Administração de Sistemas (ASIST)

Autenticação
KERBEROS, GSS-API, SASL e TLS

Outubro de 2014

KERBEROS

O Kerberos foi desenvolvido pelo MIT (“Massachusetts Institute of Technology”) para uso interno, a versão 4 foi disponibilizada para uso externo em 1980 a actual versão 5 está documentada da RFC4120. O nome provém do ser da mitologia grega Cérbero.

Trata-se de um sistema desenvolvido com o objectivo de garantir a segurança num ambiente de rede potencialmente inseguro garantido a autenticação dos intervenientes e a confidencialidade das trocas de dados no contexto das comunicações segundo o modelo cliente-servidor.

O Kerberos foi desenvolvido usando exclusivamente cifras simétricas, mas actualmente pode também suportar cifras assimétricas (chave pública).

Na sua essência é um sistema de distribuição de chaves simétricas de sessão temporárias. Para o conseguir baseia-se em chaves simétricas pré-partilhadas que garantem a autenticação, confidencialidade e integridade.

O Kerberos 5 está disponível para Linux e outros sistemas operativos Unix. Com a introdução do “Active Directory” no Windows 2000, passou a estar integrado nos sistemas operativos da Microsoft.

KERBEROS – Componentes

O Kerberos é um sistema de distribuição de chaves centralizado, as aplicações (clientes e servidores) e os utilizadores que usam este sistema são designados “principals”.

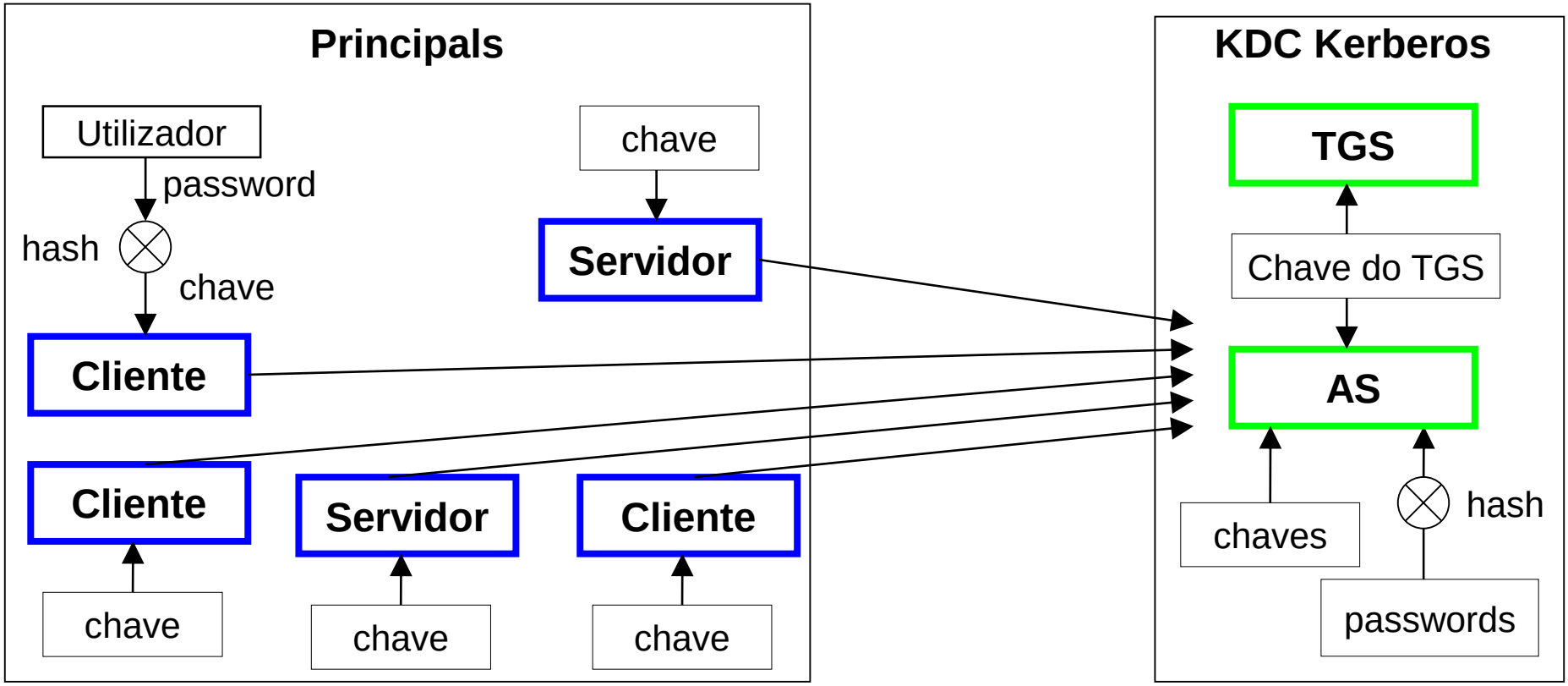
O KDC (“Key Distribution Center”) do Kerberos é constituído por dois serviços paralelos:

