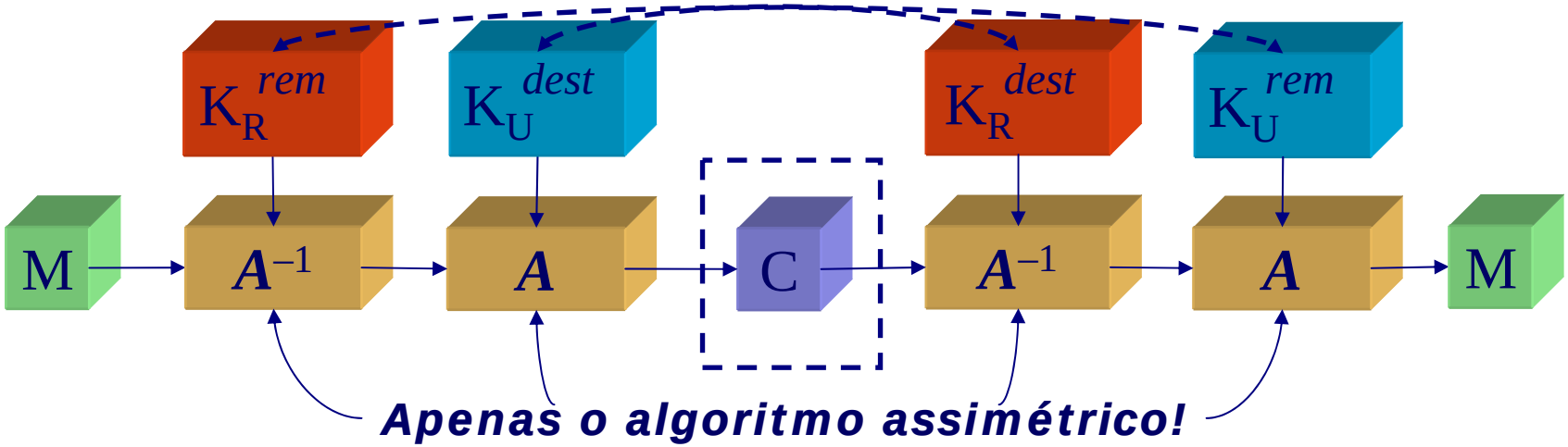
- Assinatura: (r, s)
  - k: inteiro aleatório entre 1 e $q-1$.
  - $h \leftarrow \text{hash}(M)$.
  - $r \leftarrow (g^k \bmod p) \bmod q$ (se $r = 0$, mudar k).
  - $s \leftarrow (h + xr) \cdot k^{-1} \bmod q$.
- Verificação:
  - $u \leftarrow h \cdot s^{-1} \bmod q$.
  - $v \leftarrow r \cdot s^{-1} \bmod q$.
  - aceitar $\Leftrightarrow (g^u y^v \bmod p) \bmod q = r$.

Envelope criptográfico



Envelope puro

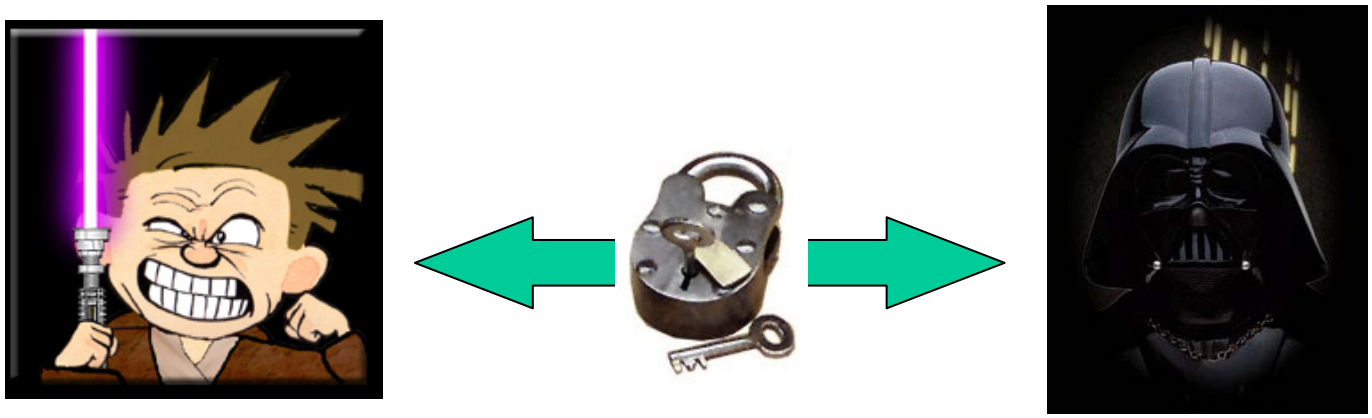
- O uso de algoritmos simétricos e funções de hash motiva-se pela maior eficiência desse algoritmos. Contudo...



Epílogo

Estabelecendo uma ICP

- Serviços básicos da segurança: algoritmos simétricos, assimétricos e auxiliares.



- *Como garantir que as informações públicas (e.g. identidade das partes envolvidas) são realmente confiáveis?*