Cifras assimétricas

Neste método são utilizadas **duas chaves**: uma privada, pessoal e intransmissível e outra pública, que é disponível a todos.

Vantagens e desvantagens

Permitem garantir a **confidencialidade** sem necessidade de troca de segredos, assim como manter a **integridade** ao autenticar os conteúdos e a **autoria** através das assinaturas digitais.

Tem como principal vantagem que em interações com N interlucotores são apenas necessárias N chaves (na simétrica seraim N^2).

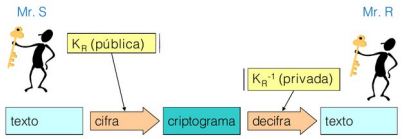
No entanto, apresentam um **desempenho baixo**, pelo que são utilizadas apenas para encriptar pequenos conjuntos de dados (nunca ficheiros!).

Problemas

Para que possamos fazer cifras com este método, precisamos de garantir a distribuição das chaves públicas, que devem ser renovadas periodicamente e consequentemente comunicadas.

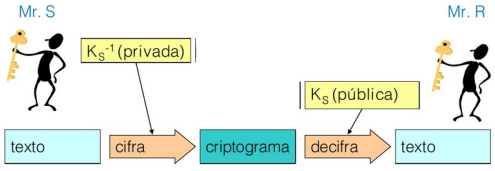
Confidencialidade a autenticidade

Para uma entidade encriptar um texto para outra entidade, a primeira faz a cifra com base na chave pública da segunda, que por sua vez realiza a decifra com a sua chave privada.



No entanto, apesar de garantir **confidencialidade**, o recetor não consegue autenticar quem lhe enviou o criptograma, uma vez que todos têm acesso às chaves públicas.

Para garantir a autenticidade, podemos inverter o processo. Assim, a entidade emissora cifra o texto com a sua chave privada e a recetora decifra com a chave pública da anterior.



Apesar de se garantir a **autenticidade**, perde-se confidencialidade, uma vez que qualquer entidade pode decifrar o criptograma, visto que se utiliza a chave pública para este fim, que é do conhecimento de todos.

De forma a garantir **autenticidade** e **confidencialidade** devem ser aplicados os dois processos.

Processo de cifra

O funcionamento da cifra está baseado em **problemas matemáticos de elevada complexidade**.

Alguns dos problemas mais utilizados são o cálculo de logaritmos discretos, fatorização de grandes números ou problema da mochila (knapsack).

Estes problemas são realizados sobre números inteiros, pelo que mesmo o texto é convertido para número inteiro.

RSA (Riverst, Shamir and Adelman)

Este algoritmo tem por base a fatorização de grandes números e o cálculo de logaritmos discretos e tem chaves com dimensão na ordem dos milhares.

$$C = P^e \bmod n \qquad P = C^d \bmod n$$

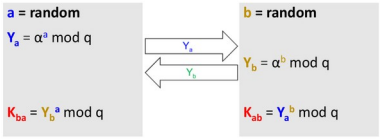
Seja 'n' um valor de grande dimensão e 'e' e 'd' valores relacionados com 'n' através de operações complexas.

ElGamal

Bastante semelhante ao anterior, mas apenas baseado no cálculo de logaritmos discretos.

Diffie-Hellman

Este algoritmo é utilizado em comunicações e tem por base a utilização de **chaves temporárias** (válidas apenas para uma sessão).

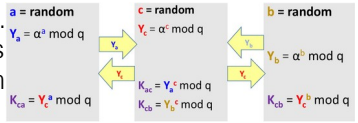


Depois de trocarem Y_a e Y_b, ambas as entidades irão ter as mesmas chaves, ou seja, $K_{ba} = K_{ab}$.

Seja 'q' um número primo de elevada dimensão e 'α' uma raiz primitiva mod q.

Assim, mesmo se Y_a ou Y_b forem intersetados, ou mesmo se os valores 'α' ou 'q' sejam descobertos, será muito difícil determinar o valor da chave.

Este algoritmo é no entanto **vulnerável** a ataques **man in the middle (MidM)**. Uma vez que não há autenticação da entidade com quem se trocam os valores Y, estes podem ser trocados com uma entidade que não é quem queremos contactar. Devem ser implementados mecanismos adicionais.



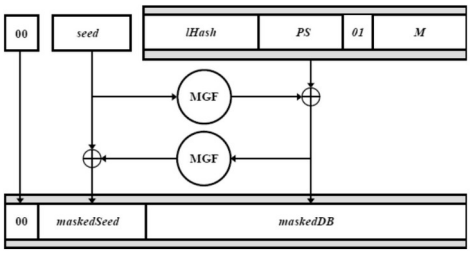
Este método é utilizado pelas tecnologias Wi-Fi, HTTPS ou por exemplo pela aplicação de mensagens Signal.

Tal como no ECB, este algoritmo é **determinístico** uma vez que os mesmos valores codificados com a mesma chave geram o mesmo criptograma.

Randomização de cifras com chave pública

De forma a que N cifras do mesmo valor produzam N criptogramas distintos, devem ser aplicada com técnicas adicionais.

A **solução ótima**, OEAP (*Optimal Asymmetric Encryption Padding*) define um conjunto de operações que dada uma *seed* (valor aleatório) e uma *Mask Generation Function* (MGF) permitem implementar uma solução.



Seja M o texto a cifrar, PS um *padding* de zeros e iHash um Digest sobre Label.

Cifra híbrida

Estas chaves juntam a **velocidade** das chaves simétricas com as **chaves públicas** das assimétricas.

Implementam um algoritmo que consiste em cifrar o texto com uma chave gerada de forma aletória e cifrar essa chave aleatória com a chave pública do destinatário, enviando os dois criptogramas.

Assim, o destinatário poderá decifrar o segundo criptograma com a sua chave privada, obtendo a chave utilizada na cifra simétrica do texto, que vai utilizar para o decifrar.

Funções de síntese (digest)

Uma **função de síntese** é uma função que cria um resultado de dimensão constante para entradas de dimensão variável. Este resultado irá identificar de forma única a entrada, pelo que é como que uma "impressão digital".

Por serem **funções de dispersão criptográficas unidirecionais** produzem resultados muito distintos para entradas similares e que não podem ser revertidos (obtida a entrada a partir do resultado).

Para serem robustas, devem apresentar três propriedades.

Resistência à descoberta do texto Dada uma síntese, é difícil encontrar entrada que a gerou

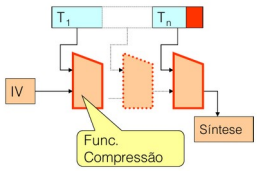
Resistência à descoberta de 2.º texto Difícil encontrar texto diferente que produza mesma síntese

Resistência à colisão Difícil encontrar dois textos que mapeiem para a mesma síntese (paradoxo do aniversário).

A **resistência à descoberta de 2.º texto** relaciona-se com a facilidade de um utilizador de forma deliberada produzir textos que sejam mapeados para a mesma síntese. Tem a ver com o tamanho da síntese, mas principalmente com o seu funcionamento interno.

A **resistência à colisão** está apenas relacionada com o tamanho da síntese. Uma síntese demasiado pequena irá facilmente produzir colisões.

Este algoritmo é implementado com recursoa uma **compressão iterativa**, onde o texto é cifrado em blocos com *feedback*. No entanto, os criptogramas originados para cada bloco só servem de *feedback* à cifra do próximo, com exceção do último, que corresponder à síntese.



Message Integrity Code (MIC)

Uma das aplicabilidades das **funções de síntese** é na deteção de erros de **comunicação** e **armazenamento**.

Para tal, quando é enviada uma mensagem encriptada, deve enviar-se também uma síntese do criptograma, também denominado por **MIC**.

$$MAC = Síntese(C)$$

O recetor, por sua vez, vai calcular a síntese do criptograma e compará-lo com o MIC recebido. Se forem iguais, então está garantida que a comunicação não teve falhas.

No entanto, este método não protege contra alterações deliberadas, uma vez que o atacante pode calcular um novo MIC para o criptograma alterado.

Message Authentication Code (MAC)

De forma a garantir a deteção de ambos os erros de **comunicação** e **armazenamento** e ainda eventuais **ataques** surgiu esta solução.

É uma pequena variação do MIC, em que na síntese, em vez de sintetizar o criptograma a enviar, sintetiza a concatenação do criptograma com uma chave que apenas o remetente e o destinatário conhecem. **Ecrypt-then-MAC**.

$$MAC = Síntese(K, C)$$

Seja K a chave conhecida pelos dois intervenientes e C o criptograma.

Assim, se algum atacante manipular o criptograma não terá maneira de gerar um novo MAC correspondente, uma vez que não tem conhecimento da chave utilizada.

O algoritmo de cifra GCM abordado anteriormente (cifras simétricas contínuas) faz uso este mecanismo para garantir a autenticação, através da introdução da chave **auth data**. O destinatário vai validar a **auth tag** e caso não seja válida não o vai decifrar.

Há más aproximações deste método de comunicação.
Encrypt-and-MAC MAC é calculado a partir do texto original e não é cifrado. Obriga à decifra para validar, pelo que se não for integro será desperdício de recursos. Dois textos iguais terão o mesmo MAC!
MAC-then-decript MAC é calculado a partir do texto original e cifrado também. Obriga à decifra para validar, pelo que se não for integro será desperdício de recursos.

Assinaturas digitais

Ao utilizar funções de síntese com chaves privadas e públicas podemos criar **assinaturas digitais**. Estas permitem garantir a integridade e autenticação, ao validar se um dado texto foi criado por uma determinada entidade.

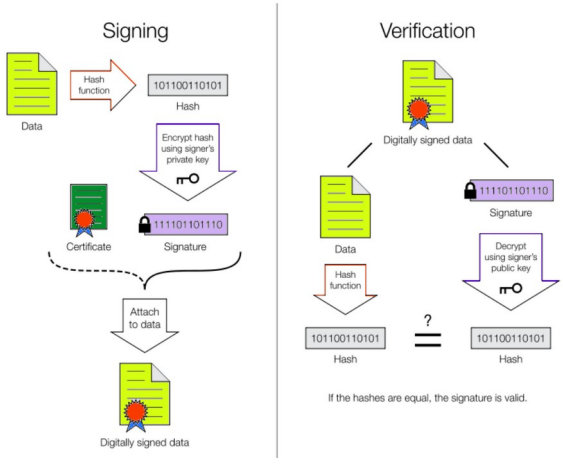
A **assinatura** consiste então da encriptação da síntese do documento e da informação da entidade que o assina com a chave privada dessa entidade.

$$A(doc) = info + Cifra(K_{privada}, Síntese(doc + info))$$

A **validação** é feita através da decifra da assinatura com a chave pública dessa entidade, que deverá ser similar à síntese do documento.

$$Decifra(K_{pública}, A(doc)) \equiv Síntese(doc + info)$$

A info será um certificado com informações sobre a entidade, como o seu nome, informação pessoal (número CC) e motivo da assinatura.



Derivação de chaves

Outra aplicação para as **funções de síntese** é a **derivação de chaves**.

Esta é utilizada quando queremos aplicar algoritmos de encriptação que requerem chaves de dimensão fixa a partir de chaves de dimensão variável e eventualmente com baixa entropia[1].

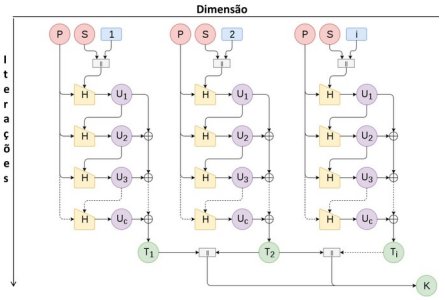
Aplica-se principalmente a cenários em que há intervenientes humanos, uma vez que estes têm tendência a definir palavras-chave relativamente simples para a complexidade necessária à garantia da segurança dos sistemas. Aumenta assim a **segurança das palavras-passe**.

Ao **aumentar o tamanho das palavras-passe** dificulta a criação de ataques por *brute force*.

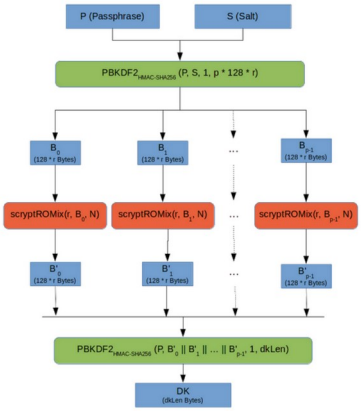
A sua **implementação** requere um sal que torna a geração única (valor aleatório), um problema complexo e um grau de complexidade parametrizável.

Nas aulas práticas foi utilizado o Password Based Key Derivation Funcion 2 (PBKDF2).
PBKDF2(função de síntese, sal, número de iterações, password, dimensão)

Na prática, o que esta função faz é executar a função de síntese para cada bloco com o sal como IV para criar uma síntese para a password com uma determinada dimensão várias vezes (número de iterações). Ver imagem ao lado.



Existem variações que apresentam **dificuldades computacionais** e outras que apresentam **dificuldades de armazenamento**.



Muitas iterações implicam muitos recursos computacionais. Podem ser ainda adicionadas dificuldades de armazenamento com a implementação de problemas que requerem muita memória para serem resolvidos. Estas duas características dificultam a decifra do texto sintetizado.

Um algoritmo que implementa estas dificuldades é o scrypt.
Scrypt(password, sal, custo, paralelização, dimensão, tamanho blocos a usar, dimensão função síntese, bytes da mistura interna)

Este faz uso da PBKDF2 para produzir blocos, que depois vêm os seus bits misturados e são novamente sintetizados na PBKDF2.

A PBKDF2 introduz as **dificuldades computacionais**, enquanto que a mistura dos bits introduz **dificuldades de armazenamento**.

1 Medida de desordem de um sistema.

5. Gestão de chaves assimétricas

Slides teóricos e aula assíncrona

De forma a assegurar a **autenticidade** e **confidencialidade** das cifras assimétricas é imperativo que seja garantida a privacidade das chaves privadas a distribuição correta das chaves públicas, respetivamente.

Há ainda outros fatores que devem ser tidos em conta como a evolução temporal do mapeamentos chave/entidades que pode ter necessidade de sofrer alterações por oerda ou roubo e ainda a geração correta dos pares de chaves, que para além de grandes devem ser realmente aleatórios e nada previsíveis.

Neste capítulo serão abordados os objetivos que devem ser cumpridos na gestão de chaves assimétricas de forma a garantir que nenhum dos problemas mencionados anteriormente se verifica.

1. Geração de pares de chaves

Devem ser utilizados geradores cuja probabilidade de cada bit seja igual para todos os bits e que não produzam chaves com padrões derivados do número de iterações ou do valor anterior (*feedback*).

Geralmente estes geradores têm de ser implementados ao nível do *hardware*. o entanto, há problemas matemáticos complexos e com grande período que permitem gerar **números pseudo-aleatórios**.

Podem no entanto ser facilitados os processos sem comprometer a segurança. Por exemplos as chaves públicas podem ter um tamanho relativamente reduzido (3 ou 17 p.e.), o que vai facilitar as operações.

É também fundamental garantir que a chave privada é gerada pelo próprio, de forma a assegurar ao máximo a sua privacidade. Se for possível, é preferível que o dono não tenha acesso à chave, mas apenas acesso aos processos com ela.

Fundamental caso haja necessidade de autenticidade. Caso contrário, esta medida pode ser relaxada.

---

2. Manuseamento de chaves privadas

No caso de ser armazenada no computador do sujeito que representa, a chave privada deve ver o seu caminho de acesso controlado e estar sempre cifrada.

A solução ideal passa no entanto pelo **confinamento** da chave, que consiste no seu armazenamento numa entidade autónoma segura. Esta entidade pode ser um módulo seguro de *harware* interno, uma partição lógica segura no CPU ou mesmo um dispositivo externo como um *smart card*.

Com esta abordagem, as aplicações nunca utilizam a chave, limitando-se a invocar ao dispositivo a realização de operações (assinaturas e verificações).

3. Distribuição de chaves públicas

De forma a garantir a disseminação confiável de chaves públicas, utilizam-se **hierarquias e grafos de certificação** que, através de uma certificação unidirecional, verifica as relações de entidade definidas de forma clara entre entidades.

Se A confia na chave pública K e B confia em A, então B vai confiar na chave pública K.

