UNIVERSIDADE DE ÉVORA

Sistemas Operativos II

Segurança e SD
(parte 3)

Algoritmos de Autenticação

- **autenticação** de um ou dois interlocutores/peers/participantes

Algoritmos

- Needham-Schroeder
- Kerberos

- Baseados em *Tickets* e *challenges* (desafios)
  - *Ticket*: mensagem encriptada pelo Servidor de Autenticação com uma chave do *principal*. Contém a identidade do interlocutor e a chave secreta gerada para usar na sessão.
  - *Challenge*: transmissão de informação (ticket) de forma a que só o verdadeiro destinatário possa ler. O processo é encarado como um desafio, porque:
    - Vencer o desafio é conseguir decifrar a informação (e continuar o processo)
    - os atacantes são afastados/eliminados e não conseguem avançar

Algoritmos de Autenticação: Needham–Schroeder

- 1978, com o surgimento dos *network file services*

há um servidor de autenticação, S, que conhece a identificação e a chave secreta de cada *principal* no sistema

Essa chave secreta é conhecida apenas pelo *principal* e pelo servidor **S**, servindo para autenticação do *principal* junto do servidor e para cifrar mensagens entre os mesmos

Nonce: valor inteiro que se adiciona a uma mensagem para demonstrar que é (ou que não é) recente

Algoritmos de Autenticação: Needham–Schroeder

Header	*Message*	*Notes*
1. A->S:	A, B, N_A	A requests S to supply a key for communication with B.
2. S->A:	$\{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_B}\}_{K_A}$	S returns a message encrypted in A's secret key, containing a newly generated key K_{AB} *(session key)* and a 'ticket' encrypted in B's secret key. The **nonce** N_A demonstrates that the message was sent in response to the preceding one. A believes that S sent the message because only S knows A's secret key.
3. A->B:	$\{K_{AB}, A\}_{K_B}$	A sends the 'ticket' to B.
4. B->A:	$\{N_B\}_{K_{AB}}$	B decrypts the ticket and uses the new key K_{AB} to encrypt another nonce N_B.
5. A->B:	$\{N_B - 1\}_{K_{AB}}$	A demonstrates to B that it was the sender of the previous message by returning an agreed transformation of N_B.

dificuldade: S ter conhecimento prévio das chaves de A e B

vulnerabilidade: B não sabe se (3) é recente ou um *replay*

- **solução**: usar um timestamp t à mensagem {Kab,A,t}Kb ; assim B pode verificar se a mensagem é atual (kerberos)

Algoritmos de Autenticação: Kerberos

- MIT, fim dos anos 80
- Objetos envolvidos
  - *Ticket*: para um cliente apresentar a um servidor (TGS ou de um serviço), prova que o cliente fez uma autenticação recente junto do Kerberos.

    Inclui identificação do **cliente**, data e hora de expiração (ou **periodo de validade**) e uma ***session key*** para usar entre cliente e servidor.

