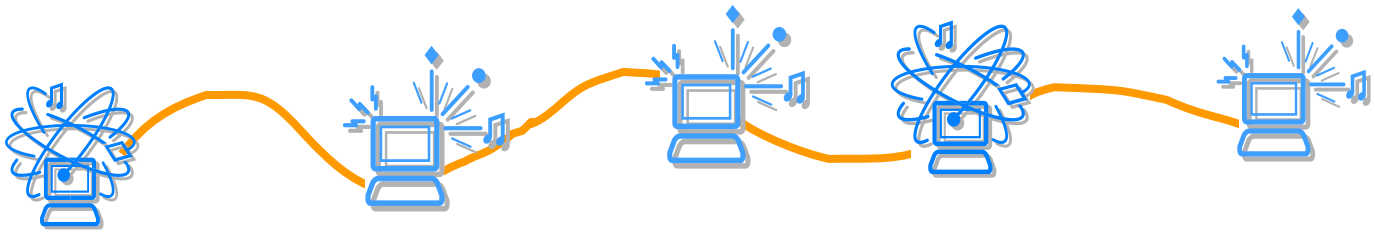
Segurança em Redes

Conceitos de criptografia 4 – Gestão de chaves



Redes de Comunicação de Dados

Departamento de Engenharia da Electrónica e das Telecomunicações e de Computadores

Instituto Superior de Engenharia de Lisboa

“Na Internet, ninguém sabe que tu és um cão!”

Isto é bom!

Não há preconceitos
Não há discriminação
O tamanho não interessa
Não são necessários fatos ou instalações vistosas
Mais eficiente

Mas....

"On the Internet, nobody knows you're a dog."

“Na Internet, ninguém sabe que tu és um cão.”

Isto é mau...

- Fraude
- Qualidade
- Contabilização
- Segurança

Soluções:

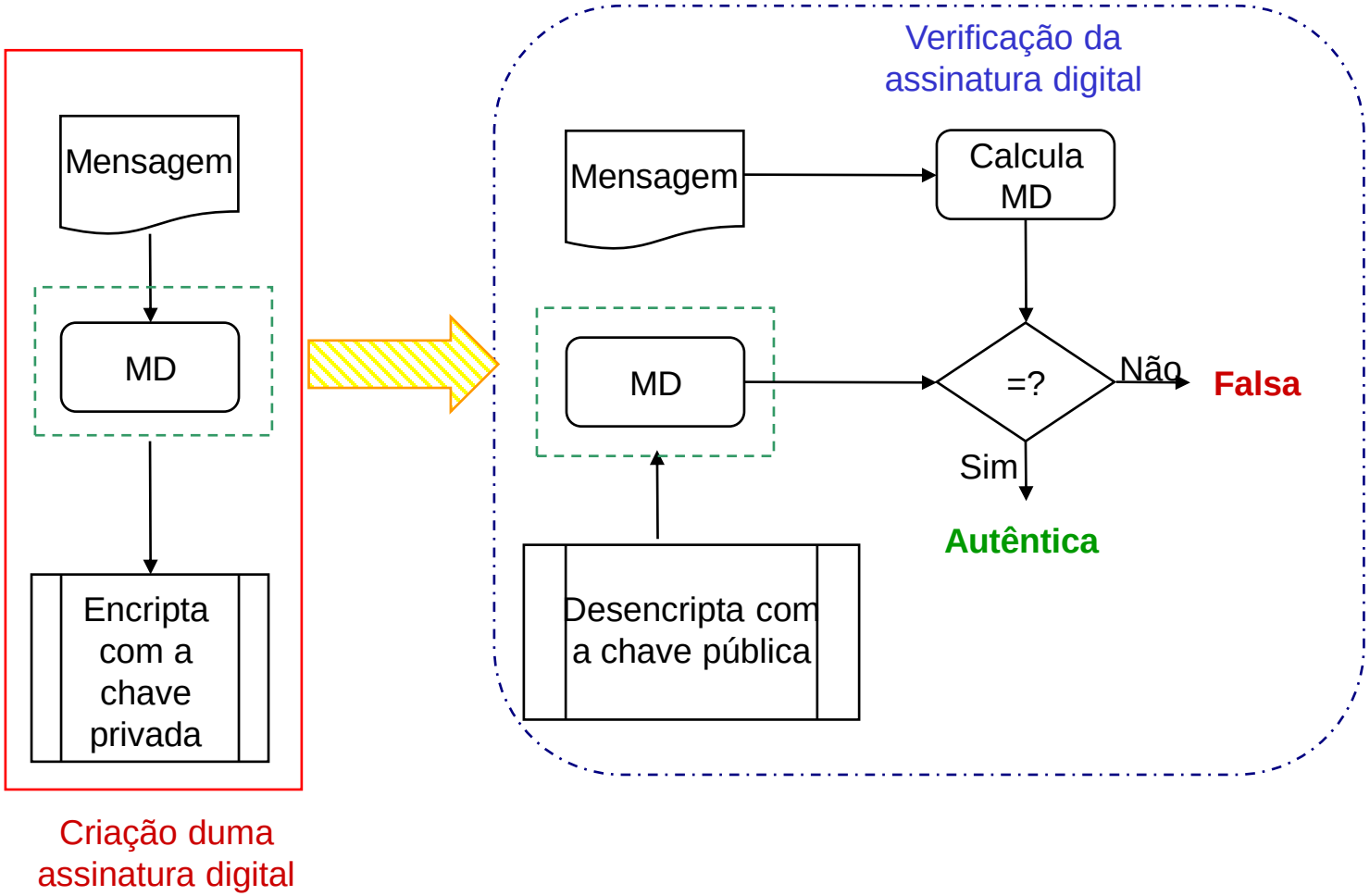
- Reputação
- Identificação

"On the Internet, nobody knows you're a dog."

Metáfora da assinatura de chave pública: Sêlo cilindrico (Pérsia antiga)

- Chave privada: Selo cilíndrico
- Chave pública: Impressão do selo
- Documento: Impresso num placa de barro
- Documento assinado: Placa de barro com impressão do selo
- Difícil criar a impressão sem o selo
- Difícil alterar a placa (sem o selo)
- Difícil copiar a impressão
- As placas de barro (c/ impressões) duram muito tempo

Hoje as assinaturas são um pouco mais complexas



Exemplo de *hash* duma mensagem

Mensagem:

Senator Hand N. Till
Washington, DC

Dear Senator:

This letter is to inform you we have opened account no. 338907021 on your behalf, and deposted therein a legislative incentive payment of $1 million. For withdrawl, you will need to use your password BJRGUD7693.

Yours very truly,

Arnold C. Creole
Vice President
Greater Caribbean Bank

***Hashes* da mensagem:**

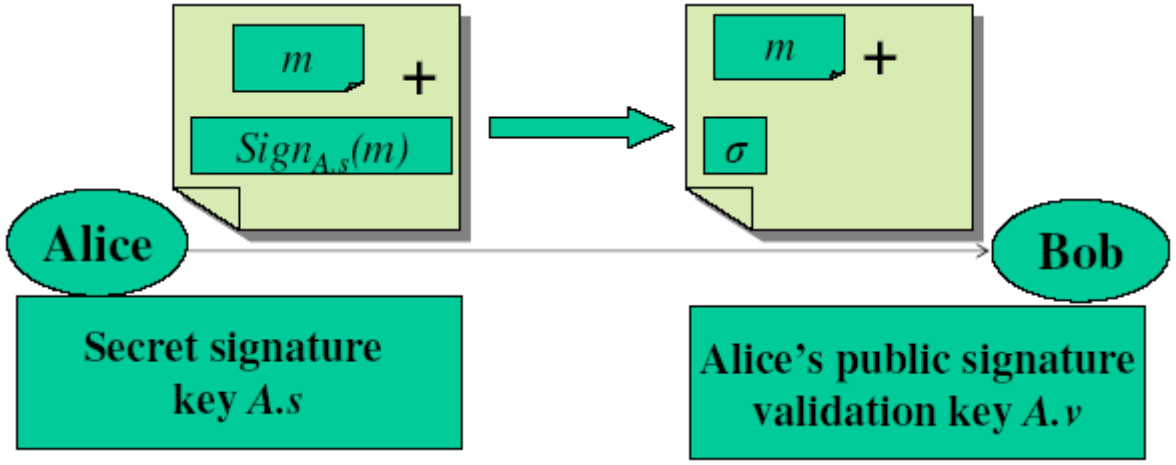
MD5	5670E64BF6CEBB46 31A25CF6990F82C0
RIPEM128	B4BB17FD0E09091A 2DF095F0B9647B41
SHA-1	1A01B56EB33FA84A 39EEDDD927977726 38331E94
RIPEM160	ABA54F46348F56D1 E492AE09A472D143 9D64E0F1

Mensagem em hexadecimal:

53 65 6e 61 74 6f 72 20 48 61 6e 64 20 4e 2e 20
54 69 6c 6c 0d 0a 57 61 73 68 69 6e 67 74 6f 6e
2c 20 44 43 0d 0a 0d 0a 44 65 61 72 20 53 65 6e
61 74 6f 72 3a 0d 0a 0d 0a 54 68 69 73 20 6c 65
74 74 65 72 20 69 73 20 74 6f 20 69 6e 66 6f 72
6d 20 79 6f 75 20 77 65 20 68 61 76 65 0d 0a 6f
70 65 6e 65 64 20 61 63 63 6f 75 6e 74 20 6e 6f
2e 20 33 33 38 39 30 37 30 32 31 20 6f 6e 20 79
6f 75 72 0d 0a 62 65 68 61 6c 66 2c 20 61 6e 64
20 64 65 70 6f 73 74 65 64 20 74 68 65 72 65 69
6e 20 61 0d 0a 6c 65 67 69 73 6c 61 74 69 76 65
20 69 6e 63 65 6e 74 69 76 65 20 70 61 79 6d 65
6e 74 20 6f 66 0d 0a 24 31 20 6d 69 6c 6c 69 6f
6e 2e 20 20 46 6f 72 20 77 69 74 68 64 72 61 77
6c 2c 20 79 6f 75 20 77 69 6c 6c 0d 0a 6e 65 65
64 20 74 6f 20 75 73 65 20 79 6f 75 72 20 70 61
73 73 77 6f 72 64 20 42 4a 52 47 55 44 37 36 39
33 2e 0d 0a 0d 0a 59 6f 75 72 73 20 76 65 72 79
20 74 72 75 6c 79 2c 0d 0a 0d 0a 41 72 6e 6f 6c
64 20 43 2e 20 43 72 65 6f 6c 65 0d 0a 56 69 63
65 20 50 72 65 73 69 64 65 6e 74 0d 0a 47 72 65
61 74 65 72 20 43 61 72 69 62 62 65 61 6e 20 42
61 6e 6b

Assinaturas digitais com chave pública

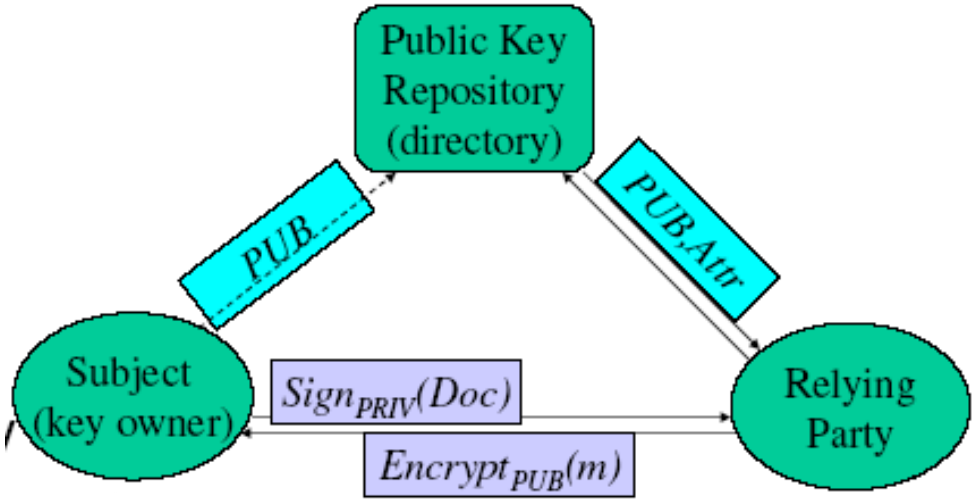
- Assinar utilizando a chave privada do par de chaves de uma cifra assimétrica
- Toda a gente conhece a chave pública de validação
- Toda a gente pode validar assinaturas em qualquer momento
  - Fornece não-repudiação - o assinante é garantido



Verificar, usando *A.v*, que σ é assinatura de Alice de *m*

Confiar em chaves públicas

- As chaves públicas são muito úteis:
  - Encriptação de dados e chaves de sessão (simétricas)
  - Assinar documentos



- Primeira aproximação: Chaves públicas serão registadas na directoria/repositório (por exemplo, X.500)
  - De confiança, centralizado
  - Sujeito identificado de forma única
  - A directoria relaciona o sujeito coma chave pública
    - Chave pública autenticada entre a directoria e a parte de confiança
  - Possibilidade de outros atributos na directoria

Problema da distribuição de chaves

Chaves simétricas:

- Como é que duas entidades trocam chaves secretas através da rede?
- **Solução**:
  - Centro de distribuição de chaves de confiança (**KDC – *Key Distribuition Center***) actuando como intermediário entre entidades
  - O KDC necessita partilhar chaves com cada entidade, trabalha sempre *online*

Chaves públicas:

- Quando Alice obtém a chave pública de Bob (dum *site Web*, por *email*, disquete) como é que pode ter a certeza que a chave pública de Bob e não da Eve?
- **Solução**:
  - Autoridade de certificação de confiança (CA – *Certification Authority*)

Gestão de chaves

A gestão de chaves controla a distribuição e uso de chaves de encriptação

- Algoritmos assimétricos revelam a chave pública e escondem a privada
  - As chaves públicas são trocadas
  - As chaves privadas são guardadas em segurança
- Algoritmos simétricos requerem um mecanismo seguro de troca de chaves
  - As chaves devem ser mantidas em segredo durante a troca das mesmas

Serviços de autenticação

- Assegurar que a identidade e a autoridade dos participantes
- Pode-se escolher:
  - Tecnologias: ***passwords,*** **frases de desafio** ***(challenge phrases),*** ***tokens hard*** **e** ***soft*** **com** ***one-time passwords*** e **certificados digitais X.509**
  - Produtos: Domínios ***NT*, NDS*, RADIUS, SDI*, Entrust*, Shiva® CA***
- Em determinados casos, por exemplo uma solução utilizada em VPNs, devia ser possível seleccionar o método de autenticação que melhor se adaptar ao caso em questão.

** Marcas registadas das respectivas companhias*

Serviços de autenticação

- O uso de certificados digitais tem sido o preferido:
  - Os certificados digitais X.509 são uma norma de facto
  - Autenticação melhor do que a dos *tokens* e *passwords*
  - Identifica sistemas e indivíduos
  - O cliente e o sistema operam mesmo quando a entidade de certificação está incontactável

Relações de confiança

- Para podermos confiar na autenticação e integridade de uma mensagem temos de ter a certeza que uma chave utilizada para encriptar uma *message digest* pertence, de facto, a quem pensamos que pertence

- Isto requer uma **relação de confiança** de forma a termos alguma certeza sobre quem possui a chave privada

- **Existem dois tipos de mecanismos de confiança**, ambos se baseiam na confiança baseada, sobretudo, na boa reputação

Reputação como recurso

- Reputação é obter e manter o bom nome entre várias partes (entidades, agentes) com (muitas vezes) interesses em conflito
- Assegurar (alguns) interesses de algumas das partes
  - Muitas vezes visto como prevenção de ameaças/riscos

- Como?
  - Parar e punir comportamentos conflituosos
  - Educação, Punição, Incentivos
  - Prevenir danos devidos a comportamentos conflituosos
- Muito geral – economia, direito, ...
- Vamo-nos focar na ciência da informação (informática)

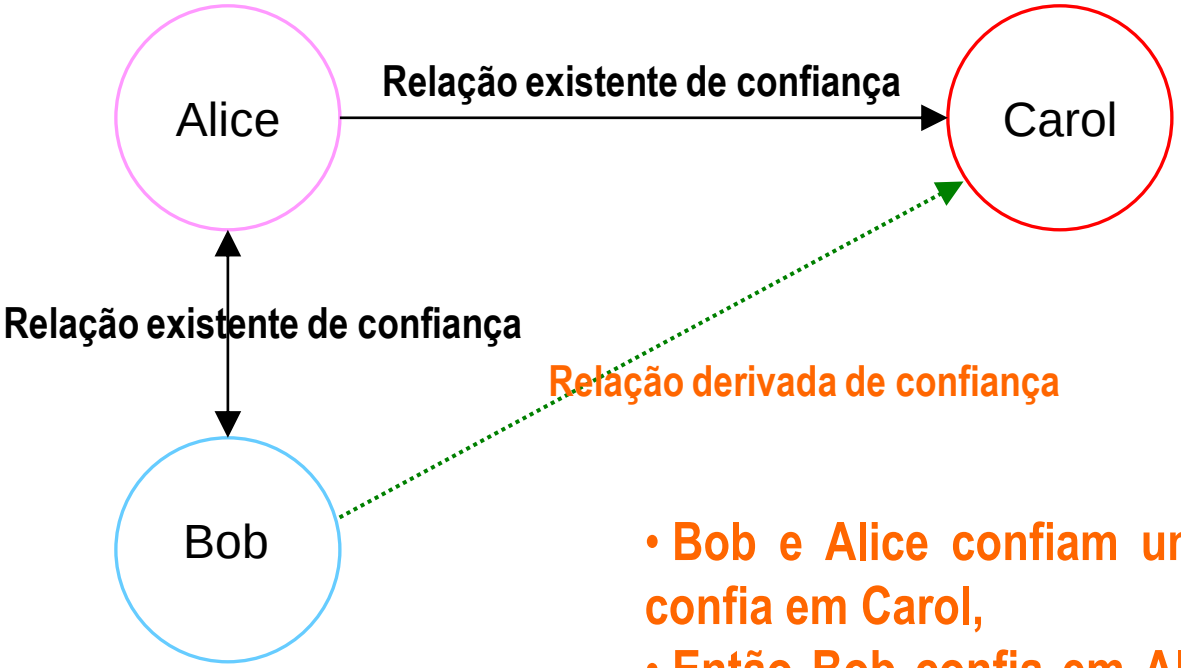
Segurança pela Reputação e pela Identificação