Este modelo ganhou forma nos certificados digitais de chaves públicas emitidos por uma **entidade certificadora** (CA). Estes certificados são **documentos públicos** que **ligam uma chave pública a uma entidade** e que são **seguros por meios criptográficos** através da assinatura do emissor (a CA).

No "mundo real", uma entidade certificadora será um notário, a pessoa que verifica a nossa identidade e certifica que fomos nós que assinámos um determinado papel.

De forma a serem processos de forma automática, a estrutura dos certificados foi estandardizada pela norma X.509v3, que define os campos obrigatórios como a versão, o **sujeito**, a **chave pública**, as **datas** de início e expiração, o **emissor** e a sua **assinatura**.

Podem ser codificados em vários formatos, como binário ou PEM (*Privacy Enhanced Email*).

É também fundamental que o certificado se aplique apenas a um perfil de utilização restrito, que é definido atravês da extensão crítica **key usage**. Assim, uma entidade terá um certificado para cada utilização.

Um certificado gerado para validar a autenticidade de um servidor *web* não pode ser utilizado para assinar *emails*!

Entidades certificadoras (CA)

Estas organizações, que podem ser governamentais, sem fins lucrativos ou até mesmo empresas devem operar de forma **segura**, **transparente** e **auditável** de forma a serem confiáveis.

A sua função não se limita a validar a identidade dos sujeitos para os quais emite certificados. Passa também pela distribuição dos certificados e pela sua revogação em caso de necessidade, providenciando para estes casos listas de certificados revogados e ainda interfaces de verificação do estado de um certificado.

Quando são **confiáveis**, o certificado de cada CA é distribuído pelos SO e navegadores e os certificados assinados por elas passam a ser considerados como válidos.

Uma CA pode ainda certificar outras CA (através de um certificado), conhecidas por **CA intermédias**, que podem também emitir certificados. Formam-se assim hierarquias de certificação, que, independente do número de níveis têm de ter no seu topo uma CA raiz confiável para serem consideradas confiáveis.

As **CA raiz** assinam os seus próprios certificados, uma vez que têm acesso tanto à sua chave pública, como privada.

Apesar de ser bastante improvável, é possível contornar estes critérios de segurança se um atacante conseguir ter acesso ao certificado e à chave privada da entidade certificada, podendo assim alterar o seu certificado.

Outra forma é a de criar uma entidade certificadora raiz falsa no diretório que armazena as CA raiz no computador do alvo.

Modelo PEM (Privacy-enhanced Electronic Email)

Esta hierarquia de certificação define um modelo com uma raiz única, a IPRA (*Internet Policy Registration Authority*), que iria certificar várias PCA (*Policy Creation Authotities*), que por sua ver certificariam as CA, que seriam responsáveis por emitir os certificados finais.

Modelo PGP (Pretty Good Privacy)

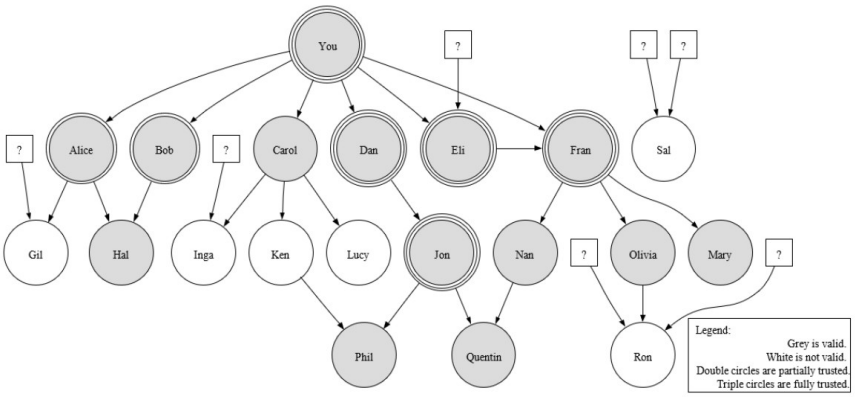
Este modelo é baseado numa rede de confiança, em oposição à autoridade central.

As entidades têm **confiança** nas chaves que conhecem porque foram validadas por si e no comportamento dos outros certificadores, assumindo que verificam as chaves que certificam. Verifica-se uma confiança transitiva, que é **unidirecional**, **pessoal** e **sujetiva**.

Divide-se em 4 níveis: ultimate (chaves próprias), complete, marginal e unstrusted

Podem ainda considerar uma entidade **válida** se confiar totalmente numa entidade que assinou o seu certificado, ou parcialmente em duas ou mais que o fizeram.

Se o certificado for assinado por apenas uma em que a entidade confia apenas marginalmente, diz-se marginalmente válida.



4. Ciclos de vida dos pares de chaves

Apesar de na teoria este modelo ser bastante robusto, existe a possibilidade de descoberta da chave privada de uma entidade, o que vai invalidar o seu certificado.

No entanto, uma vez que os certificados são copiados e distribuídos de forma livre, não existe uma noção de quem são os seus pussuidores de forma a revogá-los em cada um deles.

Para resolver este problema foram criadas duas abordagens. A primeira, menos imediata, mas que evita o problema a longo prazo, é a **validade temporal**, que define o período durante o qual o certificado pode ser considerado válido.

A outra, com efeitos mais imediatos, são as **listas de revogação de certificados (CRL)**. Estas listas são disponibilizadas publicamente pelas CA, que cruzam as suas listas entre si de forma a facilitar a distribuição.

Listas de revogação de certificados (CRL)

A lista completa de todos os certificados revogados diz-se no **formato Base CRL**.

No entanto, dada a dimensão do mundo da internet e a validade dos certificados poder chegar aos 20 anos, a ordem de grandeza do número de certificados revogados está numa ordem com demasidados zeros para ser exequível consultar a lista completa para cada verificação. Foram então propostos outros formatos.

O **Delta CRL** é um formato cuja lista contém as diferenças desde a última Base CRL.

São usadas em complemento aos Base CRL. Os navegadores podem fazer o *download* da **Base CRL** uma vez por dia ou por semana e todos os dias ou todas as horas irem buscar a **Delta CRL** com as diferenças em relação à última.

A **OSCP (Online Certificate Status Protocol)** é uma API que permite validar certificados indidualmente junto das CA. Para tal, o cliente envia o número de série do certificado e recebe uma resposta assinada pela CA, que pode guardar durante a sua validade (**OSCP Stapling**[2]).

Apesar de requerer menos largura de banda por parte do cliente, que deixa de ter a necessisade de descarregar a Base CRL completa, implica maior largura de banda para as CA, que se forem lentas a responder irão prejudicar a performance dos clientes.

Representam ainda uma perda da privacidade para o cliente, uma vez que as CA passam a saber em que momento cada sistema acede a cada serviço.

Uma vez que o subject dos certificados SSL são o domínio, ao utilizar um OSCP para validar o certificado, a CA vai saber a que *site* estamos a aceder.

A Apple utiliza no MacOS o OSCP para validar os ficheiros binários antes de os executar, pelo que a empresa tem conhecimento de todos os programas que são corridos no seu sistema operativo, quando e por quais utilizadores.

2 Agrafar

Public Key Infraestructure (PKI)

Este é o nome dado ao processo de gestão de um conjunto de certificados.

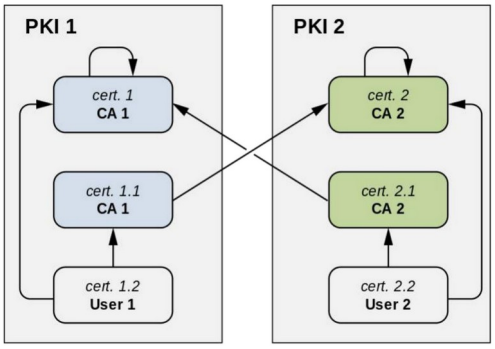
Este define a **criação segura de pares de chaves assimétricas**, a **criação e distribuição de certificados de chaves públicas**, a **definição e uso de cadeias de certificação**, a **atualização, publicação e consulta de certificados revogados** e por fim o **uso de estruturas de dados e protocolos que permitem a interoperação entre componentes**.

Esta infraestrutura deve ser disponibilizada pelas CA de forma a serem auditadas livremente pelo público.

Relações de confiança

De forma a agilizar o processo de criação de novas CA e a aumentar a sua abrangência, é bastante comum que as CA se certifiquem entre si, gerando relações hierárquicas, cruzadas, ou até mesmo ad-hoc[3] (meshed, grafos mais ou menos complexos).

Por exemplo uma CA nova vai ter dificuldade em ser considerada como de confiança por sistemas mais antigos ou que sejam atualizados com pouca frequência. No entanto, se esta CA for certificada por outra que já exista há mais tempo e que tenha uma abrangência mais consolidada, será amplamente aceite.



Pinning

O **pinning** consiste na geração de uma impressão digital através de uma função de síntese que é associada ao pedido HTTP no código compilado.

Assim, quando for recebido o certificado para esse pedido, para além da validação da hierarquia de certificação, é verificado se o certificado recebido corresponde ao esperado.

Esta técnica permite prevenir ataques através da injeção de certificados maliciosos no diretório local de certificados raiz.

Transparência de certificação

Definida pela norma RFC 6962, este é um sistema que regista todos os certificados públicos emitidos, garantindo que só são publicados certificados que levam a raízes legitimas.

3 Particular, específico

6. Autenticação: mecanismos e protocolos

Slides teóricos e aula assíncrona

A **autenticação** é um processo que tem como objetivo provar que uma entidade possui um atributo que diz ter. Possibilita a aplicação de políticas e mecanismos de autorização.

Geralmente divide-se em duas fases, a primeira onde o sujeito se identifica e a segunda onde o prova.

A autenticação (Authn) leva à autorização (Authz).

Para efeitos de prova podem ser consideradas várias alternativas, nomeadamente **algo que sabemos**, como uma palavra-passe, **algo que temos**, um *token* possuído por uma entidade como um *smart card*, ou **algo que somos**, por dados biométricos.

De forma a aumentar a segurança, por vezes estes tipos de prova são combinados na **autenticação multifatorial**, que utiliza em simultâneo dois ou mais tipos.

Há vários requisitos fundamentais para a autenticação.

Confiança uma vez que a prova deve ser robusta e difícil de subverter

Secretismo das credenciais utilizadas pelas entidades

A dificuldade de subverter uma palavra-passe é muito maior do que uma autenticação por reconhecimento facial, porque no segundo a maior parte das vezes basta imprimir uma fotografia do sujeito em questão.

Se este reconhecimento for feito com recurso a infravermelhos, como passa a haver noção de profundidade, esta torna-se mais robusta. No entanto, continua a ser possível a subversão com recurso a um busto.

Robustez na troca de dados para impedir ataques, assim como DoD

Simplicidade tanta quanto possível, para evitar que os utentes escolham simplificações perigosas.

Lidas com vulnerabilidades vindas de pessoas têm tendência para facilitar

A sua implentação divide-se em vários eixos, nomeadamente no **temporal**, podendo a autenticação dar-se ao início da interação, ou ser feita continuamente ao longo da mesma e no **direcional**, havendo autenticação uni e bidirecional (mútua).

Aproximações

Existem duas aproximações possíveis, a **direta**, que consiste na simples apresentação de credenciais e esperar pelo veredicto e a com **desafio-resposta**, onde é mostrado um desafio ao utilizador que apenas ele sabe responder, sendo o veredicto feito com base nesta resposta.

Autenticação direta com senha memorizada

Segundo esta aproximação, o utilizador fornece um nome de utilizador e uma palavra-passe, que é confrontada com o valor guardado na base de dados para a entidade correspondente. A palavra-passe nunca pode ser guardada em claro na BD, devendo ser transformada com recurso a uma função unidirecional, como a *passkword based key derivation function*.

Apesar de simples, estes sistemas têm alguns problemas, nomeadamente os ataques por *brute force* (a menos que implementados mecanismos em contrário), principalmente se as palavras-passe forem fracas e pouco complexas e a transmissão de senhas em claro em canais inseguros, que pode levar à sua interseção por parte de atacantes.

Autenticação direta com biometria

Com esta aproximação deixamos de ter problemas com as credenciais definidas pelo utilizador, uma vez que este não tem escolha. As credenciais não podem ser roubadas.

No entanto, tem alguns problemas associados, como a incipiência[4] da maioria dos métodos, que podem ser ultrapassados com facilidade, o facto de não poderem ser alterados, havendo um impacto maior da sua exposição, a impossibilidade de transferir as credenciais a outros em casos de necessidade como cenários de emergência, a possibilidade de colocar o sujeito em risco, a dificuldade de aplicação em sistemas remotos, devido à necessidade de garantia de que se está a utilizar um sistema seguro para a aquisição da biometria e por fim o facto de que a biometria pode revelar informação pessoal como hábitos, doenças...

O sujeito pode ser colocado em risco seja devido ao método em si como no caso da utilização de lasers para autenticação com a iris ou do risco de lhe cortarem um dedo para utilizarem a sua impressão digital.

Como já foi descrito anteriormente, o reconhecimento facial é fácil de ultrapassar com uma fotografia. Até mesmo a impressão digital se torna pouco robusta, uma vez que utilizamos o dedo da autenticação no ecrã do telemóvel e noutras superfícies, que pode ser a partir daí obtido com técnicas relativamente simples e utilizado por atacantes.

Autenticação direta com senhas descartáveis

Esta aproximação pode ser feita através de matrizes, de códigos pré-distribuídos ou de mecanismos como a TOTP (Time-based One Time Password).

Estes mecanismos podem ser implementados sobre canais inseguros, uma vez que mesmo intersetados não vão poder ser novamente utilizados, podem ser escolhidos pelo autenticador, passando este a definir o grau de segurança, podem depender de uma senha que o utilizador saiba e de um dispositivo, algo que o utilizador tem.

4 Que começa. Que está no início.

No entanto tem algumas desvantagens, uma vez que necessita de uma autenticação inicial, para identificar qual o utilizador e qual a senha temporária a pedir, obriga à existência de algo para armazenar/calcular as chaves, seja um pedaço de papel ou uma aplicação no telemóvel, que pode ser roubado.

Aproximação por desafio resposta

A aproximação por **desafio resposta** consiste na disponibilização de um desafio pelo autenticador, que o sujeito transforma e envia de volta ao autenticador, que valida o seu resultado, utilizando o mesmo método ou algo pré-partilhado, como a chave pública da entidade. A transformação do desafio pela entidade pode ser feita com recurso a uma credencial (palavra-passe), ou por exemplo à sua chave privada.

O desafio pode ser por exemplo um NONCE, um número arbitário que só pode ser utilizado uma vez.

Tem como vantagens a não exposição das credenciais, uma vez que elas nunca circulam no canal de comunicação (apenas circula a sua transformação), o que leva à robustez contra ataques de MITM. Este mecanismo é também compatível com outras aproximações, como dispositivos físicos, chaves (as)simétricas e permite ao autenticador definir a transformação e complexidade do desafio.

Apresenta também algumas desvantagens, como a necessidade de utilizar um token para responder aos desafios, a necessidade do autenticador guardar os segredos em claro, a possibilidade de calcular todas as respostas possíveis por brute force e a necessidade do autenticador faça uma boa gestão dos NONCE, uma vez que estes não podem ser reutilizados.

Uma implementação bastante robusta desta aproximação pode ser feita com recurso a **smart cards**, através da cifra do desafio com a chave privada do utilizador, que tem o seu acesso protegido por um PIN. Para validar, o autenticador decifra com a chave pública do mesmo. É robusto contra ataques por brute force, pelo roubo dos dados da BD e devido à utilização de canais inseguros.

Quando não há smart cards, pode recorrer-se a **segredos partilhados**. Neste caso, o desafio é transformado (pode ser encriptado) com base num segredo que é do conhecimento do sujeito e do servidor (pode ser partilhada a transformação em vez do segredo em claro). O autenticador valida a transformação através do mesmo cálculo, ou da decifra do valor enviado pelo sujeito.

Alguns algoritmos que permitem fazer a implementação com segredos partilhados são o CHAP, MS-CHAP e S/KEY.

Protocolos

Há vários protocolos que implementam as aproximações descritas anteriormente.

PAP (Point-to-poin Authentication Protocol)

Consiste na validação direta do par UID/password.

Está sujeito à transmissão insegura.

CHAP (Challenge-response Authentication Protocol)

Este protocolo implementa a aproximação por desafio resposta.

Aut → U : authID, challenge
U → Aut: authID, MD5(authID, secret, challenge), identity
Aut → U : authID, OK/not OK

O MD5 é um algoritmo de síntese.

