universidade de aveiro deti departamento de electrónica, telecomunicações e informática

Segurança Informática e nas Organizações

Teóricas

Resumos

Gonçalo Matos, 92972

Licenciatura em Engenharia Informática

3.º Ano | 1.º Semestre | Ano letivo 2020/2021

Última atualização a 31 de janeiro de 2021

Este documento é baseado nos *slides* teóricos dos professores João Paulo Barraca e André Zúquete. Fontes adicionais são referenciadas no início dos capítulos onde foram utilizadas.

Índice

1. Introdução à segurança

Slides teóricos

Contrariamente ao que é muitas vezes assumido, a segurança aplicada à informática não se resume à proção dos sistemas contra ataques de piratas informáticos. Na verdade, existem muitos outros problemas que podem comprometer a integridade de um sistema de computação.

Catástrofes como relâmpagos, picos de energia, inundações, radiação...;

Degradação dos sistemas físicos por erros nas células da RAM ou SSD, falhas na fonte de alimentação…;

As soluções para evitar o impacto destes fenómenos passam por fazer ***backups*** da informação e/ou **replicá-la**.

Falhas de energia, internas aos SO, ou erros de *software*;

Aqui torna-se fundamental a **redundância** dos componentes (que podem ser alimentados por mais do que uma fonte de energia, p.e.), podendo ao nível das comunicações implementar **sistemas transacionais** com **encaminhamento dinâmico** e **retransmissões**.

Atividades não autorizadas tenham origem externa ou interna à organização

Exemplos: DoS, acesso/alteração da informação, utilização de recurso (mineração), vandalismo

É nesta área que a unidade curricular de SIO se foca, procurando explorar a segurança da informação e das infra-estruturas, em particular ao nível das organizações.

Segurança nos sistemas computacionais

Este é um problema de **complexidade crescente**, não só porque com a evolução da tecnologia faz com que os computadores processem cada vez mais informação num menor espaço de tempo (+ estragos + rapidamente), mas também pela complexidade incremental dos sistemas, que atualmente estão em constante desenvolvimento.

Atualmente os sistemas de informação das organizações não são estáticos, estando constantemente a ver desenvolvidas novas funcionalidades, que podem representar novas falhas para o sistema. É ainda de destacar que cada dependência do sistema é uma potencial falha, uma vez que a garantia da sua integridade não é responsabilidade da organização.

Há ainda a vulnerabilidade introduzida pelas **redes**, que permitem ataques anónimos e distribuídos (ou seja, com grande capacidade computacional) de qualquer ponto do planeta.

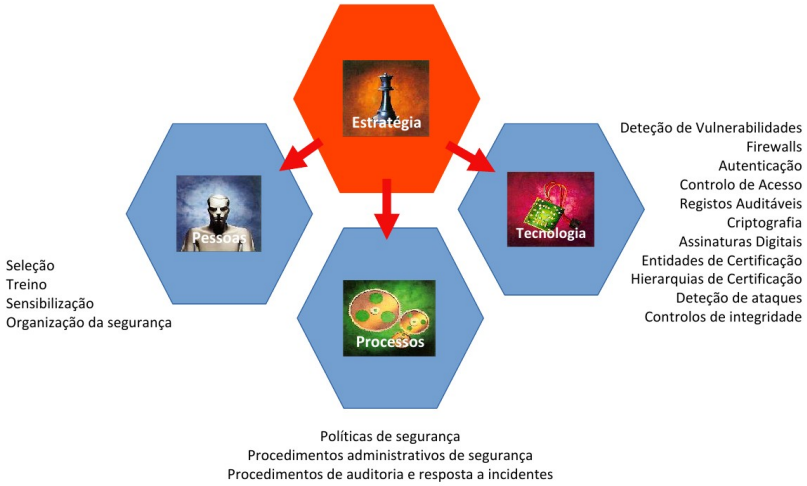
Os **usuários** são também uma vulnerabilidade, porque muitas vezes não possuem noção do risco, podendo facilmente ser induzidos a explorar uma falha sem intenção.

Apesar de todas estas variáveis, é importante entender que é impossível desenvolver um sistema 100% seguro, principalmente porque a segurança tem custos elevados e por isso deve ser feito um **balanço entre o risco e o impacto** de forma a tornar o sistema o mais robusto possível.