- **Deter e punir comportamentos conflituosos/premiar bons comportamentos**
  - Pela identificação + prova (por exemplo, para o tribunal)
    - Provar o mau comportamento (o que foi mal feito)
    - Identificar o adversário (quem o fez)
  - Pela reputação (artigos, história)

- **Prevenir os danos** independentemente do comportamento dos adversários
  - Pela identificação (lista de bons/maus)
  - Pela reputação: Evitar os "maus" (trabalhar apenas com parceiros com boa reputação)

Rede de confiança

Aproximação *peer-to-peer*



- Bob e Alice confiam um no outro, e Alice confia em Carol,
- Então Bob confia em Alice para apresentar Carol
- Então Bob passa a confiar em Carol

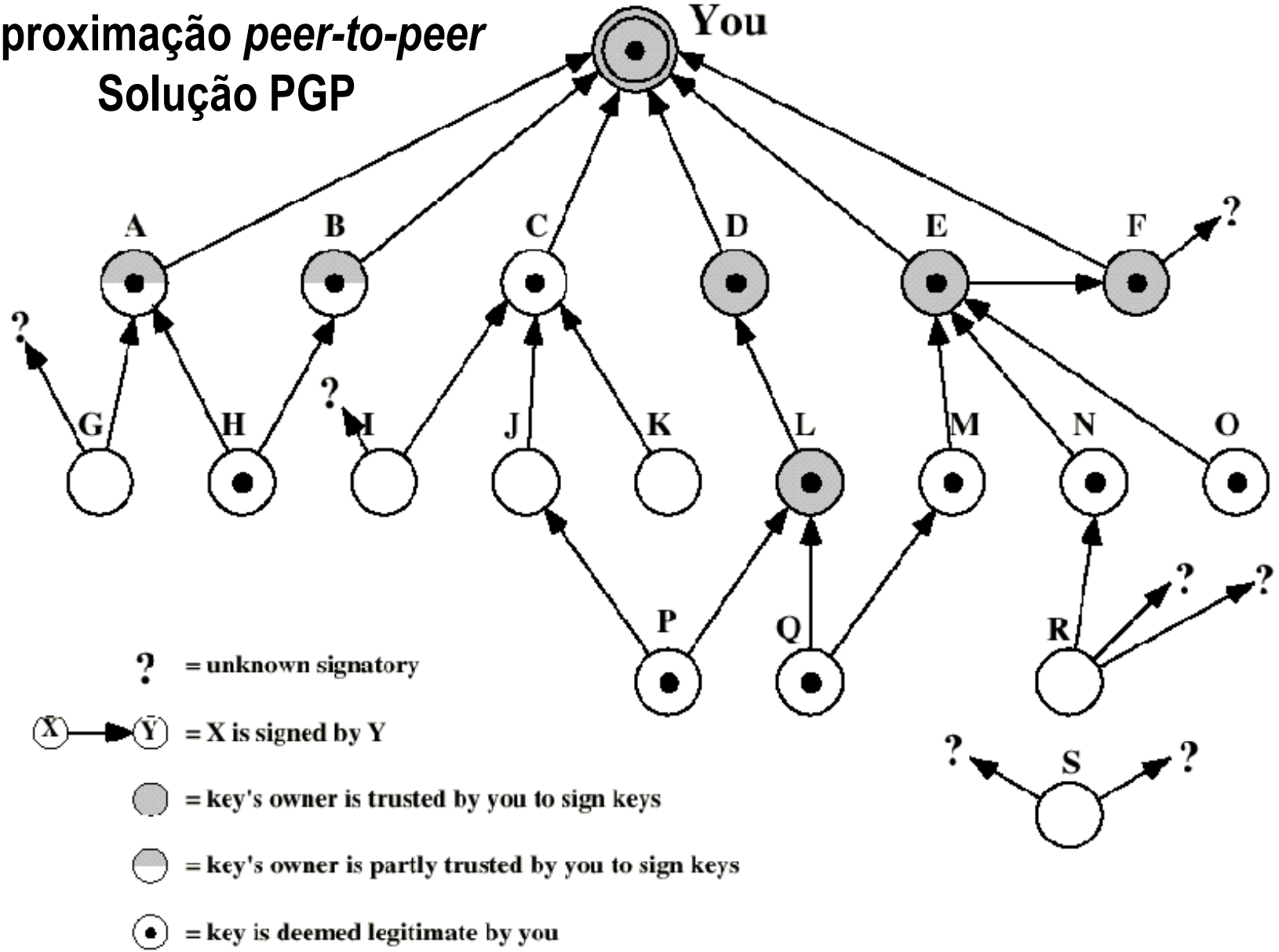
Rede de confiança

Aproximação *peer-to-peer*

- Não lida facilmente com desconhecidos de terceiro nível e acima
  - Bob confia em Carol para introduzir Don na rede?
  - Então e Earl, que ninguém dos acima conhece?
- Pode ser utilizado?
  - Sim – é o modelo utilizado pelo PGP (*Pretty Good Privacy*)
- Escala bem?
  - Não – escala exponencialmente

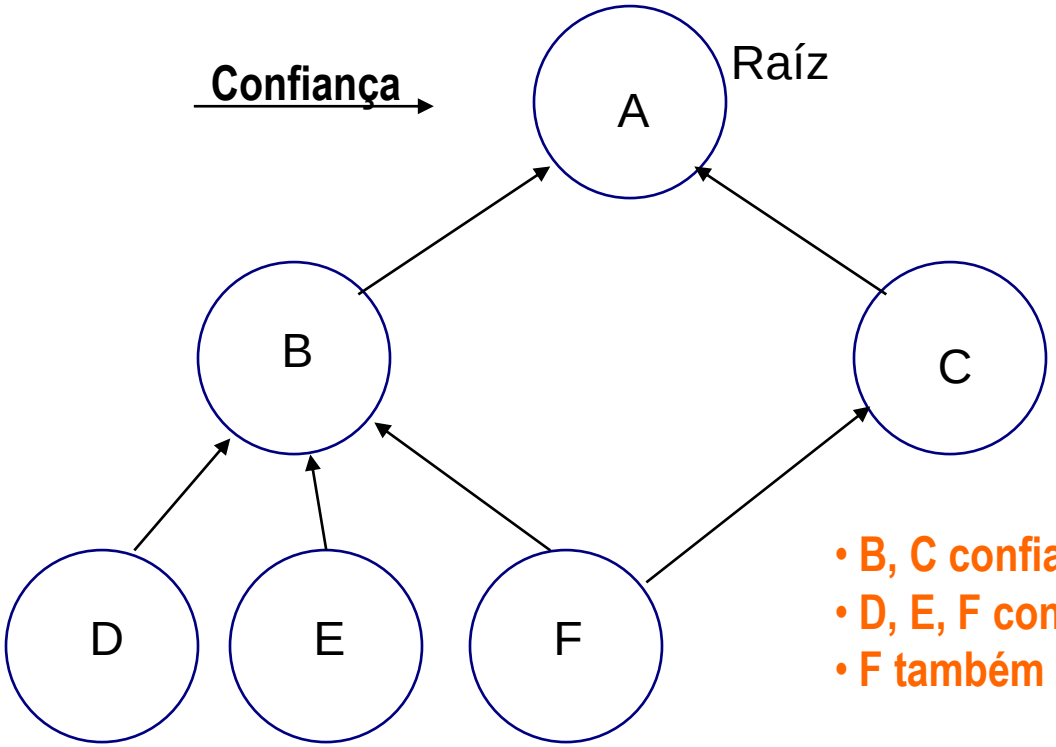
Hierárquia de confiança

Aproximação *peer-to-peer*
Solução PGP



Hierárquia de confiança

Aproximação hierárquica

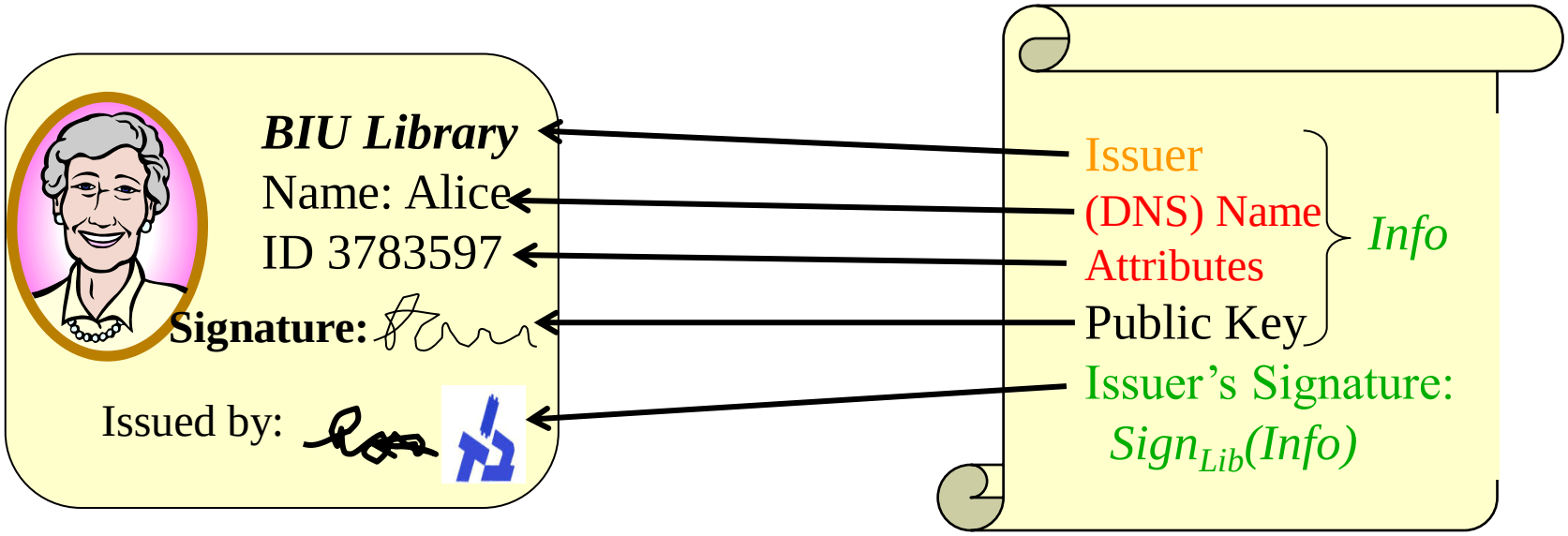


- B, C confiam em A
- D, E, F confiam em B
- F também confia em C

- *Então, TODOS confiam em A*
MAS, D, E não confiam em C

Certificados

- Semelhante ao passaporte ou à licença de condução
- Relaciona uma chave pública com um nome e/ou outros atributos do dono da chave, por exemplo, nome DNS do *site Web*
- Assinado por uma parte de confiança (emissor/Autoridade de Certificação [CA])
- Permite a uma parte de confiança (Bob, cliente) validar o nome, atributos do dono da chave (Alice, *site Web*)



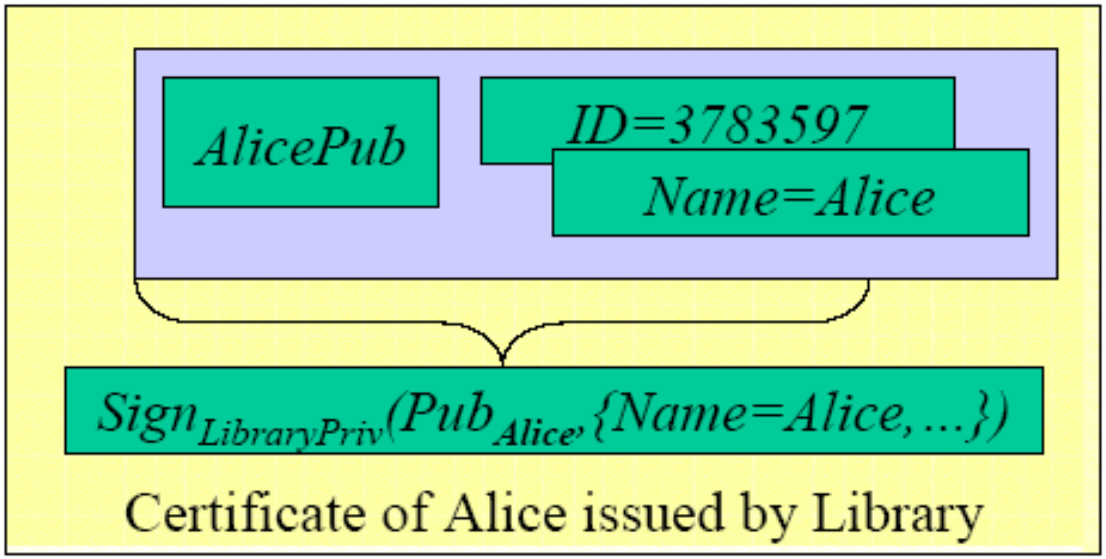
Certificados

- **Os certificados são documentos digitais que atestam a autenticidade de uma chave pública relativamente a um individuo ou a uma entidade**
- Os certificados permitem a verificação de que uma chave pública pertence realmente a uma determinada entidade ou individuo
- Os certificados contêm, no mínimo:
  - Uma chave pública e um nome
  - Data de expiração
  - Nome da entidade certificadora que emitiu o certificado
  - Um número de série
  - Outras informações

- **Importante - os certificados contêm a assinatura digital do emissor do certificado**

Certificados - Exemplo

- Um cartão de identificação de biblioteca da Alice permite a sua identificação pessoal – associando-a ao seu registo (pelo ID)
- O certificado permite a validação do pedido da Alice via rede (assinado com a sua chave pública)
- Este é um certificado de chave pública de identificação (*Identity* PKC – *Identity Public Key Certificate*)
- Natural: Semelhante aos cartões de identidade, passaportes, etc.

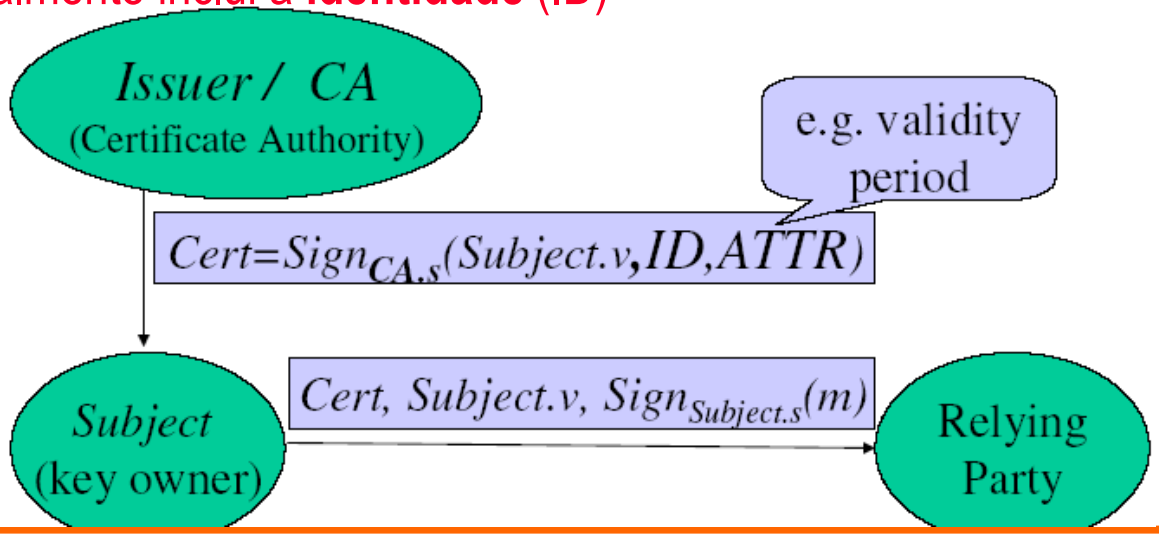


Certificados

- Um certificado de chave pública (*Public Key Certificate* (PKC)) é 4-tuplo
  - <$Emissor_{pub}$, $Sujeito_{pub}$, Atributos, Assinatura>, onde:
    - $Emissor_{pub,}$ $Sujeito_{pub}$ são chaves públicas
    - Atributos – podem ser vários e são descritos adiante
    - Assinatura é realizada usando o $Emissor_{priv}$ sobre o $Sujeito_{pub}$ e os Atributos, nomeadamente …
      - O certificado é válido se:
        - Válido $_{Emissorpub}$ (assinatura, {$Sujeito_{pub}$, Atributos})

Certificados de chave pública

- O emissor (CA) assina certificando com esta acção a relação entre a chave pública e diferentes atributos com o dono da chave
- Um parceiro de confiança valida o certificado emitido pelo CA
- O que são os atributos?
  - Devem ajudar o parceiro de confiança (*relying party*) a decidir sobre o sujeito
  - O emissor (CA) deve ser capaz de os validar (responsabilidade)
  - Normalmente inclui a **identidade** (ID)



Pormenores

- **Como é que nós sabemos que A é de confiança?**
  - Porque A assim o afirma! 
  - Quais são os critérios para **estabelecer relações de confiança**?

