

Resumo de

SIO - 1

Escrito por Diogo Silva

**adaptando os slides e imagens fornecidos para a
disciplina**

Index

- I. Introdução
- II. Vulnerabilidades
- III. Criptografia
- IV. Gestão de Chaves Assimétricas
- V. SmartCards e Cartão de Cidadão

Introdução

I. Objetivos e adversidades da Segurança

Sistemas têm que enfrentar diversas adversidades:

- **Catástrofes**
 - Fenómenos Naturais
 - Temperatura Anormal
 - Relâmpagos
 - Picos de Energia
 - Inundações
 - Radiação
 - ...
- **Degradação dos Sistemas Informáticos Físicos**
 - Setores Degradados
 - Falhas na fonte de alimentação
 - Erros em Células RAM ou SSD
 - ...

Como tal, é necessária a implementação de manobras e soluções para combater estas falhas:

- Pervenção Realista
 - Focarmo-nos nos eventos mais prováveis
- Replicação

- Cópias (Backups) de Informação
- Informação
- Recursos Computacionais

Alguns dos **erros mais comuns** consistem em:

- Falhas de Energia
- Falhas internas aos OS
 - Linux Kernel Panic / Windows Blue Screen
 - Bloquios
 - Consumo anormal de recursos
- Erros no Software
- Erros nas Comunicações

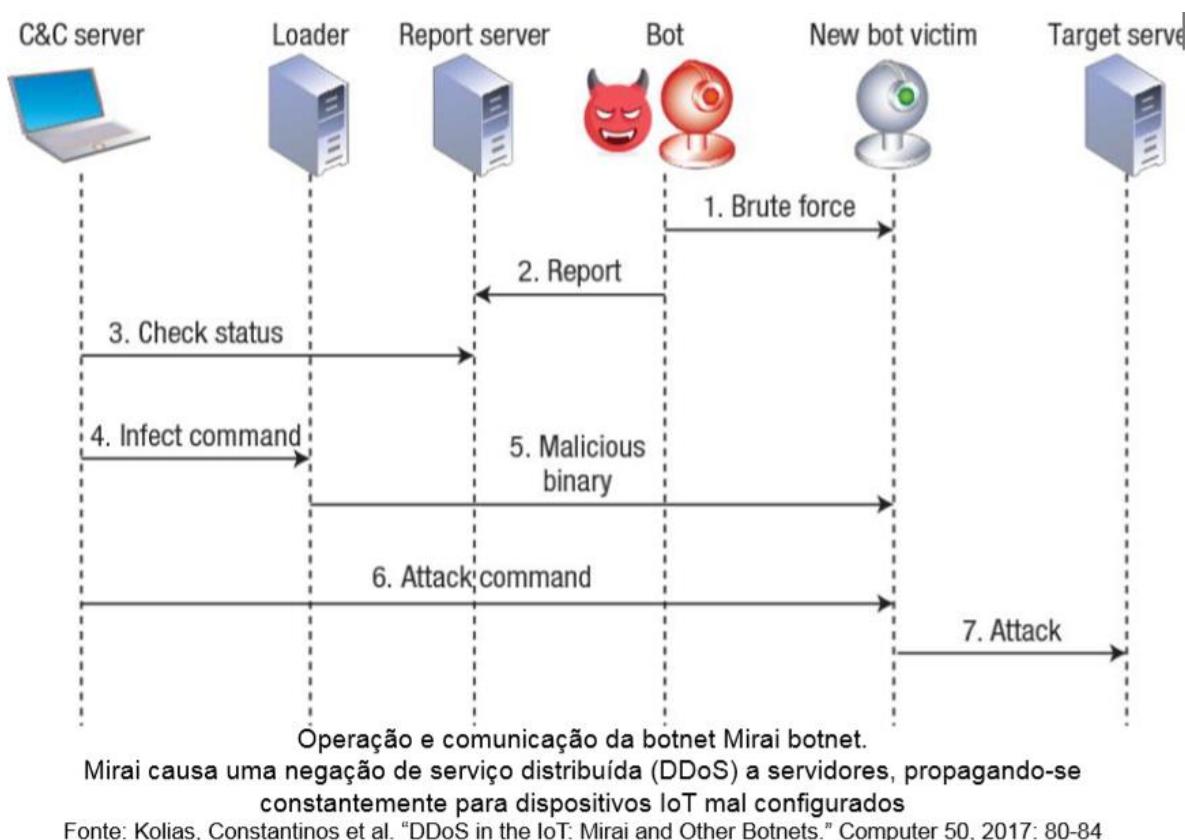
E as suas **soluções** incluem:

- Redundância de componentes
- Sistemas Transacionais
- Encaminhamento dinâmico
- Retransmissões

Para além destes acontecimentos mais “naturais”, temos também que nos proteger e **defender contra atividades não autorizadas** (sendo que estas podem ser instigadas por alguém de dentro ou de fora do sistema)

Alguns tipos de **atividades não autorizadas** incluem:

- Acesso a Informação
- Alteração de Informação
- Utilização de recursos (CPU, memória, impressão, rede, ...)
- Negação de Serviço (DoS ou DDoS)
- Vandalismo
- Interferencia do funcionamento normal sem beneficio direto para o atacante



Img 1.1 – Exemplo de um DDoS attack

II. Catalisadores dos problemas de Segurança

Como foi dito, há várias adversidades que os Sistemas Informáticos e Computacionais têm que enfrentar.

Isto deve-se também devido ao facto que:

- **Computadores podem fazer muitos estragos em muito pouco tempo**
 - São capazes de processar grandes quantidades de informação de forma rápida
- **O numero de vulnerabilidades está sempre a aumentar**
 - Complexidade incremental dos sistemas
 - Pressões do mercado
- **Redes permitem novos mecanismos de ataque**
 - Ataques anônimos de qualquer ponto do planeta
 - Ataques distribuidos sobre varias geografias
 - Exploração de aplicações e sistemas inseguros
- **Usuários não possuem noção dos riscos**
 - Não estão informados sobre o problema de segurança informatica, o impacto, boas práticas ou as soluções e medidas a tomar

- **Usuários são desleixados**

- Tomam riscos
- Não querem saber
- Não estimam o risco de forma adequada
- São esúpidos

III. Aproximação Pragmática

É impossível proteger de forma perfeita um sistema

Segurança tem **custos elevados**:

- Necessita de tecnologia dedicada, recursos adicionais, profissionais treinados, processos específicos, ...
- Organizações querem instalar **o mínimo necessário**

O que devemos fazer é encontrar um **equilíbrio entre custo e eficiência da segurança**.