Deve ser tido em atenção que por vezes segurança a mais pode na verdade tornar o sistema mais débil.

P.e. se introduzirmos palavras-passe com 20 caracteres aleatórios, a probabilidade dos utilizadores terem um *post-it* com ela escrita colada ao monitor é grande. Assim, uma palavra-passe robusta que protegeria o sistema de utilizadores não desejados torna-se quase pública para todos os que frequentem o espaço físico do escritório (mais as pessoas que vejam fotos descuidadas do mesmo).

Deve ainda ser adotada uma **política de punição das violações** à integridade do sistema da empresa, de forma a impedir a existência de uma noção de impunidade.



Glossário

Vulnerabilidade	Uma fraqueza do sistema que o torna sensível a ataques;
Ataque	Ações que levam à execução de atividades ilegais (exploram vulnerabilidades);
Risco/Ameaça	Dano resultante de um ataque;
Defesa	Conjunto de mecanismos e tecnologias com vista a: - Reduzir o número de vulnerabilidades; - Detetar ataques passados, atuais ou futuros; - Reduzir o risco para os sistemas.

As **vulnerabilidades** podem estar presentes em qualquer ponto do ciclo de vida do sistema!

Riscos da segurança

A segurança procura colmatar as vulnerabilidades, de forma a minizar os riscos, que podem ser classificados em vários níveis.

Informação, tempo e recursos Destruição ou alteração de informação;

Confidencialidade Acesso não autorizado a informação;

Privacidade Recolha/distribuição de informação pessoal;

Desponibilidade de recursos Disrupção de sistemas, comunicações ou processos;

Impersonificação Exploração não autorizada de perfis de identidade.

Fontes de vulnerabilidades

No desenvolvimento de qualquer sistema devemos considerar que sempre que recorremos a fontes externas de dados, estes podem estar comprometidos.

Por isso, sempre que há interação com **usuários** externos ou internos, **serviços externos** que cuja gestão está fora do nosso domínio, **bases de dados partilhadas** com outros serviços, ou mesmo **comunicações sobre ligações não controladas** devem ser tomadas medidas para prevenir ataques inesperados.

A serem passíveis de ser exploradas, as vulnerabilidades podem introduzir aplicações hostis no nosso sistema, que controlados por um atacante podem ter efeitos nefastos.

Políticas de segurança

São conjuntos de orientações relativas à segurança que regeem um domínio, procurando definir o **poder de cada sujeito**, os **procedimentos de segurança** que estes devem executar e quando, **os requisitos mínimos de segurança** (níveis/grupos de segurança), a **estratégia de defesa e resposta** (arquitetura defensiva, monitorização e reação a ataques) e por fim o que é **correto e ilegal**.

Estas podem ser aplicadas em domínios distintos na mesma organizações, em hierarquia que pode inclusive levar a sobreposições. O importante é serem coentes entre si.

Mecanismos de segurança

Enquanto que as **políticas** definem a teoria, os **mecanismos** representam a sua aplicação prática.

Alguns exemplos são os de autenticação, controlo de acesso, execução priveligiada, filtragem, registo, protocolos criptográficos, auditorias, …

Níveis de segurança

O Departamento de Defesa dos EUA criou um conjunto de critérios que permitem avaliar o nível de segurança de um sistema. É denominado por **Trusted Computer System Evaluation Criteria (TSEC)** e define 7 classes entre a D, mais insegura e a A1, mais segura.

Um critério alternativo é o **Information Technology Security Evaluation Criteria (ITSEC)**, criado pela Comissão Europeia, que estipula 6 níveis de especificação formal e correção da implementação entre o E1 e E6.

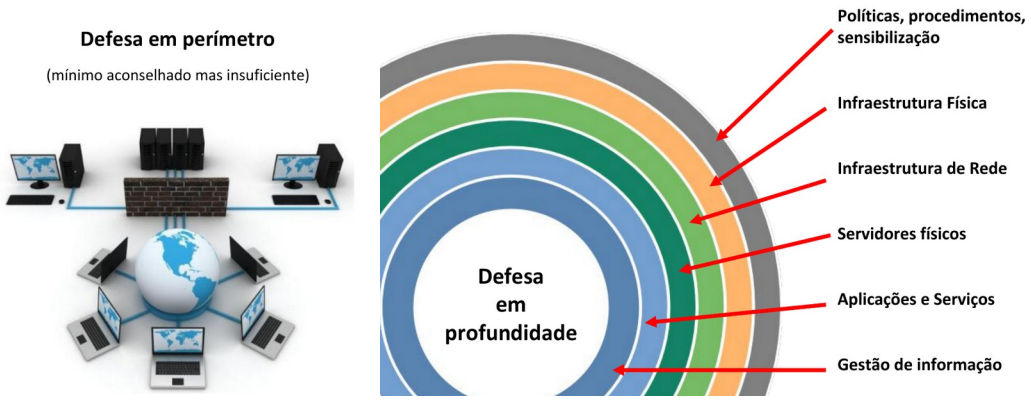
Políticas de segurança em Sistemas Distribuídos