- É útil estender a confiança a entidades que desconhecíamos anteriormente?

- Escala bem?
  - Sim, linearmente

A hierarquia é a base dos serviços de directoria X.500

- O X.500 teve inicio como uma resposta à necessidade de harmonizar as directorias telefónicas em todo o mundo
  - Na sua origem, o X.500, é uma especificação de base de dados
  - A sua implementação originou o *Directory Access Protocol*, DAP
  - Isto levou, por sua vez, à *Lightweight Directory Access Protocol* (LDAP)

- O X.509 foi desenvolvido como uma forma de estruturar as hierarquias de confiança

Autoridade de Certificação (*Certification Authority*)

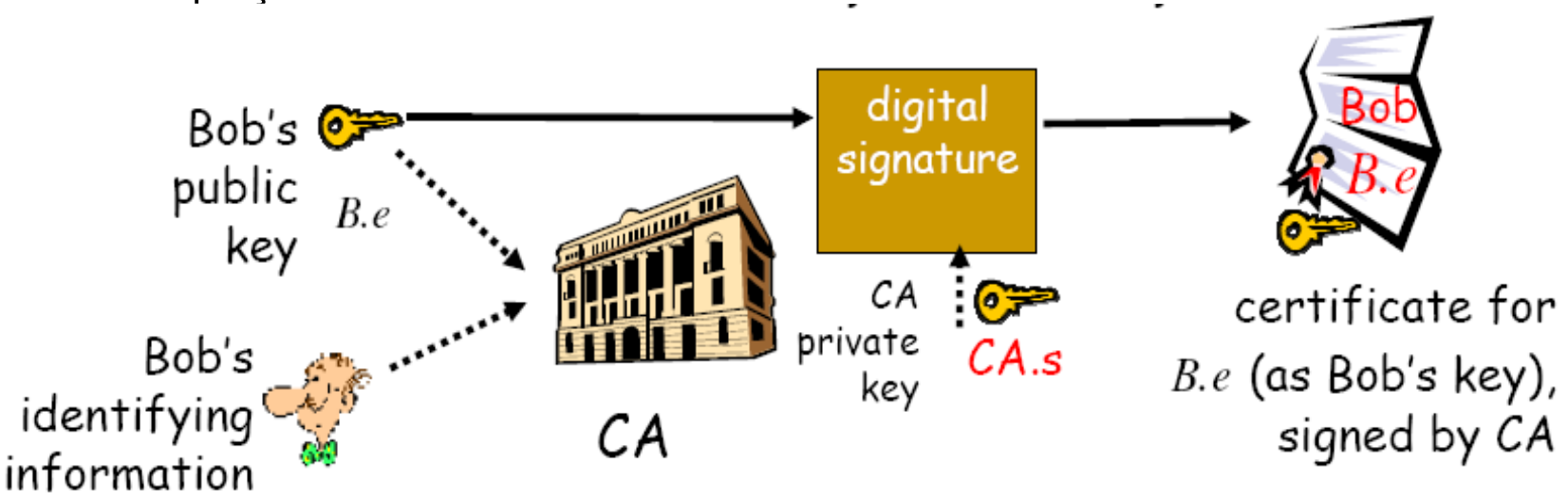
- Uma Autoridade de Certificação (***Certification Authority* (CA)**) é uma terceira parte em quem se confia e que emite Certificados Digitais que relacionam um utilizador com a respectiva chave pública

  - **A CA assina digitalmente o certificado digital** de maneira a que qualquer alteração (como a substituição por outra chave pública) se torne óbvia

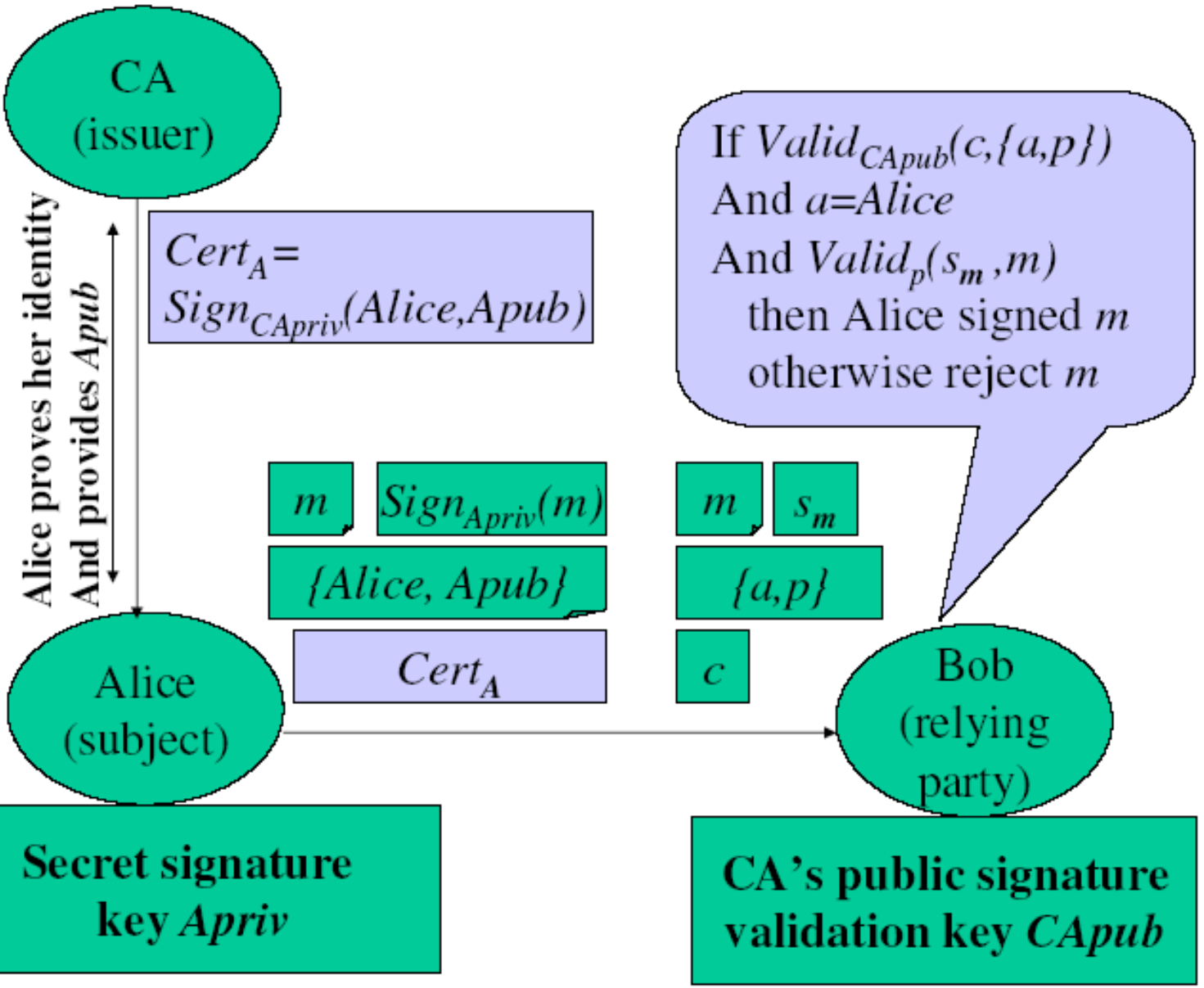
  - A CA não tem conhecimento da chave privada do utilizador

Autoridades de Certificação (CA)

- Autoridade de Certificação: Relaciona a chave pública (por exemplo: **B.e**) com o identificador (por exemplo **B.name**="BOB").
- O Bob (pessoa, servidor) regista **B.e** na CA
  - Bob convence CA que o seu nome é Bob, envia B.e
  - A CA cria o certificado relacionando "Bob" a **B.e**
  - O certificado é assinado digitalmente por CA – A diz "**B.e** é a chave pública de encriptação de Bob"

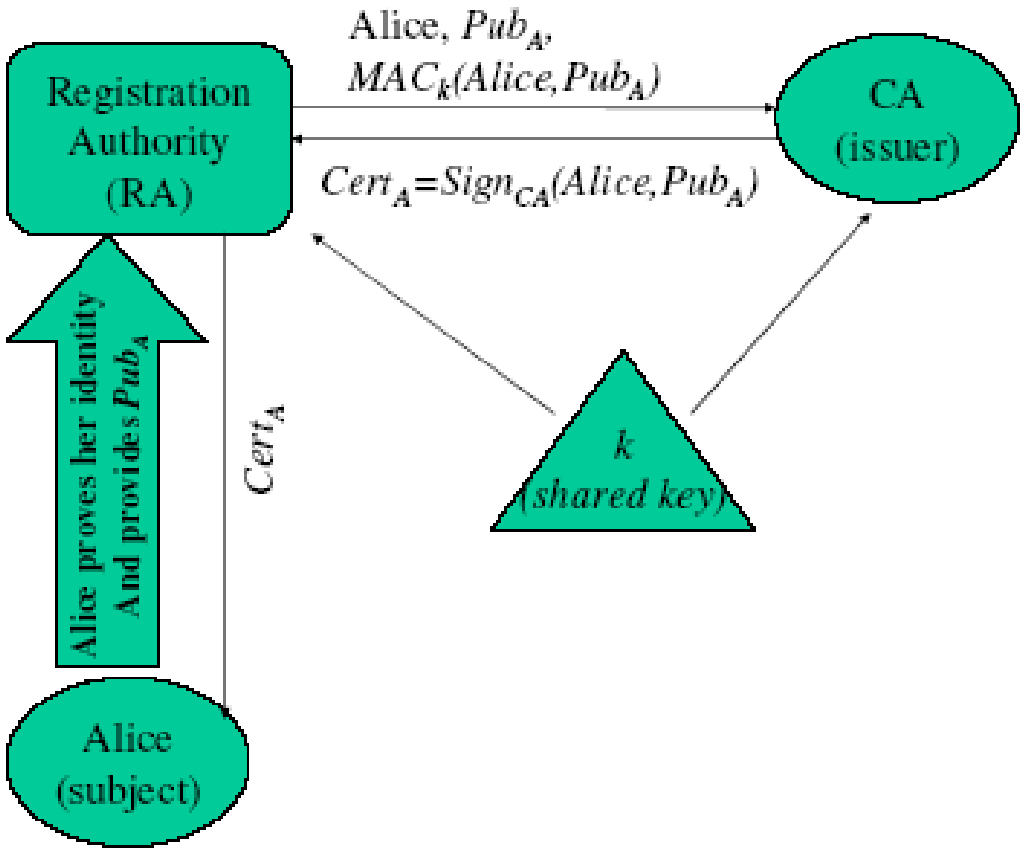


Autoridade de Certificação



Autoridade de Registo (RA)

- A RA combina duas funções:
  - Valida a identidade da origem da chave pública
  - Relaciona a chave pública com atributos (identidade,...)
- A chave secreta da CA só é necessária para assinar os certificados
- Identificação por autoridade de registo separadas

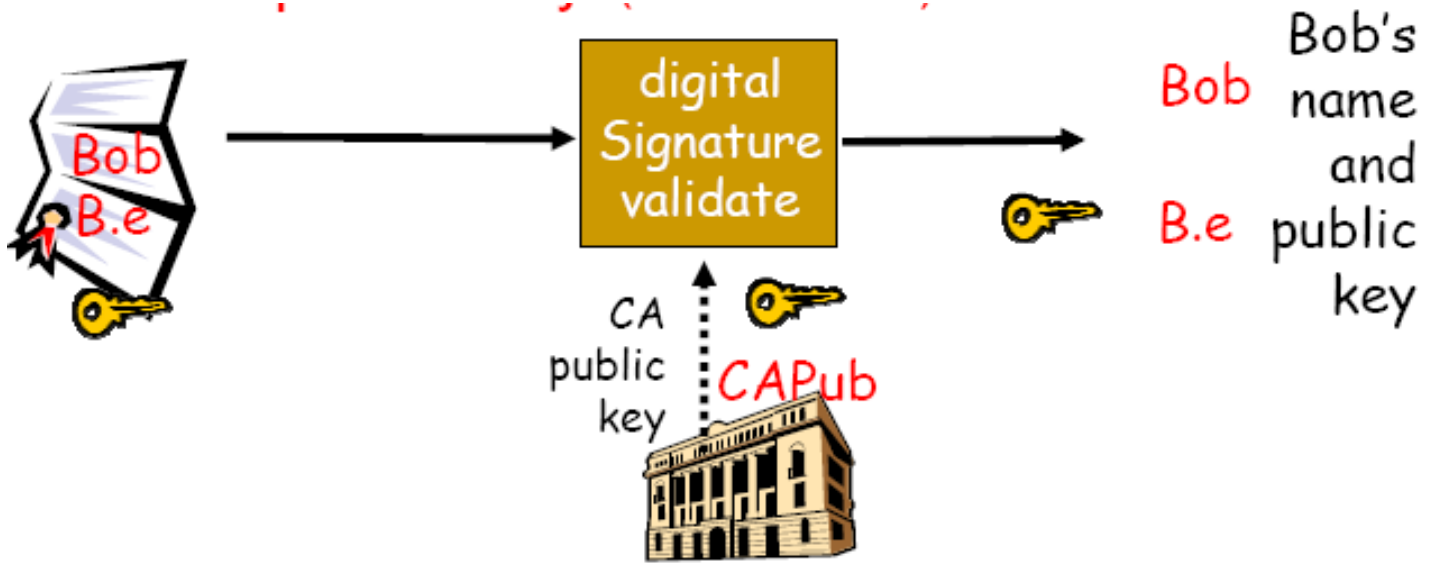


Certificados e colisões do *hash*

- Certificados (com atributos, codificação) são compridos
- Utilizar *hash* e então fazer as assinaturas, por exemplo RSA_SHA1, RSA_MD5, DSA [sempre SHA 1]
  - $RSA\_MD5\_Sign_{A.s,\ A.n}\ (m) = \langle m, [MD5(m)]^{A.s} \bmod A.n \rangle$
- Problema: colisões encontradas recentemente no MD5
  - Cuidado com os certificados que usam MD5!
    - Ataque READ: dois certificados (chaves públicas diferentes) com a mesma assinatura
    - Mais importante: Engana-se a CA para assinar uma colisão -> credenciais falsas
- Note-se que o SHA-1 também sofre colisões (ainda 2^{69})

Uso de certificados de chave pública

- Quando a Alice quer a chave pública de Bob (para encriptar uma mensagem para Bob ou para validar a assinatura deste):
  - Obtém o certificado deste (de Bob ou de outro sítio qualquer)
  - Aplica a chave pública do CA ao certificado de Bob validando assim a chave publica de Bob



Certificados Digitais

- Claramente, para isto funcionar, todas as partes têm de usar o mesmo formato para os certificados

- A norma mais popular (mas não a única) actualmente em uso é o X.509v3

- Um certificado X.509 tem um formato fixo e contém alguns itens obrigatórios numa ordem predefinida, de maneira a ser fácil a um computador pesquisar e verificar

- O X.509 foi introduzido em 1988, tendo sido revisto em 1993. Foi introduzida uma terceira versão em 1995 tendo esta sido revista em 2000.