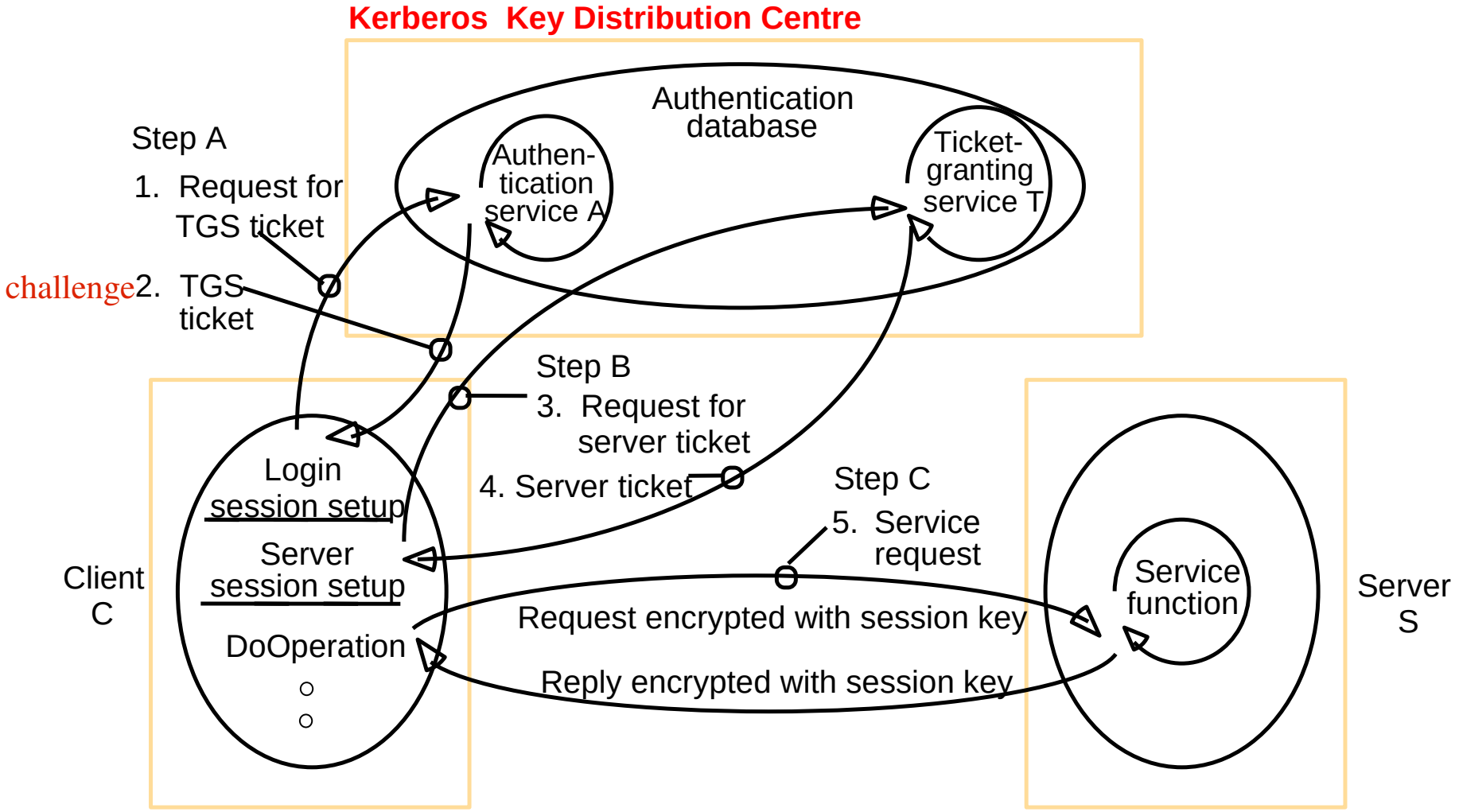
    O ticket é **encriptado** com a chave secreta do interlocutor do cliente (o próximo servidor a que se ligará)

    O *Ticket* pode ter uma duração de várias horas... para diversas interações cliente/servidor

  - *Token* de Autenticação: construída pelo **cliente** para apresentar a um **servidor** e provar a sua identidade. Pode ser usada uma só vez. Inclui identificador do cliente e um timestamp, tudo encriptado com a *session key*

  - *Session Key*/Chave de Sessão: chave secreta gerada pelo Kerberos para um cliente comunicar (com encriptação) com um servidor, e também para encriptar as Tokens de Autenticação

Algoritmos de Autenticação: Kerberos



Algoritmos de Autenticação: Kerberos

- 1- o primeiro nível de autenticação (AS) consiste numa verificação segura de utilizador/password. O cliente pede ao servidor de autenticação A que lhe forneça um *Ticket* para a comunicação com o servidor TGS

- 2- Em resposta obtém um ticket e a chave de sessão para comunicar com o TGS, tudo encriptado com a sua chave secreta.
  - Esta mensagem inclui um *Nonce* encriptado com Kc, o que significa que é proveniente do servidor.
  - O Ticket está encriptado com a chave do servidor TGS, contendo:
    - identidades C e TGS
    - timestamps de validade
    - chave de sessão entre C e TGS, Kct