Em SD, as políticas têm de englobar múltiplos sistemas e redes, destacando-se a introdução do conceito de **gateways de segurança**, as interações de entrada e saída de um domínio.

Tipos de defesa

A defesa mais básica deste ssistemas é feita em **perímetro**, cobrindo apenas a gateway entre o domínio do sistema e a internet (WAN).

A mais completa denomina-se em **profundidade**, apresentando diversas camadas que garantem a segurança desde os elementos físicos, à interação entre os vários domínios internos e externos.



Estes sistemas podem ser alvo de **ataques específicos**, concebidos para um sistema/rede particulares, ou **genéricos ou autónomos**, que exploram vulnerabilidades conhecidas e comuns.

Mecanismos de segurança

Alguns **mecanismos de segurança** para este tipo de sistemas são a utilização de **SO confiáveis** (em ambientes seguros), ***firewalls*** e sistemas de segurança para monitorizar tráfeho nas redes e **VPNs**, que permitem comunicações seguras sobre redes públicas/inseguras.

Ao nível da minimização do risco de intruzão temos a **autenticação** robusta é outro mecanismo importante, assim como a **cifra de ficheiros e dados em sessões**.

Quanto à prevenção de ataques, deve ser feita uma **deteção de intrusões**, a par da **inventarização de vulnerabilidades** e de **testes de penetração**.

2. Vulnerabilidades

Slides teóricos

Uma empresa é tão mais suscetível de ataques quanto maior a sua dimensão, uma vez que ataques bem sucedidos serão mais rentáveis.

De forma a prevenir **ataques**, que exploram **vulnerabilidades**, as organizações devem investir na **defesa** dos seus sistemas, de forma a garantir a segurança da informação que armazenam.

Segurança da informação

No entanto, **defesa** é um conceito abstrato, que na realidade ganha forma em cinco medidas.

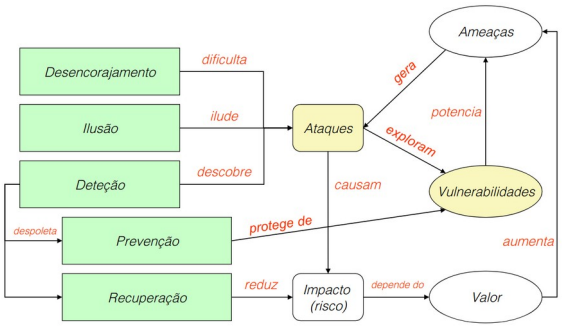
Desencorajamento através da punição dos infratores e utilização de barreiras de segurança;

Deteção de intrusões em tempo real, ou através de auditorias e análises forenses;

Ilusão dos atacantes com *honeypots* ou *honeynets* (como que *pishing* para atacantes);

Prevenção através de políticas de segurança, deteção e correção de vulnerabilidades;

Recuperação com *backups*, ou sistemas redundantes.



Cada uma influencia a segurança de maneira diferente, no entanto, enquanto que o desencorajamento, a ilusão e a deteção atuam sobre **falhas conhecidas**, a perceção e a recuperação são **universais**.

Prontidão

Mesmo com todas estas medidas implementadas, na existência de vulnerabilidades desconhecidas por parte da organização, todo o sistema pode ser facilmente comprometido.

Assim, como os sistemas funcionam em permanência e dada a facilidade que as redes criaram em orquestrar ataques (baixo custo, fácil coordenação), é também necessário apresentar uma capacidade permanente de reação a ataques, com moninotiração em permanência e equipas de segurança prontas a atuar de imediato.