Caso o atacante pode intersetar o authID, o challenge e a síntese, o protocolo torna-se vulnerável a ataques por *brute force*.

S/Key

Este protocolo também é uma aproximação por desafio resposta e utiliza um processo engenhoso que permite que um autenticador e um sujeito consigam pré-gerar uma quantidade de tokens que são geradas de forma sequencial e nunca são repetidas.

O autenticador guarda a raiz, a última OTP usada pelo sujeito e o seu índice (i).

Aquando da autenticação, o autenticador envia a raiz e o índice. O sujeito, com base na sua palavra-passe, vai gerar a cadeia de sínteses até ao índice i-1, enviando o resultado da última ao antenticador.

Este, vai pegar na síntese recebida e sintetizá-la mais uma vez (índice i), validando que esta corresponde à síntese que tem armazenada.

Caso seja válida, atualiza a síntese que tem armazenada e o seu índice (i-1).

Em alternativa à função de síntese MD4 pode ser utilizada a MD5 ou o SHA-1.

Caso um atacante intersete o índice, a raiz e a síntese, pode descobrir a palavra-passe por engenheria reversa através de brute-force. Para evitar este tipo de ataques, os índices devem ser suficientemente grandes e deve ser dado *reset* ao protocolo quando deixam de o ser.

Casos práticos

GSM (Global System for Mobile Communications)

A autenticação do subscritor dos serviços de comunicação móvel é feito com uma aproximação de desafio resposta com base num segredo partilhado entre o HLR e o subscritor, uma chave simétrica de 128 bits, denomida de Ki. Esta chave nunca é transmitida, mas utilizada em operações de transformação realizadas internamente pelo cartão SIM.

O HLR, sigla para Home Location Register, é uma base de dados que centraliza os detalhes de cada subscritor de comunicações móveis que está autorizado a utilizar a rede GSM. Armazena detalhes de cada cartão SIM.

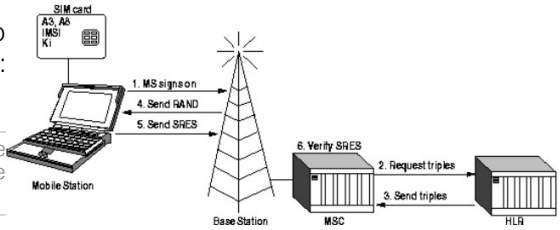
O Ki é armazenado no cartão SIM aquando do seu fabrico e armazenado no HLR de forma segura.

As transformações são feitas com recurso ao cartão SIM, sem nunca haver acesso direto ou conhecimento da Ki.

Existem vários protocolos associados a este sistema, o **A3** par autenticação, **A8** para geração da chave de sessão e **A5** para comunicações com cifra contínua.

Para **autenticar** um telemóvel na rede, o MSC pede ao HLR/AUC (*authentication center*) os valores do subscritor: RAND, SRES e Kc.

Os MSC, Mobile Switching Center, são os nós periféricos das redes de comunicações, aos quais as torres de telecomunicações estão diretamente conectados.



Este RAND é enviado ao terminal, que com base neste valor e no seu Ki gera o seu SRES, que devolve ao MSC.

$$SRES = A3(Ki, RAND)$$

Para **cifrar as comunicações**, a par da autenticação, este RAND é também utilizado para gerar a chave de sessão.

$$Kc = A8(Ki, RAND)$$

Combina algo que se tem, o cartão SIM, com algo que se sabe, o PIN, que desbloqueia o anterior, com um desafio.

Autenticação de sistemas

A autenticação de sistemas pode ser feita através de vários métodos.

Nome (DNS) ou endereço (IP/MAC)

Métodos fracos e sem provas criptográficas.

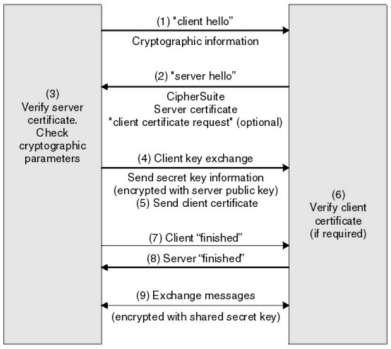
Chaves criptográficas

Chaves secretas partilhadas entre entidades que comunicam frequentemente

A autenticação por chaves criptográficas pode ser feita com recurso a **chaves assimétricas**, através da pré-partilha da chave pública de cada sistema entre as entidades que comunicam frequentemente. Em vez da pré-partilha, pode ser utilizada uma chave pública certificada por uma CA (X509).

TLS (Transport Layer Security)

Este é um protocolo que é utilizado em todas as interações através da internet, permitindo a gestão de sessões web seguras sobre TCP/IP, através da garantia de **confidencialidade**, **integridade** e **autenticação** das entidades intervenientes.



A comunicação começa com o envio do cliente ao servidor de uma mensagem de "hello" com algumas informações criptográficas associadas (p.e. as cifras que suporta).

O servidor responde identificando-se numa mensagem que inclui o seu certificado, podendo também pedir de volta o certificado do cliente (opcional), que necessitar de o autenticar. Nesta mensagem o servidor escolhe também as cifras a serem usadas (*CipherSuite*).

Validada a identidade do servidor através do seu certificado assinado com a sua chave privada (para provar que tem posse sobre esta), que por sua vez deverá ser assinado por uma CA de confiança.

Nesta fase o protocolo de cifra está negociado. Segue-se um processo de negociação das chaves segundo o método de Diffie-Helman.

Terminado este processo, está estabelecido um canal de comunicação segura entre o cliente e o servidor. No navegador é identificado por HTTPS.

O processo de negociação das cifras é fundamental e permite a criação de **ciphersuites**, um conjunto de algoritmos que inclui um algoritmo de negociação de chaves, autenticação, cifra,chaves e modo e de controlo de integridade.

Esta negociação é fundamental, uma vez que não é esperado que todos os clientes suportem um algoritmo implementado num servidor. Devem haver assim várias hipóteses, de forma a garantir compatibilidade com clientes mais novos e antigos, poderosos ou mais limitados. Desde a versão 1.3. do TLS que a escolha é feita pelo servidor.

SSH (Secure Shell)

No que toca a acessos remotos, o SSH é um protocolo que permite a gestão de sessões seguras interativas sobre TCP/IP. Tal como o anterior, fornece **confidencialidade**, **integridade** e **autenticação**.

O seu funcionamento distingue-se, no entanto, uma vez que dada a natureza do ambiente em que é utilizado (mais restrito e seguro), não há certificação das chaves. Assim, na primeira

conexão o cliente é questionado se aceita a chave do servidor, sendo em caso afirmativo esta armazenada e considerada em sessões futuras.

Assume-se que dada a especificidade da utilização deste protocolo, na primeira interação entre o cliente e o servidor, este terá algum meio de validar junto do servidor que a chave que recebeu é a esperada, antes de a aceitar.

A autenticação do cliente pode variar entre a simples palavra-chave, chaves assimétricas, *smart cards…*

Caso o cliente também se autentique por chaves assimétricas, o servidor terá de manter um conjunto de chaves públicas associadas a cada utilizador, validando assim a sua identidade. Neste caso o cliente não necessita de inserir palavra-passe, sendo a autenticação um processo transparente para si.

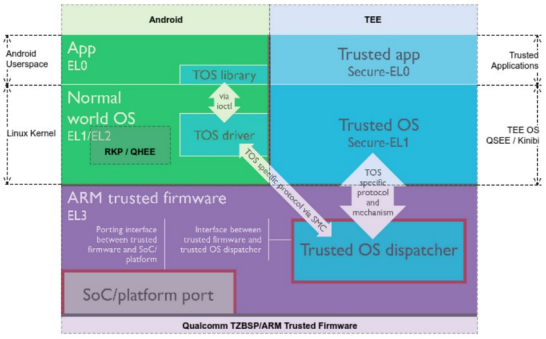
Autenticação em sistemas específicos

A teoria nem sempre se aplica da mesma forma na prática, variando de acordo com o contexto, a criticidade dos dados a que esta dá acesso e até do tipo de dispositivos, destacando-se neste último aspeto quatro grandes grupos: **dispositivos móveis**, **computadores pessoais**, **computadores em redes** em ambientes empresariais ou universitários e por fim **dispositivos de suporte** como routers, consolas, eletrodomésticos.

Dispositivos móveis

Estes dispositivos têm evoluído no sentido de serem cada vez mais **pessoais**, pelo que é comum apresentarem autenticação com 2 fatores que hoje pode incorporar vários métodos desde os mais básicos como senhas, PIN ou padrões, até aos mais avançados, como dados biométricos. Fazem ainda uso de um **cartão SIM**, também ele encriptado, que na maior parte das vezes está também associado a um sujeito identificado.

Apesar de serem sistemas pequenos e aparentemente simples, na realidade cada dispositivo tem pelo menos quatro ambientes de execução distintos.



REE Corre aplicações instaladas pelos utilizadores.

Baseband Executa código para comunicação (comunica com operador). Closedsource, fornecido pelo fabricante.

SIM Cartão Java, que corre aplicações lá dentro.

TEE Trusted Execution Environment. Pode ser um chip ou uma partição virtualizada do CPU responsável pelo armazenamento de chaves e execução de operações criptográficas. Funciona como um *smart card*, mas dentro do telemóvel.

Trusted Execution Environment

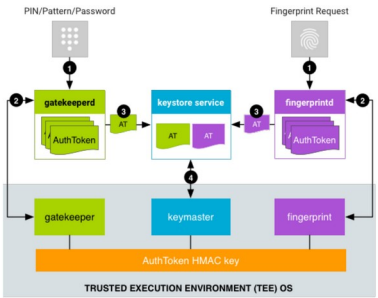
Quando o sistema é inicializado divide a memória entre o REE e o TEE. O TEE funciona como um smart card, mas dentro do telemóvel. É um ambiente limitado, que corre um SO distinto e que apresenta uma interface de acesso altamento restrita e controlada. Dentro deste ambiente correm pequenas aplicações autotizadas, denominadas por trustlets.

É através deste subsistema que é possível realizar pagamentos com o telemóvel, sem necessidade do cartão de multibanco.

Hoje em dia é considerado tão robusto que já há telemóveis que dispensam o cartão SIM, utilizando em vez disso o iSIM, que recorre aos mecanismos de autenticação e criptografia do telemóvel (através do TEE). Para tal, basta que o utilizador crie um perfil, um conjunto de parâmetros e chaves para que este se possa associar à rede.

As credenciais são associadas a um sujeito, podendo até ser específicas para cada aplicação, sendo as chaves desbloqueadas apenas quando apresentadas as credenciais corretas. No entanto, estas nunca saem para fora do TEE.

Para autenticar um utilizador, as aplicações autenticam-se no TEE através dos **daemons** respetivos, dos quais obtêm um *token* de sessão, que utilizam no **keystore** para validar as credenciais. Há vários *daemons*, para diferentes tipos de credenciais.



Gatekeeperd para pins, padrões e palavras-passe

Fingerprintd para impressões digitais

As chaves são geradas através de procedimentos que têm em conta os elementos físicos do sistema, como o MAC, pelo que não é possível trocar o TEE de dispositivo e aceder-lhe.

Keymaster

A gestão das chaves é feita no **keymaster**, em quem o **keystore** delega a validação das credenciais. Para garantir que as chaves provêm do TEE implementado em *hardware* e são autênticas, associadas a determinada aplicação e numa determinada versão do TEE, este passou a suportar o processo de **key attestation**.

$$attestKey(keyToAttest, attestParams): X.509\ Certificate$$

Este processo retorna um certificado X.509, que é assinado por um certificado raiz para este fim.

Com esta certificação as aplicações garantem que não há uma aplicação a correr entre o TEE e elas e a falsificar chaves.

Posteriormente foi disponibilizado o serviço de **ID attestation**, que consiste no inverso: na validação das aplicações que acedem ao TEE, garantindo assim que não há apks instalados manualmente a fazer-se passar por aplicações que não são.

É muito utilizado em aplicações de jogos e de *home banking*.

Gatekeeper

O **gatekeeper** armazena PIN entre 4 e 16 dígitos e senhas com dimensões semelhantes. São vulneráveis a ataques por força bruta e por canais paralelos.

O ataque por canais paralelos consiste na análise do osciloscópio do telemóvel aquando do processo de introdução do código secreto, associado a cada vibração uma tecla específica. Ver *paper* "There goes your PIN", por David Berend, 2018.

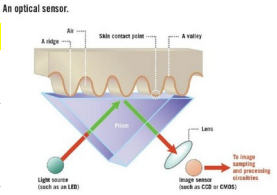
Os padrões são muito menos seguros que os anteriores, sendo vulneráveis a ataques por visualização da inserção (*over the shoulder*) e a análise das marcas dos dedos.

Fingerprint

Para cada **impressão digital** são guardados vários modelos associados a uma conta de utilizador. Para a sua validação o perfil é obtido por um sensor e validado no TEE. A segurança varia com a implementação.

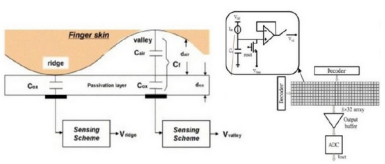
Os modelos nunca são extraídos. O que acontece é o envio do perfil ao TEE para validação.

Os leitores mais básicos são os **leitores óticos**, que utilizam um LED para iluminar o dedo e um CCD[5] para capturar a imagem. Desta captura resulta uma imagem 2D.



É possível duplicar impressões digitais ou mesmo criar padrões que levem à construção de um modelo do lado do TEE que valide a autenticação. A gordura no vidro pode deixar impressões digitais completas, bastanto realçá-las com pó para serem validadas pelo sistema.

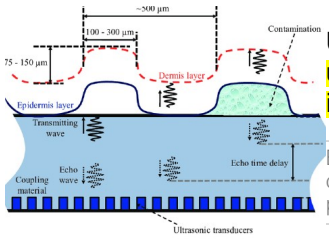
A estes seguiram-se os **leitores capacitivos**, que têm uma matriz que mede a capacidade, ou seja, a distância à pele, determinando os seus vales e montes.



5 Charged Coupled Device. Sensor utilizado em câmaras digitais para capturar imagens.

É vulnerável a modelos físicos.

O suor, água e cremes interferem com a leitura.



Uma alternativa são os **leitores ultrassónicos**, que emitem impulsos de ultrassons e através da reflexão dos sinais recebida criam um mapa da impressão digital.

Estes modelos são muito mais resilientes e precisos. Penetram cremes e água e podem até ser calibrados para detetar a pulsação do dedo, garantindo que o dedo não foi cortado a quem pertence.

Todos os métodos descritos anteriormente são passíveis de ser contronados. Ver vídeo.

Reconhecimento facial

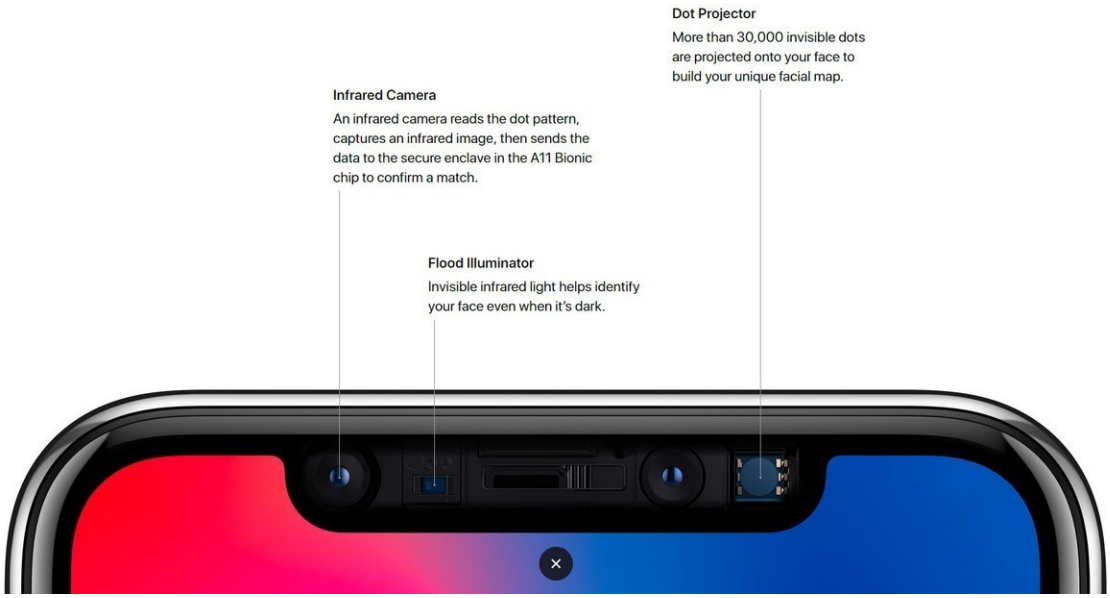
Este tipo de autenticação tem como objetivo verificar a correspondência entre uma imagem e um modelo treinado. Dependendo do método de captura da imagem da cara, pode ser mais ou menos facilmente contornado.