Algoritmos de Autenticação: Kerberos

- 3- C comunica com servidor TGS, enviando
  - um Token de Autenticação, encriptado com chave secreta Kct
  - o ticket para TGS
  - a identificação do servidor S para o qual pretende um Ticket
  - nounce

- 4- TGS verifica o ticket apresentado. Se é válido então gera uma chave de sessão Kcs e devolve:
  - chave de sessão Kcs e nounce, encriptados com Kct
  - ticket* para S (*encriptado com chave secreta do servidor S, Ks)

Algoritmos de Autenticação: Kerberos

- 5- C comunica com o servidor S (do serviço pretendido), enviando:
  - token de autenticação, cifrado com a chave de sessão secreta Kcs
  - ticket para S, cifrado com Ks (secreta do servidor S)
  - nonce, encriptado com Kcs
  - o pedido ao servidor
    - (encriptado com Kcs quando se requer confidencialidade)

- 6- S responde. A resposta pode incluir:
  - o nonce N, encriptado com Kcs
    - (para autenticação do servidor, opcional)

Algoritmos de Autenticação: Kerberos

- muito semelhante ao protocolo Needham and Schroeder, com a adição de timestamps (inteiros para data e hora para os nounces), para:
  - prevenir *message replaying* (reenvio de mensagens antigas interceptadas na rede) ou aproveitamento de *Tickets* anteriores encontrados em memória...
  - atribuir um limite temporal (*lifetime*) aos *tickets*, facilitando a revogação de direitos a um utilizador

Autenticação para pessoas nos dias de hoje

- A autenticação por login+password é insuficiente em sistemas críticos
  - Se A sabe a password de B, poderá aceder ao seu perfil…
- Autenticação Multifactor (*Multi-factor authentication* (MFA))
  - Combinar password com outro elemento diferente, em cada autenticação
  - Outro elemento: código enviado por SMS, impressão digital (…)

- https://www.nist.gov/itl/applied-cybersecurity/tig/back-basics-multi-factor-authentication
- https://www.google.com/landing/2step/index.html

WiFi Protected Access (WPA)

- Protocolo de segurança para rede sem fios
  - Evita ataques do tipo replay
  - Pacotes têm um contador (nonce) que facilita deteção de duplicados
- Temporal Key Integrity Protocol (TKIP)
  - Protocolo de cifra baseado na troca frequente de chave
    - WEP tinha chaves fixas
    - Prevê bits de verificação da integridade da mensagem
      - Message Integrity Check (MIC/MAC) de 64 bits
- WiFi Protected Access II (WPA2)
  - Evolução de WPA
  - Counter Cipher Mode Protocol
    - Mais seguro que o TKIP de WPA
    - Block Cipher
    - AES e chaves de 128 bits