Práticas que **devemos seguir** incluem:

- Devemos adotar uma proteção **suficientemente boa** para os ataques mais comuns
- Provocar menos interferências nas tarefas do que o dano causado pelos atacantes

- Reportar ataques às entidades de justiça
- Não permitir a existência de uma noção de impunidade

IV. Glossário

Vulnerabilidade

- Fraqueza do sistema que o torna sensível a ataques
- Pode estar presente em qualquer ponto do ciclo de vida

Ataque

- Série de ações que levam à execução de atividades ilegais
- Exploram vulnerabilidades

Risco / Ameaça

- Dano resultante de um ataque

Defesa

- Conjunto de políticas, mecanismos e tecnologias que têm os objetivos de:
 - Reduzir o número de vulnerabilidades

- Detetar ataques passados, atuais ou futuros
- Reduzir o risco para os sistemas

V. Riscos da Segurança

Informação, tempo e recursos

- Destrução ou alteração de informação

Confidencialidade

- Acesso não autorizado a informação

Privacidade

- Recolha não autorizada de informação pessoal
- Armazenamento (ou distribuição/venda) dessa informação

Disponibilidade de recursos

- Disrupção de sistemas, comunicações ou processos

Impersonificação

- Exploração não autorizada de perfis de identidade

VI. Principais fontes de Vulnerabilidades

Aplicações hostis ou erros em aplicações

- Root Kits
 - Inserem elementos no OS
- Worms
 - Programas controlados por um atacante
- Vírus
 - Código executável para infetar ficheiros (i.e Macros)

Usuários

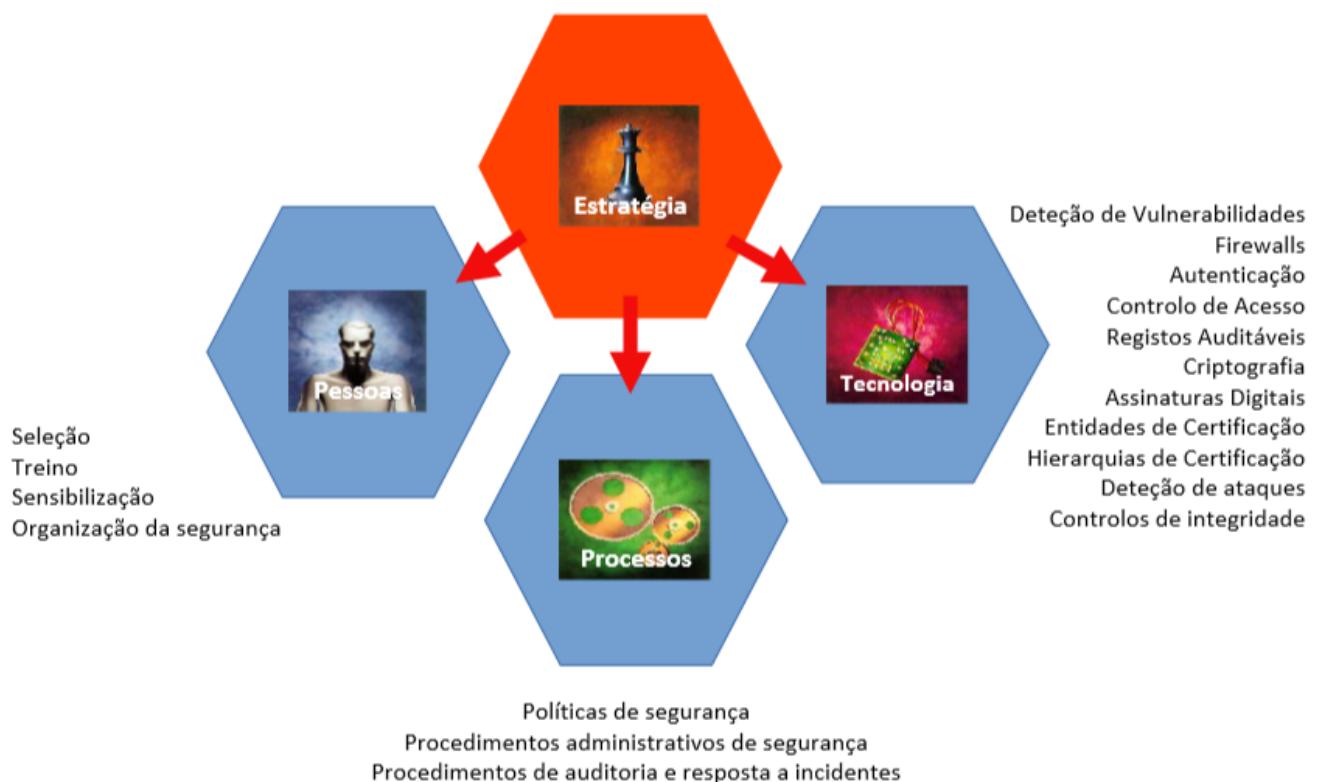
- Ignorantes e descuidados
- Falsa noção de segurança (Lol, tenho um anti-virus claro que estou protegido oh bro)
- Hostis

Administração deficiente

- Configuração por omissão raramente é a mais segura
- Restrições de Segurança vs Operações Fléxiveis
- Exceções a indivíduos

Comunicações sobre ligações não controladas/conhecidas

VII. Dimensões da Segurança



Img 7.1 – Dimensões a considerar para segurança

VIII. Políticas de Segurança

Conjunto de orientações relativas à segurança que regem um domínio

Organizações possuem uma hierarquia de políticas

- Aplicáveis a cada domínio particular
- Podem existir sobreposições
- Podem possuir ambitos e níveis de abstração distintos

As políticas devem ser coerentes entre si

Definem o poder de cada sujeito

- Princípio do privilégio mínimo
- Hardening

Definem os procedimentos de segurança

- Quem fez o quê e quando

Definem os requisitos mínimos de segurança nos sistemas

- Níveis de segurança
- Grupos de segurança

- Autorizações e Autenticação correspondentes (fraca/forte, simples/multifatorial, remota/presencial)

Definem as estratégias de defesa e de resposta

- Arquitetura defensiva
- Monitorização de atividades críticas
- Deteção de sinais de ataques
- Reação a ataques ou outras disruptões

Definem o que é correto e incorreto (legal/illegal)

- Modelo de lista negra (Blacklist)
- Modelo de lista branca (Whitelist)

IX. Mecanismos de Segurança

Mecanismos implementam políticas

Enquanto que políticas definem as orientações, os mecanismos **tornam as políticas efetivas** (/implementam as políticas)

Mecanismos de Segurança Genéricos

- Confinamento

- Autenticação
- Controlo de acesso
- Execução Privilegiada
- Filtragem
- Registo
- Algoritmos e protocolos criptográficos
- Auditorias

X. Níveis de Segurança

São definidos por

- **Políticas de segurança existentes**
- **Correção e efetividade da sua especificação/implementação**

Critério de Avaliação NCSC TCSEC, Orange Book

- Divisão por classes: D, C(1,2), B(1,2,3) e A(1)
 - D: Inseguro
 - A1: Mais seguro
 - Políticas de proteção existentes e dispendiosas
 - Procedimentos formais de validação da especificação
 - Controlo rigoroso da implementação

- **C1 – Discretionary Security Protection**
 - Identificação e Autenticação
 - Separação de utilizadores e dados
 - Controlo de acesso discricionário (DAC), capaz de aplicar limites de acesso por utilizador
 - Necessário existir documentação do sistema e manuais
- **C2 – Controlled Access Protection**
 - DAC com mais detalhe
 - Rastreio individual das ações através de mecanismos de login
 - Registos para auditorias
 - Limpeza de objetos ao serem re-usados (**Object Reuse**)
 - **Política:**
 - Todas as autorizações para a informação contidas dentro de um storage object devem ser revocadas antes do initial assignment, alocação ou realocação para um sujeito através de TCB pool of unused storage objects
 - Nenhuma informação, incluindo representações criptadas desta, produzidas por ações anteriores de um

sujeito devem ser available a qualquer sujeito que obtenha acesso a um objeto que tenha sido lançado de volta para o sistema

- **Storage Object:**
 - Objeto que suporta tanto read como write accesses
- Isolamento de recursos

Critério de Avaliação ITSEC

- Divisão por níveis: E1 até E6
- Nível de especificação formal e correção de implementação

XI. Políticas de Segurança em Sistemas Distribuídos (SD) - Definição

Têm que englobar múltiplos sistemas e redes

Domínios de segurança:

- Definição de um conjunto de sistemas e rede
- Definição de um conjunto de usuarios aceites/autorizados

- Definição de um conjunto de atividades aceites/não aceites

Gateway de segurança:

- Definição das interações de entrada e saída de um domínio

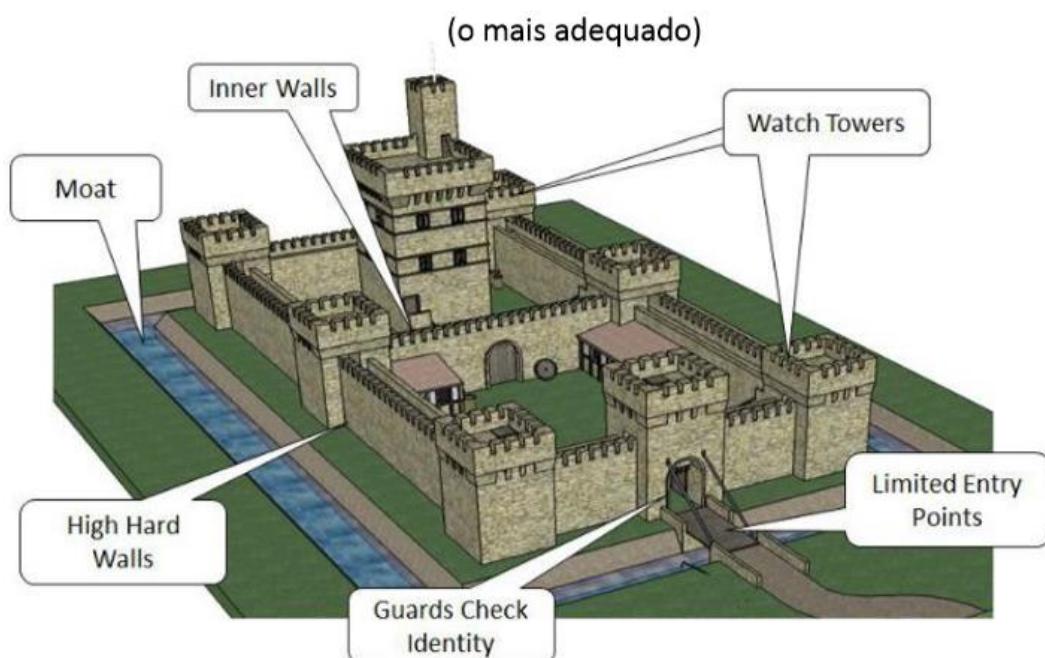
Defesa em perímetro

(mínimo aconselhado mas insuficiente)

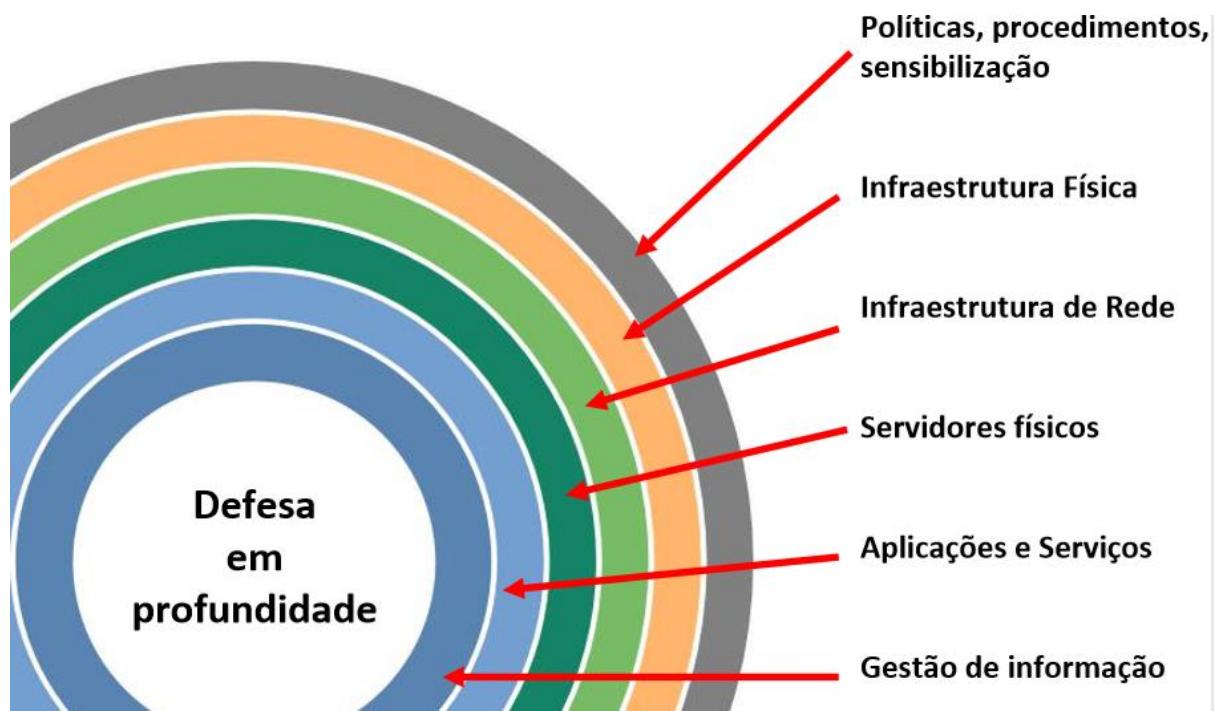


Img 11.1 – Defesa em perímetro

Defesa em profundidade



Img 11.2 – Defesa em profundidade – Metáfora do Castelo



Img 11.3 – Defesa em profundidade

XII. Políticas de Segurança em Sistemas Distribuídos (SD) – Tipos de Ataques

Ataques específicos:

- Concebidos para um sistema/rede particulares
- Idealizados e concebidos in real time por especialistas

Ataques genéricos/autónomos:

- Exploram vulnerabilidades conhecidas/comuns
- Implementadas para muitos sistemas
- Afetam o tempo médio de sobrevivência
 - Duração entre dois ataques automáticos consecutivos
 - Existe uma rede de sensores de rede a calcular isto
- Executados por profissionais, curiosos, estudantes, trolls, show-offs, fdp's, etc

XIII. Mecanismos de Segurança para SD

Sistemas Operativos Confiaveis:

- Níveis de Segurança
- Certificação
- Ambientes de execução Segura

- Sandboxes / Máquinas Virtuais

Firewalls e sistemas de segurança:

- Controlo de tráfego entre redes
- Monitorização (carga de tráfego, comportamento, ...)

Comunicações Seguras / VPNs:

- Canais seguros sobre redes públicas / inseguras
- Extensão segura das redes da organização

Autenticação:

- Local
- Remota (sobre a rede)
- Single Sign-On
- Segredos, tokens, biometria, dispositivos, localização

Entidades de Certificação / PKI:

- Gestão de chaves públicas e certificados

Cifra de Ficheiros e dados em sessões:

- Privacidade / Confidencialidade de dados transmitidos
- Privacidade / Confidencialidade de dados armazenados

Deteção de Intrusões:

- Deteção de atividades proibidas ou anómalas

- Baseado na rede / Baseado nos sistemas

Inventariação de Vulnerabilidades:

- Pesquisa para resolução de problemas ou exploração
- Baseado na rede / Baseado nos sistemas

Testes de Penetração (^_>^_):

- Avaliação das vulnerabilidades
- Demonstração de tentativas de penetração
- Teste de mecanismos de segurança instalados
- Determinação da existencia de politicas de segurança mal aplicadas

Monitorização de Conteúdos:

- Deteção de virus, worms e outras Ciber-Pragas

Administração da Segurança:

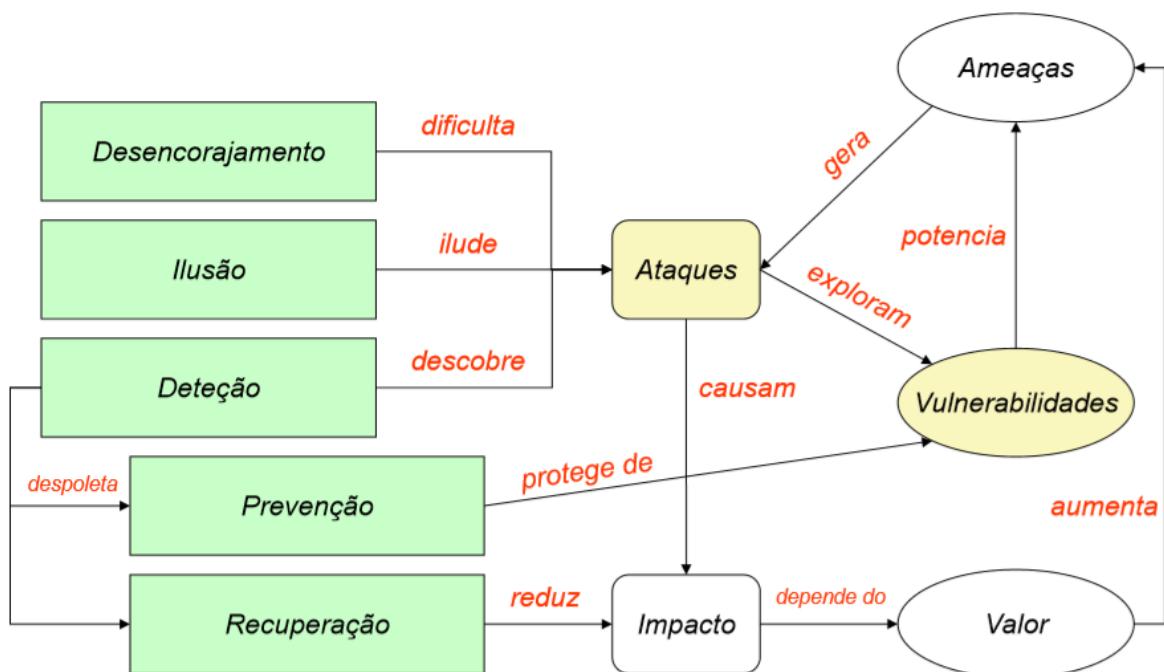
- Desenvolvimento de politicas de segurança
- Aplicação das politicas de forma distribuida
- Co-Administração / Contração de Equipas Externas

Respostas a Incidentes / Seguimento em Tempo Real:

- Capacidade para detetar e reagir a incidentes em tempo real
- Meios para resposta rápida e efetiva a incidentes

Vulnerabilidades

I. Segurança da Informação – Medidas e relações



Img 1.1 – Relações sobre Segurança

Desencorajamento:

- Punição
 - Restrições Legais
 - Provas Forenses
- Barreiras de Segurança
 - Firewalls
 - Autenticação

- Comunicação Segura
- Sandboxing

Ilusão:

- Honeypots / Honeynets (Bait attackers)
- Acompanhamento Forense

Prevenção:

- Políticas Restritivas
- Deteção de Vulnerabilidades
- Correção de Vulnerabilidades

Deteção:

- Sistemas de Deteção de Intrusões
- Auditorias
- Análise Forense

Recuperação:

- Backups
- Sistemas redundantes
- Recuperação forense

II. Prontidão – Security Readiness

Medidas de **Desencorajamento, Ilusão e Deteção** endereçam maioritariamente **Vulnerabilidades Conhecidas**

- Tentativas de Reconhecimento

- Ataques genéricos
- Ataques específicos

Medidas de **Prevenção** endereçam **Vulnerabilidades conhecidas e desconhecidas**

- Vulnerabilidades Genéricas
- Vulnerabilidades Específicas

A aplicação das medidas requer conhecimento específico

Conhecimento sobre:

- **Vulnerabilidades Conhecidas**
- **Padrões de atividades dos ataques**
- **Padrões anormais de atividade**

As ameaças em rede de computadores são diferentes de outros tipos de ameaças

- Ataques podem ser lançados em qualquer hora, de qualquer local
- Podem ser coordenados (ex. DDoS)
- São baratos
- Podem ser automatizados
- São rápidos

Por todas estas razões, **requerem uma capacidade permanente (24/7) de reação a ataques**

- Equiaps de especialistas em segurança
- Alertas de ataque na hora
- Teste e avaliação dos níveis de segurança existentes
- Procedimento de reação expeditos

III. Zero Day Attacks

Ataques que usam vulnerabilidades desconhecidas de terceiros e que não são comunicadas ao fornecedor de software

Ocorre **no dia zero do conhecimento dessas vulnerabilidades** (para as quais não existe correção ou não está aplicada)

Podemos ter 0 Day attacks que existam por meses ou até anos

- Conhecidos por atacantes mas não utilizadores
- Parte frequente de arsenais de ataques informáticos
- Comercializados em mercados específicos

IV. Shadobrokers (is that a motherfucking Mass Effect reference?)

São atores estatais que possuem arsenal para explorar vulnerabilidades desconhecidas ao público

A parte integrante das suas atividades, por muitos anos, nunca são reveladas

Agosto 2016: Shadowbrokers publicam um grande quantidade de ferramentas deste atores

- Usando canais públicos: Twitter, Github, PasteBin, Medium
- Apresentam outros conjuntos de ferramentas: fazem um leilão, fazem uma venda de Black Friday, etc...
- Objetivo: vender ferramentas que exploram 0 days a quem pagar mais

Março 2017: Microsoft lança atualizações para várias versões de Windows

- mas não lança para o W7, W8, XP e Server 2003
- poderá ter existido dica de investigadores ou atores estatais
- gravidade da atualização não é realçada

Img 4.1 – Exemplos de atividades dos Shadowbrokers

V. Detecção de Vulnerabilidades

Existem **Ferramentas Específicas** que podem **Detetar Vulnerabilidades**

- Exploram vulnerabilidades conhecidas

- Testam padrões de vulnerabilidades (e.g SQL Injection, XSS, ...)

Existem também **Ferramentas Específicas** capazes de **Replicar Ataques Conhecidos**

- Utilizam exploits conhecidos para vulnerabilidades conhecidas
- Permitem implementar correções mais rapidamente

Todas estas ferramentas são vitais para aferir a robustez das aplicações e sistemas em operação

Por norma contratam-se pessoas que saibam utilizar estas aplicações para fazer um relatório de segurança

Podem ser aplicadas a:

- Código desenvolvido – Análise Estatística
- Aplicação a executar – Análise Dinâmica
- Externamente como um sistema remoto

Não devem ser aplicadas à toa e de forma cega e sistemas em produção porque acarretam riscos de:

- Potencial perda/corrupção de dados
- Potencial negaão de serviço

- Potencialmente ilegais

VI. Sobrevivência

Como se sobrevivem a ataques de dia zero?

Como é que podemos reagir a um massive zero day attack?