Se for uma fotografia simples, facilmente é contornado com uma fotografia. Temos também o problema dos gémeos. Nem sempre é robusto a alterações de luminosidade, nem a alterações do sujeito (óculos, barba, …) ou de direção.

A Apple desenvolveu para os seus dispositivos o Face ID, que consiste num sistema que projeta um conjunto de pontos invísiveis ao olho humano, lidos por uma câmara de infravermelhos, permitindo criar um modelo 3D da cara.

A utilização de infravermelhos facilita a leitura, uma vez que não é afetada por iluminação visível, funcionando bem tanto em ambientes bem como mal iluminados.

É vulnerável à impressão 3D da cara (busto).



Computadores portáteis

Estes são dispositivos menos pessoais, sendo potencialmente utilizados por múltiplos utilizadores. Podem incluir ambientes seguros simples, designados por **TPM**, ou **Trusted Platform Module**. Estes módulos são muito mais simples que os dos smartphones.

Não suportam por exemplo o armazenamento generalizado de chaves. A autenticação é nativa, feita pelo próprio SO.

Podem apresentar uma grande variedade de ferramentas de autenticação, desde leitores de impressões digitais, sensores de reconhecimento facial (mais simples ou mais complexos), leitores de smartcards (em ambientes empresariais), ou mesmo apresentar interação com outros dispositivos como smartphones, yubikeys, etc.

Windows

O **Windows** suporta vários métodos de autenticação, incluindo remotos (Active Directory). As credenciais são guardadas de uma forma que permite que sejam facilmente removidas (ficando o utilizador sem palavra-passe). Só desde 2006 inclui o **User Access Control**, que pede dados de autenticação em operações que podem colocar em risco o sistema operativo (como instalar uma aplicação).

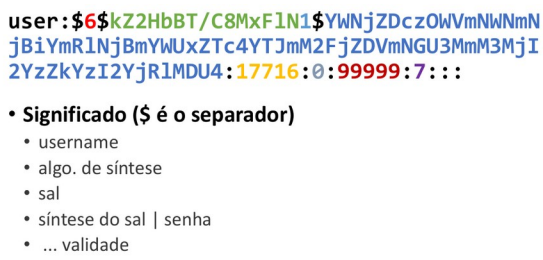
As senhas são armazenadas na forma de síntese. O PIN era suportado pelo TPM, que ao ser inserido desbloqueava as chaves (simples e pouco rubosto, abandonado). O Windows Hello é semelhante ao Face ID da Apple, fazendo recurso de um projetor e um sensor de infravermelhos para mapear a cara do sujeito a autenticar.

Linux

Do ponto de vista de mecanismos é bastante semelhante. Inicialmente cada aplicação era responsável pela sua autenticação. No entanto, este modo de funcionamento foi substituído pelo **PAM**, ou **Pluggable Authentication Modules**, um middleware que permite aos administradores definirem métodos de autenticação costumizados, que podem até combinar dois ou mais fatores.

As senhas (síntese) são armazenadas em claro no ficheiro /etc/shadow. Os dados dos utilizadores em /etc/passwd.

Validação é feira pelo PAM, que vai ao /etc/passwd verificar se o ID de utilizador dado é válido. De seguida, vai ao /etc/shadow buscar o salt e o método de síntese, calcula a síntese do salt concatenado com a senha e compara com o valor armazenado no ficheiro. São usados alguns milhares de iterações.



As aplicações fazem uso dos PAM para pedir a autenticação do utilizador, vendo-se dispensadas dessa tarefa.

Sistemas distribuídos

Nestes cenários é comum a utilização de sistemas de autenticação centralizada, através de repositórios comuns de credenciais e informação de utilizadores, conhecido por **IDP** ou **IDentity Provider**. As aplicações fazem uso destes sistemas sem necessidade de manter qualquer tipo de registos de palavras-passe.

A UA fornece um sistema deste género através do sistema do utilizador universal (idp.ua.pt).

Este tipo de autenticação **Single Sign On (SSO)** explora sistemas externos de confiança (**TTP**, ou **Trusted Third Party**), que sendo internos ou externos à organização, são assumidos como confiáveis pelas aplicações que deles fazem uso.

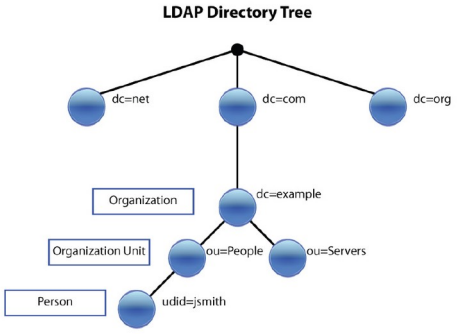
O serviço fornecido é conhecido por **AAA**, ou **Autentication, Authorization and Accounting**, uma vez que mantêm registo de quem é que entrou onde.

Tem como principais vantagens a reutilização das credenciais de acesso para vários serviços, o que facilita a vida ao utilizador e também aos desenvolvedores das aplicações, que não têm de investir recursos na construção de um sistema de autenticação.

No entanto, ao centralizá-la, a falha do sistema de autenticação pode comprometer todos os serviços que dele dependem, podendo bloquear o acesso a todos os serviços ou introduzir atrasos nos processos de autenticação.

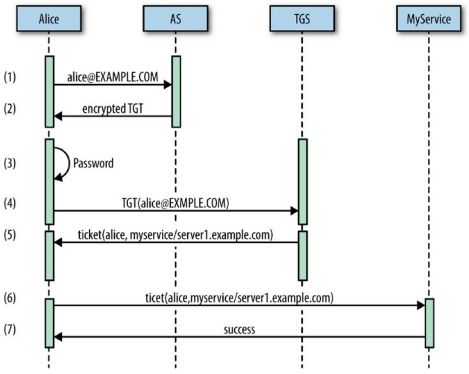
Em sistemas locais, é necessária a instalação de um agente que exponha os utilizadores remotos. Geralmente utiliza sistemas de cache para acelerar operações.

Esta autenticação pode ser feita através do protocolo **Lightweight Directory Access Protocol (LDAP)**, que mantem um diretório de informação hierárquico com informação sobre utilizadores, sistemas e serviços.



Antigamente os *e-mails* da UA tinham no domínio o departamento. Por exemplo nome@deti.ua.pt. Ao iniciar sessão no IDP com este endereço, o mapeamento era feito diretamente para os utilizadores do DETI, restringindo o domínio de pesquisa e evitando a análise de todos os utilizadores registados na UA.

Uma alternativa é o **Kerberos**, que funciona por tickets com validade limitada que são atribuídos a entidades que são validadas perante um serviço.



Envolve 4 entidades: o cliente que pretende aceder ao serviço, o service server (SS) que fornece o serviço, o ticket granting server (TGS) que fornece acesso aos serviços e o authentication server (AS) que fornece acesso ao TGS.

Para se autenticar num serviço, o cliente envia um pedido ao AS com o seu nome de utilizador, recebendo como resposta um token cifrado com a palavra-passe do utilizador e outro cifrado com a chave do serviço que pretendemos utilizar.

Localmente, o cliente vai decifrar o primeiro token. De seguida, envia-lo ao TGC, mais o segundo token e o identificador do serviço a que pretende aceder. Validados os dados recebidos, o TGC responde com um ticket de acesso ao serviço, que depois é utilizado nas comunicações com este.

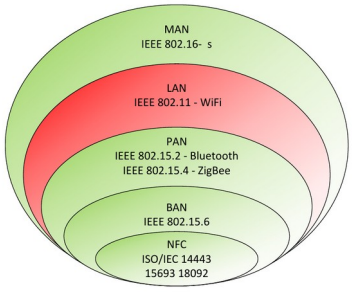
O processo de autenticação é distinto do de autorização. Um utilizador pode estar autenticado e não ter autorização para aceder ao serviço (por isso na imagem não é um traço contínuo).

Não há comunicação direta entre os sistemas. O utilizador interage com cada um de forma isolada.

7. Segurança em redes IEEE 802.11

Slides teóricos e aula assíncrona

As redes **WiFi** são uma infraestrutura importante das organizações, pelo que deve ser tida em consideração aquando da definição das suas políticas de segurança.



Dentro das várias tecnologias que implementam **redes sem fios**, esta enquadra-se nas LAN e tal como o nome indica fornece uma rede local sem fios, num raio de aproximadamente 50 metros (sem parede) do AP.

Apesar de este raio ser visto por alguns como uma limitação, na realidade é uma virtude, pois por questões de segurança não há interesse em que a rede abranja mais do que a área estritamente necessária.

Redes sem fios

Como em qualquer **rede sem fios**, as comunicações são feitas em **broadcast** para o meio, sendo responsabilidade das placas de cada máquina analisar os pacotes e determinar se a têm como destinatário. Torna-se assim bastante fácil a atacantes dentro no domínio da rede intersetar pacotes sem serem detetados, ou mesmo interferir com as comunicações legítimas atrvés de um ataque de DoS com o envio de pacotes ilegítimos na mesma frequência.

Estes ataques podem ser mitigados ao nível físico com o **Phy:**

- Codifica o canal com uma chave secreta, à qual apenas os membros da rede têm acesso;
- Evita que o canal seja monopolizado por transmissores através de políticas de acesso ao meio físico:
  - Frequência alterada segundo um padrão conhecido apenas pelo emissor e recetor, negociado aquando do emparelhamento. Bluethooth FHSS (Frequency Hopping Spread Spectrum);
  - No Wi-Fi cada rede utiliza uma frequência específica.

Ao nível dos dados há também algumas medidas que podem ser tomadas, nomeadamente a cifra dos cabeçalhos e utilização de endereços temporários, que evitam que os atacantes compreendam os dados e construam perfis de consumo para os **endereços MAC**[6] e ainda que forjem tramas válidas, uma vez que há autenticação do emissor.

Pela análise dos cabeçalhos dos pacotes sem estas medidas, é possível definir padrões de navegação aos endereços MAC, sendo depois apenas necessário fornecer uma rede pública de utilização gratuita em que o utilizador tem de se registar e a partir desse momento associa-se uma pessoa e os seus dados aos dados armazenados para o MAC. Esta prática é utilizada por exemplo para criar publicidade direcionada.

6 Media Access Control

IEEE 802.11

O **IEEE 802.11** é o standart que rege o **Wi-Fi** e define um conjunto de conceitos importantes à compreensão da sua arquitetura.

Estação (STA) Dispositivo que se liga à rede sem fios, com um identificador único (MAC)

Ponto de acesso (AP) Dispositivo que permite a ligação de dispositivos sem fios, interligando a outras redes (com ou sem fios)

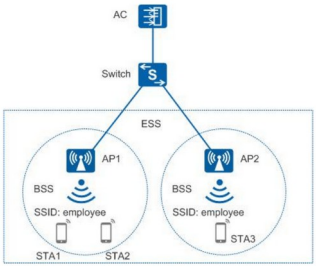
Rede sem fios Conjunto de STAs e Aps associados entre si e comunicando

SSID Identificados de uma rede sem fios

A organização da rede também deu origem a novas terminologias.

Basic Service SET (BSS) Rede formada por STAs ligadas a um AP

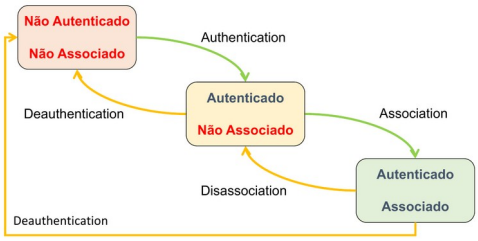
Extended Service Set (ESS) Rede formada por várias BSS interligadas por um **Distribution System (DS)**



Numa ESS, os vários AP de cada BSS fornecem o mesmo SSID (mesma rede).

Também é possível que o mesmo AP forneça vários SSID.

No acesso à rede o cliente transita entre vários estados que combinam a **autenticação** e a **associação** (autorização).



O protocolo define vários tipos de **mensagens de gestão**.

Beacon[7] Pacote com a informação da rede enviado periodicamente (10 vezes por segundo) para o ambiente, que permite aos clientes detetar que a rede existe

Probe[8] **request/response** Para Aps escondidos (que não emitem beacons), o cliente faz um probe request para o ambiente a perguntar se há um AP com o SSID procurado. A resposta é um probe response.

Authentication request/response | Deauthentication

(Re)Association request/response | Desassociation

Há ainda **mensagens de controlo** (ACK) e **mensagens de dados**.

7 Farol

8 Sondagem

A **segurança do meio físico** foi evoluindo ao longo do tempo para algoritmos mais robustos.

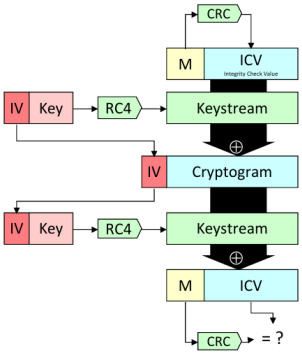
Tipo de Rede / Funcionalidade		pre-RSN	RSN (Robust Security Network)		
		WEP	WPA	802.11i (ou WPA2)	WPA3
Autenticação		Unilateral (STA)	Bilateral com 802.1X (STA, AP enetwork)		Bilateral com 802.1x
Distribuição de Chaves			EAP ou PSK, 4-Way Handshake		WP2 + OWE e SAE
Política de Gestão de IVs			TKIP	AES-CCMP	AES-GCM
Cifra dos Dados		RC4		AES-CTR	AES-GCM e EC
Controlo de Integridade	**Cabeçalhos**		Michael	AES CBC-MAC	SHA-384 HMAC
	Corpo	CRC-32	CRC-32, Michael		

WEP (Wired Equivalent Privacy)

Este tipo de rede pode funcionar em dois modos, fornecendo autenticação unilateral e facultativa, com recurso ao algoritmo RC4.

Open System Authentication (OSA) Sem qualquer autenticação.

Shared Key Authentication (SKA) Autenticação unilateral da STA com chave distinta por cliente com base no seu endereço MAC. Dados cifrados, mas com pouca robustez critográfica.



O facto de não haver autenticação do AP faz com que seja bastante simples a execução de ataques MitM (man in the middle) bastando para isso saber a chave.

Esta pode ser determinada por técnicas de engenharia reversa, por análise de pacotes cujo conteúdo é previsível. Como o IV tem poucos bits e não é gerido, vai repetir-se em algum momento e nesse caso vamos ter dois pacotes cifrados com a mesma chave (chave + IV, mas Ivs iguais), pelo que vai ser possível determiná-los.

WPA (Wi-Fi Protected Access)

De forma a mitigar os problemas do WEP, foi criado o WPA. Este tipo de rede utiliza uma chave diferente por mensagem, evitando as fracas e implementa um controlo de integridade mais robusto, assim como controlo de IVs. Faz uso do **protocolo TKIP**.

Para cada mensagem a chave resulta de um conjunto de várias operações, que misturam vários elementos, como por exemplo uma **chave temporal**, que permite evitar ataques por engenharia social.

O controlo de integridade é implementado através de um **MIC** que combina os endereços de origem e destino do pacote, entre outros elementos.

De forma a suportar estas alterações os campos do WEP nos cabeçalhos dos pacotes foram extendidos para integrarem os valores que permitem o controlo de integridade.

Está alinhado coma a especificação IEEE 802.11i, que define a arquitetura de segurança atual. Foi pensado de forma a trabalhar no hardware do primeiro, pelo que na prática implementa os mesmos algoritmos, utilizando apenas mecanismos mais robustos de variação das chaves e controlo.

De forma a trabalhar no hardware do primeiro, sua implementação inicial foi ao nível do firmware[9].