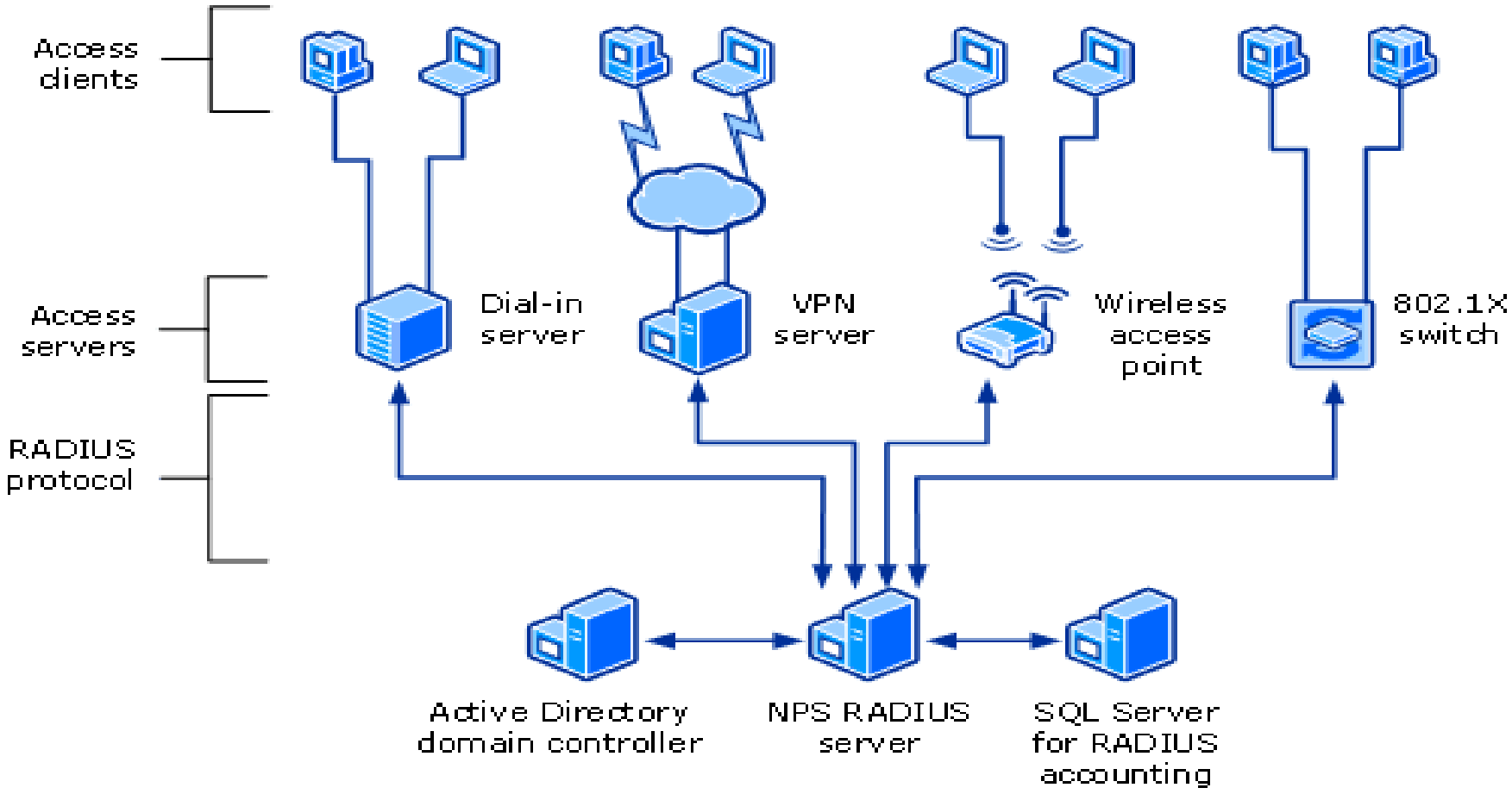
MSCHAP-v2

- Microsoft Challenge-Handshake Authentication Protocol, version 2
  - Usado em autenticação RADIUS, WPA-Enterprise e VPNs
  - Autenticação de ambas as partes na comunicação
  - Chaves diferentes para a cifra no envio e recepção de dados
    - Chaves geradas com base na password do cliente e num valor arbitrário
      - Em cada sessão a chave é diferente
  - Processo de autenticação
    - <- Authenticator Challenge
    - Peer Response/Challenge ->
    - <- Success/Authenticator Response
    - If authenticator Response verification succeeds, call continue

http://tools.ietf.org/html/rfc2759

RADIUS

- Remote Authentication Dial In User Service (RADIUS)



Assinaturas Digitais

- Dificuldades
  - documentos digitais
    - fáceis de copiar e modificar
  - o emissor pode deliberadamente divulgar a chave privada e alegar que não é o autor da mensagem (repúdio)

- Garantias desejáveis:
  - autenticidade de um documento (integridade)
  - impossibilidade de forjar uma assinatura (autenticação)
  - não repúdio