Diversidade de sistemas poderia ser uma solução, porém a produção distribuição e atualização de software vai no sentido contrario a esta:

- Porquê que o Windows é um grande alvo de ataques enquanto que o MacOS nem por isso?
- Android até pode estar na linha da frente de alvos de ataques, mas como cada produtor tem a sua versão específica e alterada há muita heterogenidade, pelo que o iOS acaba por ter mais riscos – devido a ser um ecossistema homogéneo

VII. CVE: Common Vulnerabilities and Exposures

É um dicionário público de vulnerabilidades e exposições de segurança

Usado para:

- Gestão de vulnerabilidades
- Gestão de correções – Patches
- Alarmística de vulnerabilidades
- Deteção de intrusões

Utiliza identificadores comuns para um mesmo CVE

- Permite a troca de informação entre produtos de segurança
- Fornece uma base de indexação para avaliar a abrangência de ferramentas e serviços

Detalhes de um CVE podem ser privados

- Parte do **processo de divulgação responsável**:
 - Espera-se que o fornecedor crie uma correção e só depois é que se publica abertamente a vulnerabilidade

VIII. CVE: Vulnerabilidade

Erro no software que pode ser usado diretamente por um atacante para ganhar acesso ao sistema ou à rede

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

- Exclui politicas de segurança abertas onde todos os utentes são de confiança ou onde não se considere a existencia de riscos para o sistema

Uma Vulnerabilidade é um estado de um sistema computacional que alternativamente permite:

- Que um atacante **execute comandos em nome de terceiros**
- Que um atacante **aceda a dados ultrapassando as restrições de acesso**
- Que o atacante **se apresente como outrem**
- Que o atacante **negue a prestação de serviços**

IX. CVE: Exposição

Problema de configuração de um sistema ou um erro no software que permite aceder a informação ou capacidades que podem auxiliar um atacante

O CVE considera um problema de configuração ou um erro como uma exposição se não permitir comprometer diretamente um sistema ou rede

- Mas se for uma componente importante para o sucesso de um ataque e uma violação de uma política de segurança expectável

Uma exposição é um estado de um sistema computacional que alternativamente:

- Permite que um atacante realize **recolhas de informação**
- Permite a um atacante **esconder as suas atividades**
- Inclui uma funcionalidade que se comporta como esperado mas que pode ser **facilmente comprometida**
- Ponto de entrada comum para atacantes **obterem acesso**
- É considerado **problemático por uma política de segurança razoável**

X. CVE: Benefícios e Limitações

Fornecem uma linguagem comum para referir problemas

Facilita a partilha de dados entre:

- Sistemas de deteção de intrusões
- Ferramentas de aferição
- Bases de dados de vulnerabilidades
- Investigadores
- Equipas de resposta a incidentes

Permite melhorar as ferramentas de segurança

- Maior abrangência, facilidades de comparação, interoperabilidade
- Sistemas de alarme e reporte

Fomenta a inovação

- Local primordial para discutir conteudos críticos das BD

Porém...

SÃO INUTEIS CONTRA 0-DAY ATTACKS!

XI. CVE: Identificadores

Também conhecidos por CVE Names, CVE Numbers, CVE-Ids ou CVEs...lmao

Identificadores únicos para vulnerabilidades conhecidas e públicas da CVE List

Estados possíveis:

- **Candidate**
 - Sob revisão para inclusão na CVE List
- **Entry**
 - Aceite na CVE List

Formato

- Identificador Número/ci CVE
 - CVE-Ano-Índice
- Estado
- Descrição sumária da vulnerabilidade ou exposição
- Referências para informação adicional

CVE-2015-1538, P0004, Google Stagefright ‘ctts’ MP4 Atom Integer Overflow Remote Code Execution

Img 11.1 – Exemplo de um CVE Name

CVE-ID	
CVE-2015-1538	Learn more at National Vulnerability Database (NVD)
	• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information
Description	Integer overflow in the SampleTable::setSampleToChunkParams function in SampleTable.cpp in libstagefright in Android before 5.1.1 LMY48I allows remote attackers to execute arbitrary code via crafted atoms in MP4 data that trigger an unchecked multiplication, aka internal bug 20139950, a related issue to CVE-2015-4496.
References	<p>Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.</p> <ul style="list-style-type: none"> • BID:76052 • URL:http://www.securityfocus.com/bid/76052 • CONFIRM:http://www.huawei.com/en/psirt/security-advisories/hw-448928 • CONFIRM:http://www1.huawei.com/en/security/psirt/security-bulletins/security-advisories/hw-448928.htm • CONFIRM:https://android.googlesource.com/platform/frameworks/av/+/2434839bbd168469f80dd9a22f1328bc81046398 • EXPLOIT-DB:38124 • URL:https://www.exploit-db.com/exploits/38124/ • MISC:http://packetstormsecurity.com/files/134131/Libstagefright-Integer-Overflow-Check-Bypass.html • MLIST:[android-security-updates] 20150812 Nexus Security Bulletin (August 2015) • URL:https://groups.google.com/forum/message/raw?msg=android-security-updates/Ugyu3fi6RQM/yzJvoTVr1QAj • SECTRACK:1033094 • URL:http://www.securitytracker.com/id/1033094
Assigning CNA	MITRE Corporation
Date Entry Created	
20150206	Disclaimer: The entry creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.
Phase (Legacy)	
Assigned (20150206)	
Votes (Legacy)	
Comments (Legacy)	
Proposed (Legacy)	
N/A	
This is an entry on the CVE List , which provides common identifiers for publicly known cybersecurity vulnerabilities.	
SEARCH CVE USING KEYWORDS:	<input type="text"/> <input type="button" value="Submit"/>
You can also search by reference using the CVE Reference Maps .	
For More Information: CVE Request Web Form (select "Other" from dropdown)	

Img 11.12 – Exemplo de um CVE

XII. CVE: Ataques

Ataques podem usar várias vulnerabilidades

Temos então um CVE para cada vulnerabilidade em todos os sistemas

Exemplo: Stagefright (Android, video em mensagens MMS)

- CVE-2015-1538, P0006, Google Stagefright ‘stsc’ MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1538, P0004, Google Stagefright ‘ctts’ MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1538, P0004, Google Stagefright ‘stts’ MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1538, P0004, Google Stagefright ‘stss’ MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-1539, P0007, Google Stagefright ‘esds’ MP4 Atom Integer Underflow Remote Code Execution
- CVE-2015-3827, P0008, Google Stagefright ‘covr’ MP4 Atom Integer Underflow Remote Code Execution
- CVE-2015-3826, P0009, Google Stagefright 3GPP Metadata Buffer Overread
- CVE-2015-3828, P0010, Google Stagefright 3GPP Integer Underflow Remote Code Execution
- CVE-2015-3824, P0011, Google Stagefright ‘tx3g’ MP4 Atom Integer Overflow Remote Code Execution
- CVE-2015-3829, P0012, Google Stagefright ‘covr’ MP4 Atom Integer Overflow Remote Code Execution

Img 12.1– Exemplo Stagefright

XIII. CWE: Common Weakness Enumeration

Linguagem comum para discutir, encontrar e lidar com as causas das vulnerabilidades de segurança de programas, do seu desenho ou da arquitetura de sistemas

Cada CWE Representa um tipo de vulnerabilidade

Gerida pela MITRE Corporation que disponibiliza uma lista de CWEs

Esta lista fornece uma definição pormenorizada de cada CWE

Os CWEs são **catalogados** segundo uma **estrutura hierárquica**

- CWEs localizados nos **níveis superiores** fornecem uma **descrição genérica** sobre o tipo de vulnerabilidade
- CWEs nos **níveis inferiores** **descrevem** problemas de **forma** mais **focada**

- 1. Validação e representação de entradas**
- 2. Abuso de API:** falhas na utilização de interfaces
- 3. Funcionalidades de segurança:** más práticas
- 4. Tempo e estado:** threads, concorrência
- 5. Erros:** má geração ou recuperação
- 6. Qualidade do código**
- 7. Encapsulamento**
- 8. *Ambiente de execução:** configurações e características

Img 13.1– Fatores abordados pelos CWEs

XIV. CERT: Computer Emergency Readiness Team

Organização para garantir que as práticas de gestão de tecnologias e sistemas são usadas para:

- Resistir a ataques em sistemas distribuidos
- Limitar o dano, garantir a continuidade de serviços críticos
- Mesmo considerando ataques realizados com sucesso, prevener e prever acidentes e falhas

CERT/CC (Coordination Center) @ CMU

- Componente do CERT Program
- Hub para questões de segurança na internet
- Tem demonstrado a crescente exposição da Internet a ataques

