Administração de Sistemas (ASIST)

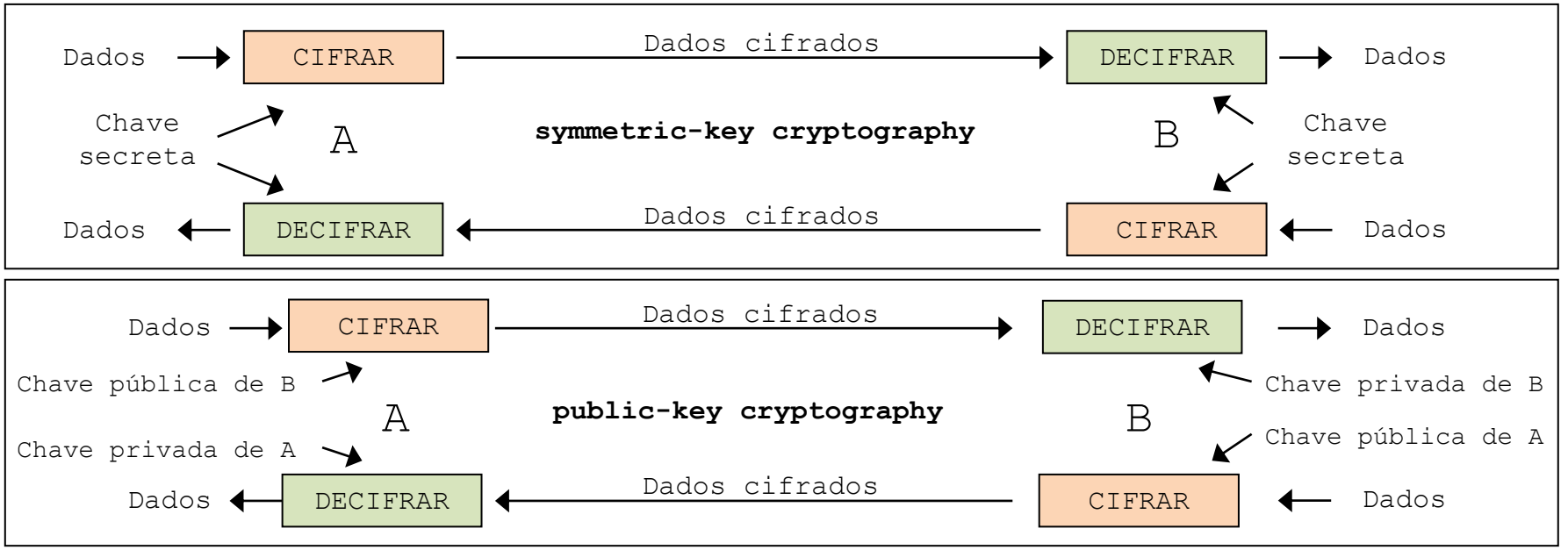
Aula Teórico Prática 09 - 2018/2019

Segurança em rede – Autenticação e privacidade.
Certificados de chave pública.
Cópias de segurança e arquivo.

Privacidade

O ambiente de rede, em especial em redes WAN, caracteriza-se pela grande dificuldade em controlar o acesso aos dados que nela circulam. A única alternativa para garantir a privacidade é recorrer a algoritmos de cifragem, o acesso aos dados não é impedido, mas os dados serão ilegíveis para entidades ilícitas.

A abordagem tradicional de chave simétrica (***symmetric-key cryptography***) exige que os dois intervenientes possuam uma mesma chave secreta (pré-partilhada). A criptografia de chaves assimétricas, também designada de chave pública (***asymmetric-key or public-key cryptography***), usa chaves diferentes para cifrar e decifrar e com isso torna a distribuição de chaves muito mais simples.



Autenticação - Symmetric-key cryptography

A criptografia torna-se totalmente irrelevante se não for aplicada juntamente com procedimentos de autenticação. Os procedimentos de autenticação em rede destina-se a garantir que as trocas de informação através da rede estão a ser realizadas com a contraparte pretendida e não com um intruso (ataques ***man-in-the-middle***). Sem esta validação, o facto da informação ser cifrada deixa de ser garantia de privacidade.

Symmetric-key cryptography

Se efetivamente temos como pressuposto que apenas as duas entidades lícitas conhecem a chave secreta, então a autenticação está desde logo garantida. Note-se que o problema fica neste caso relegado para um processo anterior de distribuição segura da chave secreta.

A distribuição de chaves secretas pode ser realizada de várias formas:

- **Manual**: o administrador define manualmente a mesma chave nos dois nós de rede.
- **Gerada nos dois nós**: ambos os nós usam um mesmo algoritmo para produzir uma chave a partir de algo que apenas ambos conhecem, por exemplo a password de um utilizador ou o respetivo *hash*. Normalmente no processo de geração da chave nos dois nós é também incluído um elemento aleatório produzido por um deles (*challenge* ou *salt*).