Assinaturas Digitais

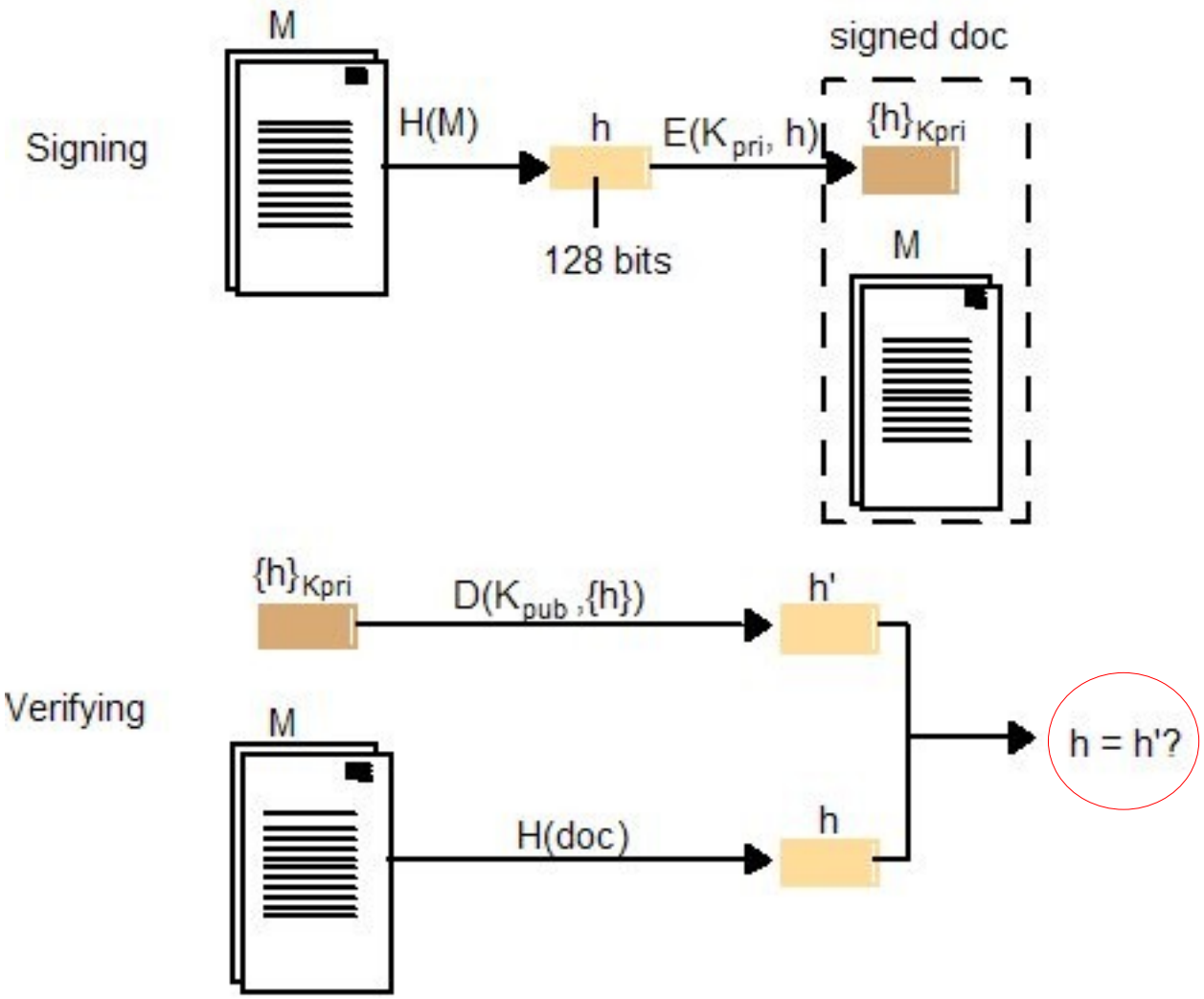
- M pode ser assinado por A
  - encriptar uma cópia de M com a chave Ka
  - anexar o ciphertext e um identificador de A a M
  - documento M assinado: $M,A,[M]_{Ka}$