Ataques de Zero Day

Este tipo de ataque caracteriza-se por explorar uma **vulnerabilidade desconhecida**.

Se for explorada de forma discreta, pode durar meses ou até anos, sendo inclusive comercializadas no mercado negro.

Deteção de vulnerabilidades

Para aferir a robustez dos sistemas, podem ser utilizadas ferramentas de deteção de vulnerabilidades e até mesmo de replicação de ataques conhecidos.

Estas ferramentas devem ser aplicadas em ambientes de teste e nunca de produção, uma vez que se realmente identificarem vulnarabilidades, os testes podem corromper a integridade do sistema.

Estas verificações podem ser **estáticas** caso consistam na análise de código, ou **dinâmicas** se testarem aplicações em execução.

Sobrevivência

Apesar de ser o oposto do que geralmente é esperado dos sistemas (estandardização, protocolos bem definidos e regulares), a **diversidade** é a chave para a sobrevivência.

Isto porque dada a sua exclusividade, operações e protocolos distintos são mais difíceis de contornar, uma vez que requerem um estudo dedicado do sistema em particular e não podem ser aplicados de forma generalizada a outros.

Dada a sua diversidade, o SO Android terá menos probabilidade de ser atacado que o iOS.

Há ainda outras ferramentas que permitem aumentar a robustez dos sistemas.

CVE (Common Vulnerabilities and Expores)

É um repositório público de vulnerabilidades, que lista e descreve vulnerabilidades e exposições de segurança.

Vulnerabilidade

Um erro só é uma vulnerabilidade se permitir que o atacante viole uma política de segurança.

Exposição

Problema de configuração que permite ao atacante aceder a informação ou capacidades que o podem auxiliar, sem conseguir no entanto comprometer diretamente o sistema.

Estes permitem a partilha de conhecimento e discussão de soluções, fomentando a inovação e permitindo melhorar as ferramentas. No entanto, também inúteis contra ataques *zero day*.

Um ataque pode explorar várias vulnerabilidades (diferentes CVE).

CWE (Common Weakness Expore)

De forma complementar temos outro repositório, mas focado na exploração das causas das vulnerabilidades, ou seja, identifica as vulnerabilidades provocadas pelos *developers* devido a uma utilização incorreta do *software*.

Um CWE podem organizar-se de forma hierárquica, havendo um pai que fornece uma descrição genérica e vários filhos, cada um focado numa parte concreta do problema.

Os erros mais comuns são validação e representação de entradas, abuso de API, funcionalidades de segurança, tempo e estado (concorrência), erros, qualidade do código, encapsulamento e ambiente de execução mal configurado.

CERT (Computer Emergy Readiness Team)

Esta é uma equipa responsável por resistir a ataques em sistemas distribuídos (em rede), limitando o dano e garantindo a continuidade dos serviços críticos.

CSIRT (Computer Security Incident Response Team)

Dentro das equipas CERT, há uma componente de sigla CSIRT, cuja responsabilidade é receber, analisar e responder a relatórios de incidente e atividade.

Centros de segurança

Estas instituições são fundamentais para o acompanhamento dos ataques, emissão de alertas e estudo das tendências. Geram conhecimento na área da segurança.

Em suma

A segurança deve ser encarada como um todo (desde aspetos normativos, legais, …).

Os riscos devem ser identificados e geridos de forma permanente.

A informação é o valor (segui-la sempre – onde está, quem a manipula, por onde circula).

Um ataque pode surgir “em qualquer lado”, por isso medidas devem ir em profundidade.

Estabelecer cultura baseada na segurança.

Confiar, mas verificar.

Partilhar experiências, regulamentação, incidentes e respostas.

3. Criptografia

Slides teóricos, Stream Cipher

Neste capítulo vamos estudar esta ciência que tem como objetivo escrever de forma “secreta”.

Glossário

Criptografia. Arte ou ciência de escrever de forma escondida/confidencial.

Criptanálise. Arte ou ciência de quebrar sistemas criptográficos.

Criptologia. Criptografia + criptanálise.

Cifra. Técnica concreta de criptografia.

Criptograma. Texto criptografado por uma cifra.

Algoritmo. Modo de transformação dos dados.

Chave. Parâmetro de um algoritmo que influencia a sua operação.

Criptanálise

Para conseguir atingir o seu objetivo de quebrar sistemas criptográficos, esta área de estudos foca-se em obter o texto original, a chave de cifra e/ou até mesmo o algoritmo de cifra, por engenharia reversa.