AS (“Authentication Server”) – responsável fornecer aos “principal” um elemento que lhes permite provar a sua identidade junto do TGS. Essa prova é conseguida através de uma chave pré-partilhada. Cada “principal” tem por configuração uma chave de cifra simétrica já partilhada com o AS. No caso de o “principal” ser um utilizador a chave é normalmente gerada por “hash” a partir da respectiva “password”.

TGS (“Ticket Granting Service”) – é o responsável por emitir chaves temporárias de sessão para comunicação entre os “principal”.

KERBEROS – Autenticação

A autenticação usa uma cifra simétrica baseada numa chave pré-partilhada entre cada "principal" e o AS. O TGS possui também uma chave pré-partilhada com o AS.



KERBEROS – diálogo com o AS

1º O “principal” envia uma mensagem ao AS. Esta mensagem não está encriptada, entre outra informação contém a sua identificação (nome do principal), o nome do serviço a que pretende aceder (nome do principal) e um valor data/hora.

- O AS procura na base de dados a chave correspondente ao “principal”, ou produz uma usando a “password” correspondente ao utilizador – (**Kpri**)

- O AS cria uma “chave de sessão” para comunicação entre o “principal” e o TGS – (**Kpri-tgs**)

- O AS cria um TGT (“Ticket Granting Ticket”) que contém: (nome-principal; endereço-rede-principal; tempo-de-validade; Kpri-tgs)

2º O AS responde ao “principal” enviando-lhe:

a) (Kpri-tgs encriptado com Kpri)

b) (TGT encriptado com Ktgs), Ktgs é a chave pré-partilhada entre o TGS e o AS

O “principal” recebe o TGT encriptado com Ktgs, algo que não consegue desencriptar pois apenas o TGS e o AS conhecem essa chave. O “principal” vai usar este TGT encriptado para provar a sua identidade junto do TGS. A encriptação com Ktgs dá ao TGS garantias da autenticidade do TGT. **O “principal” pode no entanto obter Kpri-tgs.**

KERBEROS – pré-autenticação

De destacar que no processo descrito de obtenção do TGT, não há qualquer autenticação. Qualquer intruso pode solicitar um TGT ao AS fazendo-se passar por um qualquer “principal”.

A verdadeira autenticação terá lugar quando o “principal” contactar o TGS e lhe enviar o TGT, esse diálogo apenas será possível para quem possuir Kpri-tgs que por sua vez exige Kpri.

Opcionalmente os “principal” podem realizar a “pré-autenticação” que reforça a segurança. Consiste na encriptação com Kpri do valor data/hora que é enviado pelo “principal” no primeiro contacto com o AS.

Quando o AS recebe o pedido procura a chave desse “principal” das suas bases de dados e desencripta o valor data/hora, se resultar numa data/hora válida então aceita o pedido e responde com as mensagens normais.

A vantagem da “pré-autenticação” é que o AS apenas fornece TGTs a “principals” já autenticados, mas sob o ponto de vista de segurança não é imprescindível. Alguns clientes Kerberos poderão não suportar pré-autenticação, nesse caso poderá ser necessário configurar o KDC para não a exigir.

KERBEROS – diálogo com o TGS

Concluído o diálogo com o AS, o "principal" possui o TGT encriptado com Ktgs que lhe foi enviado pelo AS. Possui também Kpri-tgs que lhe foi enviado pelo AS encriptado com Kpri e por isso pôde desencriptar (um intruso não conhece Kpri).

O "principal" envia uma mensagem ao TGS contendo: o TGT, a identificação do serviço que pretende usar ("principal") e o conjunto (identificação + data/hora) encriptado com Kpri-tgs

O TGS usa a sua chave (Ktgs) para desencriptar o TGT, obtendo assim Kpri-tgs. O TGS tem garantias de que o TGT foi emitido pelo AS e que Kpri-tgs é conhecida apenas pelo "principal" indicado. O TGT cria uma chave de sessão para comunicação entre os dois "principal" (Ksess). Cria um "Ticket" que contém: (nome-principal; endereço-rede-principal; tempo-de-validade; Ksess). O TGS obtém das bases de dados locais a chave do serviço pretendido (Kserv).