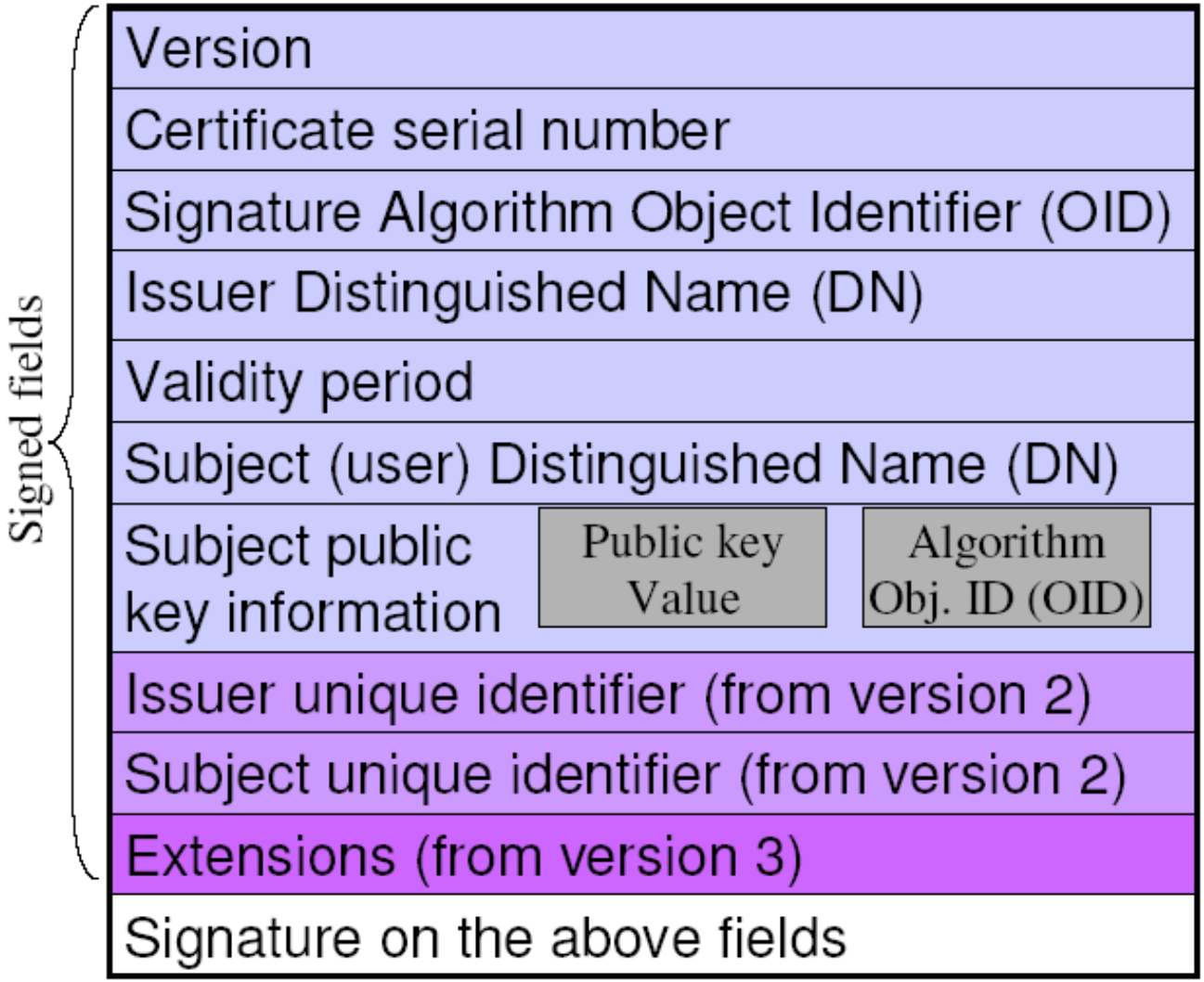
Certificados Digitais

- O X.509 baseia-se em criptografia assimétrica e assinaturas digitais.
- Não impõe algoritmos especificos de encriptação mas recomenda o RSA.
- Não impõe também nenhum algoritmo de *hash*

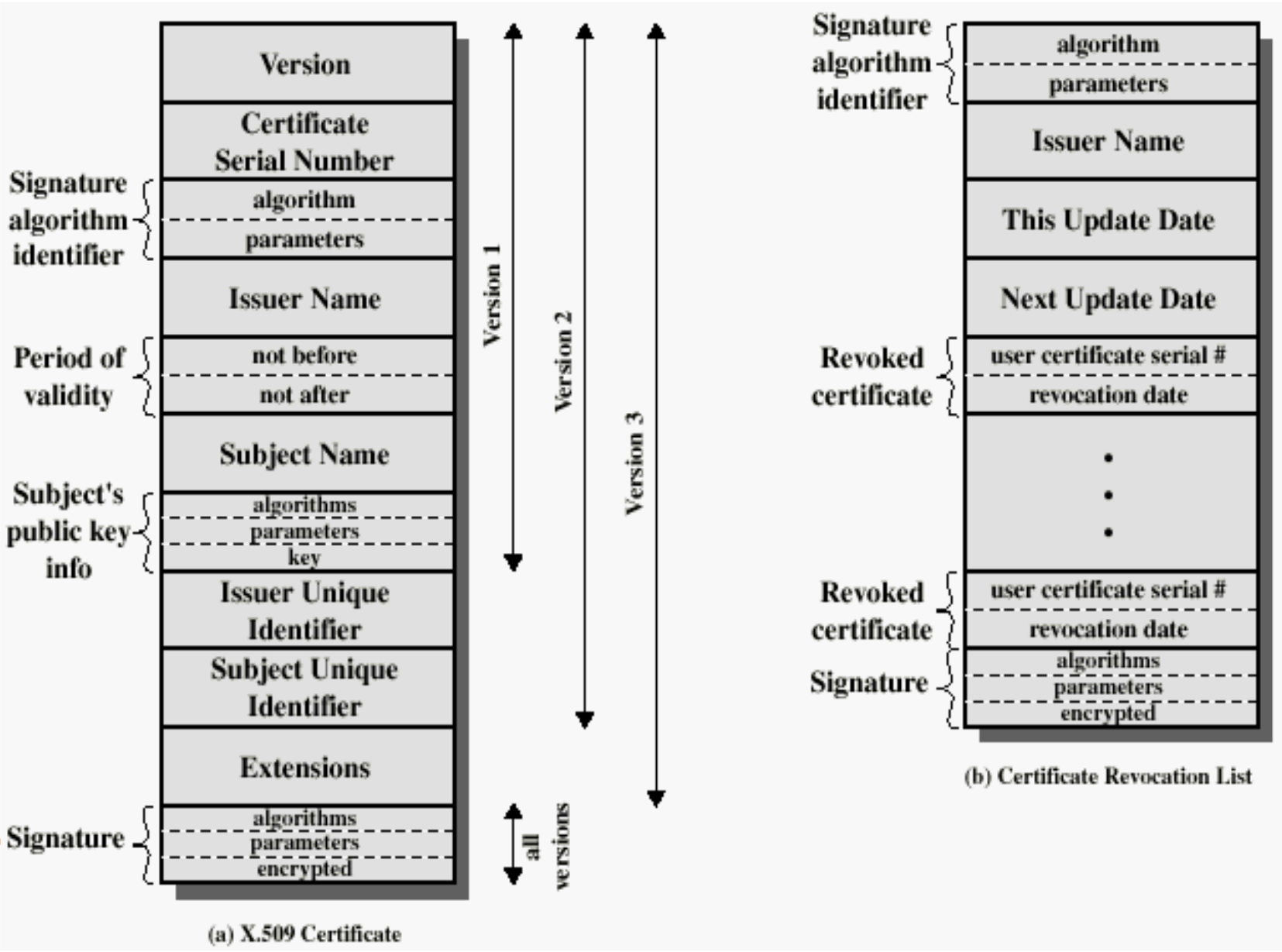
Serviço de autenticação X.509

- Conjunto de servidores distribuidos que mantêm uma base de dados sobre os utilizadores
- Cada certificado contém a chave pública de um utilizador e é assinada com a chave privada da CA
- É usado, por exemplo, no S/MIME, IP Security, SSL/TLS e SET
- O RSA recomenda o seu uso.

Certificados de chave pública X.509



Certificados de chave pública X.509 - Formatos



Identificadores de objectos (OID)

- Utiliza a norma ASN.1 (*Abstract Syntax Notation*) para identificar objectos
- Global, identificadores únicos, por exemplo: algoritmos
- Sequência de números, por exemplo: 1.16.840.1.45.33
- Números do nível de topo: 0-ITU, 1-ISO, 2-joint
- Cada organização atribui os identificadores dos níveis abaixo
- Uso no X.509: Identificação dos algoritmos e extensões

Mecanismos de extensão X.509

- Usados para certificados e para as listas de revogação de certificados (*Certificate Revocation Lists* (CRL – ver adiante))
- Cada extensão contém:
  - Identificação da extensão (OID)
  - Indicador se é critica:
    - Se for critica, todas as partes com mútua confiança devem entender a respectiva extensão para usar o certificado
    - Se não for critica, pode usar o certificado
  - Valor da extensão

Extensões X.509v3 normalizadas

- Identificador e utilização da chave (assinar, encriptar, etc.)
- Nome alternativo do sujeito e do emissor (por exemplo: email)
  - Por exemplo: dNSname do sujeito na extensão subjectAltName
- Identificador e qualificadores da politica do certificado
  - Qual é a politica do CA (e desresponsabilização)?
- Restrições do caminho de certificação
  - Restrição básica: CA ou entidade final, comprimento do caminho
  - Nome e politica de restrições (em certificados emitidos pelo sujeito)

Extensões X.509v3 normalizadas (cont.)

- Relação entre politicas
  - Como interpretar os atributos nos certificados emitidos pelo sujeito (se for CA)
- Extensões da *Certificate Revocation List* (CRL)
- Outras extensões para outros atributos (não normalizados)
  - Por exemplo: função, departamento, grau, qualidade, ...
  - Ou: uso separado "*atribute certificate*"

X.500, X.509 e *Distinguished Names*

- X.500: Recomendação ITU (norma) para directoria *on-line* (lista telefónica) global, distribuida e de confiança
  - Identificador único: *Distinguished Name* (DN)
  - Nunca foi implementado - muito complexo, muito revelador
  - Atributos diferentes, incluindo chave pública
- X.509: Autenticação relacionada com o X.500
  - Permite relacionar a chave pública com o nome (DN)

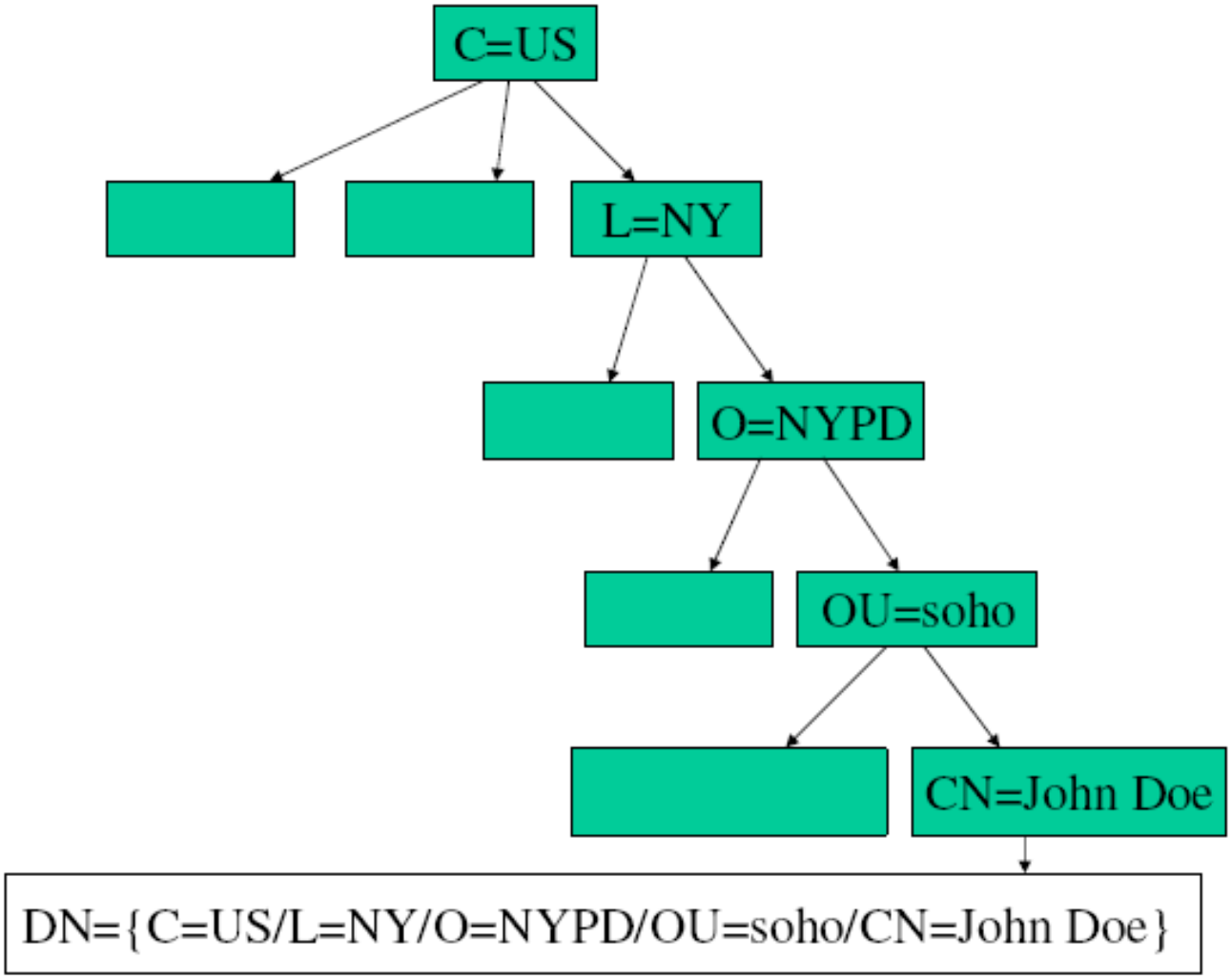
- Estabelecidos: IETF PKIX, SSL, PGP, S/MIME, IPSec,...

Distinguished Names (DN)

- Nomes globais únicos que toda a gente pode utilizar quando se refere a uma entidade – com significado legal
- Sequência ordenada de chaves (predefinidas e outras) e uma *string* correspondente a cada um deles
- Directoria distribuída, responsabilidade -> DN hierárquico

Keyword	Meaning
C	Country
L	Locality name
O	Organization name
OU	Organization Unit name
CN	Common Name

Distinguished Names (DN) - Hierárquia



Distinguished Names (DN) - Problemas

- Objectivo: Identificação legal, não ambígua e única
- Os nomes não são não-ambiguos; outros identificadores (por exemplo: número de série, SSN) não são entendidos universalmente
- A mesma entidade pode ter múltiplos DN
- Fornecer e validar detalhes é dispendioso e invasivo
- Os campos do DN podem expor informação sensível da organização
- Privacidade, possibilidade de permitir o roubo de identidade
- Hierarquia de chaves DN mal definida
- As pessoas movem-se; deve o seu DN mudar quando mudam de emprego? E a chave pública deve mudar?
- Roubo de identidade

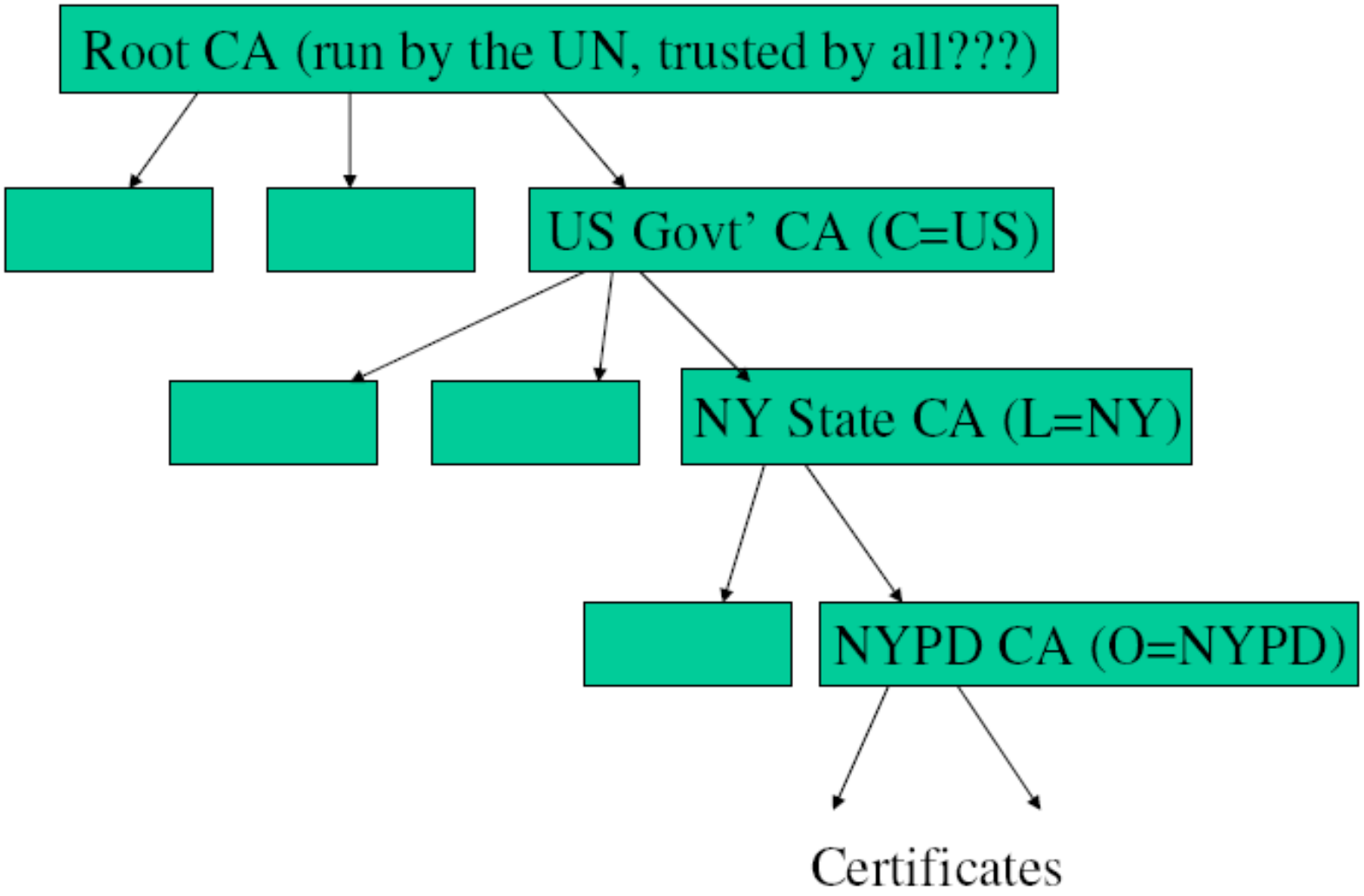
Distinguished Names (DN) – Na prática

- Identificadores aceites legalmente em alguns países
- Para assegurar que são únicos os emissores colocam frequentemente uma *string* aleatória, um número de série como parte do DN
- A partir da versão 2, os certificados X.509 contêm "identificadores únicos" adicionais para o sujeito e para o emissor
- A partir da versão 3, os certificados X.509 permitem extensões gerais, as quais são frequentemente utilizadas para adicionar identificadores
- Usado para identificar *sites Web* com SSL
- Inseguro: Utilizadores não verificam/percebem os URLs ...
- Ou, usado para identificar o utilizador para autorização