De forma a obter esta informação pode recorrer a **ataquer por força bruta**, pesquisando exaustivamente sobre todo o espaço de chaves até encontrar uma adequada.

Torna-se difícil para espaços de chaves com dimensão grande.

Pode ainda procurar realizar **ataques mais inteligentes**, tentando reduzir o espaço de chaves a determinados caracteres ou expressões e identificando padrões em algumas operações.

Atualmente a complexidade dos algoritmos torna-los dificeis de decifrar por *brute force*, sendo esta última opção a mais comum atualmemte.

Evolução das cifras

A criptografia faz parte da história antiga da humanidade. Os primeiros vestígios da sua utilização datam da Antigo Egipto, tendo sofrido várias mutações até aos dias de hoje.

Começaram por se basear na **transposição de caracteres** (relação direta), inicialmente manual e depois mecânica. Nos dias de hoje assumem **complexos problemas matemáticos**, que são utilizados de forma comum no dia a dia, muitas vezes sem nos apercebermos.

Transposição

Esta técnica baseia-se na transposição do texto para uma matriz na forma de colunas, sendo o criptograma dado pela concatenação das linhas (transposição).

Substituição

Consiste na substituição de um símbolo por outro. Pode assumir vários tipos.

Mono alfabética quando um símbolo é substituído por outro;

Divide-se ainda em substituição mono alfabética com **frase-chave**, onde ao abecedário "normal" se faz corresponder um novo abecedário, ou **aditiva**, quando fazemos um *shift* (A + 5 = E).

Em qualquer uma destas duas cifras, os padrões do texto original são repetidos, assim como a frequência das letras, sendo assim fácil de decifrar por análise estatística.

Poli-alfabética quando muitos símbolos são mapeados para um único;

Usam **n alfabetos de substituição**.

Assim, vai apresentar um período n, pelo que pode ser decifrado considerando n encriptações mono alfabéticas e recorrendo aos métodos estatíticos.

Tiveram a sua aplicação prática nas **máquinas de rotores**, que ao acumular vários em sequência e rodando-los de forma diferenciada permitia a geração de uma chave poli-alfabética bastante complexa. Uma letra nunca poderia ser codificada para ela própria. A chave era o conjunto de rotores, a ordem relativa e as posições originais e de avanço.

Homofónica quando um símbolo é mapeado para vários.

Aproximações à criptografia

Na teoria, a **cifra perfeita** (cifra de Vernam) é aquela que é (des)codificada com uma chave aleatória única (*one-time pao*) e infinita (maior que a mensagem, de forma a não ser periódica) através da operação XOR.

No entanto, na prática é impossível fazer esta implementação, sendo uma cifra neste âmbito classificada como segura quando a única forma de a decifrar seja através de força bruta, mas que esta seja de tal dificuldade computacional que na prática seja impossível (tanto tempo como a idade do universo).

Para garantir a segurança da cifra, existem 5 **critérios de Shannon**, que definem:

1. A quantidade de secretismo oferecido (comprimento da chave);

Geralmente as chaves tem geralmente à volta de 128 bits.

O fator humano não é considerado neste critério. Se obrigarmos utilizadores a saber uma chave complexa, provavelmente eles irão escrevê-la em algum local visível, comprometendo a segurança do sistema.

2. A complexidade da escolha das chaves (o espaço da chave);

Se limitarmos o espaço das chaves a algarismos numéricos, temos um espaço da chave bastante reduzido.

3. A simplicidade da implementação;

Deve ser facilmente compreendida pelos programadores, de forma a que não sejam cometidos erros na sua implementação.

4. A propagação de erros;

Algumas cifras garantem que no caso de um erro toda a informação é perdida, outras apenas perdem a informação corrompida. Há ainda as que apresentam mecanismos de redundância e permitem recuperar de falhas (por exemplo em canais de comunicação ruidosos).

5. A dimensão do criptograma;

Nunca será menor que o texto original, mas pode ser maior.

No caso da encriptação de discos rígidos (divididos em partições com tamanho definido), os dados encriptados não podem ocupar mais espaço que os originais.

Há ainda dois conceitos importantes quando nos referimos a aplicações práticas da criptografia.