O TGS responde ao "principal" enviando-lhe:

a) (Ksess encriptado com Kpri-tgs)

b) (Ticket encriptado com Kserv)

KERBEROS – acesso ao serviço

O “ticket” obtido do TGS permite o acesso ao serviço de forma semelhante à que foi usada com o TGT para aceder ao TGS. Dentro do período de validade do TGT o “principal” pode solicitar novos “tickets” ao TGS para acesso a outros serviços.

O “principal” (cliente) envia uma mensagem ao serviço contendo: o “ticket” (que está encriptado com a chave do serviço - Kserv) e o conjunto (identificação + data/hora) encriptado com Ksess.

O serviço usa a sua chave (Kserv) para desencriptar o “ticket”, obtendo assim Ksess. O serviço tem garantias de que o “ticket” foi emitido pelo TGS e que Ksess é conhecida apenas pelo “principal” (cliente) indicado.

O serviço usa Ksess para desencriptar e verificar o conjunto (identificação+data/hora) recebido.

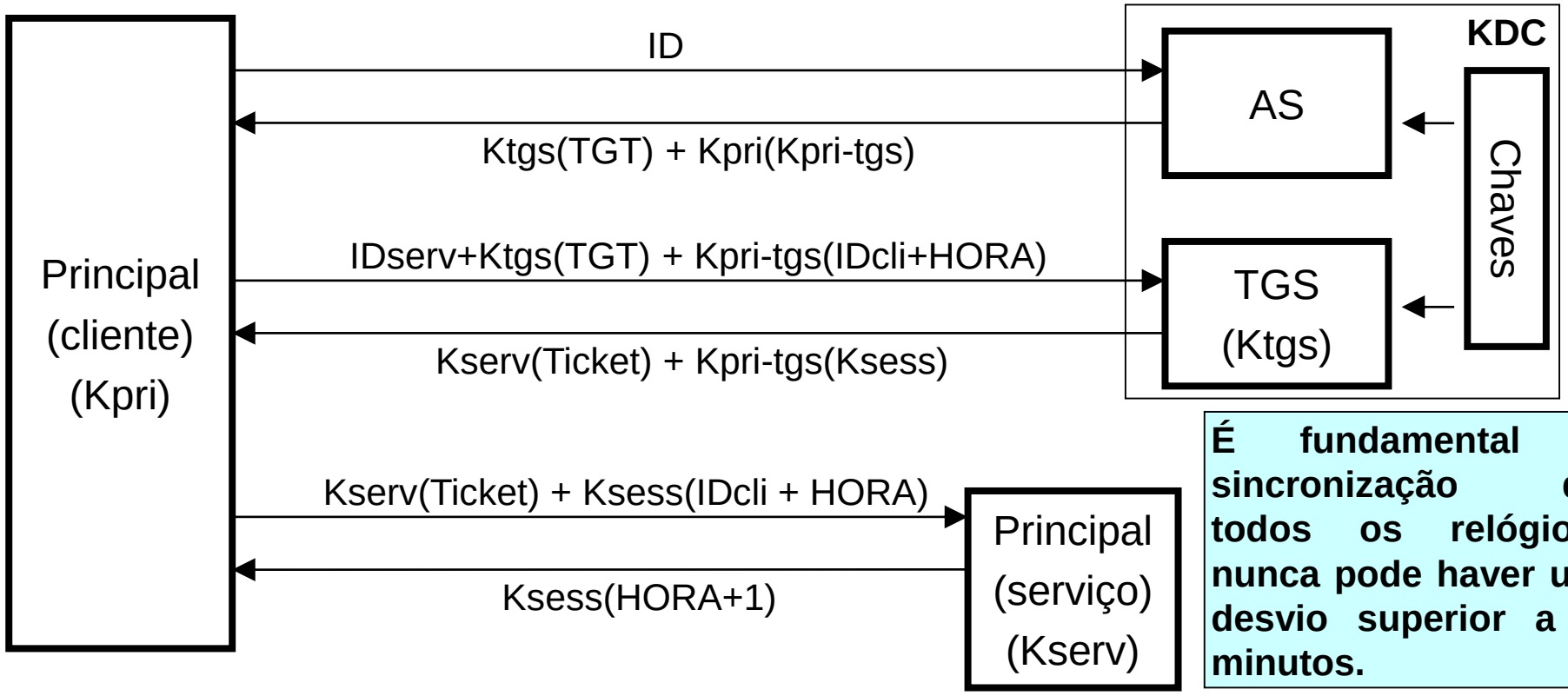
O serviço incrementa o valor data/hora recebido do “principal” e devolve-o encriptado com Ksess.

O “principal” desencripta a mensagem e verifica se o valor data/hora foi correctamente incrementado. Desta forma autentica o serviço.