XV. CSIRT: Computer Security Incident Response Team

Organização responsável por receber, erver e responder a relatórios de incidentes e atividade

- Fornece serviço 24/7 para usuarios, companhias, agencias governamentais e organizações

- Fornece um ponto unico de contacto fiavel e confiavel para reportar incidentes de segurança a escala global
- Fornecem os meios para reportar incidentes e disseminar informação relativa a incidentes

CSIRTs Nacionais

- CERT.PT: <https://www.facebook.com/CentroNacionalCibersegurancaPT>
- National CSIRT Network: <https://www.redecsirt.pt>
- CSIRT @ UA: <https://csirt.ua.pt>

Img 15.1– Canais sociais dos CSIRTs Portugueses

XVI. Alertas de Segurança e Tendências de Atividades

Vitais para a disseminação rapida de conhecimento sobre novas vulnerabilidades

US-CERT Technical Cyber Security Alerts

US-CERT (non-technical) Cyber Security Alerts

SANS Internet Storm Center

- Aka DShield (Defense Shield)

Microsoft Security Response Center

Cisco Security Center

Img 16.1– Exemplos de Alertas

XVII. Regras e Considerações a ter para propiciar a Segurança

Endereçar a Segurança como um todo

Considerar frameworks existentes:

- Melhores práticas e recomendações

Considerar requisitos normativos

- Adicionar verificações de risco e estratégias de resolução

Considerar os aspectos legais

- Leis a obedecer, regulamentos e questões contratuais

Criar controlos e garantir que estes endereçam os requisitos

Avaliar o funcionamento do programa de segurança

Identificar e Gerir o Risco

Considerar o risco específico para sistema/negócio/operações:

- Ter em conta os aspetos operacionais e tecnologias em utilização
- Ter em conta os dispositivos e interações com terceiros (e.g pagamentos com cartões)

Identificar o risco em todas as áreas da organização:

- Tecnologia, relações com terceiros, pessoas

Definir medidas preventivas para reduzir o risco:

- Considerar o ataque e o impacto na organização

Avaliar periódicamente o risco

Seguir a informação

Informação contém valor

- Atacantes: Ataques focam-se em áreas com maior valor
- Regulamentar: Fugas podem implicar multas altas
- Negócio: Fugas/manipulações podem implicar perdas elevadas

Conhecer bem onde está a informação em cada momento

- Quem a manipula
- Onde é armazenada
- Por onde circula

Classificar informação de acordo com risco/visibilidade

- Confidencial, privada, pública, dados pessoais

Aplicar medidas de defesa em profundidade

A superfície de ataques é extensa

- Adversários externos ou que ganham acesso interno
- Colaboradores

Garantir que existem controlos adequados e suficientes

- Conciliar deteção de fugas / manipulação e alteração
- Considerar colaboradores, terceiros, público em geral

Considerar métodos físicos

- Air Gaps, Portas, Infraestruturas, etc...

Aplicar requisitos de segurança na linguagem da organização

Alinhar a segurança com objetivos, produtos, serviços

Mandatório para garantir que a segurança acompanha a organização

- Continua relevante, existe e tem impacto

Evoluir da simples proteção do que é obrigatório

- Considerar todos os dados

Ter conhecimento de como a organização opera, produtos são desenvolvidos/vendidos/operados

- Saber como aplicar a segurança

Ter conhecimento da geração de lucro

- Saber como calcular o impacto de um ataque

Antecipar, Inovar e Adaptar

Ataques, negócio e vulnerabilidades evoluem

- Necessário que a segurança acompanhe a evolução

- Seguir CSIRTS, CERTS, ...

Foco nos pontos onde existe um maior retorno da proteção

- O que é mais fácil de proteger
- O que possui maior dano (e é razoável de ser efetuado)

Considerar ataques persistentes avançados (APT)

- Não acontecem só às empresas grandes

Estabelecer uma cultura baseada na segurança

Fornecer formação aos colaboradores

- Para entenderem os riscos, impactos e mitigações
- Para conhecerem as boas práticas, mecanismos e soluções
- Que seja apoiada e aplicada em todos os níveis hierárquicos

Construir políticas que se apliquem a toda a organização

- Como parte integrante da empresa e não como um extra
- Possuir políticas que incluem a segurança no design dos produtos
- Possuir políticas que incluem fornecedores, colaboradores e clientes

Promover atividades periódicas

- Revisão das políticas
- Treino e troca de experiências

Considerar a existência de dias menos bons

Confiar mas verificar

Instalar os controlos adequados

- Tanto para atividades externas como internas

Auditorias externas são vitais

- Garantir que os mecanismos são efetivos
- Garantir que as políticas cobre os aspetos devidos
- Garantir que as leis são observadas

Testes de Invasão (PenTest) são uma ferramenta importante

- Avaliar a existência de fraquezas na aplicação das tecnologias
- Avaliar a existência de fraquezas nos colaboradores e processos

Partilhar experiências, regulamentação, incidentes e respostas

Possuir a capacidade de analisar incidentes

- Sobre toda a cadeia ou stack de processos e aplicações
- Requer a existência de registos confiáveis
- Discutir internamente de forma alargada na empresa

Exercer influência externa

- Demonstrar como aplicam a regulamentação e detetam incidentes
- Demonstrar como respondem a incidentes
- Aumentar a confiança em clientes e fornecedores

Criptografia

I. Terminologia

Criptografia

- Arte ou ciência de escrever de forma escondida/confidencial
- Inicialmente usada para garantir a privacidade da informação

Criptanálise

- Arte ou ciência de quebrar sistemas criptográficos ou informação criptografada

Criptologia

- Criptografia + Criptanálise

Cifra

- Técnica concreta de criptografia

Operação de uma Cifra

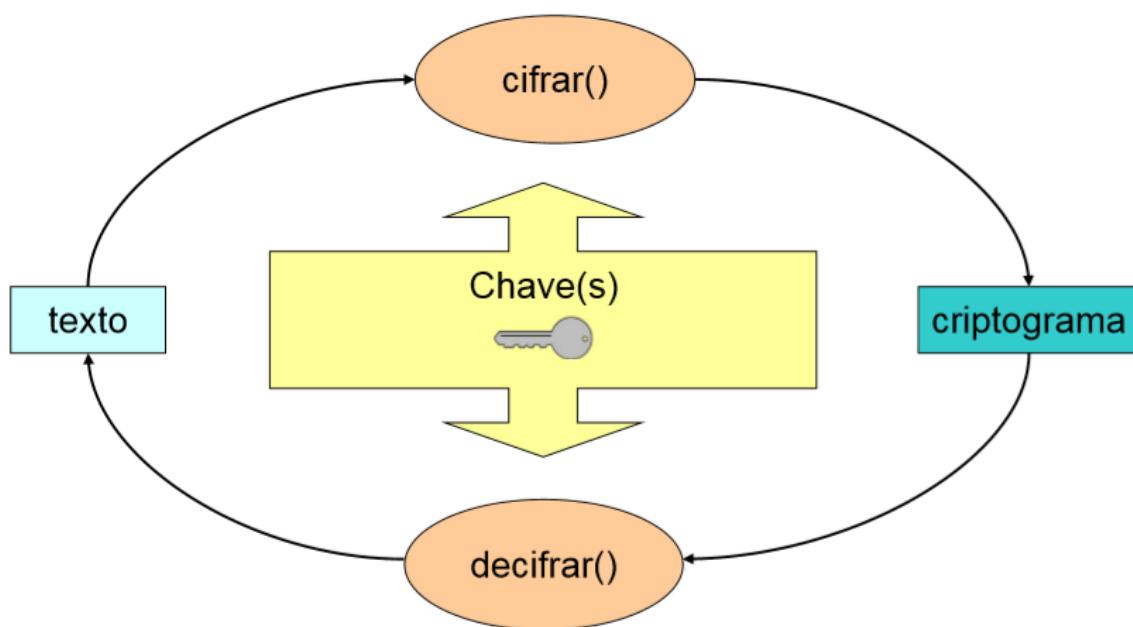
- **Cifra:** Texto normal -> Criptograma
- **Decifra:** Criptograma -> Texto normal