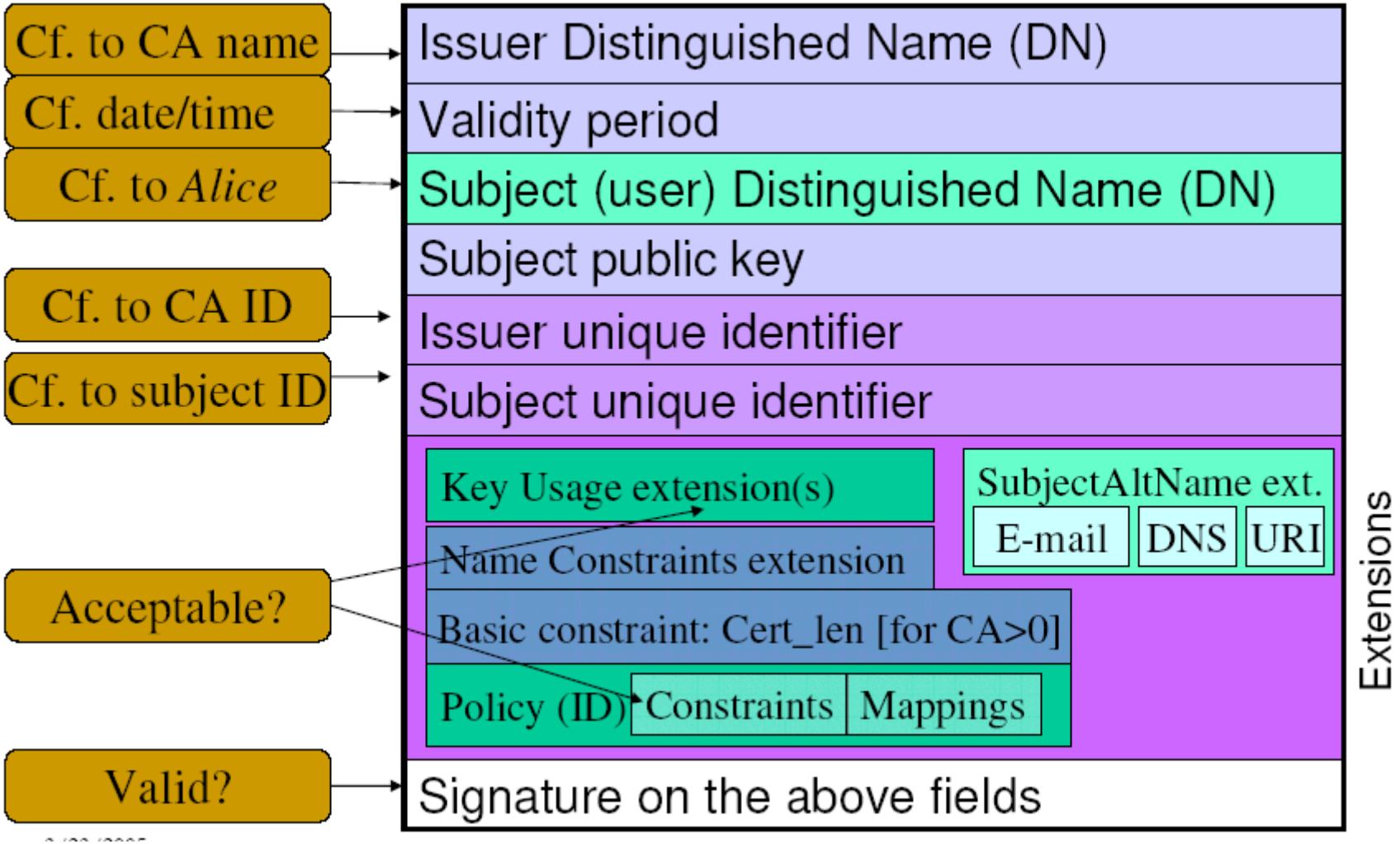
Obtenção dum certificado para um utilizador

- Características do certificado gerado pelo CA:
  - Qualquer utilizador com acesso à chave pública do CA pode aceder à chave pública que foi certificada
  - Ninguém para além do CA pode modificar o certificado sem ser detectado.

X.509 - Hierárquia global

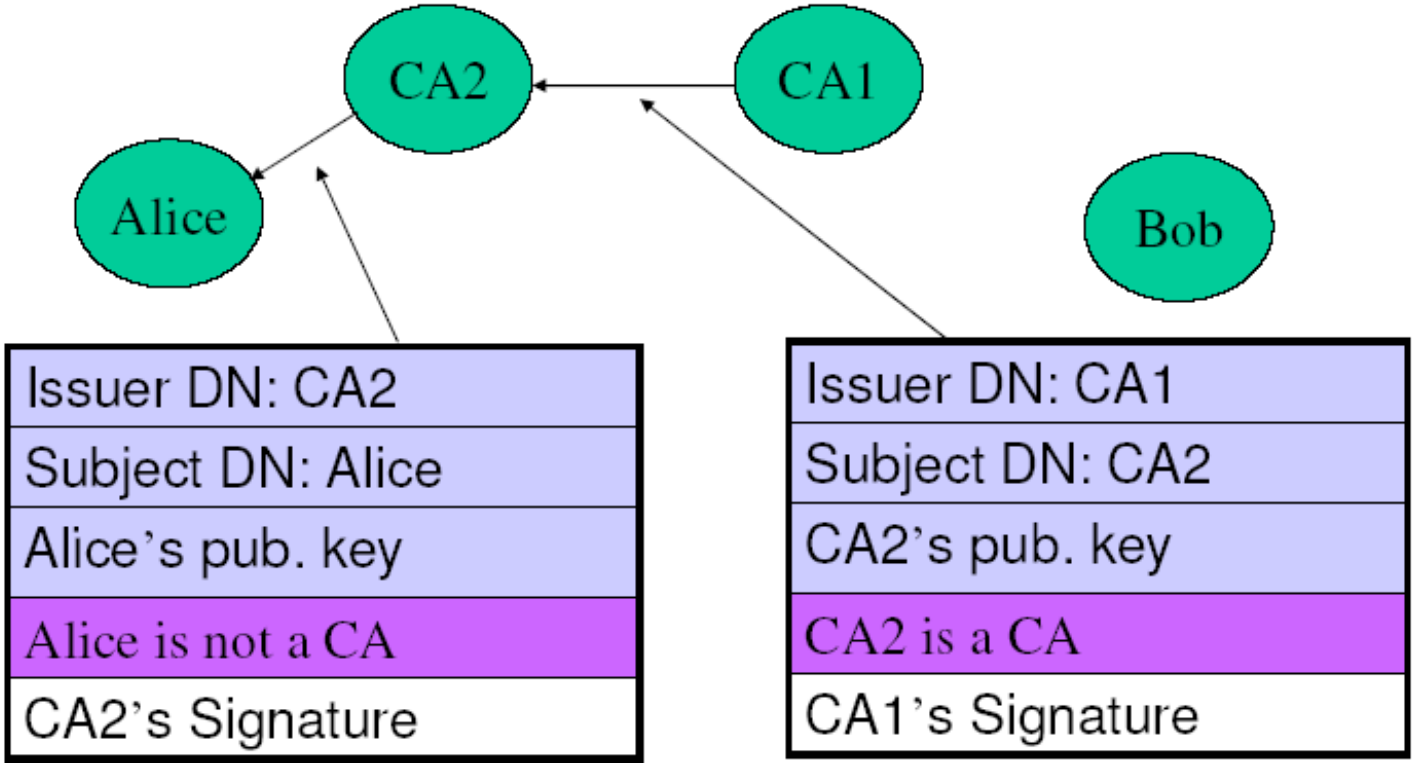


Validação do certificado $Valido_{CApub}(c,\{a,p\})$

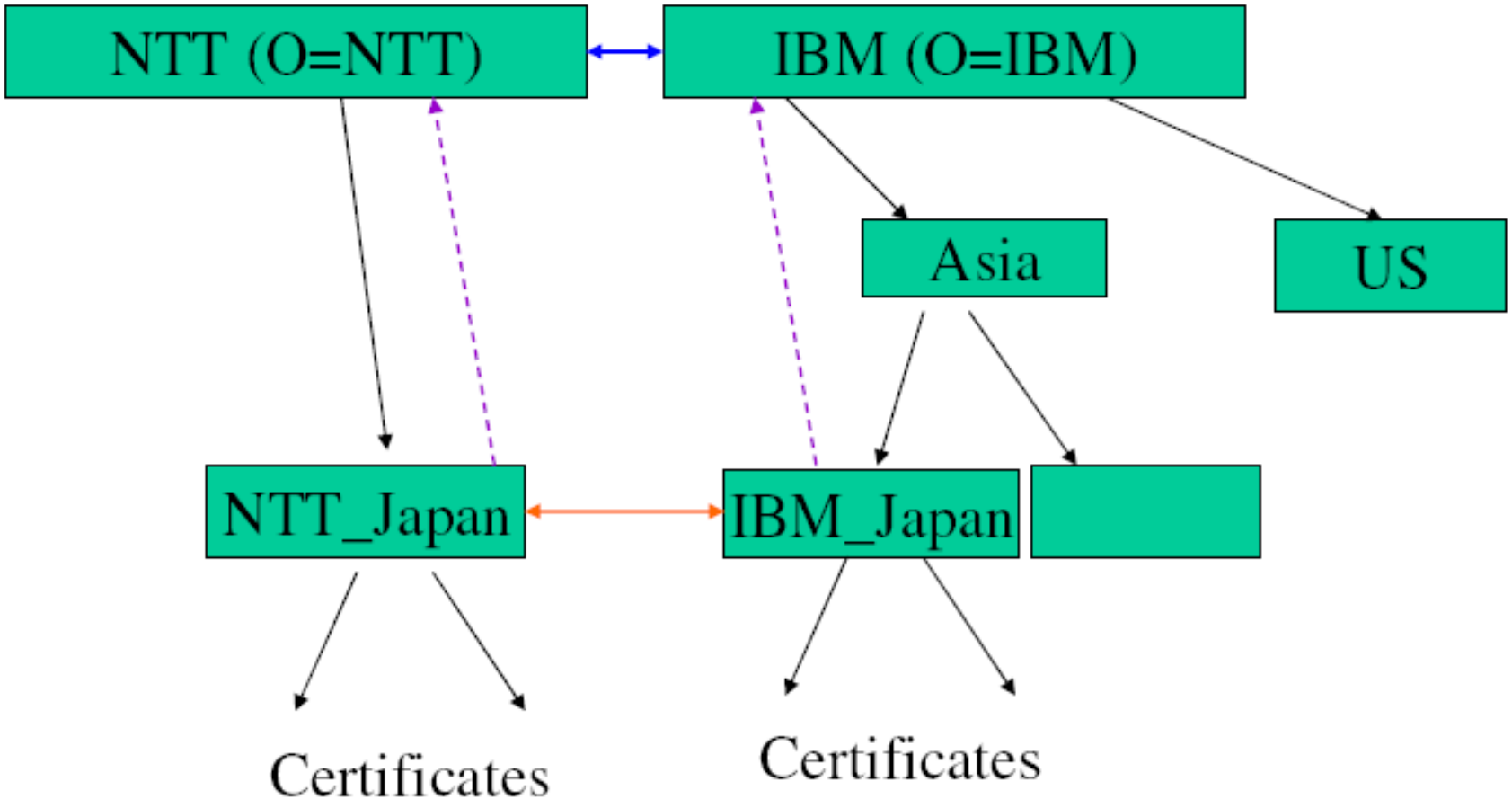


X.509 - Caminho do certificado

- E se Bob não conhecer o CA de Alice?
- Solução: Caminho do certificado – um CA conhecido e da confiança de Bob certifica o CA de Alice.



X.509 - Certificação cruzada



Grafo do caminho do certificado

- Vértices V: <pub_key, DN, CA_flag>
- CA_flag=CA para um CA e N para a entidade final

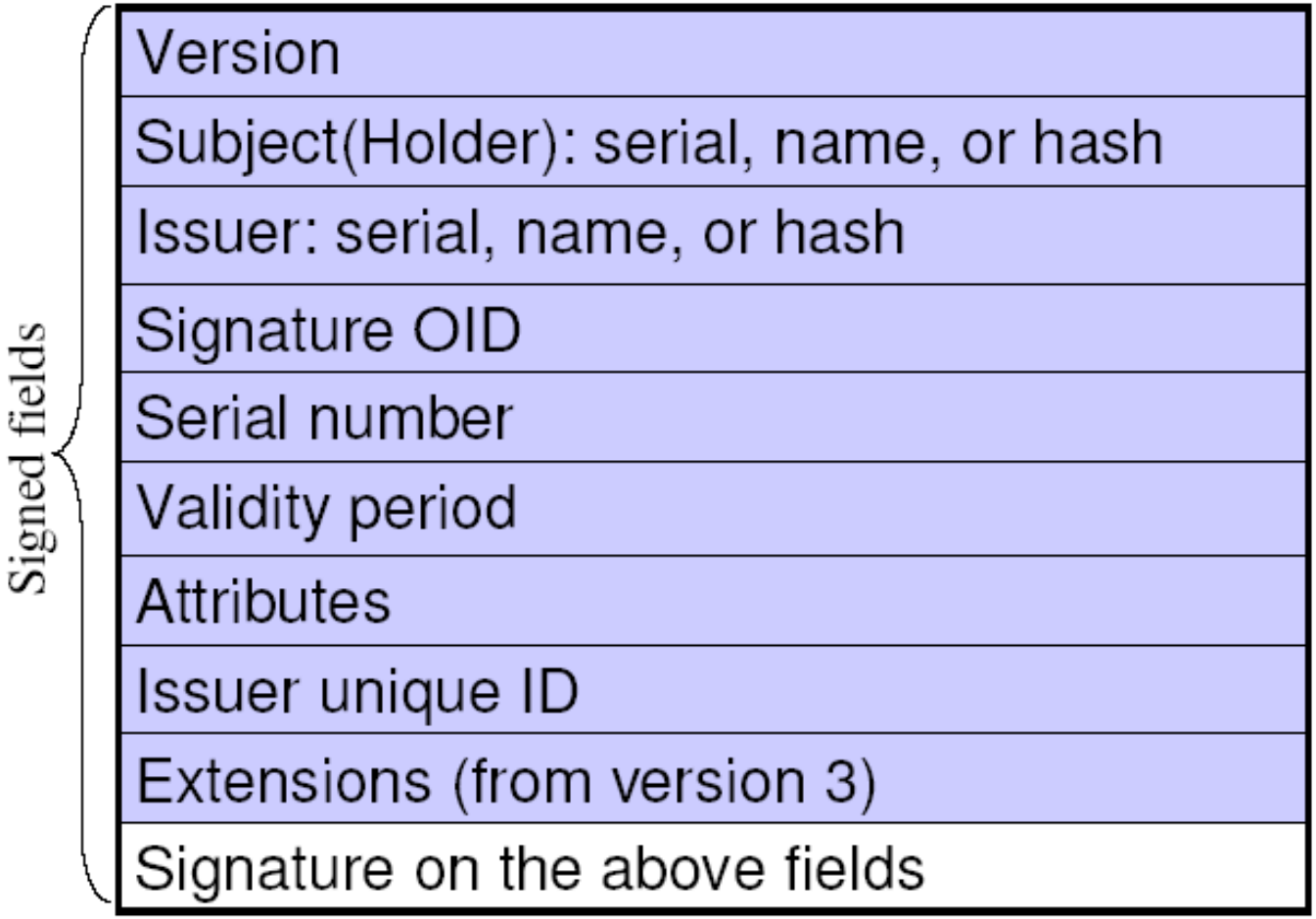
- Arestas E: Ligam de <p, n, CA> a <p', n', f> se existir um certificado:
  - Assinado pela chave pública p
  - Com emissor DN=n
  - Com sujeito DN=n'
  - Com sujeito PK=p',
  - Com CA_flag=f

