Segurança Computacional

Resumos, assinaturas e certificados

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Julho de 2019

Conteúdo

Resumo criptográfico

Resumo criptográfico (*hash*)

Função matemática:

- **Entrada**: sequência de bytes com tamanho variável
- **Saída**: sequência de bytes com tamanho fixo
- É uma "impressão digital" dos dados de entrada

Resumo criptográfico (*hash*)

Matematicamente: função unidirecional

- $y = hash(x)$ é simples de calcular
- $x = hash^{-1}(y)$ é impossível ou inviável

Usos:

- identificação única de dados
- atestação de integridade de dados

Exemplos:

- MD5 - *Message Digest 5* (obsoleto)
- Família SHA-1, SHA-2, SHA-3 - *Secure Hash Algorithm*

Exemplos

```
maziero:~> md5sum *
62ec3f9ff87f4409925a582120a40131 header.tex
0920785a312bd88668930f761de740bf main.pdf
45acbba4b57317f3395c011fbd43d68d main.tex

maziero:~> sha1sum *
742c437692369ace4bf0661a8fe5741f03ecb31a header.tex
9f9f52f48b75fd2f12fa297bdd5e1b13769a3139 main.pdf
d6973a71e5c30d0c05d762e9bc26bb073d377a0b main.tex

maziero:~> sha256sum *
ff87f0cb0c06240ad4adea80bd62dc16f60442ed089bd777a07a7985 header.tex
02250c6539288738be30d2f8c4644469d621b063ae575a8642b93078 main.pdf
96f6c8ffde131f98e6f9aec1909dc6a7b5b412db9c7161077d837681 main.tex
```

Propriedades

Determinismo: para m, $hash(m)$ é sempre o mesmo

Rapidez: calcular $hash(m)$ é rápido para qualquer m

Resistência à pré-imagem: dado x, é difícil encontrar $m \mid hash(m) = x$

Resistência à colisão: é difícil encontrar duas mensagens $m_1 \neq m_2 \mid hash(m_1) = hash(m_2)$

Propriedades

Sensibilidade: uma pequena modificação nos dados de entrada gera grandes mudanças no resumo

Espalhamento: uma mudança localizada nos dados de entrada altera várias partes do resumo

Exemplo: "*Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination* - Albert Einstein"

MD5 = `46a412936254ab00d08d4880601370ce`

Trocando *fast* por *gast* (mudança de **1 bit** na frase):

MD5 = `4820ccf457ef5af11dd57794eaffcebf`

Message Authentication

MAC: *Message Authentication Code*

- Permitem testar integridade e autenticidade de dados
- Código anexado a cada mensagem por Alice
- Código verificado por Bob

Funcionamento:

1. Alice calcula $mac(m)$ para uma mensagem m
2. Alice envia $[m, mac(m)]$ a Bob
3. Bob calcula $mac'(m)$
4. Se $mac(m) = mac'(m)$ a mensagem está integra

Hash-based MAC

MAC simples é fraco:

- Suscetível a ataques MITM - *man-in-the-middle*
- Mallory pode interceptar $[m, mac(m)]$
- Ela modifica a mensagem, recalcula $mac(m)$ e envia a Bob

HMAC (Hash-based Message Authentication Code):

- MAC usando hash criptográfico
- usa uma chave secreta para evitar ataques MITM
- Usado nos protocolos IPsec e SSL/TLS

Hash-based MAC

Definição (RFC2104):

$$HMAC(m, k) = H\{(k \oplus opad) \parallel H[(k \oplus ipad) \parallel m]\}$$

- m: mensagem a autenticar
- k: chave secreta (conhecida somente por Alice e Bob)
- H: hash criptográfico (MD5, SHA1, ...)
- *ipad*: *inner padding* – "enchimento" interno (0x5c5c5c...)
- *opad*: *outer padding* – "enchimento" externo (0x363636...)
- $\oplus$: OU exclusivo
- $\parallel$: concatenação

Assinatura digital

Assinatura digital

Objetivo:

- Atestar a autoria e integridade de um documento.

Construida a partir de:

- Criptografia assimétrica
- Resumo criptográfico

Definição:

A assinatura digital de um documento é um **resumo criptográfico** do mesmo, **cifrado com a chave privada** de quem o assina (geralmente o autor).