- **Utilização de um canal seguro:** a chave secreta é gerada por um dos nós e transmitida ao outro através de um canal seguro (autenticação e privacidade) pré-existente, por exemplo um canal de comunicação que utiliza criptografia de chave pública. A motivação para este procedimento é o facto de os algoritmos de ***symmetric-key cryptography*** serem muito mais rápidos do que os algoritmos de ***public-key cryptography***.
- **Utilização de um centro de distribuição de chaves:** um KDC (*Key Distribution Center*) é um serviço de rede que funciona como intermediário para o estabelecimento de uma chave secreta entre dois nós. Para isto ser possível é necessário que ambos os nós possam um canal seguro (autenticação e privacidade) estabelecido com o mesmo KDC. Uma das implementações mais divulgadas deste tipo de sistema é o Kerberos, usado nos sistema Linux e no *Active Directory*.

Asymmetric-key or public-key cryptography

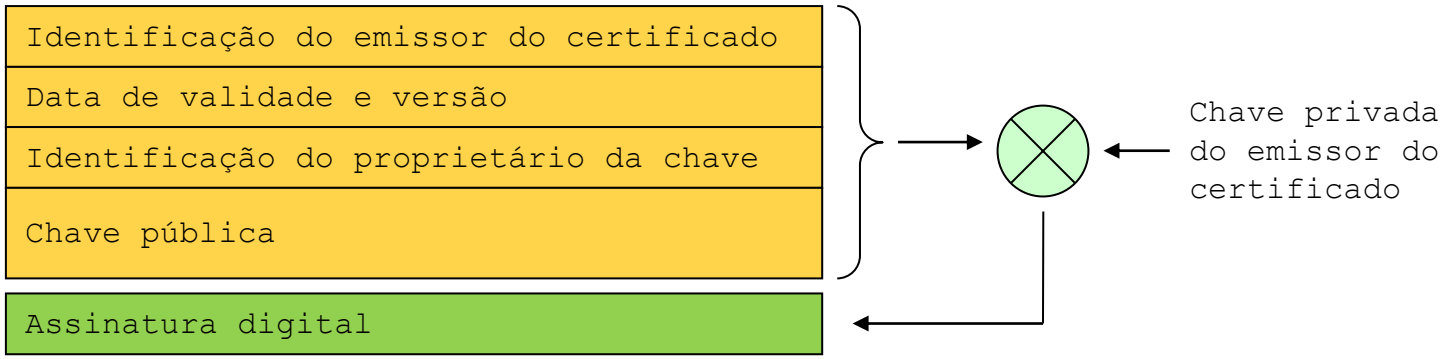
A autenticação não é garantida porque a chave usada para cifrar é pública, logo qualquer um a pode usar.

No entanto como a chave privada correspondente é conhecida apenas pela contraparte, mais ninguém poderá decifrar. Atendendo a que as transações vão ocorrer nos dois sentidos entre os dois nós, se cada nó tiver a certeza absoluta que está a usar a chave pública da contraparte, a autenticação fica garantida.

Certificados de chave pública

Ao usar um algoritmo de chave pública para cifrar uma mensagem temos a garantia que apenas o detentor da chave privada correspondente a poderá decifrar, consequentemente garantir a autenticidade dos nós resume-se a garantir a autenticidade das chaves públicas que usamos.

A autenticidade de uma chave pública pode ser comprovada através de um **certificado de chave pública**. Um certificado de chave pública é um bloco de dados que contém: a identificação do proprietário da chave (***Subject***), normalmente será um nome DNS de um nó de rede; a identificação do emissor do certificado (***Issuer***) que será uma entidade certificadora, a chave pública propriamente dita e uma assinatura digital dos elementos anteriores produzida pela entidade certificadora:



O certificado será considerado válido se a entidade certificadora é considerada fidedigna, ou seja se existe confiança na entidade certificadora. Conhecendo a chave pública da entidade certificadora pode facilmente comprovar-se se a assinatura foi produzida por ela.

Exemplo:

Entidade certificadora

Datas de validade

O proprietário da chave pública que se está a certificar, tipicamente o nome DNS do servidor

A chave pública que se está a certificar

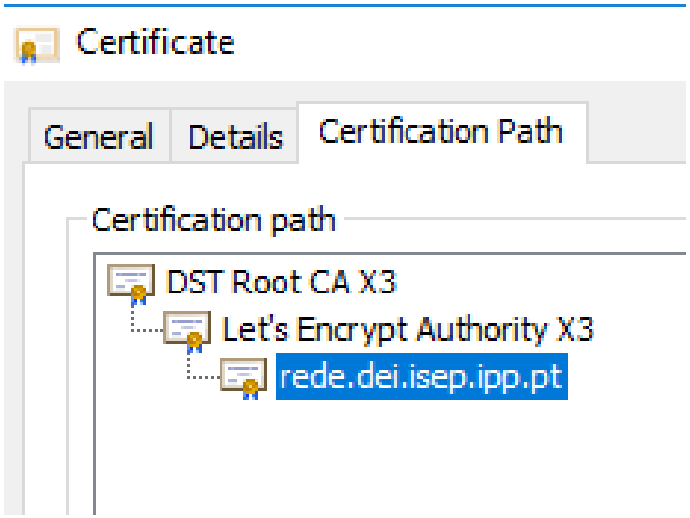
Assinatura digital da entidade certificadora