- Questão: Seja V contido em V' os CAs de confiança. Existe um caminho de um vértice em V' para <p, n, f>? Encontrar o caminho mais curto!
- Resposta: Usar BFS; trabalho = O(|E|)

Identidade – Riscos no PKI

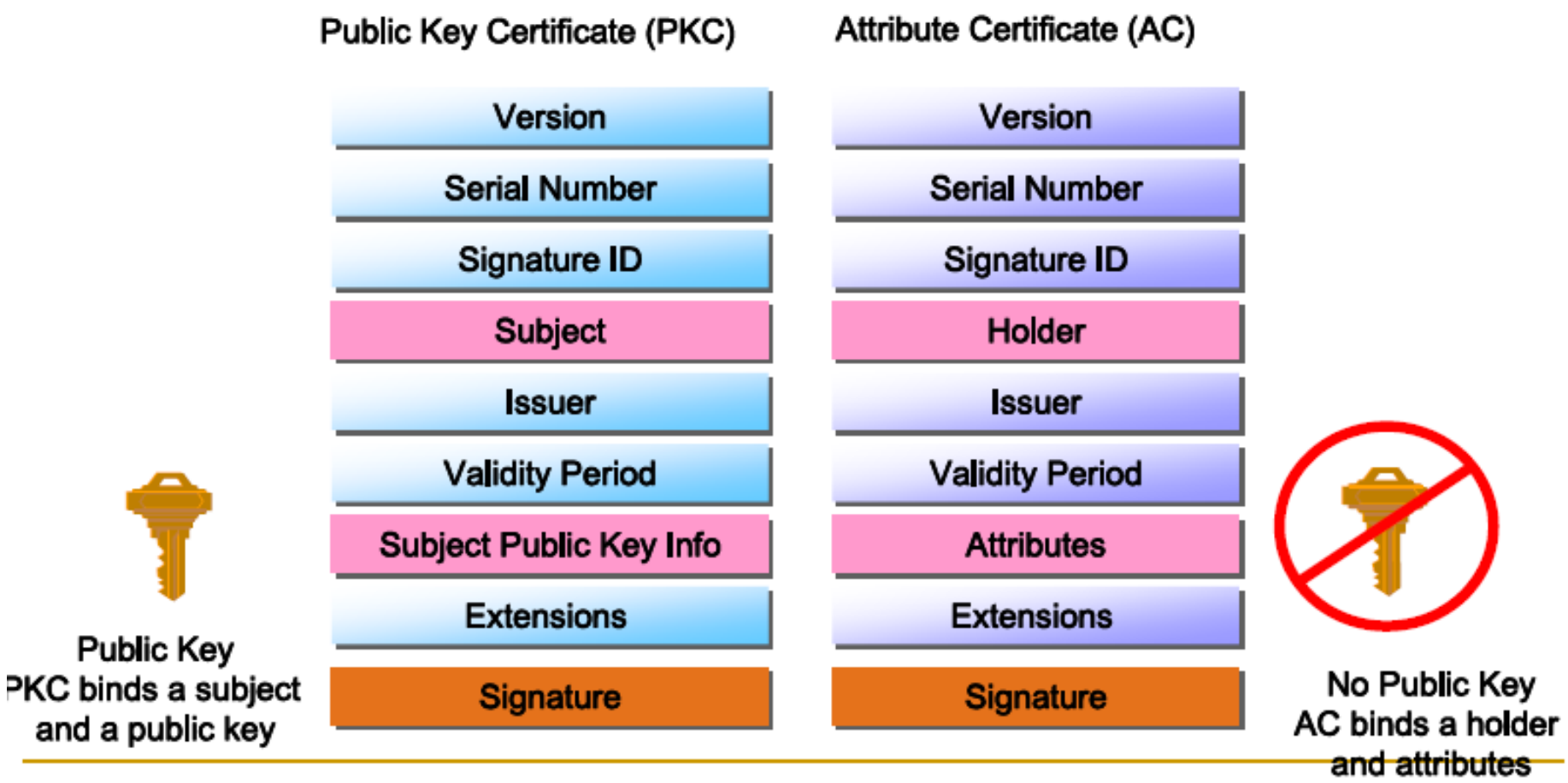
- Para as partes que confiam
  - Identidade fraudulenta
  - Disputas (reclamações de identidades fraudulentas)
- Para a entidade identificada – roubo de identidade
- Para a CA – responsabilidade
  - Potencialmente sem limites – aplicação desconhecida -> altos custos
  - Limitar através de políticas (de emissão, uso, responsabilidade)
  - Na realidade, muitas vezes descartar extrêmo de responsabilidades
  - Mesmo sem descartar de responsabilidades: como se prova a negligência???
- **Principais ameaças**:
  - Exposição da chave privada de assinatura do CA
  - Emissão de certificados para identidades falsas
    - Um problema com a emissão remota

Atibutos de certificados X.509 (v2, 2000)