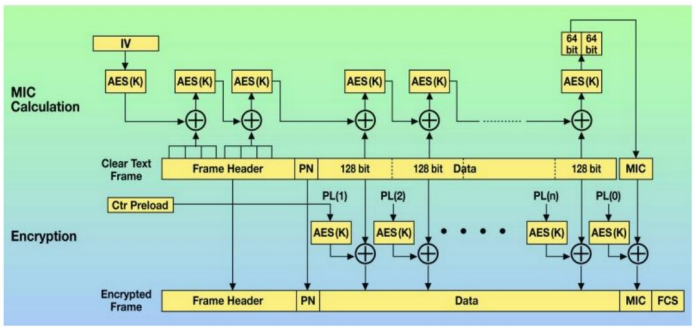
Relativamente ao WPE, o WPA não deixou de ter por base uma cifra contínua aplicada a dados que mudam de forma determinística, pelo que continua a ser possível por brute force enviar pacotes cuja resposta é conhecida e pela sua análise tentar adivinhar as chaves.

WPA2

Sucessor do WPA, este é o protocolo mais utilizado hoje em dia e implementa mecanismos avançados para a proteção das mensagens e autenticação. Utiliza o algoritmo de cifra AES.

Utiliza o 802.1x para autenticar clientes em ambientes organizacionais. Em ambientes menos exigentes e mais simples, como os domésticos, utiliza o método SOHO.

Para gerar o MIC de forma a garantir o controlo de integridade, em vez de utilizar um algoritmo de síntese dedicado, utiliza o AES.



A geração do MIC inicia-se com um IV e a cada iteração dos blocos do pacote, se faz XOR do bloco com a cifra do resultado da operação anterior. Este procedimento é ilustrado na imagem ao lado. Trata-se então de um processo de cifra com AES em modo CBC (cypher block chaining).

A encriptação do pacote e feita com AES em modo CTR (counter mode). Apenas os dados são encriptados!

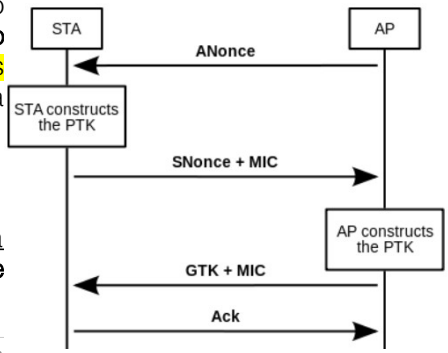
Processos de autenticação

As **chaves** utilizadas na ecriptação são negociadas.

Em redes pequenas, este processo dá-se pela troca de **nonces**[10] entre o AP e o STA, a partir das quais geram a chave através de uma **Pseudo Random Function (PRF)**, que tem como argumentos os nonces, os MACs do AP e da STA e ainda uma **Pairwise Master Key**, que consiste numa síntese da password com o SSID utilizando o algoritmo **PBKDF2**.

$$PRF(PMK \mid ANonce \mid SNonce \mid AP\ MAC \mid STA\ MAC)$$

Assim, cada cliente na rede terá uma chave distinta e exclusiva da sua ligação ao AP, pelo que que estas chaves são designadas por **Pairwise Transient Keys (PTK)**.



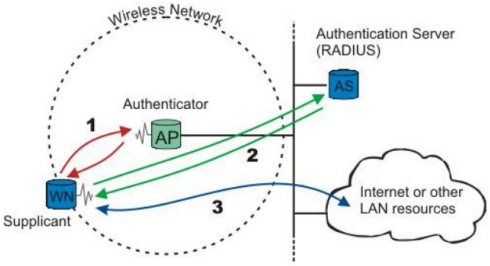
A **PMK** diz-se uma **Pre-Shared Key (PSK)**, pois é um segredo pré-partilhado. Existe ainda uma **Group Temporal Key (GTK)** que é gerada de forma semelhante e utilizada para comunicações em **broadcast**.

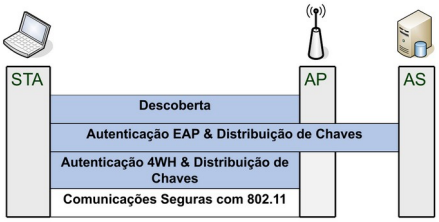
Este processo de negociação de chaves tem semelhanças com o Diffie-Hellman. É designado por **4WH, 4 Way Handshake**.

9 Software que fornece controlo de baixo nível para o hardware específico do dispositivo.

10 Palavra criada para uma única ocasião para resolver um problema imediato de comuniucação.

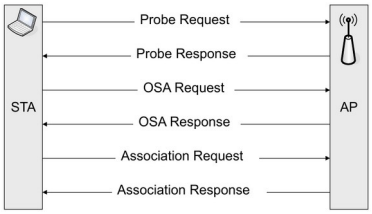
A **autenticação** em redes grandes é feita com recurso ao protocolo de gestão de acessos **IEEE 802.1X** que permite a geração de chaves de forma dinâmica com métodos de autenticação de nível mais alto de forma a que hajam chaves diferentes para cada utilizador, mas que essas chaves estejam disponíveis em todos os AP da rede.





A **arquitetura** distingue os **WPA supplicant** (clientes) que acedem a um **AP** e se autenticam num **servidor de autenticação** (AS).

Este processo inicia-se com a **descoberta** do AP, através da análise dos beacons ou de probe requests.



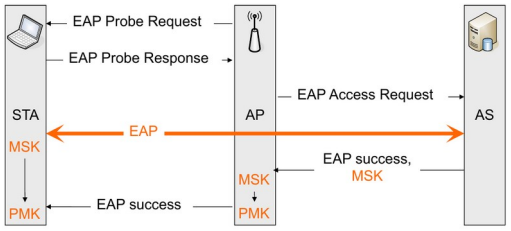
As portas controladas pelo 802.1X continuam fechadas! Não há dados do utilizador, apenas mensagens de controlo para autenticação.

Segue-se a **autenticação e distribuição de chaves** com recurso ao protocolo EAP que é feita diretamente com o AS. Quando a autenticação é bem sucedida, o AS fornece uma **Master Session Key (MSK)** tanto ao STA como ao AP, o que permite que estes continuem com o processo de autenticação.

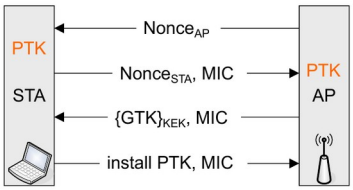
Nas nossas casas já temos a Master Key instalada no AP e a instalação do lado do STA faz-se com a inserção da palavra-passe da rede por parte do utilizador.

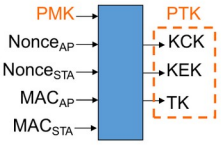
No 802.1X é necessário configurar estas chaves de forma dinâmica, através da inserção da chave pelo AS no AP, depois da autenticação bem sucedida do STA.

Os portos de dados continuam fechados.



Este processo de **autenticação local** do STA no AP faz-se depois como explicado anteriormente, com a troca de nonces de forma a negociarem uma **Pairwise Transient Key**.





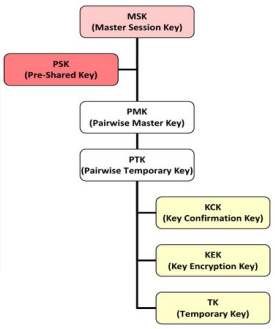
Ambos confirmam que têm os valores corretos da PMK e PTK através da troca de MICs.

As portas controladas passam a permitir tráfego unicast.

É por fim atingido o objetivo das **comunicações seguras** entre o STA e o AP.

IEEE 802.1X: Hierarquia de chaves

O protocolo **IEEE 802.1X** define uma hierarquia de chaves.



MSK Resultado direto de um processo de EAP na arquitetura enterprise. É diferente para cada processo de autenticação de cada cliente.

PSK Chave de longo termo partilhada entre SA e STA na arquitetura SOHO.

PMK Chave recente utilizada para autenticação mútua entre o AP e a STA, gerada com PBKDF2.

PTK Chave para proteger interações entre o AP e a STA.

Extensible Authentication Protocol (EAP)

Este é o protocolo de base à autenticação do **IEEE 802.1X** genérico que permite a definição de métodos de autenticação para um cliente se validar junto de um servidor de autenticação de forma transparente para o AP (autenticador).

	EAP-MD5	LEAP	EAP-TLS	EAP-TTLS	PEAP
AS	N/A	H(desafio, senha)	Chave Pública (certificado)		
Autenticação	H(desafio, senha)	H(desafio, senha)	Chave Pública (certificado)	EAP, Chave Pública (certificado)	PAP, CHAP, MS-CHAP, EAP
Gestão de Chaves	Não	Sim			
Riscos	- Exposição de identidade - Ataques por Dicionário - Host-in-the-Middle - Roubo de ligações	- Exposição de identidade - Ataques por Dicionário - Host-in-the-Middle	- Exposição de identidade		- Exposição de identidade (fase 1)

No **EAP-MD5**, o cliente identifica-se junto do AP quando pede para se ligar (exposição da identididade), ao que este responde com um desafio, utilizado pelo cliente para gerar uma hash em conjunto com a sua palavra-passe. É assim também vulnerável a ataques Man in the Middle e ainda a ataques por dicionário de geração da hash, sabido o desafio.

No **PEAP (Protected EAP)** é criado um túnel seguro entre o STA e o AP e depois é feita autenticação com MS-CHAP. Há uma exposição de identidade nos pacotes iniciais, antes da criação do túnel.

A rede das universidades (eduroam) utiliza o último método (com o MS-CHAPv2). Assim, quando alguém se liga a um AP (autenticador), este encaminha o pedido para o servidor de autenticação correspondente ao seu domínio na sua instituição (por exemplo @ua.pt para o RADIUS da UA, @uc.pt para o da UC, @up.pt para o da UP).

A transparência para o AP é garantida, pois este apenas precisa de saber qual o servidor de autenticação para o qual encaminha os pedidos e esperar que lhe seja fornecida a MSK. Abstrai-se completamente do processo de autenticação.

Problemas de segurança

Problemas

Mesmo com o melhor protocolo de segurança disponível para proteção das redes **IEEE 802.11**, há problemas de segurança que persistem.

A existência de **PSK baseados em palavras-passe** é vulnerável a ataques por dicionário.

A **desproteção das mensagens de gestão** permite a personificação por parte dos atacantes.

A **personificação** pode ocorrer por exemplo por um atacante que envie uma mensagem de gestão a desautenticar uma máquina. Como estas mensagens não são protegidas, o AP não tem maneira de autenticar que foi o sujeito a desautenticar que enviou a mensagem.

Há ainda o **protocolo de controlo de acesso ao meio (CSMA)**, que obriga os clientes a esperarem um tempo aleatório no final da transmissão de cada pacote, que permite a atacantes ter janelas de acesso maiores em relação aos outros, caso escolham sempre o menor tempo.

Vulnerabilidades

A **falta de segurança futura** é uma vulnerabilidade, pois os ataques por dicionário, apesar de levarem tempo, são possíveis e a durabilidade das PSK no modo SOHO permite-o.

O WPA-Enterprise não apresenta esta vulnerabilidade se a PMK for diferente a cada autenticação, que acontece se a MSK for sempre diferente em cada autenticação (geralmente aleatória). Como depende da MSK, está fora do âmbito do WPA e é responsabilidade do servidor de autenticação.

Estes ataques por força bruta acrescentam ainda à lista a **descoberta de chaves** e do **PIN WPS**.

Durante o 4WH, o atacante consegue obter imensa informação: SSID, A/SNonce, MAC AP/STA.

Para a descoberta da **PTK** falta apenas descobrir a **palavra-passe** de forma a gerar o **PMK**. No entanto, este processo é demorado, porque a função de síntese utilizada é robusta e é utilizada com muitos rounds. Ainda mais difícil é se a senha for grande e aleatória.

Há ainda alguns AP que enviam o campo **PMKID** nos seus beacons para acelerar o processo de autenticação, permitindo ao STA validar as credenciais inseridas pelo utilizador antes de iniciar o processo de autenticação, evitando a inicialização do processo com credenciais erradas.

No entanto, este valor resulta de uma síntese mais simples, com menos parâmetros e novamente com vários conhecidos, com exceção do PMK. É por isso mais fácil de decifrar.

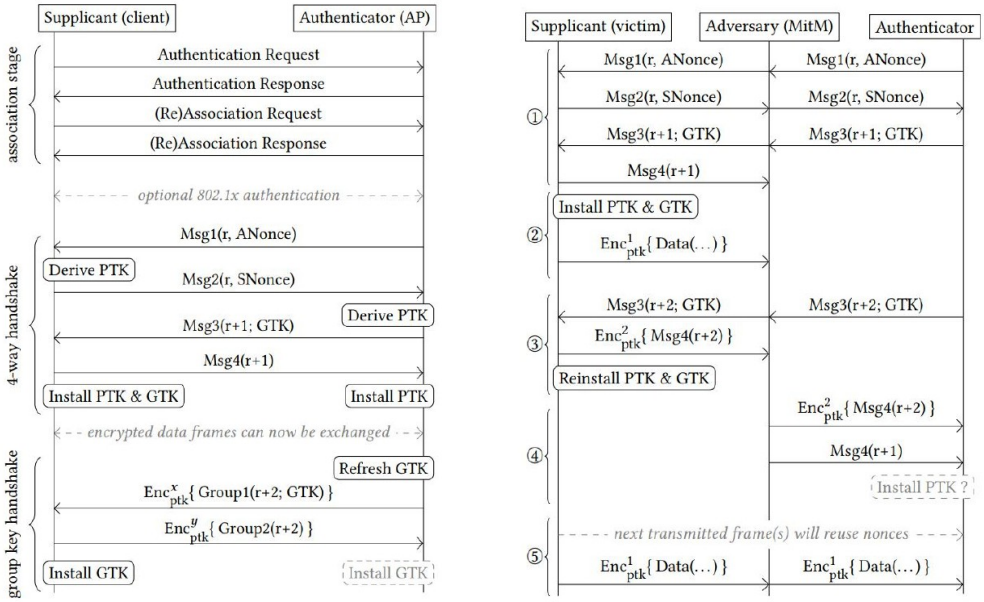
$$PMKID = HMAC - SHA\,1 - 128\left(PMK \mid PMKNAME \mid AP\ MAC \mid STA\ MAC\right)$$

Qualquer um destes ataques são ataques **offline**, que trabalham com pacotes capturados em determinado momento, permitindo que o ataque seja executado sem conhecimento da organização gestora da rede.

A **reinstalação de chaves** é também uma vulnerabilidade, que tem como objetivo forçar a vítima a reutilizar chaves.

Este ataque consiste na autenticação de um cliente numa STA com o bloqueio da 4.ª mensagem do 4WH, que envia o ACK da STA ao AP. Este bloqueio pode ser feito por um MiiM ou através do envio de um pacote em simultâneo pelo atacante, que colida.

Isto vai obrigar o AP a retransmitir a mensagem 3, repetindo a chave. Vamos então ter pacotes diferentes cifrados com chaves iguais, o que numa cifra contínua é um problema.



Os AP podem prevenir este ataque, reiniciando o processo de autenticação no caso da STA não enviar ACK, em vez de reenviar a 3.ª mensagem.

8. Firewalls

Slides teóricos e aula assíncrona

As **firewalls** são elementos indispensáveis na ligação de um domínio de rede, permitindo o controlo dos acessos, fluxo e conteúdos transacionados, através da análise de pacotes.

Controla o **acesso** a recursos da rede, podendo restringir-lhes o acessso.
Controla o **fluxo** dos pacotes, atribuindo determinada largura de banda a cada cliente.
Controla os **conteúdos**, podendo filtrar páginas com base na informação que esta contém.

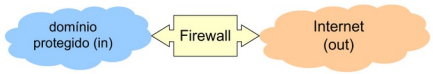
Esta pode ser instalada em **perímetro**, nos pontos de contacto entre a rede interna da organização e o exterior, ou em **profundidade**, se for aplicada de forma generalizada aos vários componentes da rede.

A instalação da firewall de forma centralizada facilita a aplicação das medidas, uma vez que é possível controlar todo o fluxo de comunicações com o exterior num único ponto, em detrimento de análises distribuídas pela rede, que serão mais complexas de configurar e manter.

Mesmo com uma única firewall é possível criar soluções em **profundidade**. Por exemplo, na UA, todas as comunicações entre todos os dispositivos conectados à rede wireless passam primeiro pela firewall.

Foi definida por Cheswick e Bellov como o elo de ligação entre redes, um perímento protegido e uma rede insegura, tomando forma num conjunto de componentes de software e hardware imune à penetração pelo qual passa todo o tráfego de entrada e saída da rede.