Algoritmo

- Modo de transformação de dados

Chave

- Parâmetro do algoritmo
 - Influencia a operação do algoritmo



Img 1.1– Operações de Criptografia

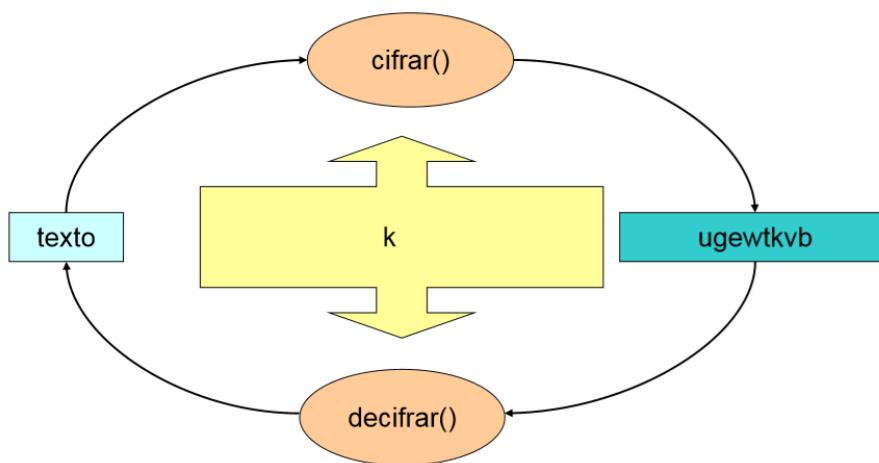
II. Exemplo de uso (com Cifras Simétricas)

Proteção usando a chave K

- Alice cifra o texto **P** com a chave **K**
 - -> Alice: $C = \{P\}_k$ (cifra **C** = **P** cifrado com **K**)
- Alice decifra **C** com a chave **K**
 - -> Alice: $P' = \{C\}_k$ (texto **P'** = **C** decifrado usando **K**)
- Se tudo tiver corrido bem, $\mathbf{P}' = \mathbf{P}$

Comunicações seguras usando a chave K

- Alice cifra o texto **P** com a chave **K**
 - -> Alice: $C = \{P\}_k$ (cifra **C** = **P** cifrado com **K**)
- Bob decifra **C** com a chave **K**
 - -> Bob: $P' = \{C\}_k$ (texto **P'** = **C** decifrado usando **K**)
- Se tudo tiver corrido bem, $\mathbf{P}' = \mathbf{P}$



Img 2.1– Exemplo da cifragem e decifragem usando a chave simétrica K

III. Criptanálise - Objetivos

Obtenção do texto original

- Relativo a um criptograma

Obtenção de uma chave de cifra

- Ou de uma equivalente

Obtenção do algoritmo de cifra

- Ou de um equivalente
- Normalmente os algoritmos não são secretos, mas existem exceções
- Pode ser feito por reverse engineering

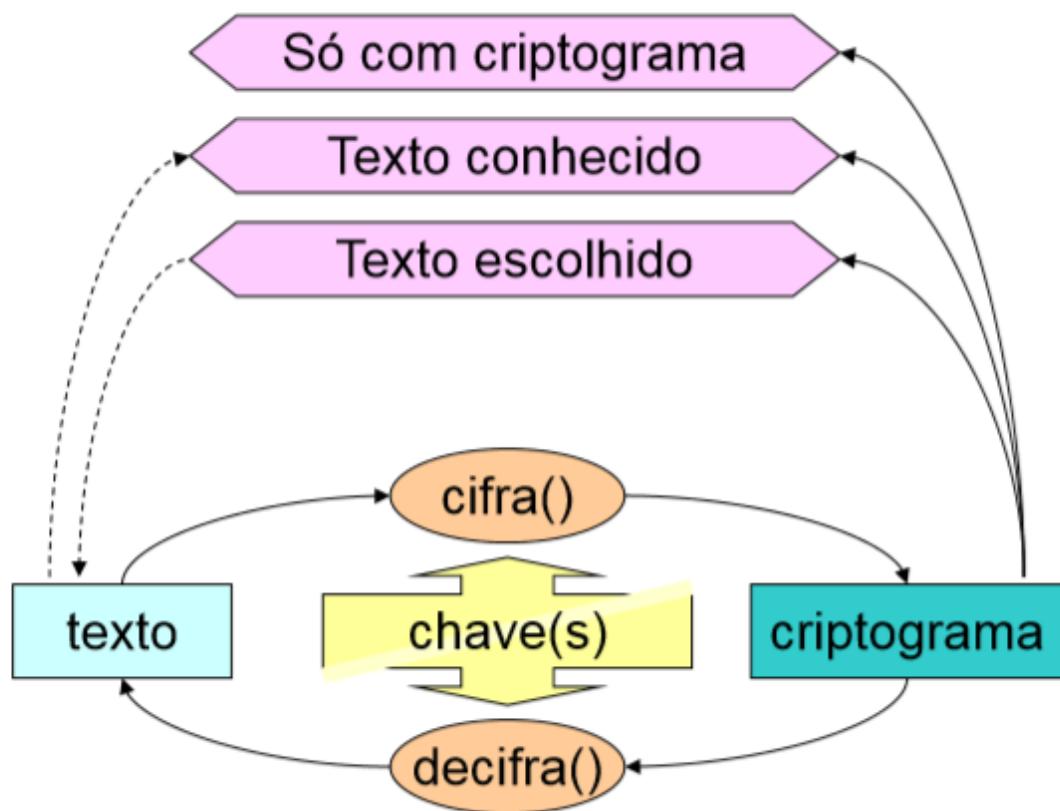
IV. Ataques por Criptanálise

Força Bruta – Ataque Genérico

- Pesquisa exaustiva sobre todo o espaço de chaves até se encontrar uma chave adequada
- Não é prática para espaços de dimensão grande
- Como prevenção deste tipo de ataques é importante que existe aleatoriedade na chave (i.e não devem haver padrões)

Ataques (mais) inteligentes

- Reduzir o espaço de pesquisa para uma dimensão menor (palavras, numeros, conjunto reduzido, alfabeto)
- Identificar padrões em algumas operações
- ...