A chave Ksess é usada nas comunicações seguintes entre cliente e serviço.

KERBEROS – Resumo de operação

Através de chaves pré-partilhadas entre os "principal" e o KDC garante-se a autenticidade de todos os intervenientes (Kpri, Kserv e Ktgs). Destaque-se que essas chaves nunca são partilhadas entre os "principal". As chaves partilhadas entre os "principals" são chaves temporárias de sessão. Os "tickets" garantem o transporte confidencial e autenticado das chaves de sessão até ao serviço.



É fundamental a sincronização de todos os relógios, nunca pode haver um desvio superior a 5 minutos.

KERBEROS – “realm”

No Kerberos um “realm” é um domínio de autenticação, é constituido por um KDC Kerberos e um conjunto de “principals” que possuem uma chave no KDC.

Vários “realms” podem coexistir, sendo possível operações entre “realms” vizinhos, para isso ser possível, cada “realm” deve ser identificado por um nome único, embora existam várias formas de definir o nome do “realm”, a mais divulgada é através da utilização do nome de domínio DNS local convertido para maiúsculas.

Por exemplo, para o nome de domínio “dei.isep.ipp.pt”, o nome de “realm” a utilizar seria “DEI.ISEP.IPP.PT”, isto é conveniente porque geralmente existe uma correspondência real entre o “realm” e um domínio DNS.

O nome de “realm” derivado do DNS é escrito em maiúsculas para evitar qualquer ambiguidade, ao contrário do que acontece com os nomes DNS, nos nomes do Kerberos letras maiúsculas e minúsculas possuem significados diferentes.

Nomes dos “principal”

Num “realm” cada principal é identificado através de um nome único, o nome de cada “principal” está sempre associado ao respetivo “realm”, por isso podem existir nomes iguais desde que pertençam a “realms” diferentes.

Os símbolos “/” e “@” são reservados pois possuem significados específicos, o símbolo “@” é usado para separar o nome do “principal” do nome do “realm”. O símbolo “/” é usado para separar componentes do nome do “principal”.

UTILIZADORES: quando o “principal” é um utilizador, o nome do principal é o nome do utilizador (“login”).

Exemplo: nome-do-utilizador@NOME-DO-REALM

SERVIÇOS: quando o “principal” é um serviço, o seu nome é habitualmente composto pelo nome do serviço seguido do nome do servidor (ambos em minúsculas).

Exemplo: ssh/megaserv@NOME-DO-REALM

O nome “krbtgt” é reservado para o TGS.

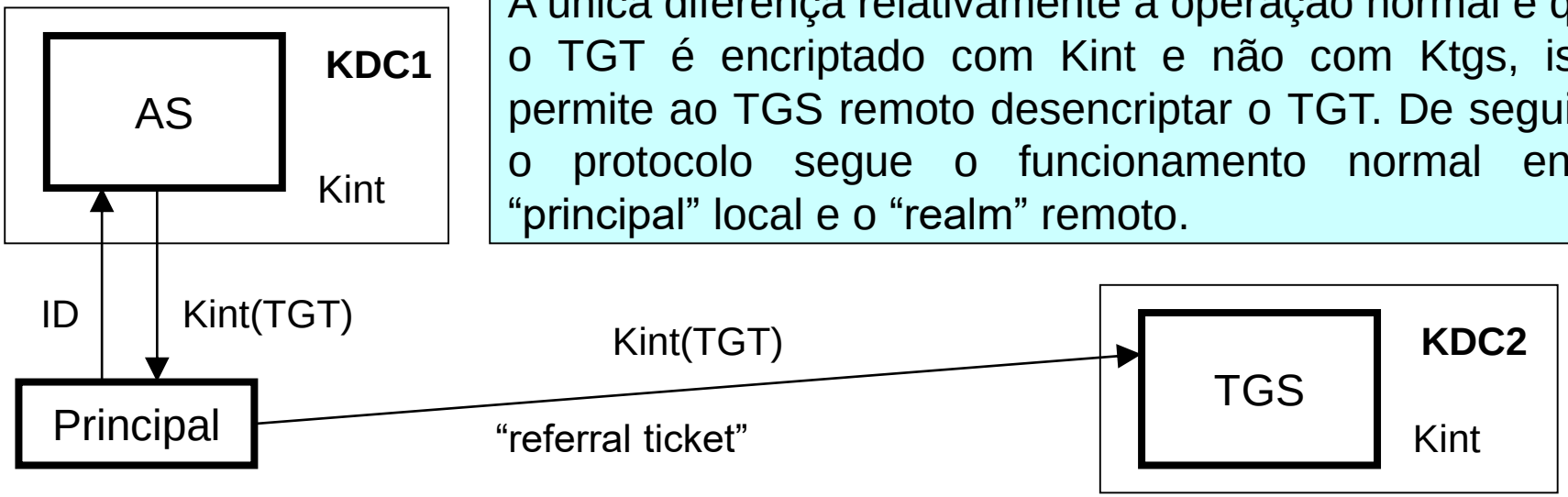
Confiança entre “realms”

É possível criar relações de confiança entre “realms” vizinhos, a grande vantagem é que isso permite a um “principal” de um “realm” aceder a um serviço no outro “realm”, mesmo sem possuir uma chave no “realm” onde se encontra o serviço.