Assinatura digital

Sendo um documento d emitido pelo usuário u, sua assinatura digital $s(d, u)$ é definida por:

$$s(d, u) = \{\ hash(d)\ \}_{kv(u)}$$

onde:

- $hash(x)$ é uma função de resumo criptográfico
- $\{x\}_k$ é a cifragem de x usando uma chave k
- $kv(u)$ é a chave privada do usuário u

Assinatura digital

Para verificar a validade de uma assinatura $s(d, u)$:

- Calcular novamente o resumo: $r' = hash(d)$
- Decifrar a assinatura: $r'' = \{s\}^{-1}_{kp(u)}$
- Se $r' = r''$ o documento foi assinado por u e está íntegro

Passos da assinatura

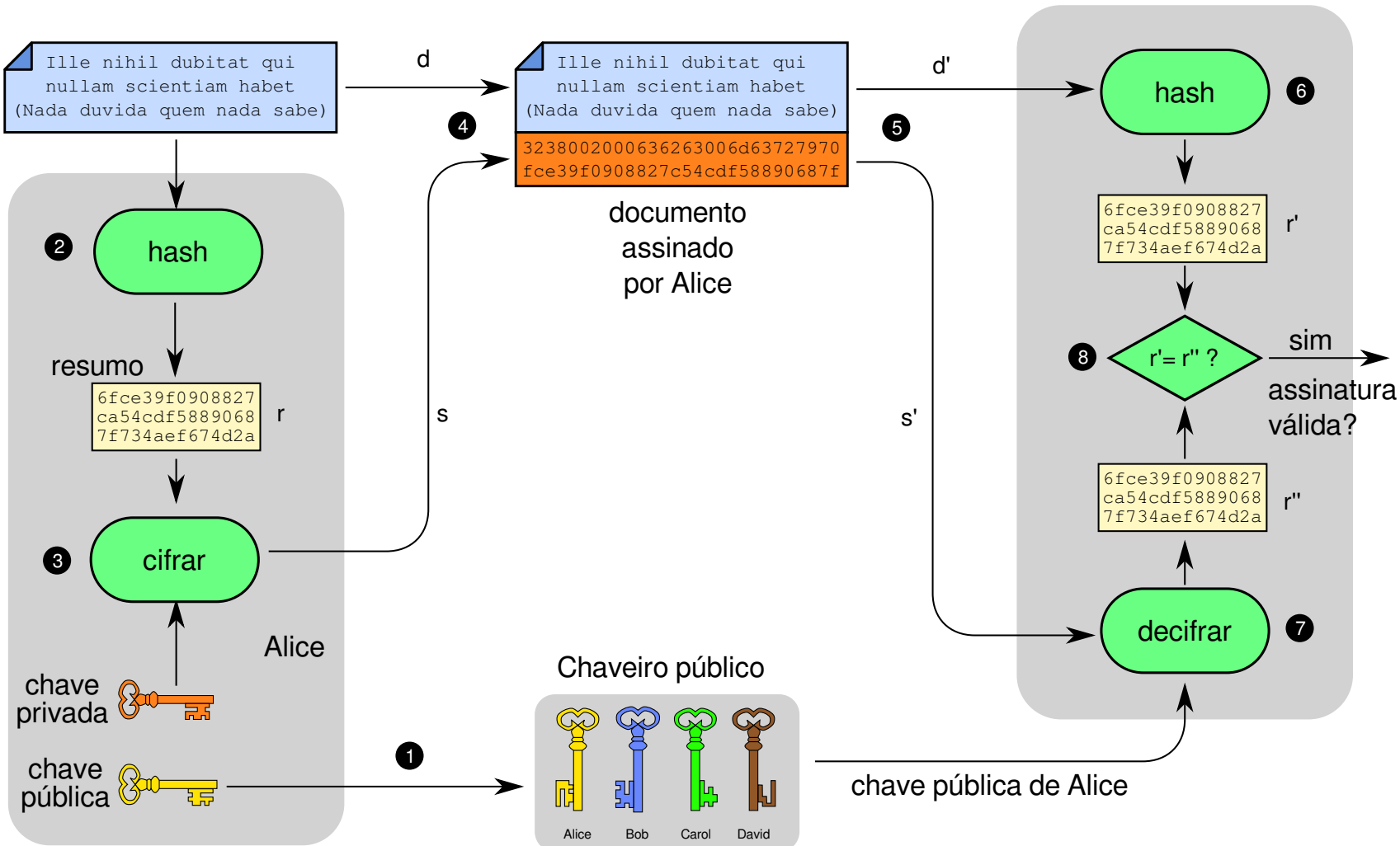
Alice tem um documento d a assinar:

1. Alice divulga sua chave pública kp_a
2. Alice calcula o resumo r do documento d
3. Alice cifra o resumo r usando sua chave privada kv_a
4. O documento original d e a assinatura s, em conjunto, constituem o documento assinado por Alice: $[d, s]$

Passos da verificação

5. Beto obtém o documento assinado por Alice $[d', s']$
   - Se ambos estiverem íntegros, $d' = d$ e $s' = s$
6. Beto recalcula o resumo $r' = hash(d')$ do documento
7. Beto obtém a chave pública kp_a de Alice e decifra a assinatura s' do documento, obtendo um resumo r''
8. Beto compara o resumo r' do documento com o resumo r'' obtido da assinatura digital
   - Se ambos forem iguais ($r' = r''$), o documento foi assinado por Alice e está íntegro

Assinatura digital



Algoritmos de assinatura digital

Teóricos

- Assinatura de Schnorr
- Assinatura de ElGamal

Assinatura RSA

- Usa o algoritmo de cifragem RSA
- *Optimal Asymmetric Encryption Padding* (OAEP)

Assinatura DSA

- Padrão do governo americano (1982)
- Baseado no algoritmo de Schnorr
- ECDSA: variante usando curvas elípticas

Certificados digitais

Como confiar na chave pública?

Chave pública do site `www.bb.com.br` (RSA):

```
Modulus (2048 bits): 85 1b 83 f1 18 ee 40 a1 a3 21 4c 7b e6 fc 8a c5 a0 0c aa 6b 92 14 1e 96
91 b1 18 e4 52 41 34 a9 a0 b3 e5 86 80 f9 ca f9 7e 0e c5 fb b6 8d 8d e8 3a 30 a1 7e 71 5b 68
f1 94 d3 82 0d 63 be 86 62 5d 82 7c ae ee ea 2b d9 5f 91 55 17 69 b0 37 5c ed 58 9c 52 98 a6
d9 17 6a b6 27 e2 19 e5 2c 8e db 03 14 5c 72 e6 94 31 40 b9 9d e5 64 f4 27 a6 6d 0a 45 1c fc
fc 6b 8d 2d 74 e6 0c 7e 6c 8b 8d ca 72 17 34 6a bb 81 9e 87 c5 e6 15 1b de 7f 35 5f 8b 76 66
6f ba 55 fd 96 85 aa f9 6f a9 b2 7e 29 e8 d2 cb d8 0c a8 f3 6e ba 5b df 4a a7 a3 6f ae 84 d2
58 9c 7b a9 42 15 46 6d 26 15 50 cf 8f c7 a0 70 ab 99 e3 7e b0 f0 be 7f 07 8c 37 f3 f5 43 84
87 42 0a b0 ee 79 cd 42 6c 67 94 b8 0c 2d 9f d2 4f 25 da c5 ef 6a 51 dd 42 28 03 50 6b 42 46
cd 36 a3 23 a2 01 ac ef 9d bb 73 9e 3b 71 39

Exponent (24 bits): 65537
```