Uma vez que se insere no perímetro da rede, será inútil caso o atacante encontre alguma forma de aceder à rede de forma interna.



A supervisão das comunicações permite a introdução de **controlo** do uso de recursos protegidos por máquinas exteriores e do uso da rede exterior pelas máquinas do perímetro protegido e de **defesa** contra ataques externos ao domínio protegido e ataques iniciados no interior do domínia lançados para o exterior.

É ainda possível adicionar-lhe mecanismos adicionais fornecidos por gateways, como **NAT (Network Address Translation)**, que permite a utilização de endereços públicos para codificar endereços privados, ou **VPN (Virtual Private Network)**, que permite estabelecer comunicações privadas sobre redes públicas.

As firewalls dos routers que temos em casa implementam o mecanismo de NAT, pelo que para as máquinas externas à nossa rede não é possível ter noção de quais os dispositivos ligados à rede.

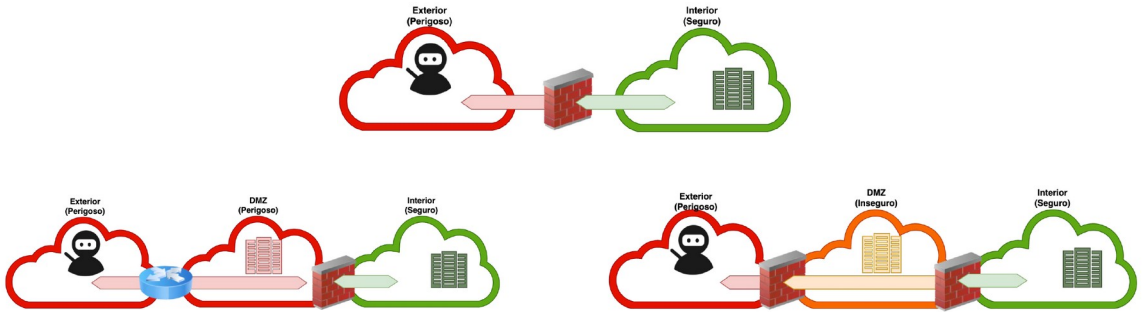
Estrutura

A forma como a firewall é aplicada à rede pode variar. A abordagem mais simples consiste na instalação de uma firewall única no ponto de contacto entre o perímetro protegido e a rede insegura, denominada por **bastião**[11].

Também conhecida por **topologia Dual-homed**. É a mais simples e económica. Pode ou não ter DMZ.

No entanto, quando a organização fornece serviços ao exterior, por vezes separam-se os recursos internos e externos em duas redes, sendo a firewall instalada entre os recursos internos e externos da organização, estando os últimos expostos à rede insegura sem qualquer proteção. Esta zona desprotegida chama-se **DMZ**, **DeMiliatarized Network**, ou **Perimeter Network**.

Há ainda uma alternativa a esta estrutura, que consiste no modelo anterior, mas com instalação de uma firewall adicional na fronteira entre a DMZ e o exterior do domínio da organização. Estas são distinguidas entre a **firewall externa**, mais permissiva que a **firewall interna**, que controla o acesso à rede interna.



Tipos de firewalls

As firewalls podem ser implementadas de várias formas.

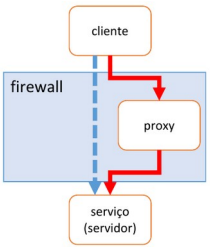
Filtro de pacotes Filtram os pacotes com base no endereço IP, na porta, dimensão dos datagramas…

Não consegue filtrar por nomes de DNS, uma vez que os pacotes apenas têm informação do IP. Ou seja, seria possível bloquear endereços IP do Facebook, mas não pacotes dirigidos ao facebook.com.

Podem analisar comportamentos de fluxos, como *scannings* de portas (com o nmap, p.e.).

Os núcleos dos SO oferecem suporte para este tipo de firewall. Em Linux recorre-se às iptables.

Gateways aplicacionais Controlam interações ao nível da aplicação, através da análise do conteúdo dos pacotes



Este controlo é feito de forma transparente para as aplicações. Normalmente há uma firewall diferente para cada protocolo de comunicação, para os quais o tráfego que entra e sai da rede é redirecionado antes de seguir o seu caminho. Para além do controlo do conteúdo, podem fazer controlo de acessos, registo (logging) detalhado e até proxying (substituição transparente de um dos interlucotores).

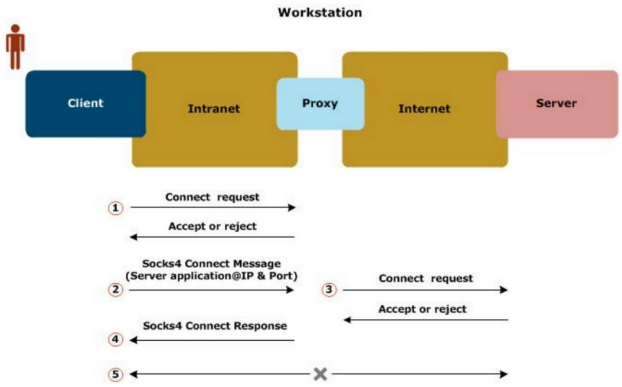
Caso as comunicações sejam cifradas, os proxis podem ter instalado um certificado de uma CA que também está instalado nos clientes, de maneira a que possam decifrar as comunicações para as analisar e voltar a cifrá-las para estas seguirem o seu fluxo.

11 Local de defesa ou núcleo defensor de algo.

Gateways de circuitos Semalhantes às anteriores, mas interposição[12] não é transparante

Contrariamente às gateways aplicacionais, em que o controlo é feito de forma transparente, com as gateways de circuitos, os clientes contactam diretamente o proxy para lhe pedir que faça pedidos para a rede externa. Os circuitos podem ser vistos como ligações.

Normalmente requer a alteração das aplicações cliente, que utilizam protocolos específicos para comunicar como o proxy, como por exemplo o SOCKS ou o Proxy HTTP.



Filtros de pacotes com contexto Permite o fluxo de pacotes dinâmico, com contexto

Também conhecido por Stateful Packet Filter/Inspection (SPI).

Ao introduzir a compreensão dos fluxos por observação dos pacotes transacionados ao longo do tempo (contexto histórico), esta filtragem consegue agregar pacotes em contextos e filtrar com base nestes contextos. Utiliza mais memória, o que pode prejudicar a performance.

Por exemplo o FTP utiliza portas distintas para controlo de ligação de dados, mas numa ligação faz uso das duas.

Outro exemplo são os pacotes fragmentados, que conseguem contornar as firewalls tradicionais, mas aqui são identificados e reconstruídos para serem analisados.

Há ainda o caso das portas, que apesar de convencionadas para os serviços mais utilizados, na realidade estas convenções não têm de ser seguidas. Por exemplo a porta 80, convencionada para HTTP pode ser também utilizada para ligações FTP ou VPN. A filtragem com contexto permite identificar o tipo de pacotes a serem transacionados em cada porta e caso este não corresponda ao tipo de tráfego esperado, bloqueá-lo.

Bastião

O **bastião** é um sistema que implementa a firewall, que por definição tem instalados apenas os serviços essenciais, de forma a diminuir o risto de ataques.

Os servidores públicos não devem executar um bastião, devendo ser executados em máquinas isoladas dentro das DMZs, preferencialmente uma por serviço. Quando nos referimos a máquinas, estas podem ser físicas ou virtuais.

Geralmente é implementado com várias técnicas, sendo muitas vezes plataforma para gateways aplicacionais, para as quais encaminha o tráfego. É fundamental que os gateways aplicacionais sejam **independentes** e **sem privilégios especiais**, de forma a que se forem comprometidos não possam interferir com o normal funcionalmento dos restantes.

12 Ato ou efeito de interpor. Posição entre duas coisas.

A utilização de um bastião único pode gerar problemas em caso de comprometimento do bastião, que vai desativar a firewall e ainda a carga de processamento que lhe é atribuída, que pode levar a que as comunicações com o exterior da rede sejam demoradas quando há grandes fluxos.

Serviços de segurança

As firewalls podem providenciar vários serviços de segurança.

Autorização de fluxos de dados (filtragem de pacotes) e de utentes (gateways aplicacionais/circuitos)

Redirecionamento do tráfego para máquinas dedicadas e representação

Com o redirecionamento para máquinas dedicadas é possível a firewall expor portas no seu endereço IP que são mapeadas internamente para máquinas distintas. Facilita a distribuição da carga de trabalho e esconde a topologia interna da rede, uma vez que de fora só é visível a máquina da firewall com um endereço IP e múltiplas portas. Esta abordagem permite facilmente a substituição do servidor responsável por determinados pedidos em casos de falha ou de necessidade de manutenção. Este redirecionamento pode também ser feito para proxies em security appliances.

No caso da representação esta pode ser explícita como no caso dos gateways de circuitos ou transparente como por exemplo na tradução de endereços NAT.

Processamento de conteúdos aplicacionais através da análise de conteúdos e/ou alteração de protocolos de alto nível

Geralmente utilizados para detetar e eliminar vírus.

Comunicação segura VPNs, tunelling, …

O tunelling consiste na extensão do domínio IP para nós distantes, podendo ser utilizados protocolos como o PPTP, L2TP, IPSec… São como VPNs permanentes, fornecidas como serviço e independentes do utilizador.

Defesa contra tentativas de DoS Deteção de ataques, filtragem de datagramas perigosos, acuinamento de medidas paliativas

Identificação de padrões de volume de tráfego anormais, de volume alto.

Defesa contra fugas de informação Deteção de tráfego anormal contra modelos conhecidos

Limitações

Apesar da diversidade de funcionalidades de proteção da rede providenciados pelas firewalls, na realidade, estas não garantem a total segurança da mesma.

Como já mencionado anteriormente, estas não resolvem o problema dos atacantes dentro da rede interna, a menos que a rede interna esteja segmentada em múltiplas sub-redes.

Nos restaurantes e pequenos serviços é habitual a palavra-passe da rede sem fios ser dada aos clientes. Esta é uma péssima política de segurança, pois, uma vez dentro da rede, um cliente com más intenções consegue capturar o tráfego e intersetar comunicações ou até mesmo atacar serviços vulneráveis. A solução passa pela divisão das redes interna e dos clientes (convidados), que até pode ser uma divisão virtual.

Têm também limitações no controlo de todas as ligações ao exterior, bastando uma WLAN, um AP ou um modem não cadastrado para abrir uma porta a ataques.

Se um funcionário de uma empresa levar um modem 4G para aceder ao Facebook, que é barrado pela firewall da empresa e estiver a utilisá-lo em simultâneo com a rede interna, para aceder aos serviços da empresa, caso o computador esteja infetado com algum tipo de malware, poderá ser feito um ataque através do modem com acesso direto e não controlado pela firewall à rede interna da empresa.

Destaca-se ainda a falta de controlo sobre interações camufladas, através de tunéis de comunicação segura (encriptada, que pode correr sobre imensos protocolos) ou VPNs.

Uma VPN pode ser configurada sobre HTTP, ICMP (pedidos PING), DNS, ou até mesmo HTTPS com web sockets. A firewall não é capaz de filtrar estes pedidos.

Por fim é ainda de notar a dificuldade de adminitrar estes sistemas de controlo em ambientes de interesses heterogéneos.

Em organizações de grande dimensão como por exemplo na Universidade de Aveiro, a diversidade de finalidades para as quais a rede é utilizada exige uma firewall com um conjunto de regras complexo e em constante adaptação. Nestes casos é preciso haver alguém ou mesmo uma equipa dedicado exclusivamente ao controlo e manutenção da firewall.

Firewalls pessoais

Este tipo de firewall foi introduzido para a proteção de máquinas individuais, distinguindo o domínio da máquina do seu exterior e permite definir políticas de defesa em profundidade. São independentes entre si e também da firewall da rede, não fazendo qualquer assunção às demais proteções.

Assumem que a rede não tem qualquer proteção, não confiando verificações de segurança a ninguém.

Pode ser gerida de forma centralizada ou localmente, permitindo a definição das aplicações autorizadas a aceder à rede, com que protocolos e as máquinas/redes que os protocolos/aplicações podem contactar.

É bastante utilizada em máquinas de empresas que migram entre redes (por exemplo um vendedor de uma cadeia de abastecimento). Em caso de penetração na firewall geral ou mesmo acesso indevido a uma máquina no interior da rede, previne o acesso a outras máquinas na mesma rede (acesso horizontal).

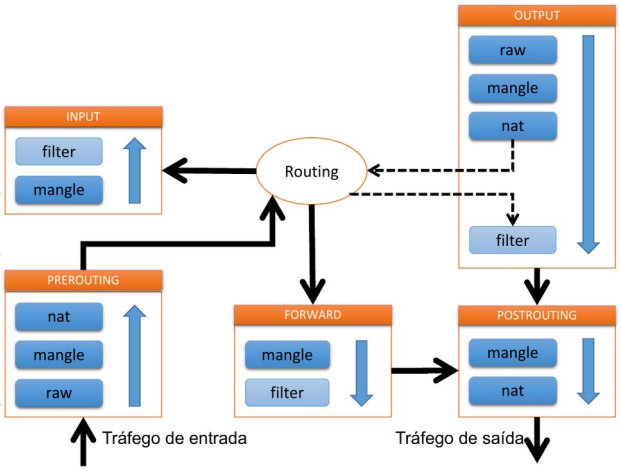
No entanto, se o seu controlo não for realizado de forma centralizada, é provável que tenha um papel pouco útil, uma vez que os utilizadores comuns, confrontados com pedidos de autorização para acesso a determinada porta ou protocolo por determinada aplicação irão provavelmente responder que sim, por falta de conhecimento de causa, sem conseguirem avalidar se a interação é normal e expectável ou suspeita.

É ainda de notar que o bloqueio de interações suspeitas pode anular funcionalidades das aplicações, que na maioria dos casos utilizam a rede sem informarem os utentes das suas necessidades de comunicação.

Tal como nas firewalls de rede, temos ainda o proble a da heterogeneidade, que implica uma grande complexidade operacional ao combinar cenários, interface de rede e interaçõies aceitáveis para cada caso, levam à necessidade de diferentes políticas.

iptables

Este é um dos mecanismos mais comuns de firewalls pessoais, utilizado nos sistemas Linux. Aplicam um filtro de pacotes com contexto, que é integrado com o protocolo TCP/IP no núcleo do SO, podendo ser estendido com módulos no núcleo ou aplicações em modo utilizador (como antivírus).



É este tipo de firewall que é utilizada habitualmente nos routers domésticos.

Tem por base **5 cadeias** de eventos que recorrem a **4 tabelas** e pode ainda ter associados vários **módulos extra**.

As quatro tabelas são:
Raw é o que o wireshark utiliza para fazer captura de pacotes.
Mangle ajusta os pacotes (pode alterar portas, conteúdos, …), otimizando qualidade do serviço
NAT Network Address Translation (altera IP e portas de origem e destino)
Filter Filtra pacotes. É bloqueado tráfego proibido de acordo com as regras.

O tráfego da rede para a máquina local percorre as cadeias de **prerouting** e **input**.

O facto de haver contexto permite que seja autorizada apenas a entrada de respostas a pedidos feitos previamente.

O tráfego reencaminhado faz **prerouting**, **forward** e **postrouting**.

O **mangle** e o **nat** são aplicados em várias cadeias dependendo da informação que há em cada uma delas. No prerouting sabemos por que interface de rede entrou, fazendo operações com base nesta informação. No forward aplicam-se regras genéricas independentes das interfaces e apenas tendo em conta os endereços de origem e destino. Finalmente no postrouting aplicam-se regras com base na interface de saída.

Por exemplo os pacotes de comunicação com o Zoom são direcionados ao nosso router (pois apenas o seu IP é público, o da nossa máquina é privado e só existe na nossa rede, não na internet). Ao entrarem no router, seguem o percurso de encaminhamento, sendo o endereço de destino dos pacotes alterado na tabela **nat** da cadeia de **postrouting**. O mesmo acontece no sentido inverso (em pacotes enviados da nossa máquina para os servidores do Zoom).

O tráfego com origem na máquina local passa pelo **output** e **postrouting**.

O facto de haver contexto permite que seja autorizada a saída apenas do tráfego relacionado com o de entrada.