Para criar uma relação de confiança, entre dois “realms” cada TGS é registado como “principal” no TGS vizinho, definindo-se uma chave pré-partilhada (Kint).

Os “principal” de um “realm” conseguem então obter um TGT para usar com o TGS do “realm” vizinho. Este TGT é normalmente designado “referral ticket”.

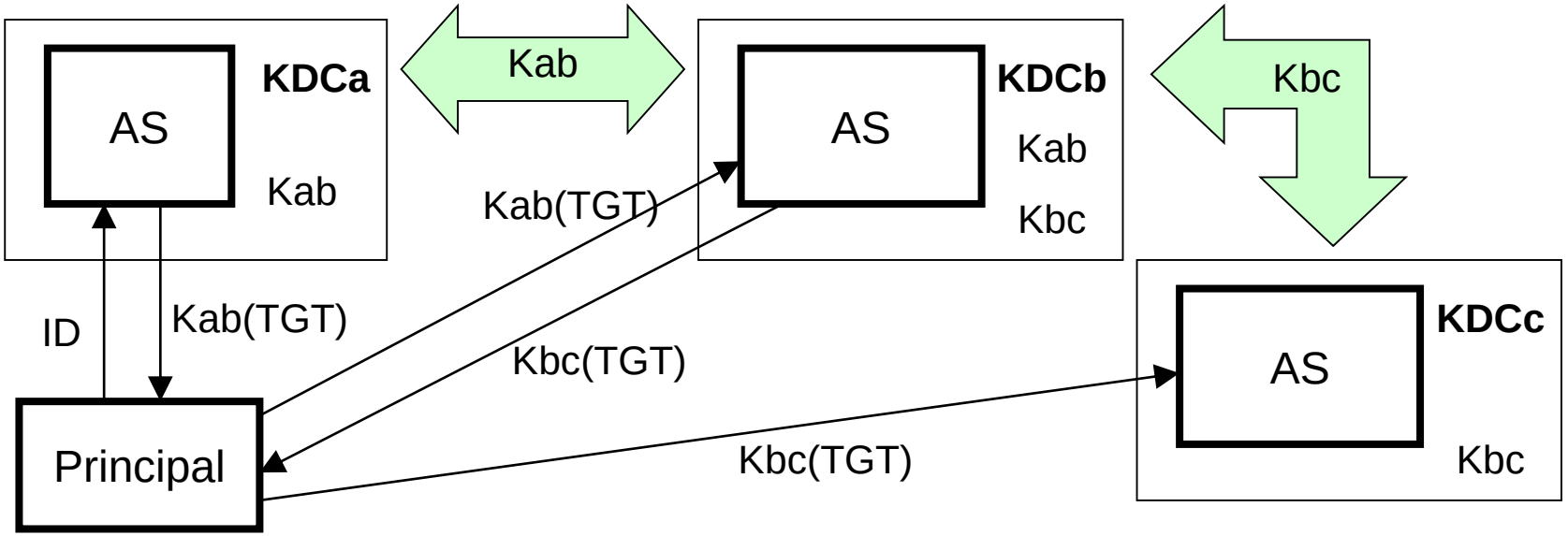
A única diferença relativamente à operação normal é que o TGT é encriptado com Kint e não com Ktgs, isso permite ao TGS remoto desencriptar o TGT. De seguida o protocolo segue o funcionamento normal entre “principal” local e o “realm” remoto.



Transferência de confiança entre “realms”

As implementações de domínios “Active Directory” da Microsoft utilizam a confiança entre “realms” para implementar a confiança entre domínios (“realms”), nomeadamente entre os domínios que fazem parte da mesma árvore (“tree”).

As implementações de relações de confiança para todos os pares de domínios de uma árvore seria morosa, mas o Kerberos permite simplificar o processo através da transferência de relações de confiança (“transitive trust”), a ideia base é que se A confia em B e B confia em C, então A deverá confiar em C.