- Verificação (de $M,A,[M]_{Ka}$) permite verificar
  - a origem (existia a chave correspondente, secreta ou privada)
  - o conteúdo não foi alterado

- A verificação da assinatura depende da criptografia usada:
  - chave secreta: só pode ser verificada por quem possuir a chave secreta
  - chave pública: verificada por qualquer principal com a chave pública do *signer*
    - mais usadas

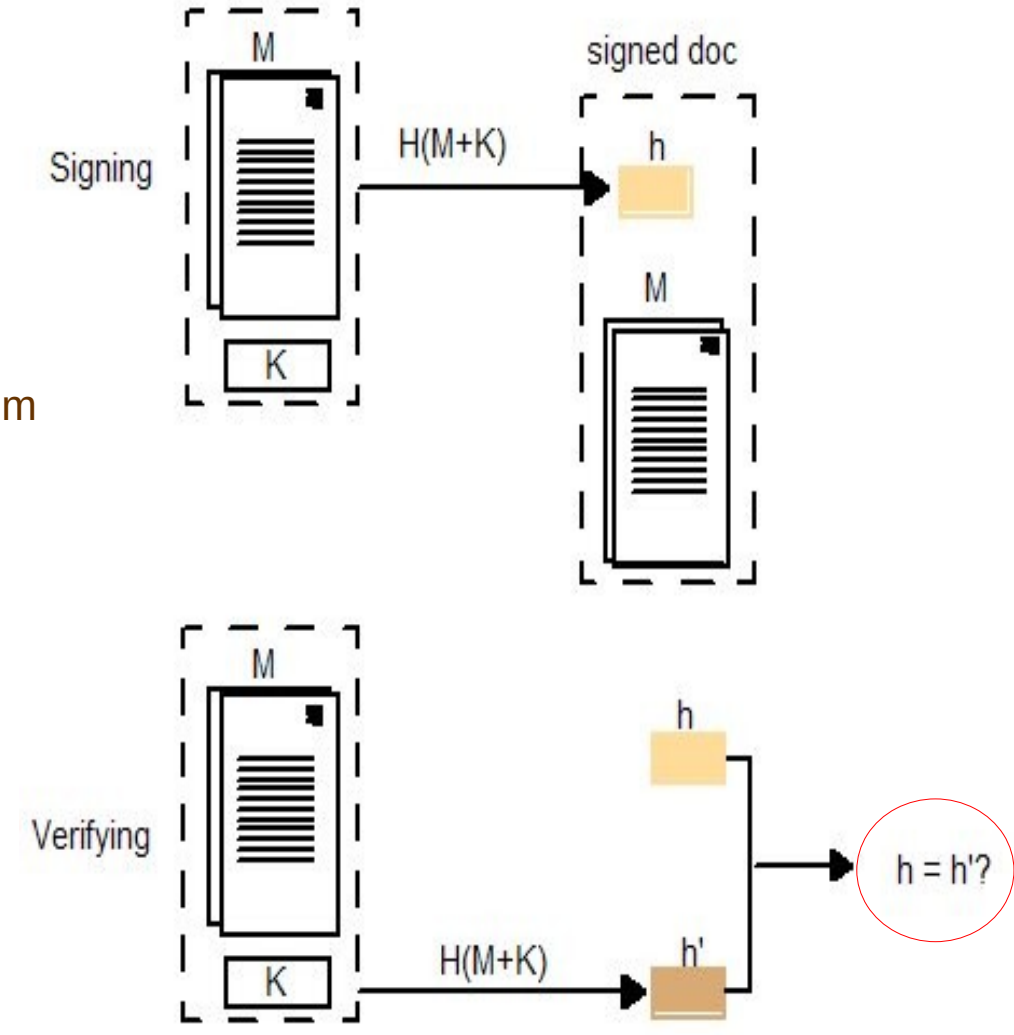
Assinatura Digital com Chave Pública

- Vantagens
  - simplicidade
  - dispensa comunicação prévia entre os intervenientes
- A encriptação é feita com a chave privada
  - o objetivo <u>não é a confidencialidade</u> da mensagem