Para cada pacote podem ser tomadas várias decisões. **ACCEPT** deixa o pacote prosseguir, **DROP** descarta-lo sem mensagem de erro, **REJECT** faz o mesmo mas com mensagem de erro, **CONTINUE** deixa-lo prosseguir na cadeia de validação… O **QUEUE** entrega o pacote a uma aplicação de alto nível (software fora do núcleo) que faz o processamento, podendo alterá-lo por completo e retorna uma decisão ao núcleo, que age de acordo.

9. Sistemas operativos

Slides teóricos e aula assíncrona

Embora possa não parecer, o núcleo é uma parte muito pequena do sistema operativo, tendo como funções a **inicialização do dispositivo**, a **virtualização do hardware** (modelo computacional), **fornecer mecanismos de proteção** (restrições de acesso aos recursos, protegendo contra erros de utilizadores e atividades não autorizadas) e **fornecer um sistema de ficheiros virtual (VFS)**, agnóstico do sistema de ficheiros realmente utilizado.

A **inicialização do dispositivo** passa pela execução do programa bootstrap, que é responsável por inicializar vários dispositivos do sistema (configurar e instalar drivers p.e.), localizando e carregando ainda em memória o núcleo (kernel) do SO e começando a sua execução.

O acesso ao hardware por parte das aplicações é feito através de portas e espaços de memória de comunicação, que são os mecanismos que o SO utiliza para **virtualizar o hardware**. Garante-se assim uma transparência nos acessos, que para as aplicações são independentes do fabricante ou modo de operação, tornando-se uniformes para todos os dispositivos do mesmo tipo.

O **VFS** é mais uma abstração para as aplicações, que interagem com ficheiros virtuais, que são depois mapeados para o sistema de ficheiros utilizado pelo núcleo.

Modos de operação

Atualmente, os sistemas operativos podem ser utilizados em dois modos.

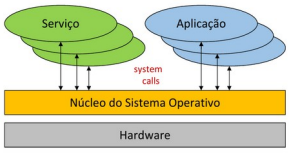
Modo utilizador Execução no modo normal do CPU, sem acesso a instruções priveligiadas

Modo supervisor Execução no modo priveligiado do CPU, com acesso a instruções priveligiadas

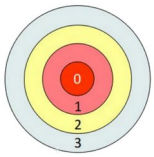
As aplicações são executadas em modo utilizador. No entanto, este modo é bastante restrito, não permitindo comunicação entre aplicações nem com o hardware (acesso ao disco por exemplo). Para estas operações, que necessitam de controlo de acesso e de concorrência, as aplicações fazem pedidos ao kernel, através das **chamadas ao sistema**.

Se uma aplicação puder aceder diretamente ao disco poderia ler ficheiros que não são dela, ou mesmo de outros utilizadores.

Outro exemplo é a captura de pacotes. Em modo de utilizador isto não é possível, por isso é que o Wireshark tem de ser corrido em modo de administrador.



Esta proteção é feita ao nível do CPU, que apresentam vários **anéis de proteção** que restringem as instruções que podem ser executadas. Atualmente os processadores apresentam quatro anéis, correspondendo ao modo supervisor o 0 e o utilizador ao 3.



Dependendo de estarem numa zona de endereçamento aplicacional ou de kernel, as aplicações são executadas nos anéis 3 ou 1, respetivamente.

ATENÇÃO! Ao executarmos uma aplicação em modo de administrador, com sudo, na realidade a aplicação continua a ser executada no nível 3. Simplesmente, as suas chamadas ao sistema passam a ser feitas por um utilizador com mais permissões.

Máquinas virtuais

Sendo uma aplicação, uma máquina virtual vai ser executada em modo de utilizador. Assim, o código em modo de utilizador é executado diretamente, mas o código priveligiado tem de ser alvo de uma tradução binária de modo a convertê-lo em instruções que possam ser executadas no anel 3. Esta tradução pode ser demorada e leva a perdas de eficiência comparativamente coma execução direta no nível 0.

Para corrigir esta perda, foi introduzido suporte no hardware dos CPU através da criação do nível -1 para o núcleo do SO, passando o hardware a poder ser virtualizado no anel 0. Neste caso, deixa de ser necessária a tradução binária.

A sua utilização permite implementar um mecanismo especial para a segurança, o **confinamento**, que para além de oferecer uma abstração quanto ao hardware em que corre, implementa um domínio de segurança restrito para um conjunto de aplicações.

Em vez de correrm os várias aplicações num único servidor podemos correr cada aplicação numa máquina virtual desse servidor. Assim, em caso de comprometimento de alguma, apenas a sua máquina virtual poderá ser adulterada, uma vez que o atacante não terá acesso ao SO principal nem às restantes máquinas virtuais.

Modelo computacional

O seu **modelo computacional** define como as aplicações e utilizadores interagem com o núcleo, sendo estas definições universais para cada sistema operativo e estáveis ao longo do tempo.

É esperado que uma aplicação desenvolvida à 15 anos em Linux possa ser corrida neste sistema operativo hoje.

Identificadores de utilizadores e grupos

Para o SO, os **utilizadores** são identificados por um número, o **UID**, que é associado a todas as atividades executadas e a partir do qual o núcleo lhes dá permissões ou não.

No Linux e Android o UID 0 é omnipotente[13]. No macOS o UID 0 é omnipotente apenas na gestão, havendo módulos críticos do núcleo que apenas podem ser manipulados com assinaturas da Apple. O Windows funciona por privilégios que são atribuídos aos utilizadores.

Cada utilizador pode ainda pertencer a vários **grupos**, também identificados por um número, o **GID**, que se define por um conjunto de utilizadores, aos quais é atribuído um conjunto de permissões.

$$Permissões_{utilizador} = Permissões_{UID} + Permissões_{GIDs}$$

Um utilizador tem de pertencer ao grupo sudo para fazer sudo, caso contrário não lhe serão dadas permissões de execução do ficheiro binário por parte do sistema operativo.

Processos

Os **processos** contextualizam atividades, ou seja, operações de leitura, escrita e execução (RWX) sobre recursos. Também são identificados por um número, o **PID**, que são associados à identidade de quem os lançou, herdando as permissões que o UID e os GID do utilizador lhe conferem.

Mas então porque é que quando uma aplicação é executada com sudo ganha permissões adicionais? Porque na realidade, é possível que um processo, quando é executado, embora se saiba que pertence a um utilizador, ele seja executado com permissões de outro utilizador.

Este tema será explorado no próximo tópico sobre "Controlo de acessos".

Memória virtual

Quando são executados, os processos são mapeados para um segmento de memória (página), do qual fazem uso como se da memória RAM se tratasse. Na realidade esta é uma **memória virtual**, que dá a ilusão de que as aplicações têm toda a memória RAM disponível para si, mas que na realidade partilham com outras aplicações.

É responsabilidade do núcleo do SO fazer o mapeamento desta memória virtual para a memória física (RAM) quando é necessário ler ou escrever. Esta gestão deve ainda evitar a fragmentação da memória e a alocação de recursos, podendo armazenar em memória física as memórias virtuais de aplicações que não estão em utilização.

13 Que pode tudo. Que tem poderes ilimitados.

Sistema de ficheiros virtual (VFS)

Os **VFS** fornecem um método para representar pontos de montagem, diretórios, ficheiros e links.

Ponto de montagem é um acesso à raiz de um file system específico

Em Windows é dado por X: sendo X uma letra. A: e B: era utilizados para as disquetes. Hoje em dia a "contagem" começa no C:, que é habitualmente o ponto de montagem do disco rígido do computador.

Em Linux no /dev no /cd-rom temos vários pontos de montagem.

Diretórios são métodos de organização hierárquica. Podem conter pontos de montagem, diretórios, ficheiros e links.

Em Linux apresenta uma estrutura hierárquica em árvore que parte do '/', denominado por raiz.

Links são mecanismos de indireção no FS

Soft links apontam para outros recursos em qualquer FS, no mesmo VFS

Hard links fornecem múltiplos identificadores para o mesmo conteúdo no mesmo FS

Em Windows os atalhos são soft links ao nível aplicacional.

Em backups sucessivos, os ficheiros não modificados podem ser representados em backups posteriores com hard links para os anteriores, permitindo a poupança de espaço.

Ficheiros são sequências ordenadas de bytes associadas a um nome, que permitem o armazenamento de dados de forma permanente. Possuem uma proteção que controla o seu uso (permissões de RWX).

Na realidade, a longevidade dos ficheiros depende do suporte físico onde estes são armazenados. Na memória RAM já sabemos que o seu tempo de vida é limitado e termina quando se desliga o computador.

Os ficheiros têm **mecanismos de proteção mandatórios** que definem o seu dono, os utilizadores e grupos permitidos e permissões de leitura, escrita e exeução. Podem ainda ser definidos mecanismos de proteção adicional.

Canais de comunicação

Os **canais de comunicação** permitem a troca de dados entre atividades distintas mas cooperantes, permitindo comunicações internas entre aplicações (pipes, streams, sockets), entre aplicações e o núcleo (syscalls e sockets) e externas (sockets TCP/IP e UDP/IP).

Controlo de acessos

O **núcleo do SO** é o **monitor do controlo de acesso**, pelo qual passam todas as interações com o hardware e entre entidades do modelo computacional.

Para visualizar as chamadas de sistema feitas por um determinado programa pode recorrer-se à aplicação **strace**.

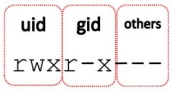
Existem dois tipos de controlos de acesso. Os **mandatórios**, que fazem parte da lógica do modelo computacional e por isso não são moldáveis pelo utilizador e os **discricionários**[14], que podem ser definidos pelo utilizador.

Por exemplo é mandatório que o root (UID 0) possa fazer tudo em Linux. Esta configuração está embebida no núcleo e a menos que este seja alterado, vai ser uma regra imutável.

ACL (Access Control List)

As **ACL** são metadados que consistem em listas expressivas que limitam o acesso a recursos. Há uma para cada objeto com controlos obrigatórios e discricionários que são verificados pelo núcleo do SO sempre que uma atividade os tenta manipular.

Esta lista atribui 3 tipos de direitos a 3 entidades: leitura, escrita e execução ao dono do ficheiro (UID), ao seu grupo (GID) e aos demais.



O dono de um ficheiro tanto pode ser um utilizador como um grupo.
Nas pastas, a leitura consiste em listar o seu conteúdo, a escrita em adicionar/remover ficheiros ou subdiretórios.

Na realidade podem ser definidas ACL discricionárias para além desta lista de 3x3. Neste caso, nas permissões retornadas pelo comando **ls -l** terão um + depois da lista da ACL. Para as consultar, recorre-se ao comando **getfacl** <filename> (get file ACL).

Elevação de privilégios

Como foi mencionado anteriormente, é possível que um processo, quando é executado, embora se saiba que pertence a um utilizador, seja executado com permissões de outro utilizador.

Isto deve-se ao facto de na prática, existirem o **real UID**, que é o UID de quem o chamou, e o **effective UID**, que pode ser diferente do anterior e que é utilizado para conferir as permissões ao processo.

Esta elevação de privilégios é feita por causa do **set-UID**, uma configuração nas permissões do ficheiro que faz que este seja executado com o effective UID do dono do ficheiro, em vez do UID de quem o executa, como aconteceria normalmente.

$$setUID = 0 \Rightarrow rUID = eUID = UID\ do\ utilizador\ que\ executa\ o\ processo$$

$$setUID = 1 \Rightarrow rUID = UID\ do\ utilizador\ que\ executa\ o\ processo \wedge eUID = UID\ do\ dono\ do\ ficheiro$$

Esta configuração pode ser definida por **chmod u+s** (**g+s** no caso de set-GID). Na ACL, em vez de x na permissão de execução do utilizador, aparece s.

14 Deixado à descrição. Livre de condições.

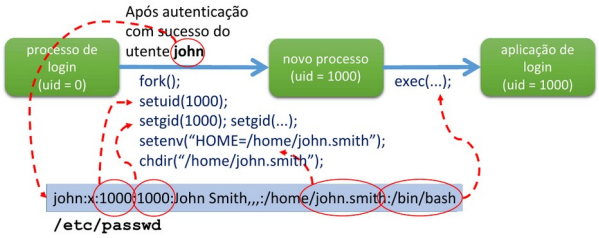
Apesar de poder ser alterada pelo utilizador, é uma configuração intrínseca ao sistema operativo, pelo que é classificado como um controlo de acesso mandatório.

É através desta configuração que podemos executar programas como o **passwd** para alterar palavras-passe, o **ping** que cria uma raw socket (apenas permitida ao root) ou o **sudo**, que permite executar uma aplicação com um eUID diferente. Todas estas aplicações têm como dono o root e são executadas com o seu UID.

Também existe o **set-GID** que funciona de forma análoga mas para o GID em vez do UID e rGID e eGID em vez de rUID e eUID.

Login

A autenticação nos sistemass operativos não é feita pelo núcleo, mas sim por uma aplicação priveligiada que corre como administrador e após validação das credenciais obtém os UID e GID apropriados, inicializando uma aplicação num processo com esses identificadores (em Linux é um shell).



A validação da senha é feita com recurso à sua síntese, armazenada no ficheiro /etc/shadow, que só pode ser lido pelo root.

Todos os processos criados pelo utilizador vão ser forks deste, pelo que vão herdar estes identificadores.

Quando o processo pai termina, a aplicação de login reaparece.

Sudo

Este comando permite executar as aplicações com eUID de outro utilizador, mas mantendo algum controlo sobre quem executa o quê, uma vez que apesar de modificar o eUID, o rUID continua a ser o do utilizador que está a executar o programa.

Por defeito executa com o UID do root, mas é possível passar-lhe como argumento o UID do utilizador que queremos utilizar como eUID e assim executar um programa com as permissões de outro utilizador.

O comando **id** permite consultar os UID e GID reais. Ao executarmos **id** e **sudo id** vamos obter valores diferentes, sendo que no primeiro são mostrados os nossos ID e no segundo os do root (UID 0).

Chroot

Contrariamente aos anteriores, este comando reduz os acessos, limitando a visibilidade do sistema de ficheiros.

Cada processo tem no seu descritor o número do i-node raiz, a partir do qual são devolvidos os caminhos absolutos. O chroot permite mudar esse número para referir o i-node de outro diretório arbitrário.

Chroot significa change root.

Permite confinar um programa a um conjunto de diretórios mais restritos que todos os da máquina, sendo por isso bastante útil em servidores públicos ou aplicações descarregadas.

Confinamento

Linux Apparmor

Este mecanismo define regras que limitam as system calls de aplicações independentemente do utilizador, mesmo que este seja o root.

As suas configurações são armazenadas em /etc/apparmor.d.

Permite a criação de confinamento para as aplicações através de **sandboxes**, um ambiente controlado com contexto isolado, onde só podem ser invocadas as atividades (system calls) definidas no ficheiro de configuração.

MasOS sandbox

O confinamento através de **sandboxes** é aplicado no iOS de forma universal a todas as aplicações, que nos seus binários têm descrito o seu comportamento esperado, pelo que uma aplicação não poderá fazer chamadas ao sistema fora do que está pré-determinado.

Namespaces

Consistem em **partições** (ambientes virtualizados) definidas dentro do **sistema operativo** que possuem uma vista restrita do sistema, nas quais podem ser disponibilizados parte dos seus recursos.

Dentro de um namespace, um processo tem a ilusão de que o sistema operativo é o interior do namespace, não conseguindo ver fora dele.

Permite por exemplo definir regras de firewall dentro do namespace que não são aplicáveis ao restante SO, ou mover uma interface de rede para dentro do namespace e restringir a sua utilização ao seu interior.

O Docker, ao gerar **containers**, não está a fazer mais do que a fazer uma orquestração ao nível aplicacional dos mecanismos de namespace e confinamento providenciados pelo núcleo do SO de forma a criar um processo isolado do restante sistema.

10. Armazenamento

Slides teóricos e aula assíncrona

O armazenamento tem a si associados vários problemas, como as **avarias** dos sistemas, a **lentidão** no acesso à informação ou os **acidentes** que levam à perda total dos dados.

Os dois primeiros são garantidos, sendo as **avarias** não uma questão de tempo.

Para os evitar há várias soluções que podem ser implementadas.

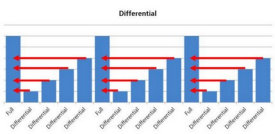
Cópias de segurança

Consistem em cópias periódicas dos dados que podem variar nos diferenciais do espaço que ocupam e do tempo que levam a ser recuperados, inversamente relacionados.