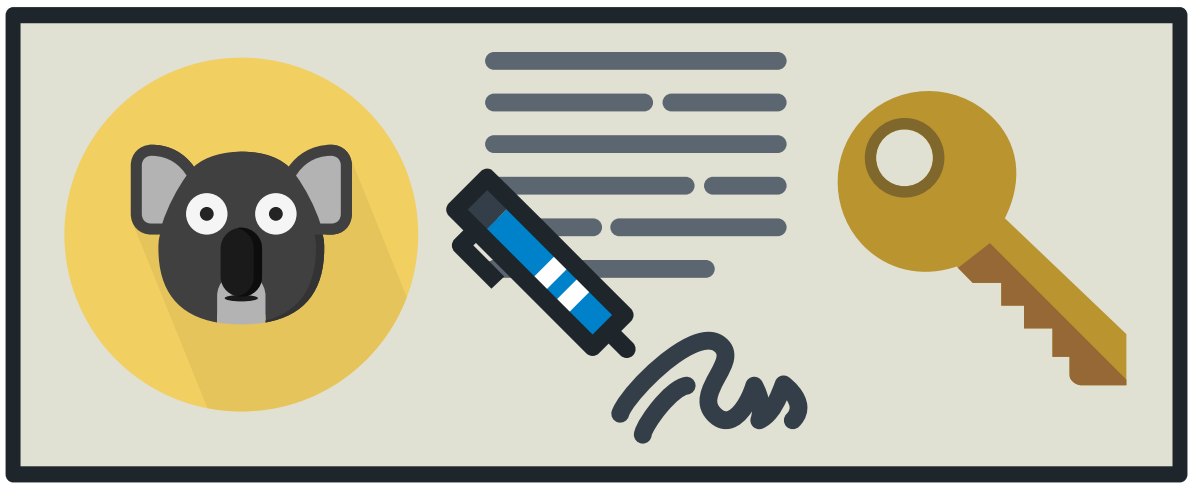
Ela serve para **autenticar** as páginas do site do Banco do Brasil

Problema: atacante pode trocar chaves no repositório público!

Problema: Como saber se essa é a chave certa?

Certificado de chave pública

- Documento que associa uma chave a uma identidade
- Emitido por uma **autoridade confiável**
- Assinatura permite atestar sua confiabilidade



Certificado de chave pública

Um certificado contém:

- Identidade do proprietário (nome, e-mail, URL, etc)
- Chave pública do proprietário
- Outras informações (algoritmos, período de validade, ...)
- Assinatura digital por uma entidade confiável

Existem vários padrões de certificados: X.509, PGP, etc

No X.509: **Autoridades Certificadoras** emitem certificados

Estrutura de um certificado X.509 v3

```
Certificate
    Version Number
    Serial Number (issuer)
    Signature Algorithm ID
    Issuer Name
    Validity period
        Not Before
        Not After
    Subject name
    Subject Public Key Info
        Public Key Algorithm
        Subject Public Key
    Issuer Unique Id (opt)
    Subject Unique Id (opt)
    Extensions (optional)
        ...
Signature Algorithm
Signature
```

Exemplo de certificado X.509 v3

Informações básicas (versão, emissor e validade):

```
Certificate Data:

    Version: 3 (0x2)
    Serial Number: 05:f1:3c:83:7e:0e:bb:86:ed:f8:c4:9b

    Issuer: C=BE, O=GlobalSign nv-sa, CN=GlobalSign Extended
            Validation CA-SHA256-G3

    Validity
        Not Before: Feb  7 12:41:03 2017 GMT
        Not After : May  9 23:59:59 2018 GMT
```

Exemplo de certificado X.509 v3

Proprietário e sua chave pública:

```
1   Subject: businessCategory=Private Organization/serialNumber=7297/
2            jurisdictionC=BR, C=BR, ST=Distrito Federal, L=Brasilia/
3            street=ST STN SN QD 716 CONJ C EDIF SEDE ASA NORTE,
4            OU=DITEC, O=Banco do Brasil SA, CN=www2.bancobrasil.com.br
5
6   Subject Public Key Info:
7      Public Key Algorithm: rsaEncryption
8         Public-Key: (2048 bit)
9         Modulus:
10           00:db:4a:0e:92:da:5b:f3:38:3f:d5:63:9d:6d:f9:
11           91:6c:16:fc:24:84:28:e8:aa:86:aa:9c:a3:aa:1a:
12           2e:b6:09:74:6a:f8:1e:31:4a:60:81:0f:ac:76:59:
13           ... (linhas omitidas)
14           8e:0b
15        Exponent: 65537 (0x10001)
```

Exemplo de certificado X.509 v3

Campos opcionais:

```
X509v3 extensions:
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    Authority Information Access:
        CA Issuers - URI:http://secure.globalsign.com/.../gsext.crt
        OCSP - URI:http://ocsp2.globalsign.com/gsext/endvalsha2g3r3
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
```

Assinatura:

```
Signature Algorithm: sha256WithRSAEncryption
    94:8e:14:c6:38:30:78:77:80:fc:92:f1:5b:8b:72:6a:b6:b6:
    95:66:c5:7b:ba:be:51:a4:b8:8a:f5:37:0a:4a:74:4d:82:27:
    ... (linhas omitidas)
    b6:44:e8:8c
```

Verificação de um certificado

Como verificar um certificado?