KERBEROS – Cifras

O Kerberos suporta um conjunto variado de cifras, atualmente o Kerberos 5 exige que sejam suportadas as cifras baseadas em AES com chave de 256 bits. Cada cifra de encriptação é emparelhada com um método de cálculo de "checksum".

Encriptação	"checksum"
AES256-CTS-HMAC-SHA1-96	**HMAC-SHA1-96-AES256**

- SHA1 com chave, resultado truncado a 96 bits (controlo de integridade)
- Variante do CBC (CTS – "Cipher Text Stealing")
- AES com chave de 256 bits

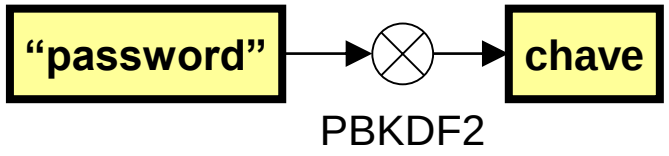
Adicionalmente devem ser suportadas as cifras:

AES128-CTS-HMAC-SHA1-96	**HMAC-SHA1-96-AES128**
DES-CBC-MD5	**DES-MD5**
DES3-CBC-SHA1-KD	**HMAC-SHA1-DES3-KD**

KERBEROS – Segurança

Desde que sejam usadas cifras com chaves de comprimento razoável (128 ou 256 bits), o sistema é considerado bastante sólido.

A produção de chaves a partir das "passwords" dos utilizadores, por exemplo com PBKDF2 ("Password-Based Key Derivation Function 2") é problemática. Abre a porta a ataques de dicionário (ataque à chave em que se procurar gerar valores de chave mais prováveis, por oposição ao ataque de "força bruta" em que são testados todos os valores possíveis).



Sabendo que a chave deriva da "password" torna-se mais simples quebrar a "password" para obter depois a chave.

O funcionamento do "realm" depende da disponibilidade do KDC que constitui assim um ponto único de falha (SPOF). O Kerberos não possui mecanismos de tolerância a falhas. É possível ter vários KDC a servir o mesmo "realm", mas:

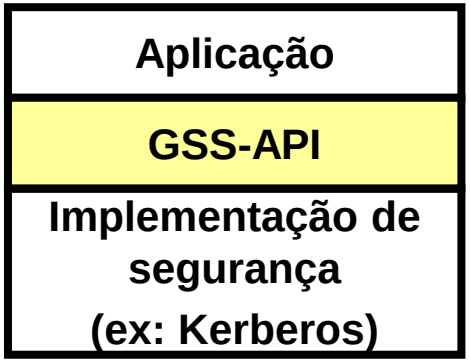
- a sincronização das respetivas bases de dados tem de ser realizada externamente
- os clientes têm de implementar um mecanismo de tentativa e erro, percorrendo uma lista de KDCs até encontrar um disponível.

(estas funcionalidade estão disponíveis por exemplo no AD da Microsoft)

GSS-API ("Generic Security Services Application Program Interface")

Como o nome indica, trata-se de uma API, o objetivo é definir um conjunto de funções que permitem às aplicações interagir com qualquer tipo de mecanismo de autenticação, confidencialidade e integridade existente. Não implementa nenhum mecanismo de segurança em particular.

A GSS-API disponibiliza um conjunto de quase meia centena de funções que permitem às aplicações interagir com a implementação de segurança sem conhecer os seus detalhes de funcionamento.



A aplicação começa por obter as suas credenciais através da função "gss_acquire_cred".

A aplicação estabelece a ligação com a aplicação remota, criando um contexto de segurança na API ("gss_init_sec_context" e "GSS_accept_sec_context").

As aplicações comunicam entre si, usando funções da API para validar ou obter acesso aos dados em forma legível ("gss_verify_mic", "gss_unwrap") e funções para proteger os dados enviados ("gss_get_mic", "gss_wrap").

Existem implementações da GSS-API para as linguagem C, Java e outras.

SASL (“Simple Authentication and Security Layer”)

O SASL permite que qualquer protocolo orientado à ligação, tipicamente TCP num modelo cliente-servidor seja protegido por um mecanismo de segurança existente.

Permite que as comunicações cliente-servidor segundo um dado protocolo sejam protegidas sem necessidade de modificação nas mesmas, novos mecanismos de segurança poderão ser implementados sem qualquer alteração no protocolo.

O SASL limita-se a adicionar ao protocolo algumas mensagens iniciais de negociação entre o cliente e servidor para acordarem o mecanismo de segurança a adoptar, de seguida esse mecanismo será usado.