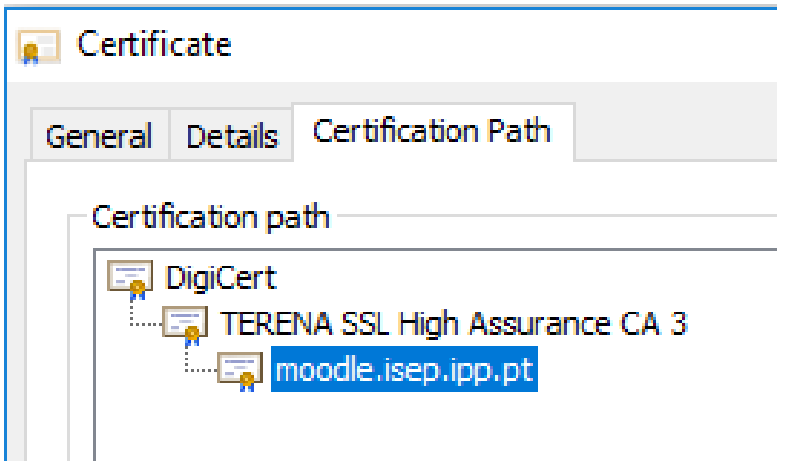
```
[root@server ~]# openssl x509 -in /etc/cert/redecert.pem -text
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 2 (0x2)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=PT, ST=PORTO, L=PORTO, O=DEI.ISEP, OU=DEI,
      CN=DEICA/emailAddress=dei@dei.isep.ipp.pt
        Validity
            Not Before: Jun 20 11:37:28 2007 GMT
            Not After : Jun 17 11:37:28 2017 GMT
        Subject: C=PT, ST=PORTO, L=PORTO, O=DEI.ISEP, OU=DEI,
     CN=rede.dei.isep.ipp.pt/emailAddress=dei@dei.isep.ipp.pt
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:cc:38:e2:e6:5d:ff:43:6d:ff:43:12:94:03:db:
                    8f:5c:28:61:1a:61:0d:d5:5e:04:dd:20:a9:d8:99:
                    0c:53:e3:c6:23:b0:6c:4d:fe:6b:9e:b8:00:ea:23:
                    1d:55:fa:e7:9a:9b:1b:fa:ef:e0:0d:2c:e0:e1:43:
                    31:2d:b1:37:b5:27:68:01:e0:3d:d8:bf:96:15:bb:
                    25:31:ec:6c:38:5e:2f:17:4c:b5:14:5e:8c:de:1b:
                    14:20:b9:9c:fb:fe:41:5e:ea:68:17:ab:50:a7:9f:
                    6d:93:b3:30:0f:c2:09:2b:7b:43:a4:06:1b:2a:e8:
                    d2:e5:ca:ff:71:e1:69:9c:fb
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        97:e9:b2:9d:f1:ca:16:37:43:21:3a:11:61:03:d7:4b:de:f9:
        06:f9:ee:02:4e:6e:08:25:bc:e3:98:e8:1d:bf:9f:43:5b:cb:
        6c:20:d0:6e:7c:d2:53:f2:29:3b:f9:6e:aa:c3:e0:ab:f8:8f:
        06:83:95:9f:dc:a8:94:bb:a8:50:67:34:de:64:0a:02:29:f0:
        1b:1d:f1:bc:51:09:37:f9:15:23:14:5d:b6:98:85:86:d4:37:
        ff:58:d0:74:24:2f:0d:da:d9:c4:02:89:7f:e6:6c:98:c3:f0:
        8e:42:77:45:3d:a1:ae:a2:8c:20:83:83:c5:2b:cc:58:7c:d7:
        2f:06:32:c6:fa:95:b3:de:5f:48:76:54:08:7a:df:b1:58:a8:
        8d:d9:74:94:57:c1:57:5b:79:32:f1:71:02:77:bb:6f:2a:9f:
        42:94:ef:37:d1:03:da:db:9f:b6:11:8a:4b:d4:e4:34:48:86:
        a0:e3:ce:9e:a3:f4:97:86:37:e1:b9:d4:af:de:a8:e9:ce:2a:
        70:aa:8f:c5:90:d9:71:48:a8:9b:8b:f0:2e:6c:42:7e:c3:bf:
        b6:8e:80:79:2f:ed:81:cb:3a:01:69:ce:c0:e8:9f:fe:ce:0a:
        bf:08:89:4b:68:d0:d5:9b:78:9a:e6:e5:19:92:6c:93:7f:c5:
        a0:89:f9:45
```

Cadeias de certificação

Um certificado de chave pública é considerado fidedigno na medida em que a entidade certificadora (CA – ***Certification Authority***) que o emitiu é considerada de confiança.

A confiança propaga-se entre as CA, uma CA será considerada de confiança se o seu certificado de chave pública foi emitido por uma outra CA considerada de confiança. Forma-se assim uma **cadeia de certificação**, mas tem de haver uma origem para esta cadeia, na origem está uma ROOT CA.