1. Identificar o emissor e (quem assinou)
2. Obter a chave pública do emissor kp_e
3. Com kp_e, conferir a assinatura do certificado

Problema: A chave kp_e é realmente do emissor e?

Solução:

1. Obter certificado da chave kp_e
2. Verificar esse certificado

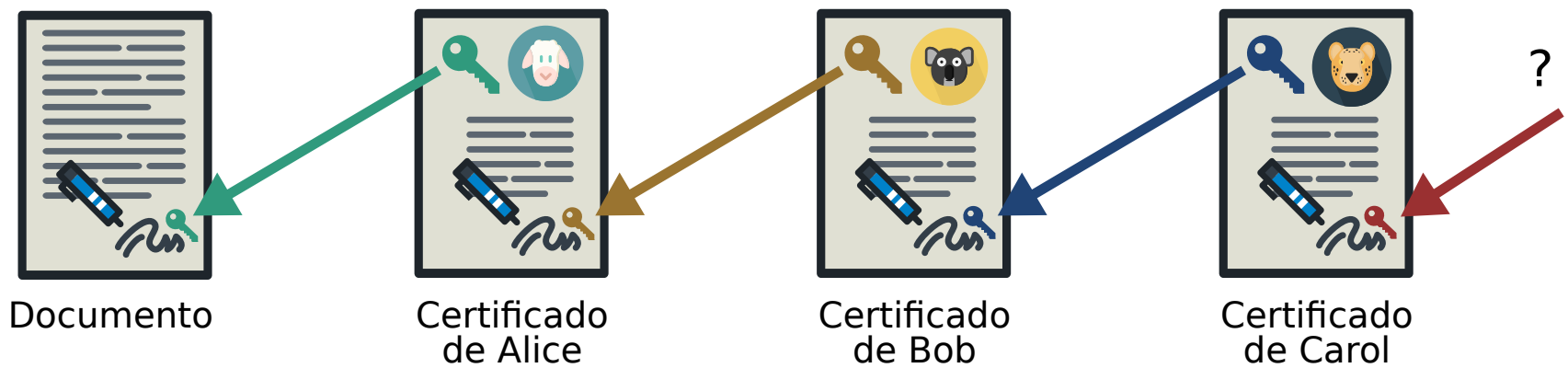
Infraestrutura de chaves públicas

Infraestrutura de chaves públicas

PKI: *Public Key Infrastructure*

- Cada entidade tem um certificado *[entidade, chave]*
- Cada certificado é assinado por outra entidade
- Assinatura define relação de confiança entre as entidades

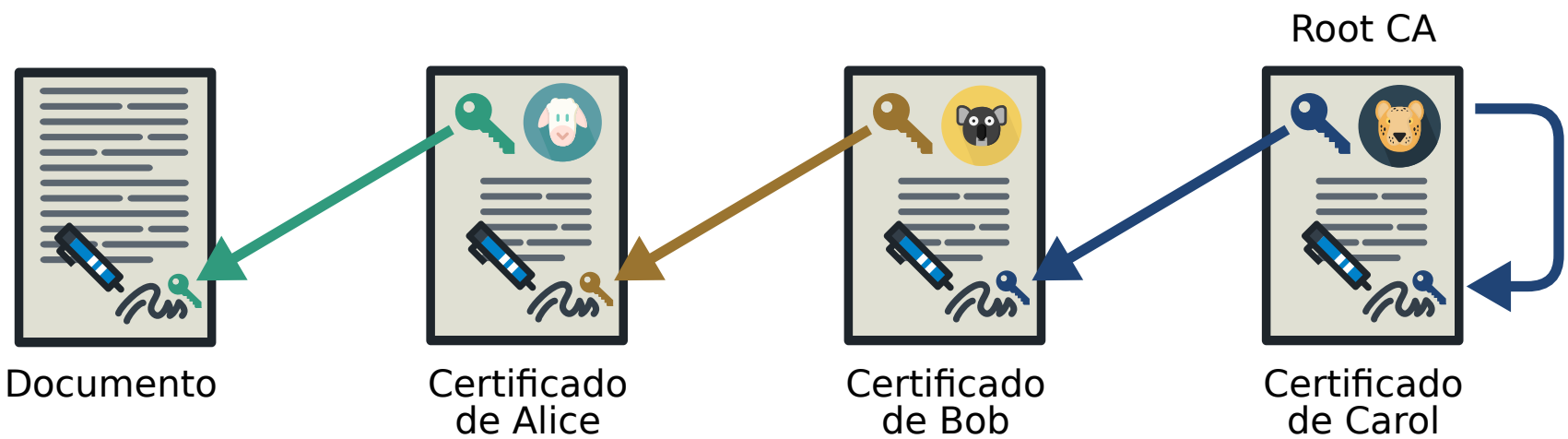
Os certificados definem **cadeias de confiança**:



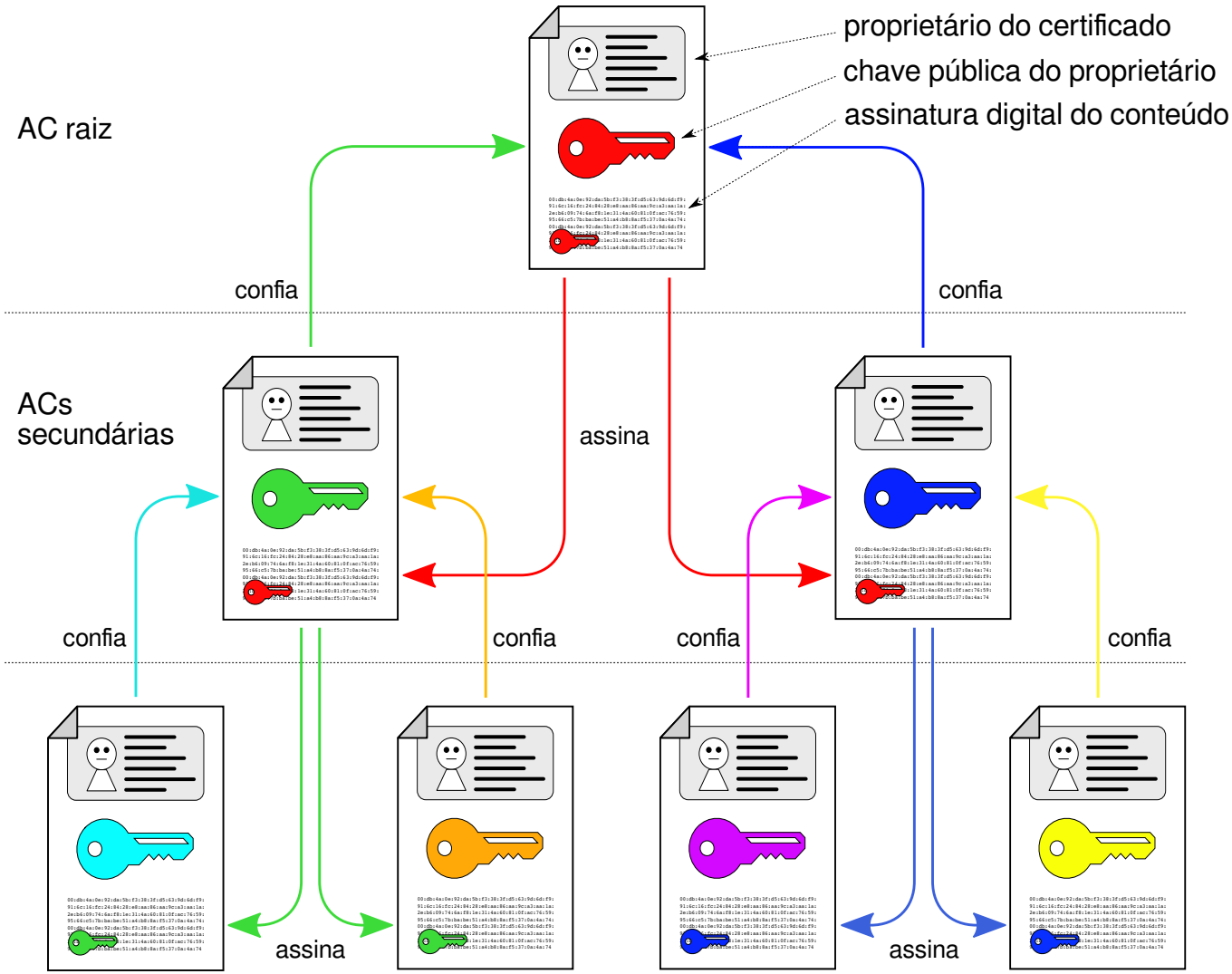
Certificados X.509

Os certificados X.509 definem uma PKI hierárquica:

- Estrutura de certificação transitiva
- No topo: AC raiz (considerada confiável)
- AC Raiz usa um certificado auto-assinado
- Clientes mantêm lista de ACs confiáveis



Infraestrutura de chaves públicas



Exemplo de verificação

`openssl s_client -connect www.itau.com.br:443`

```
1  depth=2   C = US,
2            O = DigiCert Inc,
3            OU = www.digicert.com,
4            CN = DigiCert High Assurance EV Root CA
5            verify return:1
6
7  depth=1   C = US,
8            O = DigiCert Inc,
9            OU = www.digicert.com,
10           CN = DigiCert SHA2 Extended Validation Server CA
11           verify return:1
12
13 depth=0   businessCategory = Private Organization,
14           jurisdictionC = BR, serialNumber = 60.701.190/0001-04,
15           C = BR, ST = Sao Paulo, L = Sao Paulo,
16           O = Itau Unibanco S.A.,
17           CN = www.itau.com.br
18           verify return:1
```

Segurança da autoridade certificadora

Em um navegador existem 100's de CAs pré-cadastradas

Cada CA é a raiz de uma cadeia de confiança

Se uma delas for comprometida, certificados podem ser forjados

Certificado forjado: associa um nome a outra chave

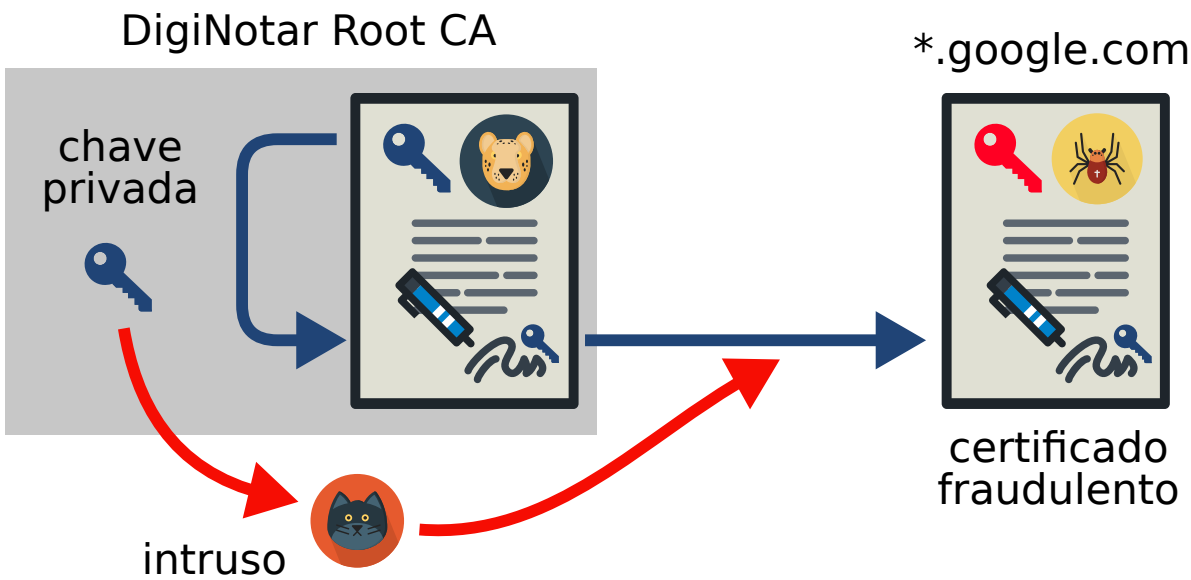
Possibilitam ataques MITM (Man-in-the-Middle)