Os identificadores de mecanismos de segurança a usar durante a negociação são vários (http://www.iana.org/assignments/sasl-mechanisms/sasl-mechanisms.xml), por exemplo:

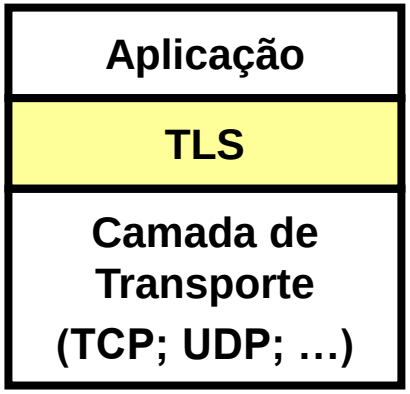
- EAP-AES128
- KERBEROS_V5
- DIGEST-MD5
- OTP
- GSSAPI
- 9798-U-RSA-SHA1-ENC
- 9798-M-RSA-SHA1-ENC
- EXTERNAL
- ANONYMOUS
- PLAIN
- GS2-KRB5

TLS (“Transport Layer Security”)

O TLS é uma evolução do SSL (“Secure Sockets Layer”) desenvolvido pela Netscape, é um protocolo que fornece às aplicações ligações seguras através da rede. As modificações necessárias nas aplicações e protocolos de aplicação para usar o TLS são mínimas.

As aplicações de rede usam o TLS substituído a invocação das funções normais da camada de transporte por funções da API do TLS (por exemplo a API OpenSSL).

A versão atual do TLS é a 1.2, as implementação TLS suportam várias versões incluindo versões de SSL.



Quando se estabelece uma ligação TLS, o primeiro parâmetro a ser negociado é a versão TLS. O cliente envia ao servidor a mensagem “ClientHello” contendo entre outros elementos o número da versão TLS mais recente que o cliente suporta.

O servidor responde com ”ServerHello” indicando qual a versão do protocolo que vai ser usada: a versão mais recente que é comum a ambos.

TLS – Negociação das cifras (“cipher suite”)

Um “cipher suite” é um conjunto da algoritmos de criptografia que é usado numa dada comunicação. Na mensagem “ClientHello” o cliente sugere ao servidor várias alternativas de conjuntos de cifras. Cabe ao servidor selecionar um e indicar de qual se trata na mensagem “ServerHello”.

Exemplos de nomes de conjuntos de cifras (“cipher suites”):

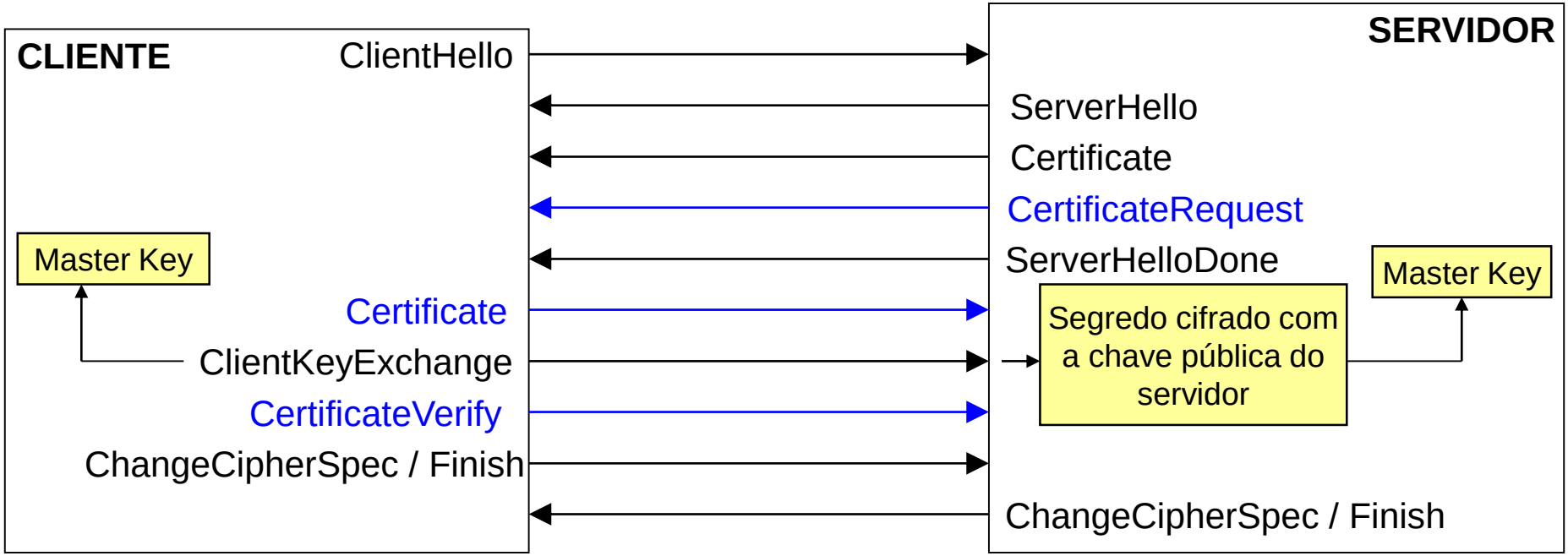
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_KRB5_WITH_DES_CBC_SHA
- TLS_PSK_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_RC4_128_SHA
- TLS_RSA_WITH_IDEA_CBC_SHA