Confusão. Complexidade na relação entre o texto, a chave e o criptograma. Inexistência de padrões.

Difusão. Pequenas alterações no texto original levarem a alterações de grande parte do criptograma (pelo menos metade dos bits).

No início da criptografia assumia-se que os algoritmos deviam de ser secretos. No entanto, nos dias de hoje é aceite que os algoritmos devem ser publicamente escurtinados, de forma a serem testados amplamente por vários especialistas.

Estamos limitados ao nosso conhecimento e recursos!

Devemos assumir que os criptanalistas conhecem tudo do algoritmo (criptograma, contexto do texto original e algoritmo). Apenas desconhecem a chave.

Cifras contínuas

Este tipo de cifra tenta replicar a noção teórica da cifra perfeita, ao realizar a operação XOR entre o texto original e uma chave contínua aleatória (*one-time pad*) ou pseudualeatória (produzida por um gerador a partir de uma chave mais pequena).

Devido à sua simplicidade, é uma cifra bastante rápida.

Tal como na noção teórica, aqui as chaves só podem ser utilizadas uma vez, caso contrário a soma dos criptogramas fornece a soma dos textos. Ao descobrir o texto original e fazer XOR com o criptograma, obtém-se a chave.

$$C1 = P1 \oplus Ks,\ C2 = P2 \oplus Ks \rightarrow C1 \oplus C2 = P1 \oplus P2$$

Apesar de oferecer **confusão**, não aplica a **difusão**.

O facto de a relação entre os caracteres do texto original, da chave e do criptograma ser direta, permite que desencriptemos apenas parte da informação caso não necessitemos de toda. No entanto, esta previsibilidade permite que atacantes alterem partes específicas do criptograma de forma a corromper a informação.

4. Cifras modernas

Atualmente existem dois eixos de classificação de cifras.

Operação	Tipo de chave
Por bloco	Simétricas
Contínuas ou *stream*	Assimétricas

Estas podem ser combinadas entre si, com exceção das cifras contínuas assimétricas.

Cifras simétricas por bloco

A chave secreta é **única,** partilhada por todos os indivíduos que devem ter acesso ao conteúdo.

Caso exista uma chave por utilizador, permitem **autenticação** caso sejam utilizadas corretamente.

Tecnicamente se conseguirmos descodificar a mensagem com a chave do utilizador X significa que foi o utilizador X a enviá-la. No entanto, como todos os recetores têm de ter a mesma chave do emissor para a descodificar, se os recetores forem mal intencionados podem codificar as suas mensagens com a chave de outro utilizador.

Os algoritmos mais usados são o DES (DataBlock=64, Key=56), IDEA (D=64, K=128), AES (D=128, K=128, 192, 256).

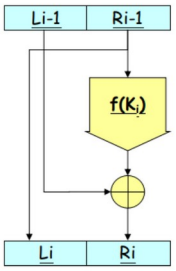
Este tipo de algoritmos é bastante comum na encriptação de objetos discretos (ficheiros, documentos, …).

Difusão e confusão

Para gerar **difusão e confusão** pode ser feita permutação, substituição, expansão ou compressão.

Redes de Feiestel

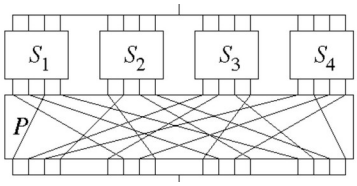
Este método consiste em aplicar a cifra apenas à segunda metade do bloco, sendo feito XOR entre o valor cifrado e a primeira metade, que vai ser a segunda metade do criptograma. A primeira metade é a segunda parte do bloco original.



Como a primeira metade do criptograma vai corresponder exatamente à segunda metade do bloco original, com uma única iteração não vamos ter um criptograma 100% seguro. A cada iteração acrescentamos segurança (no mínimo devem ser 2!).

Redes de substituição-permutação

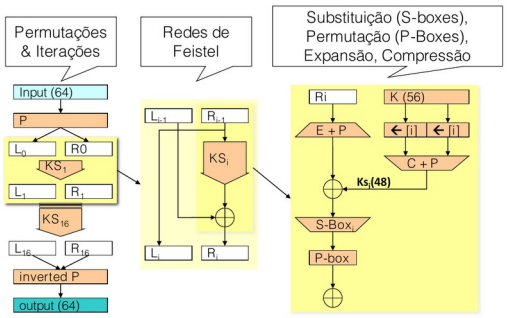
Este é um método alternativo para criar difusão e confusão. Tem por base a **substituição** dos *bits*, seguida da sua **permutação**.