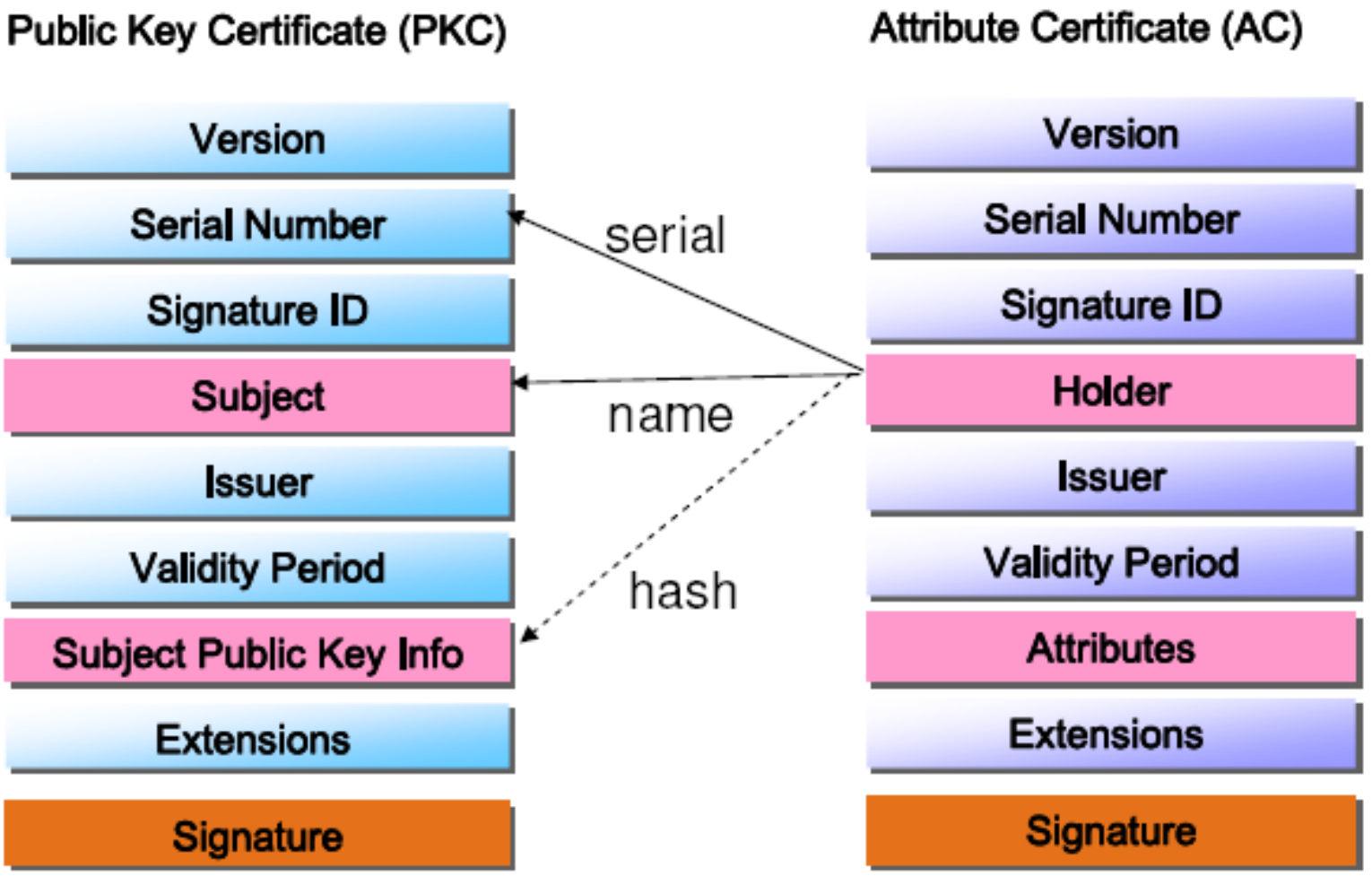


Diferenças entre PKC e AC

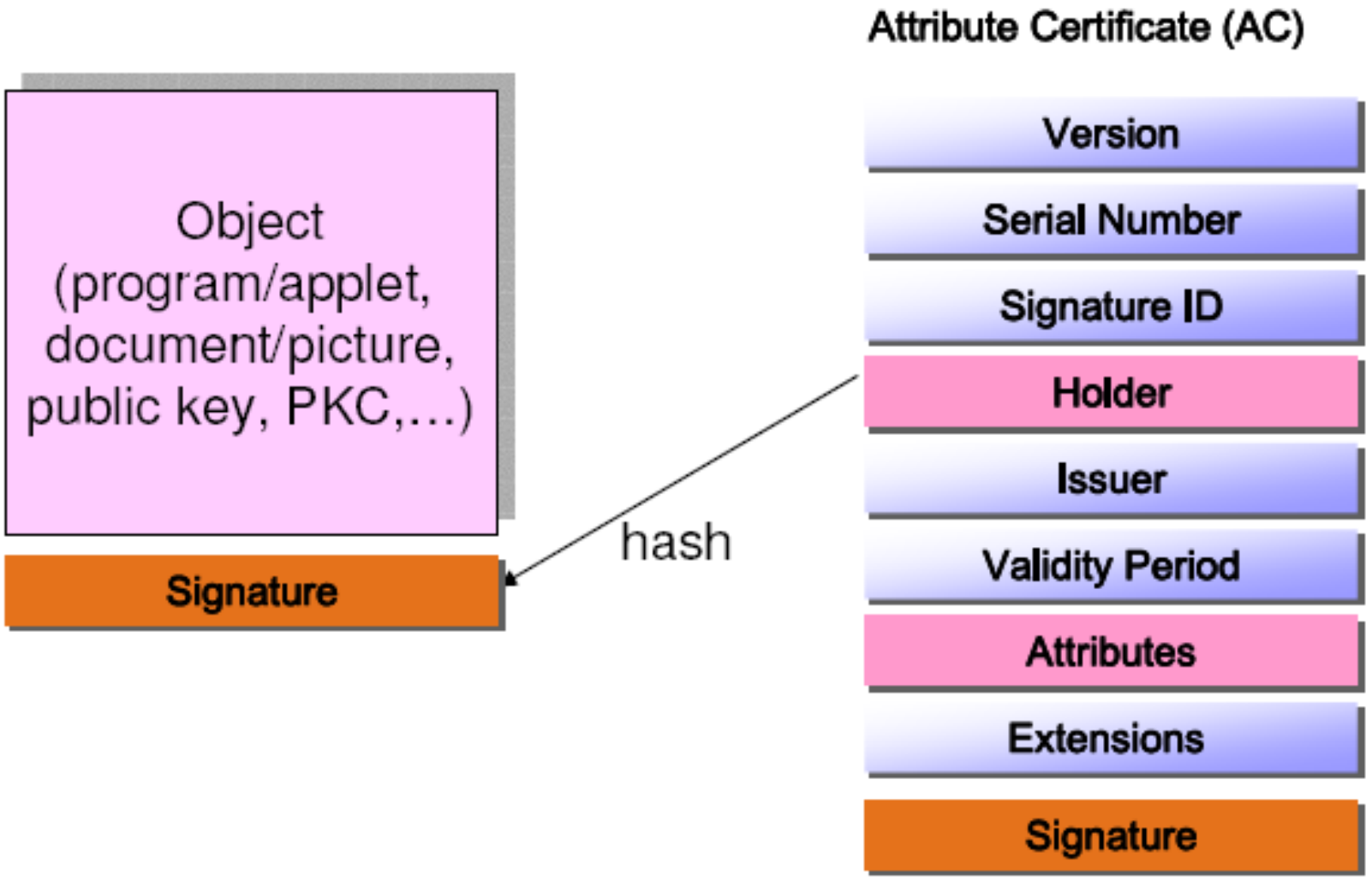
- PKC é o passaporte e AC é o visto



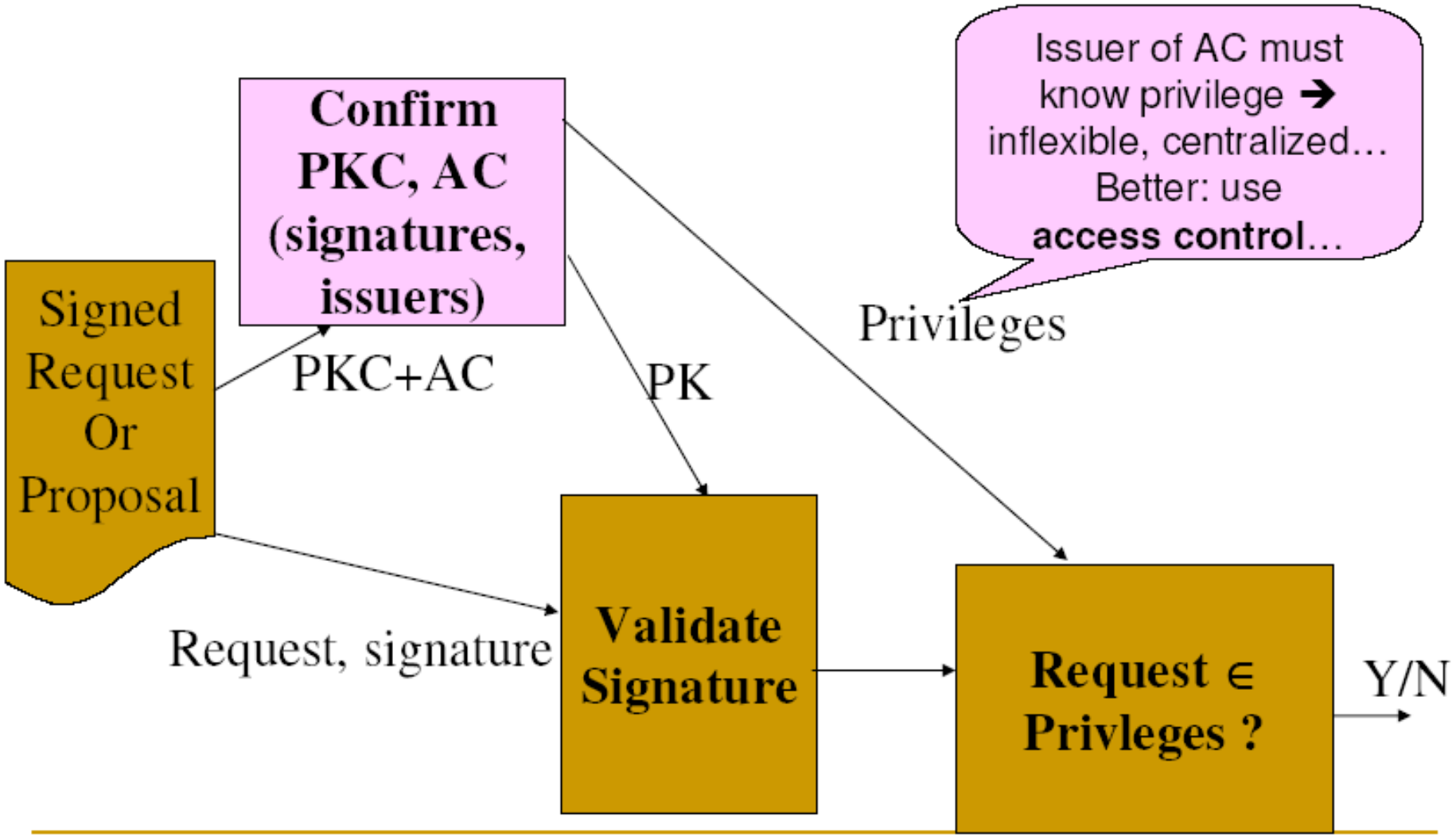
Associação de PKC e AC



Associação AC a um objecto por Hash

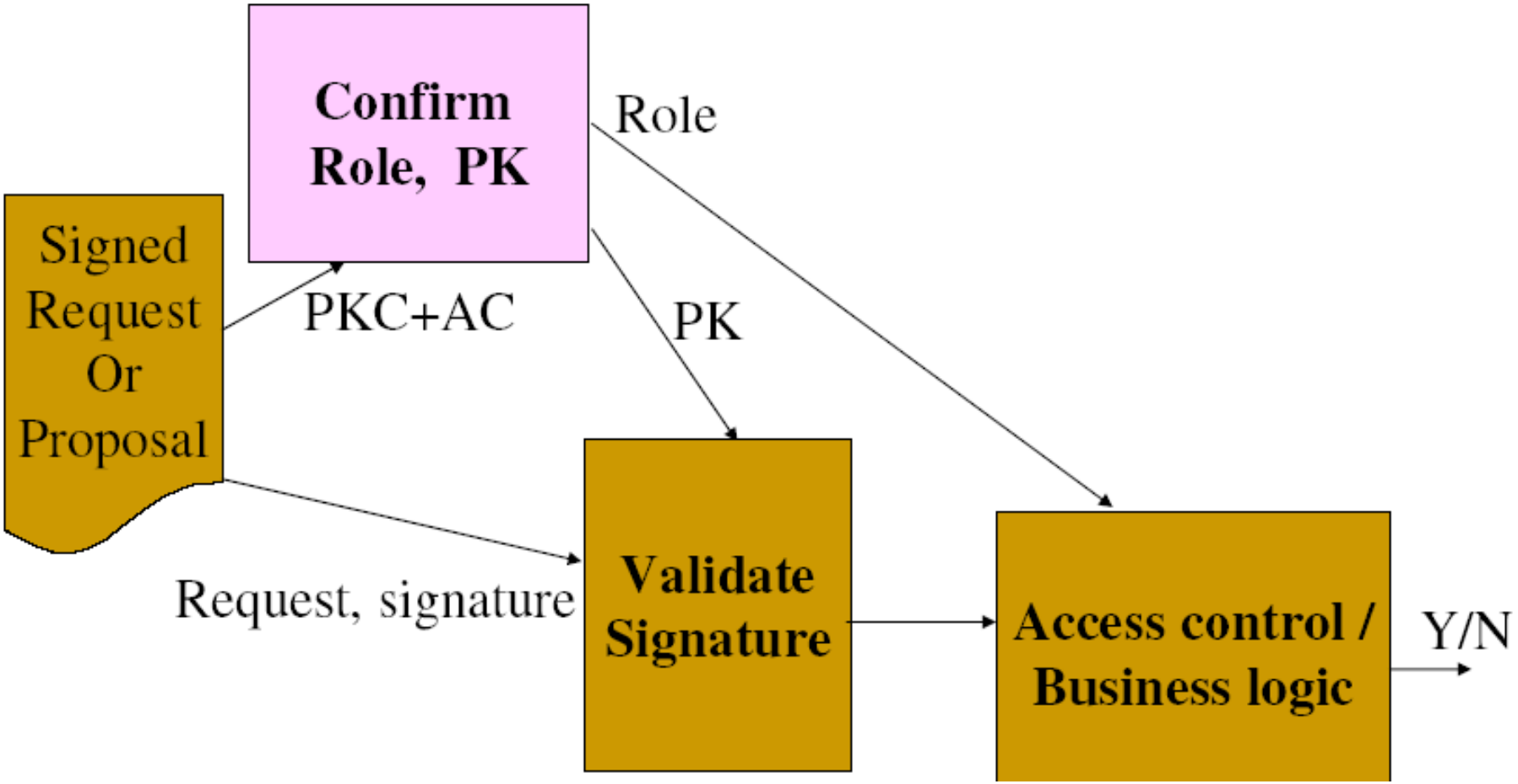


Controlo de acessos ao AC: Privilégio como atributo



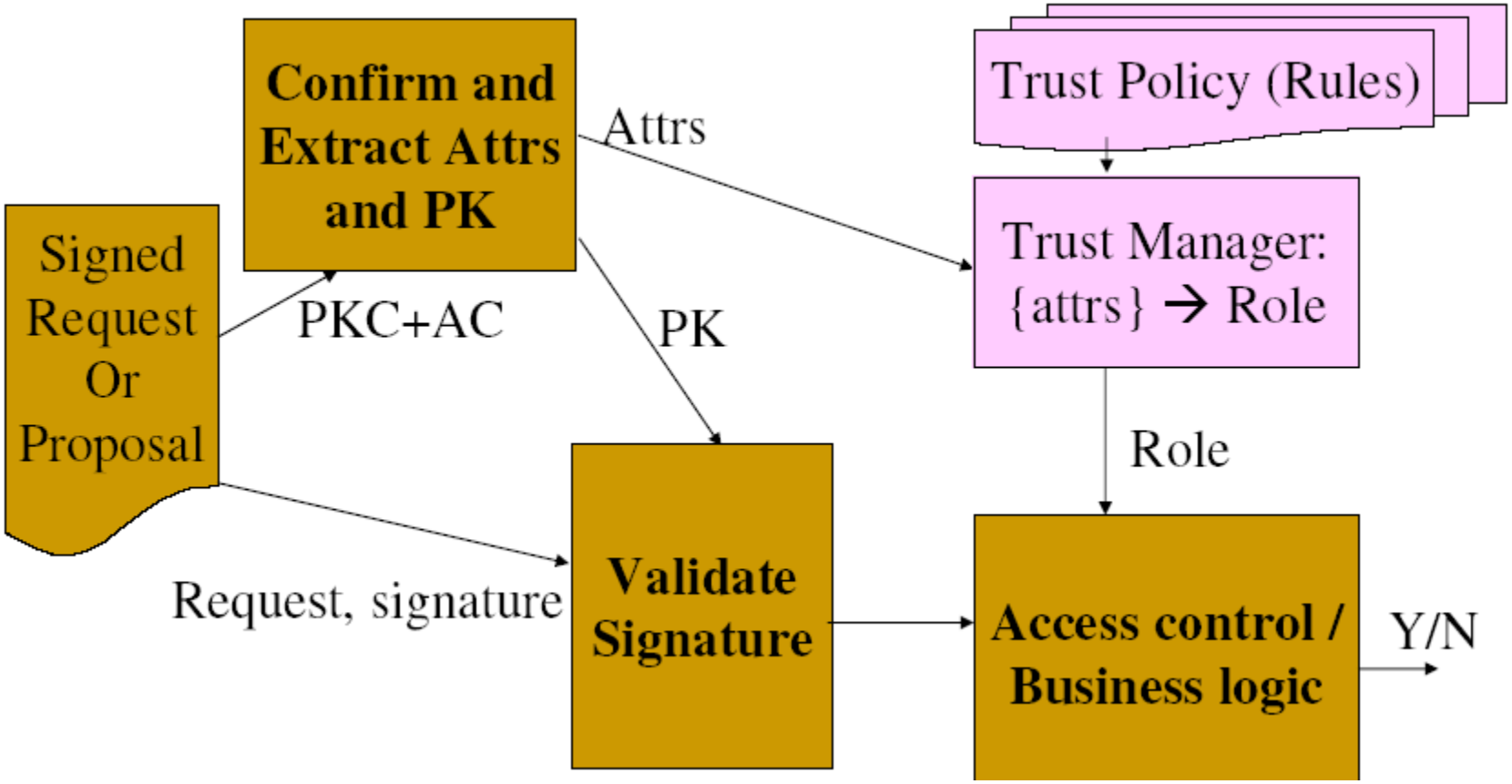
Controlo de acessos "*Role-based*" com AC

Caso I: "*Role*" como atributo



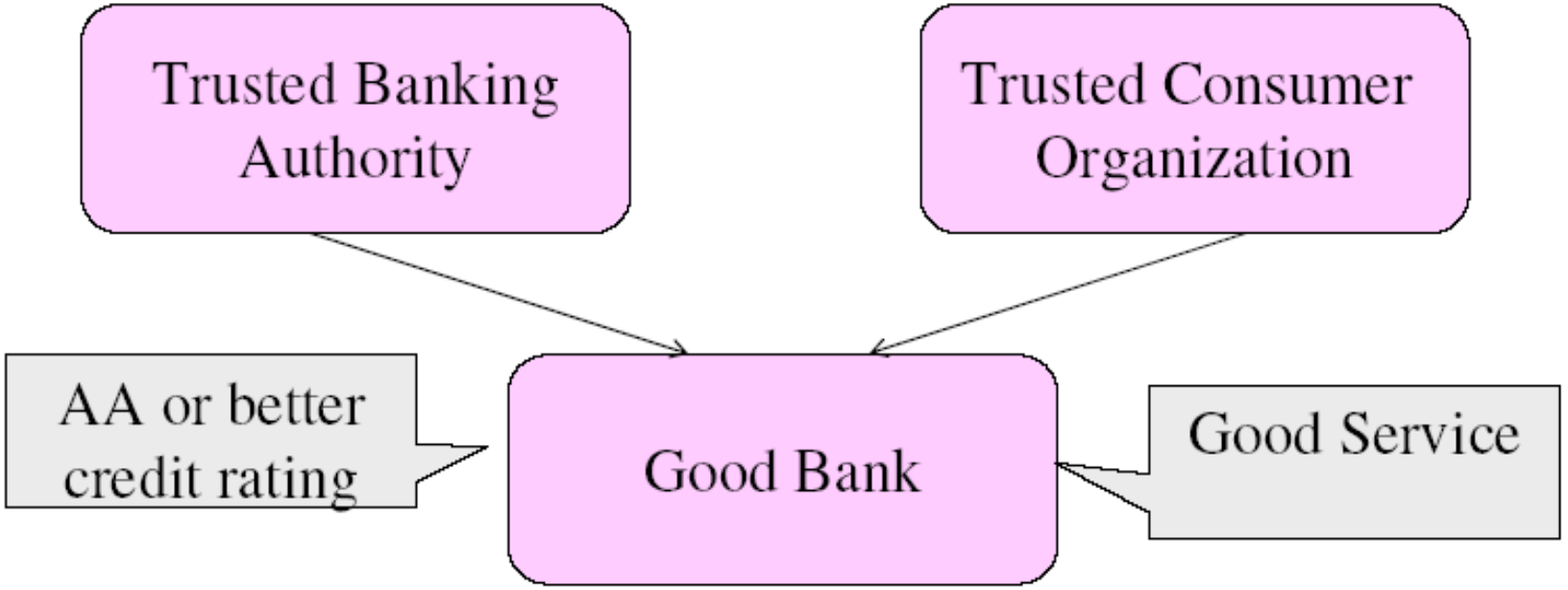
Controlo de acessos "*Role-based*" com AC

Caso II: "*Role*" derivado dos atributos

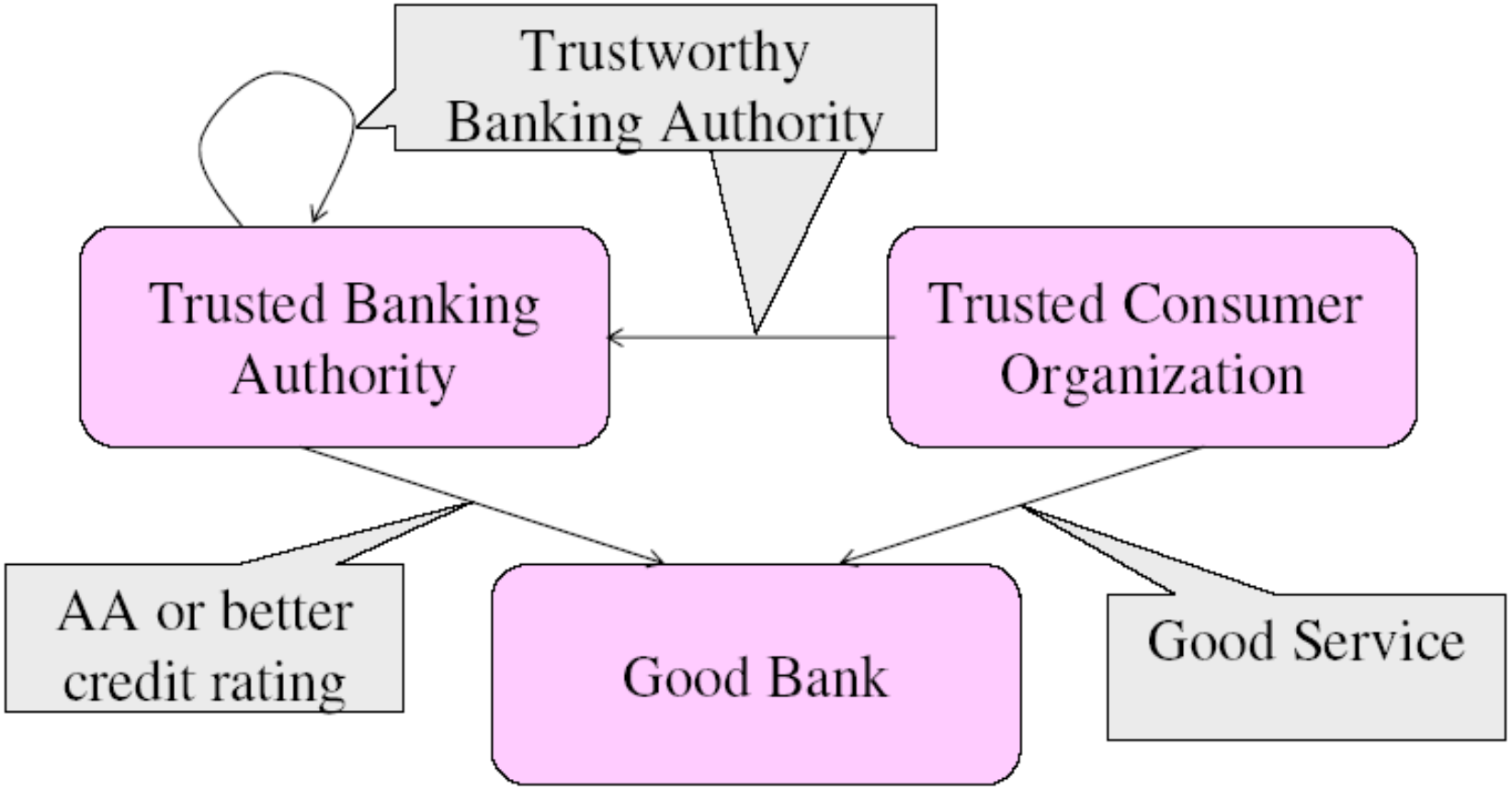


Política de confiança (*trust*) e regras

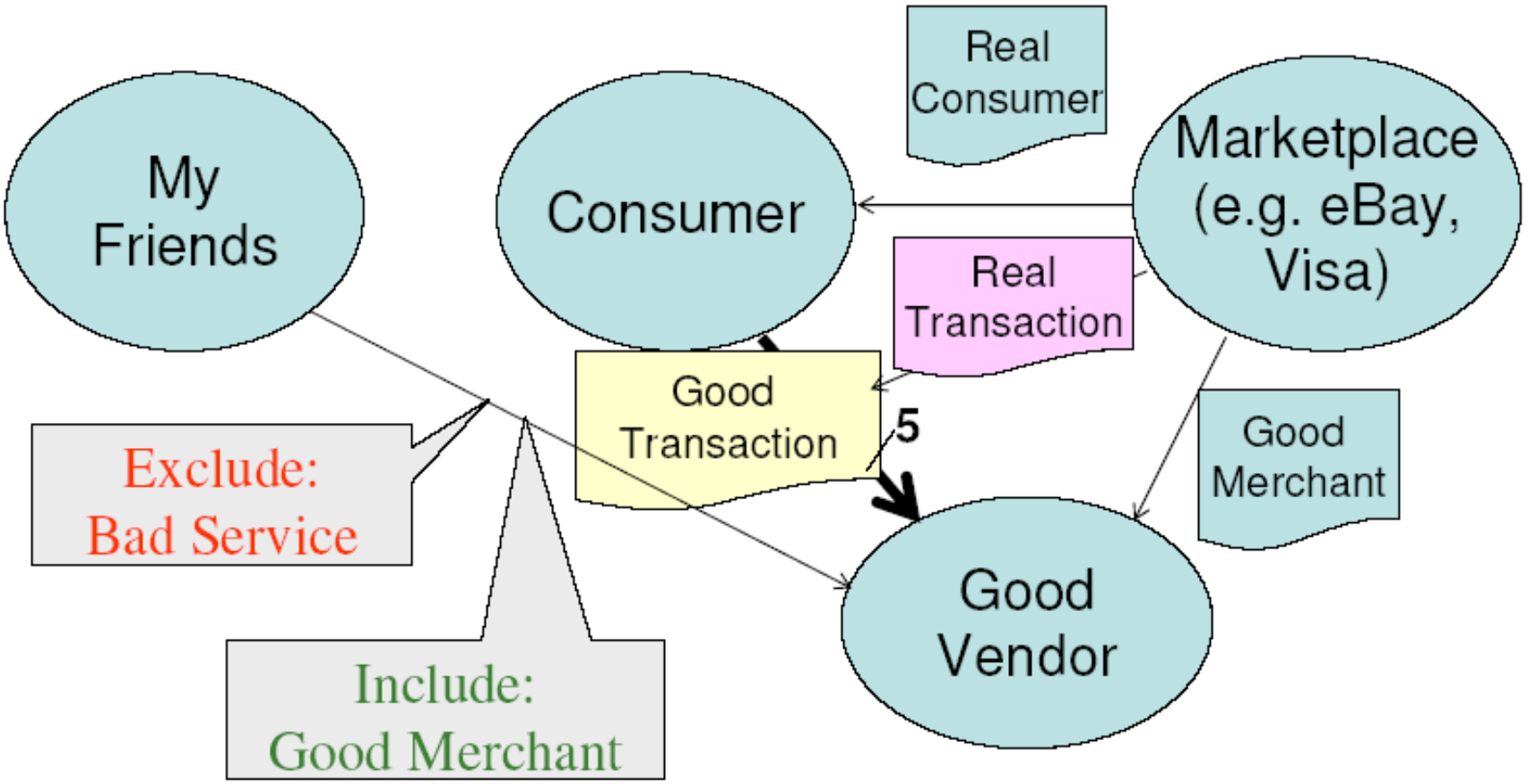
- Como definir e avaliar uma política de segurança?
- Muitos trabalhos sobre o assunto, sem respostas claras, assuntos em aberto
- Alguns exemplos:



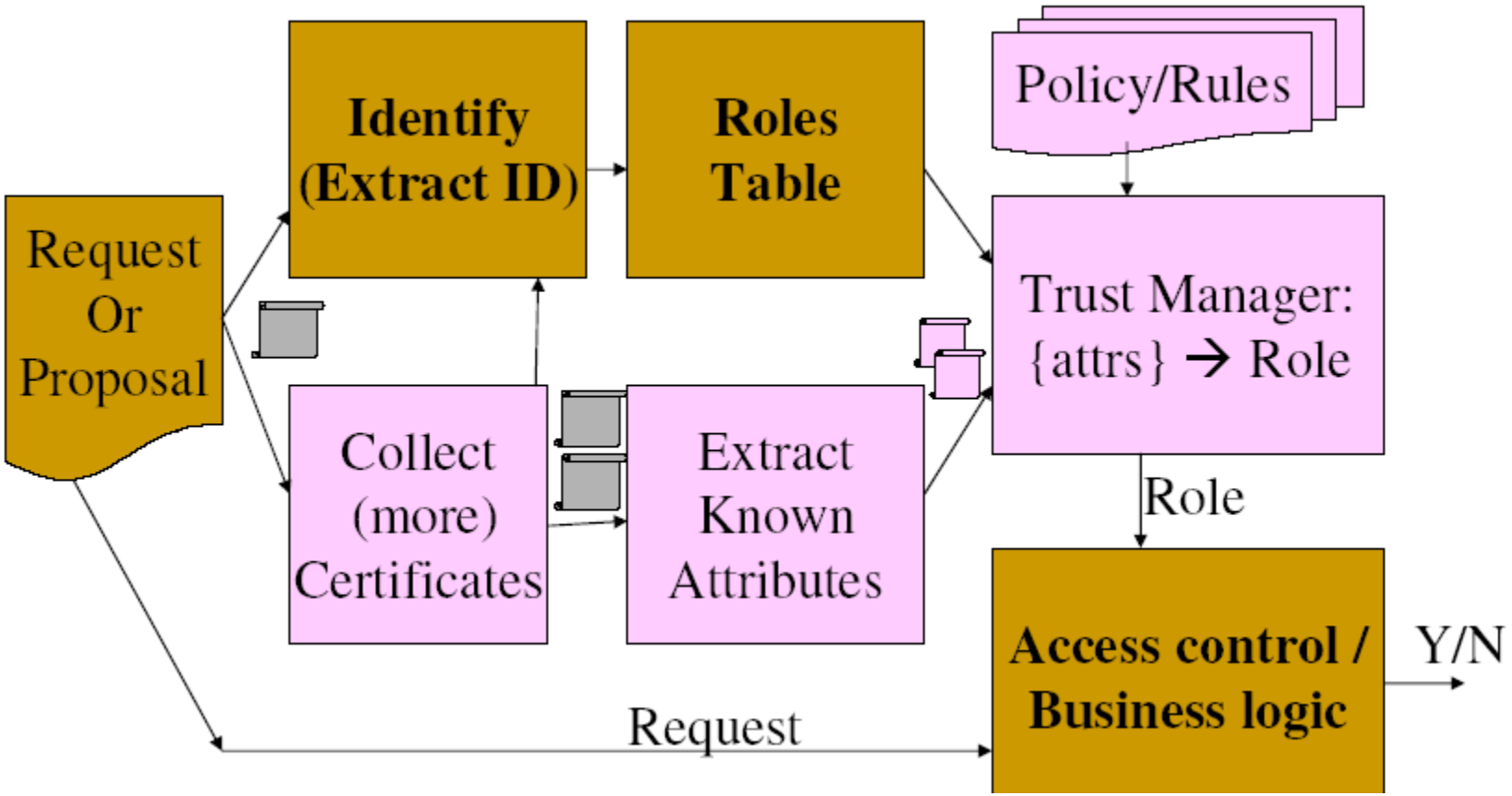
Autoridades dinâmicas



Exemplo de certificado multi-partes



PKC+AC+Collect+Extract+...



Privacidade das credenciais

- Outra motivação para os atributos de certificados:
  - Nem todas as partes confiantes necessitam de todos os atributos
  - Não enviar -> melhora a privacidade e o certificado fica mais curto
- **Solução 1**: Enviar apenas os atributos do certificado necessários
- **Solução 2**: o Certificado contém apenas o *hash* dos atributos
  - $Cert_A = Sign_{CA.s}$(A.s, "Alice", h(A.grades), h(A.jpg))
  - Uso: Alice envia $Cert_A$, A.grades para se propor a um emprego
  - E: Alice envia $Cert_A$, A.jpg para o *chat*
  - Podemos utilizar este método para esconder o género?
    - Com $Sign_{CA.s}$(A.s, "Alice", h(A.gender), h(A.jpg))

Privacidade para pouca informação: Compromisso

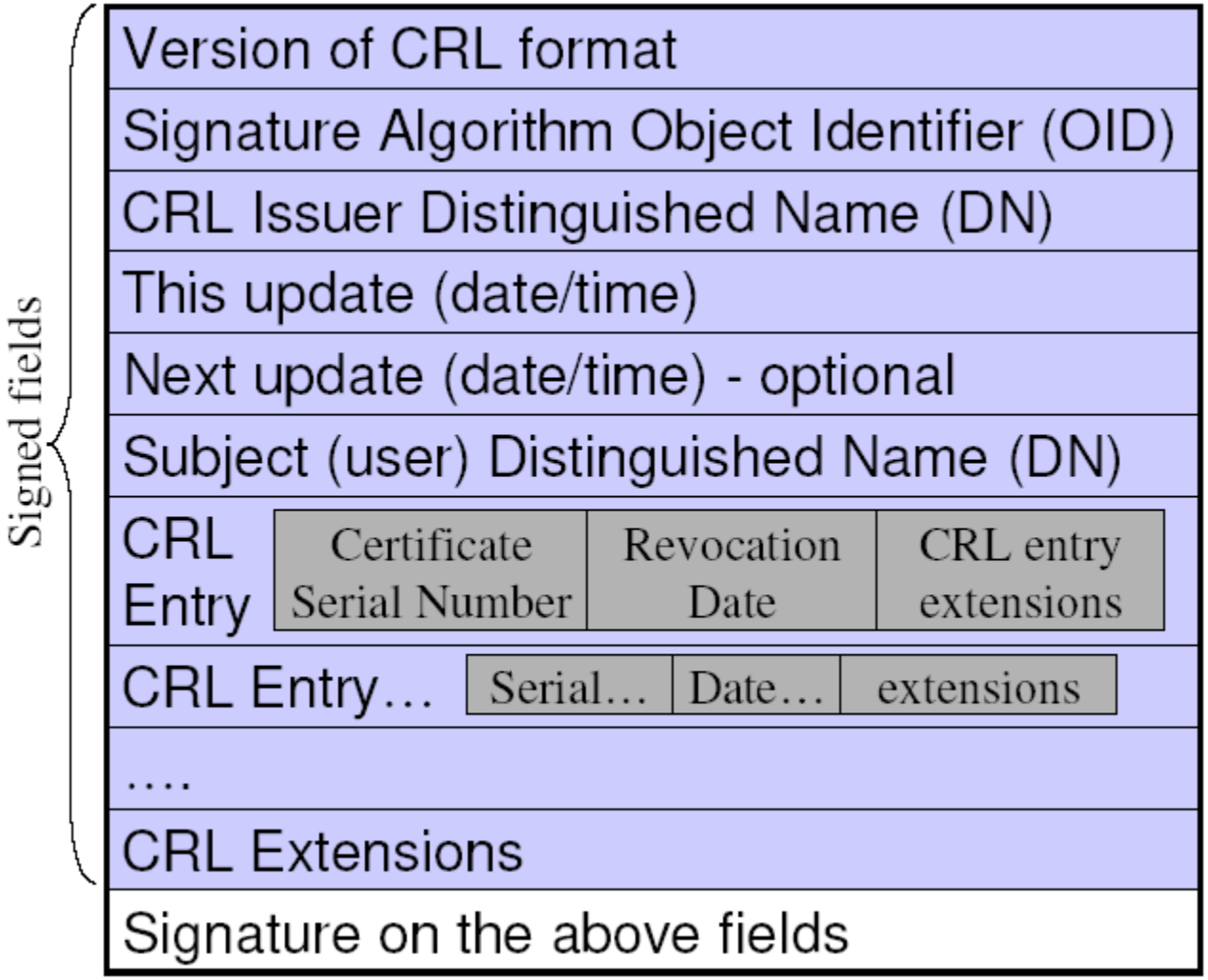
- Pode esconder até um bit, usando '*salt*'....
  - Nomeadamente; $Sign_{CA.s}$(A.s, "Alice", h(A.gender, salt))
  - Para fazer prova do género (*gender*): enviar $Cert_A$, A.gender, salt
  - Segurar o *hash*
- Este é um caso especial de esquema de compromisso
  - Funções: commit(m), dcommit(m), valid(m, c, d)
  - Tal que: valid(m, commit(m), decommit(m))=True
  - E duas (em competição) propriedades de segurança:
    - Privacidade: commit(m) não revela informação
    - Não encontra c, d, d', m, m' s.t valid (c, d, m), valid (c, d', m'), m≠m'
  - Muitas vezes o compromisso é definido com uma chave pública

Revogação de certificados

- Razões para a revogação de certificados:
  - A chave secreta do utilizador está compromentida
  - O utilizador já não é certificado por este CA
  - O certificado do CA está comprometido
  - Substituido
  - Cessação – já não é necessário

- Como informar as partes confiantes?
  - Não informar – esperar pelo fim (pequeno?) do período de validade
  - Distribuir CRL
  - Perguntar – *Online Certificate Status Protocol* (OCSP)

Formato dos CRL X.509



Certificados - São difíceis de revogar

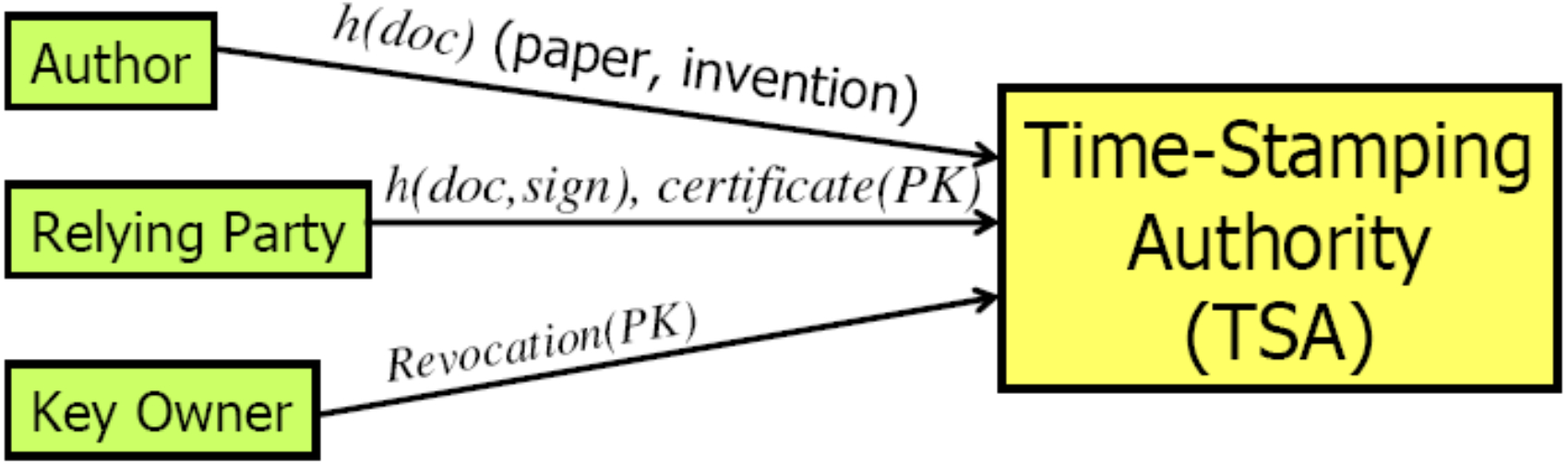
- Se as CRLs contiverem todos os certificados revogados (que não tenham expirado) ... podem tornar-se enormes!
- As CRLs não são imediatas
  - Quem é responsável até a CRL ser distribuida?
  - Qual é o impacto na não-repudiação?
- Soluções:
  - *Online Certificate Status Protocol* (OCSP)
  - Esquemas de CRLs mais eficientes (usualmente extensões às CRLs)
    - Ponto de distribuição de CRLs – separar os CRLs por vários CRLs
    - *Authorities Revocation List* (ARL): lista apenas CAs revogados
    - *Delta* CRLs - apenas novas revogações desde o último CRL 'base'
    - Certificate Revogation Tree
  - Validade curta dos certificados – sem CRLs

Prevenir a invalidação de assinaturas

- Problema: Chave pública revogada ... as assinaturas tornam-se inválidas?
  - Se não... a revogação não previne a assinatura com chave roubada
  - Se sim... o assinante pode revogar a chave (afirmar que foi exposta) para negar a assinatura
  - Solução justa: as assinaturas realizadas antes da revogação mantêm-se válidas
- **Solução 1**: *Time-stamping* da assinatura e da revogação
  - A assinatura é válida se a data/hora da assinatura < data/hora da revogação
  - Evidência por terceira parte (*timestamp*) do momento da assinatura/ revogação

Time-stamping de assinaturas e revogações

- **Objectivo**: Prova não-repudiada da data de criação do documento
- Prova que o documento/assinatura/revogação existia na data
- Se a assinatura do contracto existia (foi validada) antes da chave pública ser revogada, então o contrato permanece válido!
- *Timestamp* assinado pela **Time-Stamping Authority (TSA)**
- *Hash* do documento para prevenir a confidencialidade



Prevenir a invalidação de assinaturas

- Problema: chave pública revogada...
- ...
- **Solução 2**: Validade curta e limitada das chaves
  - Dividir tempo em períodos, por exemplo dia/mês
  - Chaves diferentes para cada período i: Priv[i], Pub[i]
  - Exposição de Priv[i] não torna possível assinar com Priv[i'], i'<i
  - -> mesmo que a chave do período i seja revogada, as assinaturas anteriores continuam válidas

Certificados de curto prazo

- **Ideia**: Certificados de curto prazo, de maneira a não ser necessário revogá-los
- **Preocupação:** Sobrecarga de assinar tantos certificados a cada curto período
- **Soluções:**
  - Abranger muitos certificados com uma assinatura: árvore de *hash*

Usos para os Certificados

- Qualquer um pode obter o certificado digital de qualquer outra pessoa com quem pretenda manter uma comunicação segura, *quer tenha tido ou não uma relação com ela anteriormente*

- O CA atesta que a chave pública no certificado é mesmo do sujeito referido no certificado. Sabe-se com quem se pretende comunicar! (ou não?)

- Isto facilita muito o comércio electrónico

Revisão sobre CAs

- Como se pode confiar num CA?
- Quem garante a "*honorabilidade*" do topo da hierárquia?
- Quais as responsabilidades envolvidas?
- Garantem *realmente* que se fica a conhecer quem é quem no mundo digital?

- A hirárquia de CAs é designada por Infraestrutura de Chave Pública (***Public Key Infrastructure - PKI***)

Revisão sobre os CA

- Os designados certificados de raíz são pré-carregados nos clientes Web

- Se lhe oferecerem um certificado para o qual possuí um certificado de raíz na sua máquina então o certificado é testado e é-lhe dito se houver algum problema, tal como o certificado ter expirado, etc.

- Escolherá então o que fazer

Confiança?

- Dado que os certificados são pré-carregados pelo cliente Web, está a confiar na Microsoft, na Netscape ou noutro

- Pode-se alterar um conjunto de certificados em qualquer cliente *Web*, *desde que esse cliente não tenha sido alterado para prevenir isso*

- É comum colocarem-se *browsers* alterados nas máquinas de maneira a prevenir que os utilizadores tomem liberdades com os parâmetros do sistema, adicção de certificados, etc.

Juntando tudo

- Até agora vimos como é que as peças são criadas para efectuarem determinadas funções criptográficas.

- Como é que estas coisas podem ser integradas para criarem um sistema que faça alguma coisa útil?

- Um desses sistemas, que se irá analisar, é o PGP.

Conclusão

- Os certificados de chave pública fazem a ligação entre a chave pública e (os atributos do) seu dono
- O X.509 foi focado nos PKCs de identidade
- PKCs de identidade são naturais – estamos habituados a cartões de identificação
- Mas mesmo o X.509 adiciona atributos não relacionados com a identidade:
  - Em extensões do PKC X.509
  - Em certificados de atributos