Assinatura com Chave Secreta - **MAC**

- Algoritmo simétrico de encriptação
- Dificuldades
  - requer processo seguro para transmitir a chave secreta até ao *verifier*
  - pode ser necessário verificar a assinatura numa fase posterior à sua criação e por *verifiers* que o *signer* não conhece e a quem não dá a chave
  - a partilha da chave secreta traz fraqueza: um detentor da chave pode forjar a assinatura do signer original
- Vantagem: performance (<u>não há encriptação</u>)
  - funções de hash são 3 a 10 x mais rápido que alg. simétricos



Assinaturas Digitais com Chave Pública e MACs

- AD de chave pública são uma solução mais conveniente na maioria dos casos

- Exceção:
  - utilização de um canal seguro para transmitir mensagens não encriptadas mas para as quais é necessário verificar a autenticidade
  - canal seguro permite a transmissão de chave secreta para uso nestas AD "de baixo custo computacional" - *Message Authentication Codes** (**MAC**)
    - Também referidos como MIC (*Message Integrity Check*)

Assinaturas digitais de chave pública

- Exemplo: as assinaturas que fazemos com:
  - Cartão do Cidadão
  - Chave Móvel Digital (um serviço inovador de desmaterialização)

- Na prática: assinaturas da mesma pessoa, em cada opção acima, usarão pares de chaves diferentes, mas o relevante é a validade das mesmas
  - No CC usam a chave privada inerente ao CC
  - Na CMD, usam outra chave privada associada ao cidadão, mas na posse do estado, alojada centralmente no serviço, e usada mediante autenticação

https://www.autenticacao.gov.pt/cc-assinatura
https://www.autenticacao.gov.pt/cmd-assinatura

Certificados Digitais

- certificado de chave pública para o Banco de Bob

1. *Certificate type*:	Public key
2. *Name*:	Bob's Bank
3. *Public key*:	K_{Bpub}
4. *Certifying authority*:	Fred – The Bankers Federation
5. *Signature*:	$\{Digest(field\ 2 + field\ 3)\}_{K_{Fpriv}}$

Formato do Certificado X509

Subject	Distinguished Name, Public Key
Issuer	Distinguished Name, Signature
Period of validity	Not Before Date, Not After Date
Administrative information	Version, Serial Number
Extended Information	

- usados em processos de autenticação e ligações SSL
- verificação da autenticidade um certificado:
  - obter a chave pública do issuer (e tem de se acreditar no issuer)
  - validar a assinatura digital do certificado (1)
- outras validações
  - Validade (2)
  - Tipo de utilização (3)
  - Listas de Revogação (4)

Considerações sobre Segurança

- Desempenho de algoritmos de Encriptação Simétrica e Digest

	Key size/hash size (bits)	*PRB optimized 90 MHz Pentium 1 (Mbytes/s)*	*Crypto++ 2.1 GHz Pentium 4 (Mbytes/s)*
TEA	128	–	23.801
DES	56	2.113	21.340
Triple-DES	112	0.775	9.848
IDEA	128	1.219	18.963
AES	128	–	61.010
AES	192	–	53.145
AES	256	–	48.229
MD5	128	17.025	216.674
SHA-1	160	–	67.977

C

Considerações sobre Segurança

- o tamanho da chave influencia o tempo/custo computacional necessário para um ataque de força bruta

- a verdadeira força está no algoritmo criptográfico, no modo como ofusca o *plaintext*

Referências

- Informações diversas e curiosidades
  - http://www.openssl.org
  - http://www.insecure.org
  - http://csrc.nist.gov/nissc/
  - Listas: BugTrack, VulnWatch...

- Referência Recomendada para aprofundar conhecimentos:

  Applied Cryptography
  Second Edition
  Bruce Schneier
  John Wiley & Sons, 1996
  ISBN 0-471-11709-9

- **Nota**: os ataques com base em ‘engenharia social’ são cada vez mais frequentes