Ambas estas operações são feitas com base em modelos matemáticos de forma determininística com base na chave dada.

Como visto anteriormente, no modelo ideal da **subsituição** uma alteração de um bit levaria à alteração de todos. Na realidade espera-se que uma alteração leve à alteração de pelo menos metade dos bits.
Quanto à **permutação**, idealmente permuta todos os bits.

DES (Data Encryption Standart)

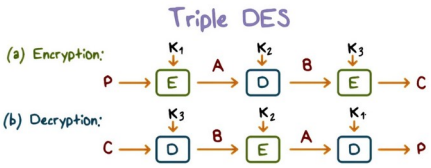


Este algoritmo é uma aplicação prática do modelo das chaves simétricas que combina os dois métodos descritos anteriormente.

De uma forma geral pode ser visto como complexo, mas na realidade consiste na junção de "caixas" com métodos bastante simples.

É importante esta simplicidade em cada componente dos algoritmos de cifra, de forma a facilitar a sua compreensão por parte de quem o implementa.

Na teoria é seguro, mas na prática há algumas chaves que comprometem a sua segurança. A utilização de chaves com apenas 56 bits faz dele **pouco seguro**, uma vez que é exequível ataques por ***brute force***.



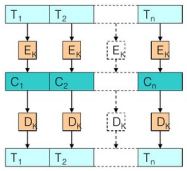
Uma solução que o torna mais robusto é a **cifra múltipla**, em que o algoritmo é realizado com 3 chaves (que equivalem a 168 bits): uma para encriptar, outra para desencriptar e outra para encriptar novamente. A desencriptação é feita no sentido inverso. Esta implementação é designada por **3DES**.

Há variações deste algoritmo em que a terceira chave é igual à primeira, sendo assim a chave considerada de 112 bits apenas.

Modos de encriptação

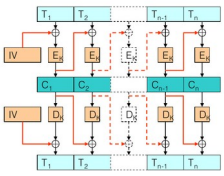
A forma como o texto é dividido em blocos e como este vai ser encriptado pode variar. Cada uma destas variações é um **modo**.

Electronic Code Block (ECB)



Este modo consiste na divisão linear dos blocos e na sua codificação individual, de forma independente.

Vantagem: Permite a desencriptação de um bloco específico apenas.
Desvantagem: Elementos iguais originam criptogramas iguais! **Padrões** são mantidos no ruido.



Chiper Block Chaining (CBC)

Este modo inclui **feedback** da encriptação anterior em cada nova encriptação. Para tal é necessário um **vetor de inicialização** (IV) para o primeiro bloco encriptado (tem de ter o mesmo tamanho do bloco).

Desvantagem: Apesar do criptograma ter o mesmo tamanho, tem de ser guardado com ele o IV, pelo que na prática o resultado ocupa mais espaço que a informação original.

Na prática antes de encriptado, cada bloco é XORed com o resultado da encpritação do bloco anterior.

Isto permite que se for perdito o criptograma até ao bloco n, pode ser feita a decifra do restante a partir do n+1, uma vez que vai utilizar como *feedback* o criptograma n.

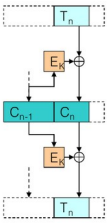
ECB/CBC: Problemas de alinhamento

Para além de no caso CBC ser necessário um IV, há outro problema que é comum aos dois modos descritos: o tamanho dos blocos é fixo, pelo que o criptograma terá sempre um tamanho múltiplo do tamanho do bloco. A forma como a mensagem original pode ser "adaptada" a este tamanho pode variar.

Uma solução é o **padding**, que consiste em adicionar ao último bloco os X bytes que faltam para ter o comprimento necessário, sendo que cada byte vai ter o valor X.

Nesta solução, se a mensagem final pode ter entre 1 e X bytes extra. Como tem de ser sempre adicionado o **padding**, caso a mensagem tenha comprimento múltiplo do tamanho dos blocos terá de ser adicionado um ploco só de **padding**, que terá o comprimento do tamanho dos blocos.

Outra abordagem é **cifrar o último bloco de forma diferenciada**, que consiste em voltar a encriptar o último bloco encriptado e fazer XOR entre este e o último bloco.