A lista de todas as ROOT CA fidedignas (respetivos certificados de chave pública) é gerida pelo sistema operativo ou aplicação e é atualizada quando se atualiza o sistema operativo ou aplicação.

ROOT CA e certificados auto assinados

Se as ROOT CA são a origem das cadeias de certificação, quem assina o certificado de chave pública de uma ROOT CA? A resposta é, a própria ROOT CA, ou seja é um certificado auto assinado.

Um certificado auto assinado é considerado fidedigno por configuração local do sistema operativo ou aplicação, através da sua colocação na lista de certificados fidedignos. Dessa forma a entidade correspondente e todas a cadeias de certificação com origem nela passam a ser consideradas fidedignas.

Validade dos certificados

Quando um certificado é usado num ambiente de rede (normalmente recebido de um servidor por um cliente) são realizadas várias validações:

- Se está dentro da data de validade.
- Se o campo **CN** do *subject* do certificado corresponde ao **nome DNS do servidor**.
- Se o emissor do certificado está na cadeia de uma CA fidedigna.

Dependendo do tipo de aplicação cliente, em caso de falha numa desta verificações poderá ser perguntado ao utilizador se ainda assim pretende aceitar o certificado e eventualmente mesmo instalar esse certificado com sendo fidedigno (por sua conta e risco).

Revogação de certificados de chave pública

Embora cada certificado contenha uma data de validade, pode em determinadas circunstâncias ser necessário impedir a sua utilização antes de expirado.

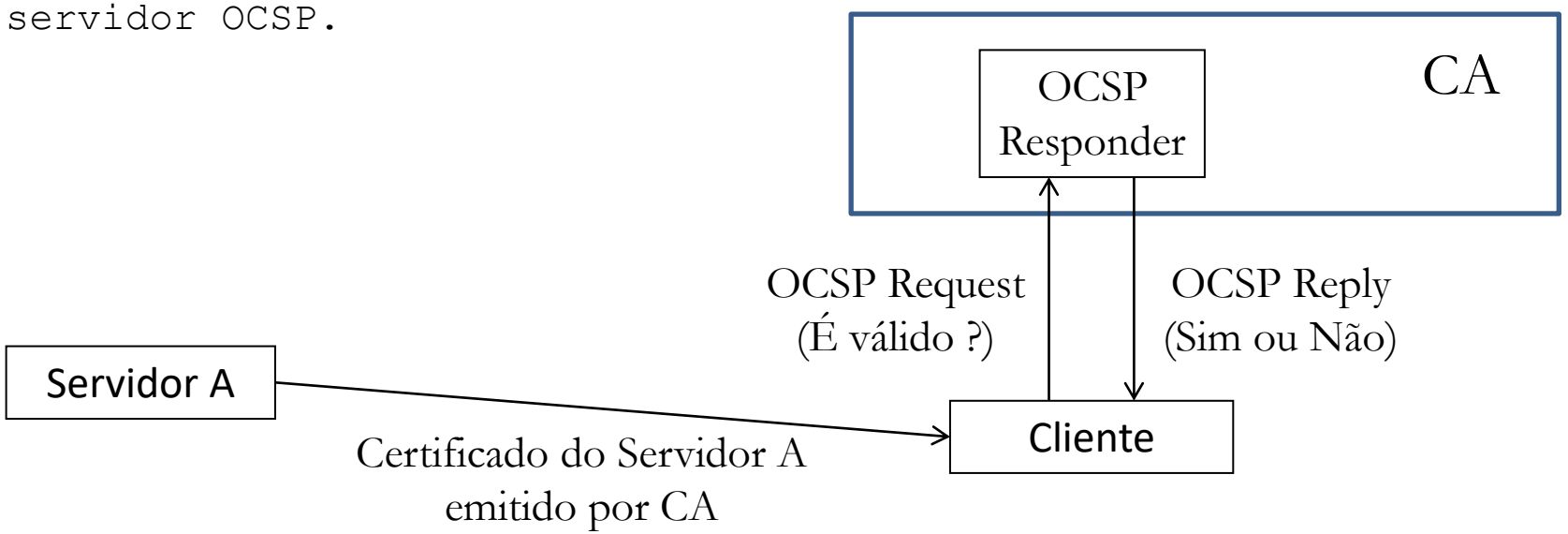
A revogação de certificados é necessária quando se toma conhecimento de que a chave privada correspondente foi comprometida e está na posse de entidades ilícitas. Se não fosse revogado, os utilizadores poderiam continuar a usa-lo e desse forma estar a comunicar com um servidor ilícito criado pelos atacantes.

A verificação da revogação de certificados exige que o utilizador do mesmo realize uma validação adicional através de um contacto direto com a entidade certificadora que o emitiu.

Para que se possa verificar se um certificado foi revogado, o emissor do certificado tem de inclui no mesmo um atributo adicional que identifica através de um nome DNS o serviço de rede disponível para esse efeito.