O caso DigiNotar - 1: violação da CA

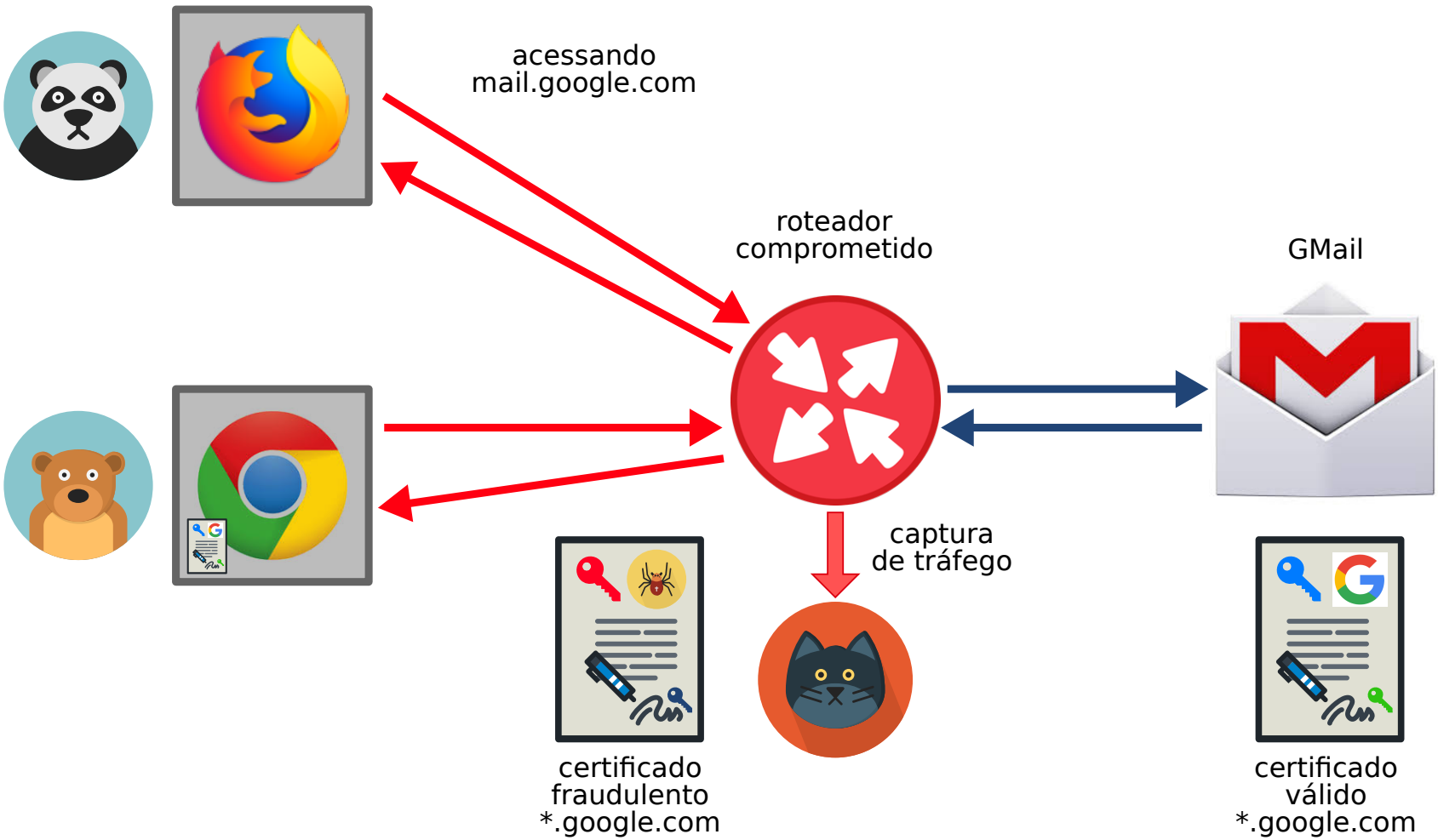
Em 2011, a CA DigiNotar (Holanda) foi invadida

+500 certificados forjados foram emitidos

Google, Yahoo, Mozilla, WordPress, Tor, ...



O caso DigiNotar - 2: ataque MITM



Revogação de certificados

- Certificados têm prazos de validade:

```
1 Not Before: Feb 22 09:20:38 2017 GMT
2 Not After : May 17 08:58:00 2017 GMT
```

- Pode ser necessário revogar um certificado antes do prazo
  - Mudanças no certificado (permisssões)
  - Fim de atividade da empresa
  - Comprometimento da chave privada de alguma CA
- Como saber se um certificado foi revogado?
- R: consultando a CA que o emitiu:
  - CRL - *Certificate Revocation List*
  - OCSP - *Online Certificate Status Protocol*