O primeiro elemento do nome do conjunto de cifras indica o mecanismo de autenticação. O resultado da negociação do conjunto de cifras vai condicionar a continuação do diálogo.

O método de autenticação mais usado é através de cifras assimétricas, tipicamente RSA. Nesse caso o servidor além do “ServerHello” envia também o seu certificado de chave pública.

TLS – Autenticação com cifra assimétrica

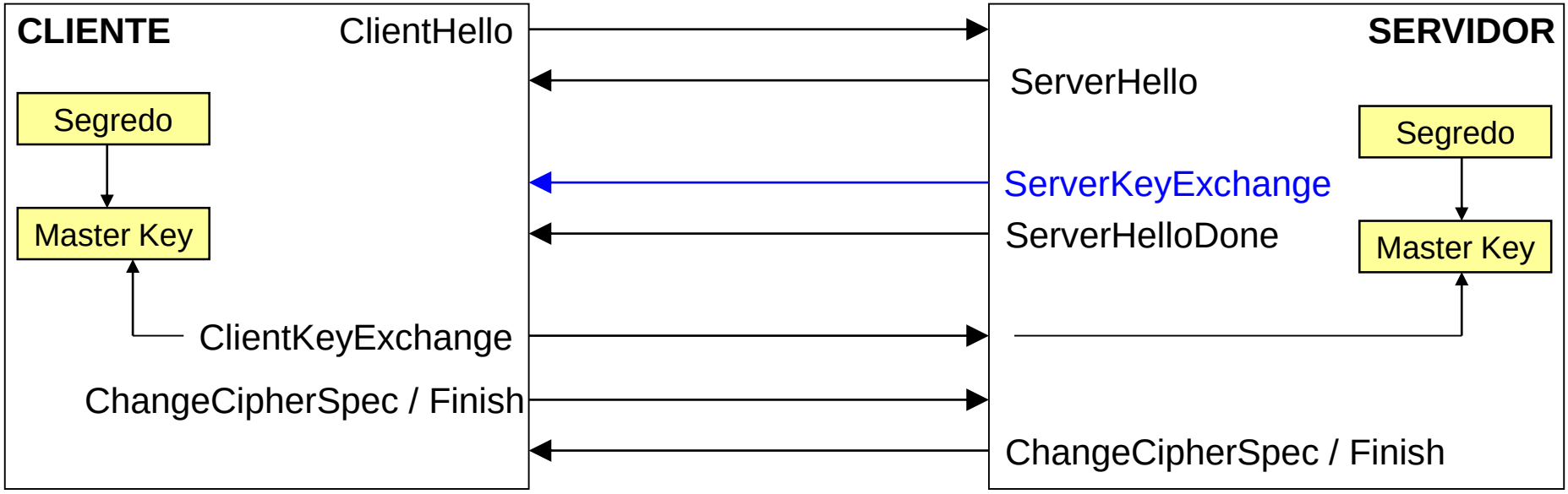
Quando o "cipher-suite" indicado na resposta "ServerHello" corresponde a uma autenticação via chave pública (ex.: TLS_RSA_*), o servidor envia desde logo o seu certificado de chave pública, de seguida pode exigir que o cliente também lhe envie o respetivo certificado.



A cifra assimétrica é usada apenas para autenticação e distribuição da chave. A "Master Key" é usada posteriormente para criar chaves para as cifras simétricas.

TLS – Chave pré-partilhada (PSK)

A utilização de certificados de chave pública tem muitas vantagens e é claramente a técnica mais usada com o TLS. Em caso pontuais usam-se outras técnicas como por exemplo segredos pré-partilhados (TLS_PSK_*).



A finalidade das mensagens "…KeyExchange" é a identificação mútua para determinarem que segredo pré-partilhado devem usar através de desafios. A "Master Key" é gerada a partir do segredo pré-partilhado e é usada posteriormente para criar chaves para as cifras simétricas.

Referências bibliográficas

[1] Lhamas A. (2010-2011). Aulas Teóricas de ASIST.

(…)