OCSP (*Online Certificate Status Protocol*)

Embora existam outros mecanismos, a forma mais divulgada de se verificar se um certificado foi revogado é através do OCSP. Para esse efeito, a CA deverá disponibilizar um servidor OCSP, normalmente designado ***OCSP responder***, adicionalmente deve definir em conformidade um atributo nos certificados que emite onde coloca o nome DNS do seu servidor OCSP.



O grau de exigência das várias aplicações clientes que recebem certificados de servidores varia, algumas podem não aceitar certificados de uma CA se ele não disponibilizar um serviço de verificação de revogação.

Linux – comando openssl

O comando openssl fornece uma interface de linha de comando para acesso à biblioteca openssl que suporta SSL v2 e v3 e TLS v1. Entre outras funcionalidades o comando openssl permite gerir certificados de chave pública.

openssl req … – a finalidade principal é gerar um par chave pública/privada e emitir um pedido de certificação da chave pública. O pedido de certificação deve depois ser processado por uma entidade certificadora que emitirá o respetivo certificado. Este comando também pode ser usado para produzir um certificado auto assinado, por exemplo:

```
openssl req -nodes -x509 -keyout pri.key -out pub.crt -days 3650 -newkey rsa:1024
```

Neste exemplo seria gerado o ficheiro pri.key contendo uma chave privada RSA de 1024 bits e o ficheiro pub.crt contendo o respetivo certificado de chave pública auto assinado.

openssl x509 … – processamento de certificados, por exemplo é usado pela entidade certificadora para produzir o certificado a partir de um pedido de certificado.

O comando openssl usa o ficheiro de configuração openssl.cnf. Algumas operações, nomeadamente de gestão de certificados como autoridade certificadora, exigem o uso deste ficheiro de configuração.

Windows Server – Certificate Authority

O Windows Server pode exercer a função de entidade certificadora (CA) no contexto do *Active Directory* através da ativação do role ***Active Directory Certificate Services*** podendo também instalar-se o ***Online Responder***.

Para operar no contexto do *Active Directory*, a CA deve ser do tipo ***Enterprise CA***. Num servidor que não esteja integrado num domínio AD a CA deve ser do tipo ***Standalone CA***.

A CA criada pode ser inserida numa cadeia de certificação (como ***Subordinate CA***), usando um certificado apropriado emitido por outra CA, ou pode ser uma ***ROOT CA***, neste caso será criado um certificado auto assinado.

Através da aplicação ***Certification Authority*** é então possível gerir a CA, receber pedidos de certificados, emitir os mesmos e revogar certificados.

Também podem ser instalados serviços Web (***Certification Authority Web Enrollment)*** que permite aos utilizadores solicitar certificados à CA através do URL **https://SERVER-DNS-NAME/certsrv**.

Cópias de segurança e arquivo

Existem duas motivos pelos quais os administradores dos servidores podem criar cópias dos sistemas de ficheiros dos seus servidores.

Cópias de segurança – vulgarmente designadas ***backups***, destinam-se a garantir que caso haja uma falha grave no servidor que leve à perda irremediável dos dados armazenados nos seus sistemas de ficheiros, exista pelo menos uma cópia a partir da qual eles possam ser repostos. A periodicidade com que estas cópias são realizadas vai determinar até que ponto o sistema recuperado vai ser diferente do estado do sistema no momento da falha (RPO – *Recovery Point Objective*). Outro aspeto a considerar é que, atendendo aos objetivos, o local de armazenamento da cópia deve ser fisicamente afastado do original, deste modo evita-se que um desastre local grave afete tanto o original como a cópia.

Cópias de arquivo – neste caso o objetivo é libertar espaço nos sistemas de ficheiros do servidor, os dados que já não são necessários são copiados para outro local e removidos do servidor. Ficam ainda assim disponíveis se necessário.

Um aspeto que deve estar presente quando se realizam cópias é que elas devem garantir o mesmo nível de privacidade que é assegurado para os dados que se encontram nos sistemas de ficheiros originais.

A privacidade deve ser garantida tanto sob o ponto de vista do local onde as cópias serão armazenadas como do ponto de vista do processo de realização da cópia. Nomeadamente se a cópia é realizada através da rede deve usar protocolos de transferência que incluam mecanismos que deem garantias de autenticação e privacidade.