Completos imagem completa da informação

Permite recuperação rápida, mas ocupa muito espaço.

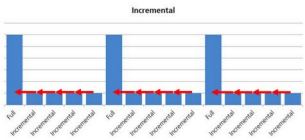
Diferenciais diferenças desde o último backup completo



A recuperação é mais lenta, mas o espaço é reduzido.

A periodicidade deve ter em conta o contexto e ser intercalada com o backup completo periódico, de modo a garantir uma gestão o mais eficiente possível do espaço.

Incrementais diferenças desde o último backup



A recuperação é muito mais lenta, mas também ocupa muito menos espaço que as anteriores.

O armazenamento da informação é feito num **disco adicional**, que não deve ser confundido com um disco externo!

Têm a si associado um conjunto de **políticas**, **mecanismos** e **processos** para realizar, manter e recuperar as cópias que devem resistir a várias situações de falha e atuar em situações de catástrofe. Devem ainda contemplar o enquadramento legal de expiração dos dados pessoais.

Compressão

Mesmo nas abordagens que ocupam menos espaço é possível implementar técnicas adicionais que permitem a sua minimização.

Compressão por algoritmos sem perdas (ZIP)

Cópia seletiva da informação

Copiar apenas ficheiros do utilizador, ignorando logs ou ficheiros do SO.

Deduplicação Cópias totais com processo de redução posterior

No processo de redução os ficheiros duplicados no último backup em relação aos anteriores são apagados e substituídos por um ponteiro (hardlink) para o ficheiro já existente. Isto requer bastante processamento e pode requerer congelamento do sistema para garantir a consistência.

Precisa de sistemas de ficheiros específicos. No Linux e no MAC funciona, mas no Windows o sistema não é compatível. Os backups do MAC são feitos com base nesta técnica.

Níveis dos backups

Os backups podem ocorrer a vários níveis.

Aplicacional Extração dos dados da aplicação

Esta abordagem é bastante comum nas bases de dados, uma vez que ao fazer a extração através da aplicação se consegue garantir uma maior consistência, uma vez que é possível o bloqueio temporário das escritas.

Ficheiros Cópia dos ficheiros individuais

Permite copiar qualquer aplicação sem necessidade de a contactar para a extração. No entanto, isto pode levar à cópia de ficheiros **corrompidos** por estarem a ser escritos naquele momento ou **inconsistentes** por estarem abertos em memória com dados não escritos em disco.

Sistemas de ficheiros Mecanismos próprios do sistema de ficheiros

Estes backups são feitos no próprio sistema de ficheiros, fazendo uso do espaço livre em disco. Apesar de não resistirem a falhas do disco permitem a recuperação de ficheiros manipulados erroneamente pelo utilizador (por exemplo apagados sem querer).

Blocos Cópia dos blocos de suporte ao armazenamento

É agnóstico do sistema de ficheiros e do sistema operativo, sendo realizado de forma **transparente** e **sem impacto** pela infraestrutura de armazenamento.

Local da cópia

As cópias de segurança podem ser feitas em diferentes locais, permitindo responder a diferentes problemas com diferentes tempos de recuperação e redundância.

Mesmo volume ou sistema

Permite uma recuperação rápida de ações indevidas. Não protege contra avarias.

O OS X TimeMachine ou o GIT são exemplos destes sistemas.

Outro sistema na mesma infraestrutura

Recuperação rápida (menos que anterior) contra falhas isoladas do armazenamento, mas não eventos de maior âmbito que afetem a infraestrutura, como incêndios ou inundações.

O Apple Time Capsule é um exemplo.

Sistema remoto (off-site)

Transporte dedicado por rede ou mesmo fisicamente para um local seguro que permite a recuperação em caso de eventos com grande impacto como incêndios, roubo… A recuperação de informação é muito mais lenta!

A Amazon permite a recuperação de grandes volumes de informação com o envio de discos físicos.

Seleção do equipamento

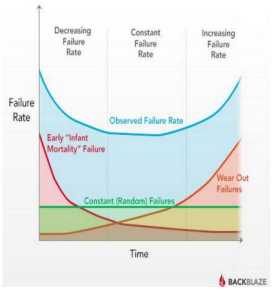
Esta é outra escolha importante no que toca ao armazenamento, devendo ter em conta a **qualidade** e **mecanismos de recuperação** através de métricas como o **MTBF (Mean Time Between Failures)**.

A BackBlaze, uma empresa de armazenamento em cloud, publica periodicamente relatórios de desempenho para os discos que utiliza, contemplando os mais variados modelos das mais variadas marcas.

É ainda importante que os discos sejam **adequados ao contexto** em que se inserem, podendo ser write ou read intensive ou mixed-usage e ainda ao **nível de desempenho** que geralmente está inversamente relacionado com a capacidade.

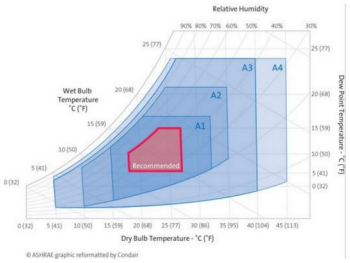
Os discos SSH têm um alto desempenho e uma baixa capacidade e os HHD são o inverso.

As falhas dos equipamentos são mais incidentes no início do seu ciclo de vida, por defeitos de fabrico e no final, por desgaste. Quando não manifestam problemas no início, geralmente mantêm-se estáveis durante o tempo “médio” de vida.



Ambientes controlados

Características do ambiente como a **temperatura** ou a **humidade** influenciam a durabilidade dos sistemas de armazenamento, sendo por isso comum o controlo destes fatores com recurso à climatização dos espaços.



Há um trade off entre o custo da climatização e a durabilidade, que representa uma poupança na compra de armazenamento por avarias causadas pelas características ambientais.

Armazenamento redundante

RAID4 com base em https://www.youtube.com/watch?v=cxkzViLf-6Q

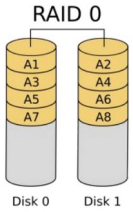
A solução mais comum para a redundância no armazenamento é a tecnologia **RAID (Redundant Array of Inexpensive Drivers)**, que consiste na **combinação lógica de discos** de forma a que possamos otimizar o sistema de armazenamento com o objetivo de oferecer mais **redundância** e/ou **velocidade**.

É uma abordagem de baixo custo que se caracteriza pela utilização de **hardware barato e falível**. **Não substitui os backups**, porque não armazena múltiplas cópias dos dados, sendo vulnerável a falhas catastróficas e não tolerante a erros dos utentes ou do sistema.

Os discos podem ser combinados de várias formas. Os mais usados são os RAID 5 e 6.

RAID 0 (stripping) Velocidade

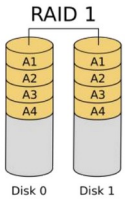
Informação lógica de um volume é subdividida em fatias (stripes), intercaladas em vários discos, que podem ser acedidos em paralelo.



Aumenta a velocidade dos acessos até N vezes para N discos.
A probabilidade de perda de informação e do número de dispositivos.

RAID 1 (mirroring) Tolerância a falhas

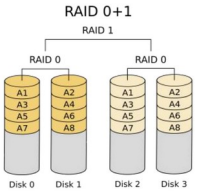
Informação é duplicada pelos discos, com escrita sincronizada e leitura com comparação ou apenas de um disco.



Diminui a probabilidade de perda de informação. Com a leitura de apenas um disco, a velocidade de leitura aumenta.
No entanto verifica-se um desperdício da capacidade de armazenamento (em pelo menos 50% para 2 discos) e o aumento do número de dispositivos.

RAID 0+1 Desempenho e tolerância a falhas

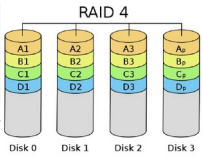
Combina as soluções anteriores de forma a oferecer desempenho e tolerância a falhas.



Mantém os contras do aumento do número de dispositivos e do desperdício da capacidade de armazenamento.

RAID 4 Desempenho e tolerância a falhas

Armazena os dados divididos (striped) por N-1 discos (velocidade), mantendo um disco para armazenar a **paridade** (tolerância a falhas).



Oferece a tolerância a falhas do RAID 1, mas com um desempenho e espaço próximos do RAID 0.

O disco de paridade mantém a combinação dos dados dos restantes discos (por exemplo o resultado de uma operação de XOR).

A tolerância a falhas é garantida, pois, sempre que um disco falhe, os N-1 restantes podem ser utilizados para recuperar os seus dados. Por exemplo, através de uma operação de XOR. Esta é no entanto uma operação mais demorada que no RAID 1, que normalmente exige hardware dedicado.

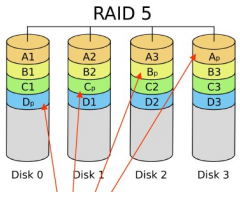
A velocidade de leitura é próxima do RAID 0 porque podem ser lidos dados de vários discos em paralelo. A escrita é no entanto pouco eficiente, uma vez que para escrever num disco é necessário ler os blosos correspondentes nos

restantes discos para gerar um novo bloco de paridade e escrever este último no disco de paridade. Portanto uma escrita implica na realidade 2 escritas e N-2 leituras.

Necessita de um mínimo de 3 discos, 2 para a divisão dos dados 1 para a paridade (redundância).

RAID 5 Desempenho e tolerância a falhas

Similar ao RAID4, mas divide blocos de paridade por todos os discos.



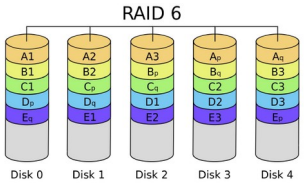
O desperdício de espaço é igual ao do RAID 4, mas em vez de termos um disco exclusivo para paridade, para cada bloco, o bloco de paridade é armazenado num dos outros N-1 discos qualquer.

As leituras podem continuar a ser feitas em pararelo. As escritas tornam-se mais eficientes, pois apesar de serem na mesma necessárias 2 escritas e N-2 leituras, as escritas não são sempre feitas sobre o mesmo bloco de paridade, pelo que pode ser possível fazer mais do que uma escrita em paralelo (desde que os discos a escrever e os respetivos discos de paridade não colidam).

Necessita de um mínimo de 3 discos, 2 para a divisão dos dados 1 para a paridade (redundância).

RAID 6 Desempenho e tolerância a falhas

Procura melhorar a fiabilidade do RAID 5, com a duplicação dos blocos de paridade, o que permite a falha de dois discos.



O desperdício do espaço é o dobro do RAID 5 (2 discos para paridade).
A concorrência das escritas é ligeiramente pior que o RAID 5, uma vez que precisa de escrever em 3 discos (disco a escrever e 2 de paridade) e ler N-3 (os restantes).

Necessita de pelo menos 4 discos, 2 para a paridade e 2 para a divisão dos dados.

Domínios de armazenamento

Tipicamente os servidores e os sistemas de armazenamento não armazenam os dados diretamente, mas fazem uso de **domínios de armazenamento** redundantes (com RAID) ligados em rede.

NAS (Network Attatched Storage)

Sistema disponível por rede, com vários discos RAID.

SAN (Storage Area Network)

Conjunto de sistemas disponíveis por rede, que pode implementar qualquer esquema de redundância.

O NAS é uma solução de mais pequena escala e por isso mais económica. O SAN pode atingir os milhões de euros.

Estas soluções permitem centralizar políticas de armazenamento, facilitando o acesso aos dados com uma interface normalizada e independente do armazenamento real.

Não têm backups!

Confidencialidade do armazenamento

Em capítulos anteriores foi abordado o controlo de acessos a ficheiros, um conjunto de mecanismos de **proteção lógica** aplicados pelo sistema operativo que gerem os acessos a um **dispositivo físico** confinado àquele dispositivo.

No entanto, estes limitam-se à fronteira do SO, pelo que há um número de situações em que esta proteção é irrelevante, como no caso de acesso direto e físico aos dispositivos ou através de mecanismos de controlo de acesso.

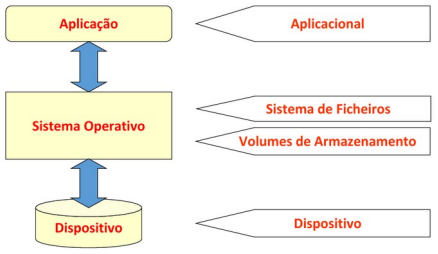
Um disco de um computador onde os seus acessos são altamente controlados pode ser montado por um administrador noutra máquina sem restrições, passando os seus dados a ser completamente acessíveis.

Os utilizadores root têm o poder de aceder a tudo e de personificarem outros utilizadores.

O **armazenamento distribuído** também se tem revelado um problema, pois requer um número maior de administradores muitas vezes anónimos com autenticação remota, o que torna o controlo de acessos mais difícil e complexo.

A solução para estes problemas é a **cifra da informação**, que permite não só a disponibilização segura sobre uma rede insegura como também o armazenamento em meios inseguros.

Ainda assim esta não é uma solução perfeita, impondo também alguns constrangimentos como a perda das chaves, que leva à perda da informação, a dificuldade da partilha de ficheiros, a interferência com tarefas comuns de administração como a análise, deduplicação, indexação e ainda o espaço ocupado.



A partilha de ficheiros, de uma forma direta, implica a libertação dos ficheiros ou das chaves. Podem ser implementados mecanismos mais complexos no entanto.

Uma boa encriptação dos dados gera cifras diferentes para as mesmas entradas, pelo que ficheiros iguais (cifrados) não podem ser agregados através da deduplicação.

A cifra dos dados pode ser aplicada em diferentes níveis.

Nível aplicacional

O processo de cifra é gerido por uma aplicação autónoma.

É difícil partilhar ficheiros internos ao pacote cifrado, porque implica a decifra, extração e nova cifra.

São exemplos o RAR, ZIP, 7zip….

O TrueCrypt era uma aplicação que pemitia a cifra ao nível aplicacional e que trabalhava com volumes que eram ocultados quando cifrados. Destacaou-se pelo seu mecanismo de **negação plausível**, pois para um conjunto de dados permitia a criação de vários volumes não detetáveis com diferentes chaves. Assim, em caso de coação do administrador a fornecer as chaves, este podia apenar fornecer a chave para decifrar volumes com informação pouco crítica e os coatores não teriam maneira de saber se todos os volumes foram decifrados ou apenas parte.

Sistema de ficheiros

O sistema de ficheiros cifra os dados dos ficheiros de maneira a que quando há escritas ou leituras só seja concedido acesso a quem tem as chaves corretas.

O processo de cifra é feito no núcleo do SO, que gere também as chaves.
Os ficheiros decifrados são guardados em memória.

São exemplos o CFS (Cryptographic File System), EFS (Encrypted File System) e NTFS (NT FileSystem).

Nível do volume

Cifra ao nível do volume/partição. Suporta qualquer sistema de ficheiros e é transparente para o SO, sendo apenas necessário um controlador.

Não permite a diferenciação entre diferentes utilizadores, uma vez que uma única chave desbloqueia todo o volume.
Também não fornece controlo de integridade, uma vez que o espaço para guardar a cifra é examente o mesmo dos dados originais, pelo que não há espaço para guardar informação adicional.

Protege contra roubo ou perda do equipamento.

São exmplos o LUKS (Linux), BitLocker (Windows) ou FileVault (iOS).

O BitLocker implementa uma cifra em bloco com uma chave derivada a partir da chave mestra com o identificador de cada bloco, de forma a que cada bloco seja cifrado com uma chave diferente.

Implementou ainda um difusor que entretanto foi removido, mas que tentava contornar ataques de manipulação de bits do criprograma, que por análise às zonas de memória de várias máquinas conseguiam manipular os criptogramas com objetivos específicos que permitiam contornar o controlo de acesso.

Nível do dispositivo

Mecanismo de cifra aplicado internamente pelo disco em firm/hardware. O dispositivo é desbloquado no boot e a cifra é mais uma vez transparente para o SO.

Não se perde performance. Quando o dispositivo é desbloqueado, dados ficam acessíveis a todos os SO e utilizadores.
Erros no processo de cifra são difíceis de detetar e corrigir.

Possuí duas chaves, uma para cifrar os dados **Media Encryption Key** e outra para a cifrar, a **Key Encryption Key**, fornecida pelo utilizador.

Na prática, quando é ligado, o disco mostra apenas o **shadow disk**, uma porção de 100Mb que tem software para o desbloqueio. Quando é desbloqueado, o disco decifra o MEK e altera a sua geometria, concedendo acesso ao **real disk**.