Cifras simétricas contínuas

Para a **implementação** destes algoritmos são utilizados **geradores pseudo-aleatórios** de chaves **sem sincronização** (ou seja, não relacionados com o texto a cifrar) e **sem possibilidade de acesso aleatório rápido**.

Os algoritmos mais usados são o A5/1, RC4 (WEP), E0 (Bluetooth), SEAL (c/ acesso aleatório uniforme), Chacha20, Salsa20.

Este tipo de algoritmos é bastante comum na encriptação de telecomunicações dada a sua simplicidade e rapidez.

Linear Feedback Shift Register

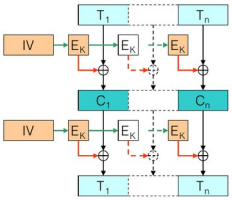
Blocos que permitem a geração de chaves pseudo-aleatórias.

Modos de encriptação

Todos os modos apresentados previnem a existência de padrões no criptograma, uma vez que introduzem confusão através da operação XOR. Não há difusão.

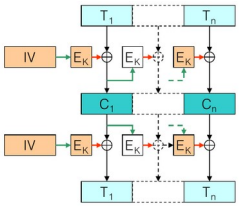
O tamanho dos blocos pode ser personalizado, pelo que não há necessidade de padding.

Output Feedback (OFB)



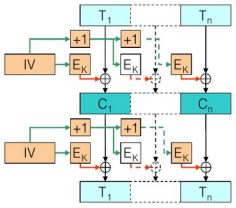
Este modo consiste em cifrar o IV e fazer XOR com o texto do bloco a encriptar, o IV encriptado é *shifted* e novamente cifrado e o processo repete-se.

Apesar de ser feita em bloco, o bloco é utilizado na criação de uma **chave contínua** (*key stream*), esta sim que vai ser utilizada na cifra do texto através do XOR.



Ciphertext Feedback (CFB)

Neste modo o criptograma *shifted* é utilizado como *feedback*.



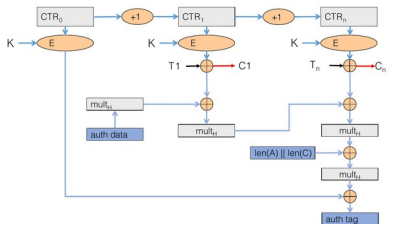
Counter (CTR)

Aqui o *feedback* é o IV, incrementado de 1 a cada iteração.

Galois w/ Counter Mode (GCM)

Esta é uma pequena variação do anterior, onde associado ao modo contador está a realização de operações de manipulação do criptograma com uma chave introduzida pelo utilizador.

Este modo permite a **deteção de criptogramas corrompidos**.



Os vários modos de encriptação

Cada modo apresenta as suas vantagens e desvantagens, sendo que as primeiras geralmente implicam as segundas, como em quase tudo no mundo da tecnologia.

	Bloco		Contínua (Stream)			
	ECB	**CBC**	**OFB**	**CFB**	**CTR**	**GCM**
Ocultação de padrões no texto		✓	✓	✓	✓	✓
Confusão na entrada da cifra		✓		✓	Contador Secreto	Contador Secreto
Mesma chava para mensagens diferentes	✓	✓	Outro IV	Outro IV	Outro IV	Outro IV
Dificuldade de alteração	✓	✓ (...)				✓
Pré-processamento			✓		✓	✓
Paralelização	✓	decifra	com pré. proc.	decifra	✓	✓
Acesso aleatório uniforme						
Propagação de erros		próximo bloco		alguns bits seguintes		detetado
Capacidade de re-sincronização	perda de blocos	perda de blocos		perda de múltiplos n-bits		detetado

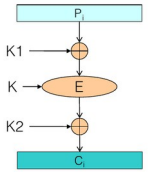
Reforço da segurança

Cifra múltipla

Cifrar o texto 2 ou 3 vezes (EDE – *Encrypt Decript Encrypt*).

A solução da cifra dupla é vulnerável!

Branqueamento



Para introduzir **confusão** podemos introduzir XORs com chaves (dinâmicas ou estáticas) antes ou depois da encriptação.

Este método facilita ainda a remoção de padrões.

XEX (XOR-Excrypt-XOR)

Este é um caso particular do anterior, onde a chave para os XOR é dinâmica.