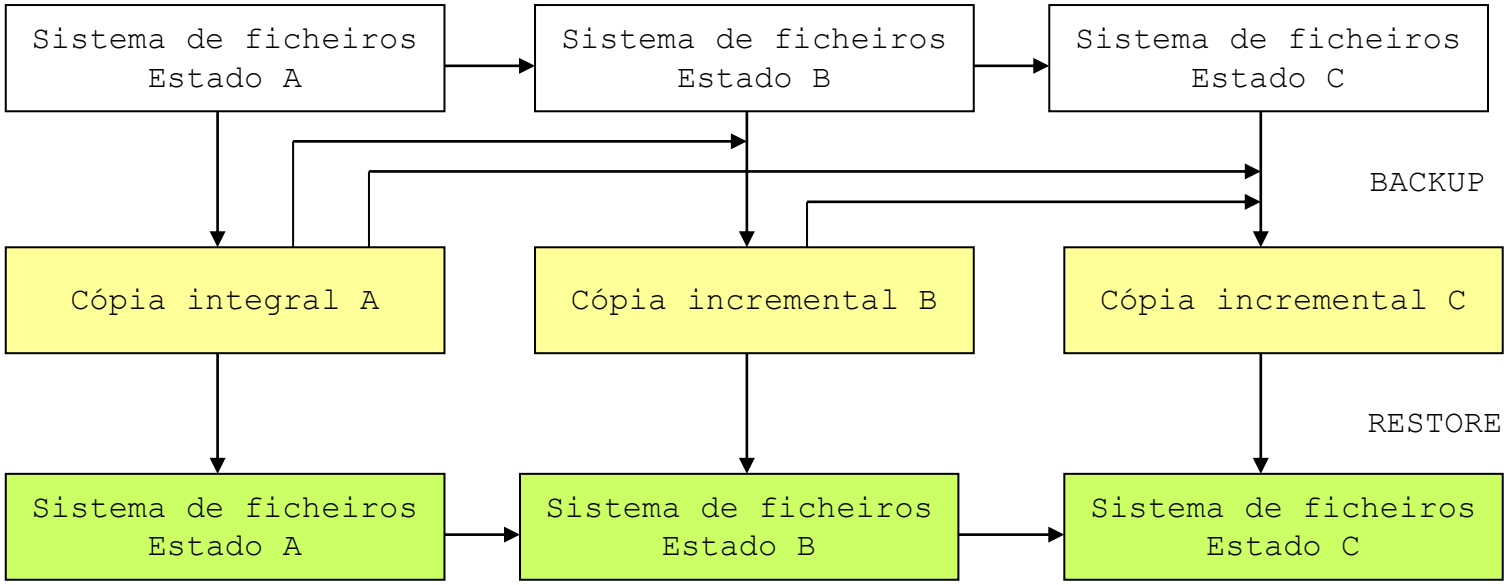
As cópias podem ser realizadas a vários níveis:

- **Disco** – cópia integral do disco, incluindo a tabela de partições, normalmente só pode ser reposta para um disco idêntico completamente vazio.
- **Partição** – cópia integral da partição, a reposição exige uma partição de tamanho idêntico à original.
- **Sistema de ficheiros** – cópia integral do sistema de ficheiros, é de certa forma semelhante à anterior, mas é realizada copiando as estruturas do sistema de ficheiros e não toda a partição.
- **Objetos do sistema de ficheiros** – eventualmente pode contemplar todos os objetos de um sistema de ficheiros, partição ou volume. Ao contrário das anteriores pode ser efetuada sobre um sistema de ficheiros que está em utilização normal porque os objetos são copiados um a um através do sistema operativo.

Cópias incrementais

A cópia de objetos através do sistema operativo acaba por ser a mais utilizada porque é a única que pode ser realizada com o sistema de ficheiros em funcionamento normal, os outros tipos de cópia exigem no mínimo que o sistema de ficheiros esteja em modo *read-only*.

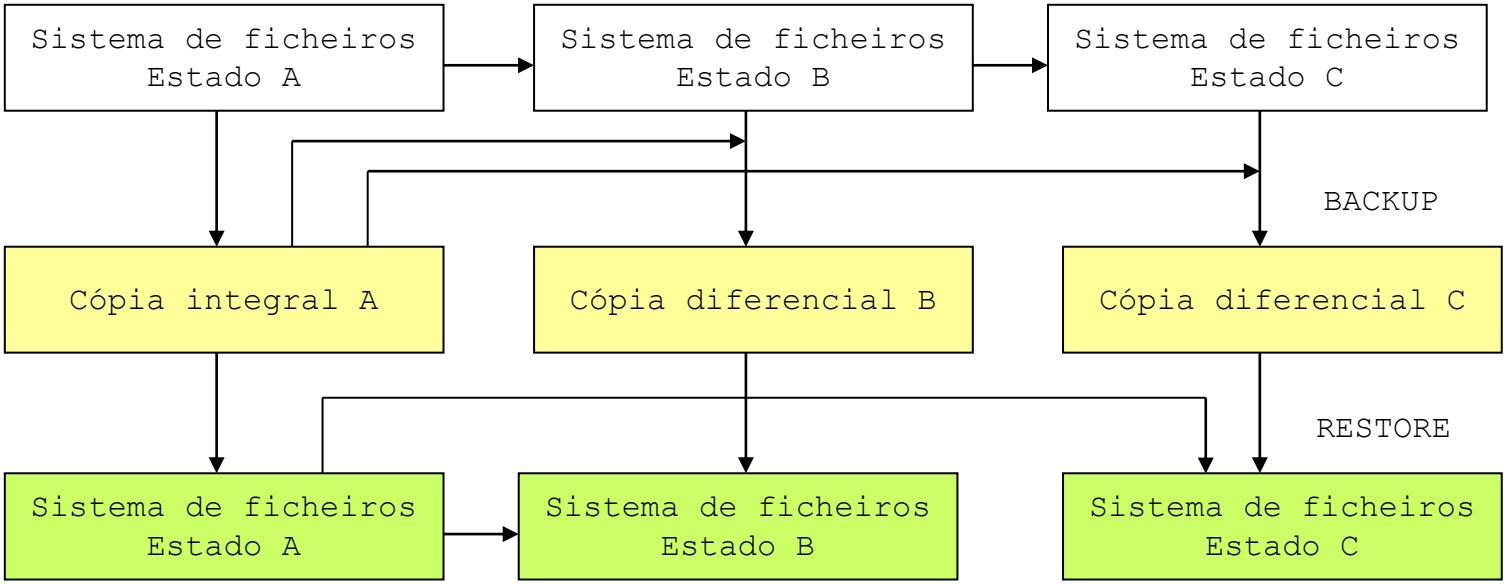
Um **cópia integral** de todos os objetos existentes num sistema de ficheiros pode ser bastante morosa. Depois de criada uma cópia integral pode recorrer-se a cópias incrementais onde se registam apenas a diferenças relativamente à cópia anterior:



Cópias diferenciais

As cópias incrementais têm dois inconvenientes: é necessário manter tanto a cópia integral como todas as cópias incrementais seguintes; a reposição (*restore*) é morosa porque obriga a aplicar a cópia integral e de seguidas as várias cópias incrementais uma a uma na ordem correta.

As **cópias diferenciais** são ligeiramente diferentes, cada cópia diferencial regista sempre as diferenças do sistema relativamente à cópia integral (e não a cópia anterior).



Apenas é necessário manter a cópia integral e a última cópia diferencial, a reposição torna-se muito mais rápida.

A vantagem das cópias diferenciais é que apenas é necessário manter a cópia integral e a última cópia diferencial, as cópias diferenciais anteriores podem ser eliminadas. A reposição torna-se muito mais rápida porque envolve apenas repor a cópia integral e de seguida a cópia diferencial.

Sejam incrementais ou diferenciais, o número de aplicações sucessivas do processo deve ser sempre limitado, periodicamente deve reiniciar-se o ciclo e realizar uma nova cópia integral.

Uma prática recomendada é realizar todos os fins de semana uma cópia integral e durante a semana realizar cópias incrementais ou diferenciais. A cópia integral, mais morosa, é assim realizada num período de menor atividade.

A realização de cópias de segurança deve ser automatizada através de serviços de execução programada, havendo toda a vantagem em que sejam realizadas em períodos de pouca atividade.

A realização de cópias terá sempre algum impacto no desempenho dos servidores e também será facilitada se não estiverem no momento a ocorrer alterações significativas no sistema de ficheiros.

Windows Server Backup

A aplicação ***Windows Server Backup*** pode ser usada no Windows Server depois de instalada a *feature* com o mesmo nome.

Permite realizar diretamente uma cópia (***backup once***) ou programar a execução de cópias (***backup schedule***). É possível realizar a cópia de drives inteiros (volumes), dados do sistema, todo o servidor (*bare metal recovery*) ou apenas alguns ficheiros e pastas.

O destino da cópia pode ser um drive local ou uma partilha de rede, no local de destino será criada uma pasta como o nome ***WindowsImageBackup*** que contém as cópias realizadas para esse destino e toda a informação associada, incluído o que contém, a máquina e sistema de ficheiros de origem e a data/hora a que foi realizado.

O Windows Server Backup faz a gestão automática de cópias integrais e diferenciais sem controlo direto do administrador. Inicialmente faz uma cópia integral e depois cópias incrementais, periodicamente volta a efetuar uma cópia integral.

Para fazer a reposição deve ser usada a opção ***recover***, e selecionar a origem (drive ou partilha de rede que contém a pasta ***WindowsImageBackup***), as várias cópias disponíveis nesse local serão apresentadas, todos ou parte dos elementos constantes na cópia podem ser restaurados para as localizações originais ou outras localizações.

Linux – comandos de cópia

No Linux não existe uma aplicação equivalente ao ***Windows Server Backup***, pelo menos não com o mesmo grau de integração com o sistema operativo e simplicidade de utilização.

Existem várias ferramentas de alto nível disponíveis, muitas delas são apenas *front-ends* amigáveis de um conjunto de comandos de cópia que estão disponíveis no sistema operativo.

Com a ajuda de alguns scripts e a execução programada, estes comandos podem ser usados para implementar mecanismos de cópia automática com os mais diversos propósitos.

Comando tar (*Tape Archive*)

O comando **tar** permite criar um ficheiro por vezes designado ***tarball*** que contém partes ou a totalidade de um sistema de ficheiros.

Uma vez criado o tarball (opção -c), sempre que necessário, podem ser extraídos (-x) todos ou parte dos ficheiros e pastas nele armazenados preservando todas as propriedades dos originais (opção -p).

Adicionalmente o tarball pode ser comprimido usando o gzip (opção -z) ou outros algoritmos de compressão. Os nomes dos objetos são guardados no tarball, mas nunca incluem a raiz do sistema de ficheiros, isso permite que os objetos possam ser extraídos para um local diferente do original.

O **tar** suporta cópias incrementais através da opção –g, através desta opção pode ser indicado um ficheiro de registo de ***snapshot***, se esse ficheiro não existe será feita uma cópia integral e o ficheiro será criado, exemplo:

```
tar –c –f /backups/backup.00.tar –p –g /backups/back.snap /etc /var/lib
tar –c –f /backups/backup.01.tar –p –g /backups/back.snap /etc /var/lib
tar –c –f /backups/backup.02.tar –p –g /backups/back.snap /etc /var/lib
```

Presumindo que inicialmente o ficheiro **/backups/back.snap** não existe, então o arquivo **/backups/backup.00.tar** será uma cópia integral com todo o conteúdo das pastas **/etc** e **/var/lib**. Os seguintes serão cópias incrementais. Recorde-se que o tar omite a raiz nos nomes dos objetos, logo serão armazenados com nomes começados por **etc/** e **var/lib/**.

Para recuperação (extração) não é necessário o registo de *snapshot*, mas é necessário usar a opção –G para indicar que se trata de uma sequência de cópias incrementais e a recuperação terá de ser feita exatamente na mesma ordem:

```
tar –x –f /backups/backup.00.tar –p –G –C /
tar –x –f /backups/backup.01.tar –p –G –C /
tar –x –f /backups/backup.02.tar –p –G –C /
```

A opção –C indica o local para onde se pretende extrair o conteúdo do arquivos, se não for especificada serão extraídos para a pasta de trabalho corrente.

Copias remotas

Normalmente as cópias serão realizadas para servidores remotos, no caso das cópias de segurança isso é um requisito.

A forma mais simples de conseguir isso é simplesmente efetuar a cópia para uma partilha de rede (NAS) com permissão ***read-write***, no Linux será tipicamente uma pasta remota montada por NFS, mas também pode ser por SMB/CIFS.

Uma alternativa é a utilização do comando **rsync**, este comando permite copiar sistemas de ficheiros para servidores remotos através de SSH ou através de um serviço próprio (rsyncd) que terá de ser instalado na máquina alvo.

Normalmente a operação será realizada *unattended* pelo que nenhuma password de acesso poderá ser solicitada. Se for usado o serviço rsyncd a password pode ser guardada num ficheiro (o username/password do rsyncd é independente dos utilizadores do sistema). Se for usado SSH, é necessário configurar esse serviço para funcionar sem password, a opção mais aconselhada é fazer a autenticação através de um par chave pública/chave privada.

Se o serviço SSH estiver configurado para operar sem passwords também se torna possível enviar o resultado do comando tar através de uma sessão SSH, por exemplo:

```
tar -c -p /etc /var/lib | ssh USER@TARGET-HOST "( cat > /backups/backup.tar )"
```

Copias parciais

Uma cópia parcial abrange apenas uma parte do sistema. Normalmente o objetivo é salvaguardar dados referentes a um dado subsistema ou serviço, tais como servidores Web, servidores de bases de dados, etc.

A cópia parcial pode ser realizada em Unix com recurso ao comando **tar**, para esse efeito basta localizar todos os ficheiros e pastas associados ao subsistema ou serviço em causa.

O exemplo seguinte cria uma cópia de segurança comprimida para uma instalação particular do servidor Apache:

```
tar -c -z -p -f /backups/apache2.tar.gz /etc/apache2 /var/www  \
/etc/ssl/certs/apache2.pem /etc/ssl/private/apache2.key
```

Neste caso estão a ser contemplados todos os ficheiros de configuração na pasta **/etc/apache2**, as páginas disponibilizadas **/var/www** e ainda as chaves usadas pelo HTTPS.

Muitos serviços dispõe de comandos próprios para realizar cópias de salvaguarda dos seus dados enquanto estão em funcionamento, especialmente no caso das bases de dados.

Exemplos: o serviço **MySQL** dispõe do comando **mysqldump**, o serviço **OpenLDAP** dispõe do comando **slapcat**.

Nestes casos é claramente vantajoso usar estes comandos especialmente concebidos para o efeito e que podem ser usados de forma segura sem necessidade de interromper o serviço.

Linux – sistemas de cópia

Embora seja possível através de scripts e os comandos anteriormente referidos automatizar as cópias, existe também algum software gratuito de nível mais elevado que deve ser mencionado.

AMANDA (Advanced Maryland Automatic Network Disk Archiver) (http://www.amanda.org/)

O AMANDA é um sistema centralizado de cópias, é constituído por um servidor que corre numa máquina Linux onde as cópias são armazenadas e clientes que realizam as cópias para o servidor.

Os clientes podem ser máquinas Linux, mas existe também um cliente para sistemas Windows. A realização das cópias programadas pode ser desencadeada a partir do servidor AMANDA.

Muitas das funcionalidades do AMANDA são implementadas através de comandos nativos dos sistemas Linux como o tar e ssh.