Img 3.1– Ataques por criptanálise

V. Evolução das Cifras

Manuais – Algoritmos de substituição ou transposição

Mecânicas – Críticos para a 2a Guerra Mundial

Informáticas

- Surgem com o uso dos computadores
- Algoritmos de substituição mais complexos
- Algoritmos matemáticos de grandes números ou problemas complexos
- Utilizados de forma comum e transparente no dia a dia atual



Img 5.1– Evolução das Cifras

VI. Cifras – Tipos Básicos

Transposição – O texto original é baralhado

Transposição: O texto original é “baralhado”



O	O	I	B	H
T	O	N	A	A
E	R	A	R	D
X	I	L	A	O
T	G	E		L

Resultado: ooibh tonaa erard xilao tgel

Img 6.1– Exemplo de transposição

Transposição – Podemos refinar esta técnica usando **permutações intra-blocos**

Transposição: Permutações intra-blocos



1	2	3	4	5
P	E	R	M	U
T	A	C	O	E
S	I	N	T	R
A	B	L	O	C
O	S			

Resultado:

- (13524) -> pruem tceao snrit alcbo os
- (25413) -> eumpr aeotc irtsn bcoal so

Img 6.2– Exemplo de transposição

Substituição – Substituição de cada simbolo original por um outro. Podemos considerar como simbolos letras, digitos e pontuação

Substituição Mono-alfabética – Um para Um

- Usam apenas um alfabeto de substituição com um número de elementos igual ao numero de elementos #A
- Alguns exemplos incluem
 - **Aditivas** (ou de translação)
 - cripto – letra = (letra + chave) mod #A
 - letra = (cripto – letra – chave) mod #A
 - Numero de chaves efetivas = #A
 - Ex: Cifra de César
 - **Frase Chave**
 - Ex:
 - ABCDEFGHIJKLMNOPQRSTUVWXYZ
 - QTUWXYZCOMFRASEHVBBDGIJKLNP
 - Número de chaves efetivas = #A!
- **Problemas**
 - Reproduzem padrões do texto original
 - Letras, diagramas, triagramas, etc...
 - A analise estatistica facilita a criptanálise

Frequência de Pares

- AO, NO, AS, OS, SO, UM, IA, NA...

Probabilidades condicionais

- $P(A | B)$ diferente de $P(Z | B)$

Frequência de Triplos

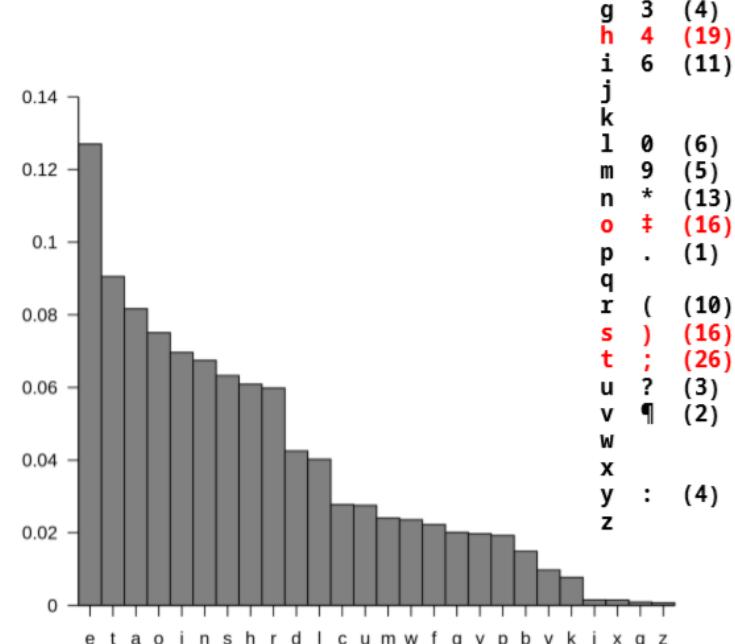
- QUE, NAO, EST, ENT, ÇÃO, TRA...

Img 6.3– Padrões comuns na Substituição Mono-alfabética

a good glass in the
bishop's hostel in the
devil's seat fifty-one
degrees and thirteen
minutes northeast and
by north main branch
seventh limb east side
shoot from the left eye
of the death's-head a
bee line from the tree
through the shot forty
feet out

53#†305))6*;4826)4#.)
4#);806*;48†860))85;1#
(;:‡*8†83(88)5*†;46(;8
8*96*?;8)*‡(;485);5*†2
:‡(;4956*2(5*-4)88*;4
069285);)6†8)4‡‡;1(#9;
48081;8:8‡1;48†85;4)48
5†528806*81(#9;48;(88;
4(#?34;48)4‡;161;:188;
‡?;

53#†305))6*;4826)4#.)
agooodglassinthebishopshostel
6*;48†8¶60))85;1#(;‡*8†83(88)
inthedevilsseatfortyonedegrees
5*†;46(;88*96*?;8)*‡(;485);5*†
andthirteenminutesnortheastand
2:‡(;4956*2(5*-4)8¶8*;40692
bynorthmainbranchseventhlimb
85);)6†8)4‡‡;1(#9;48081;8:8‡1
east sideshootfromthellefteyeof
;48†85;4)485†528806*81(#9;48
thedethsheadabeelinefromthe
;(88;4(#?34;48)4‡;161;:188;‡?
treethroughtheshotfiftyfeetout



Img 6.4– Exemplo de Cifras Mono-Alfabéticas

Substituição Poli-alfabética – Muitos para Um

- Usam N alfabetos de substituição
- Tem período N

- Exemplo
 - **Cifra de Vigenère**
- **Problemas**
 - Conhecido o periodo podem ser analisadas como N Mono-Alfabéticas
 - Podemos descobrir o periodo usando estatística
 - Método de Kasiski
 - Fatorização de distâncias entre blocos iguais do criptograma
 - Índice de coincidência
 - Fatorização de deslocamentos relativos que produzem mais coincidências na sobreposição do criptograma

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y	

Exemplo de se cifrar a letra M com a chave S, resultando no criptograma E
Criada por Blaise Vigenère (final séc XVI)

• le chiffre indéchiffrable!

Quebrada no séc XIX por Charles Babbage e Friedrich Kasiski

- **Texto:**

Eles não sabem que o sonho é uma constante da vida
tão concreta e definida como outra coisa qualquer,
como esta pedra cinzenta em que me sento e descanso,
como este ribeiro manso, em serenos sobressaltos
como estes pinheiros altos

- Cifra com o quadrado de Vigenère e chave “poema”

Img 6.6– Exemplo de Cifragem usando cifra de vigenere

- Teste de Kasiski

- Consiste em localizar padrões comuns no criptograma
 - Calcular o afastamento entre os padrões
 - O maior divisor comum sugere a dimensão da chave

tzienpcwmbtaugedqszhdsyyarcetpbxqdpjmpaiosooocqvqtphqfxbmpa

mpa	$20 = 2 \times 2 \times 5$
tp	$20 = 2 \times 2 \times 5$

- Com o texto indicado:

$$\begin{aligned}175 &= 5 \times 5 \times 7 \\105 &= 3 \times 5 \times 7 \\35 &= 5 \times 7 \\20 &= 2 \times 2 \times 5\end{aligned}$$

Imagem 6.7 – Exemplo de Cifragem usando cifra de Vigenère

• Índice de Coincidência

- Sobreposição de uma copia com afastamento
- Contagem dos caracteres que se repetem

D	I	P (%)	D	I	P (%)	D	I	P (%)	D	I	P (%)	D	I	P (%)
1	6	3.2	31	9	5.7	61	1	0.8	91	4	4.1	121	4	5.9
2	6	3.2	32	7	4.5	62	5	3.9	92	0	0.0	122	3	4.5
3	5	2.7	33	6	3.8	63	6	4.8	93	3	3.1	123	0	0.0
4	7	3.8	34	5	3.2	64	6	4.8	94	2	2.1	124	3	4.6
5	15	8.2	35	17	11.0	65	11	8.9	95	3	3.2	125	7	10.9
6	3	1.6	36	5	3.3	66	7	5.7	96	2	2.2	126	1	1.6
7	6	3.3	37	4	2.6	67	6	4.9	97	2	2.2	127	1	1.6
8	5	2.8	38	4	2.6	68	6	5.0	98	2	2.2	128	2	3.3
9	10	5.6	39	7	4.7	69	5	4.2	99	4	4.4	129	2	3.3
10	6	3.4	40	14	9.4	70	14	11.8	100	2	2.2	130	6	10.2
11	8	4.5	41	5	3.4	71	5	4.2	101	0	0.0	131	1	1.7
12	6	3.4	42	6	4.1	72	6	5.1	102	6	6.9	132	4	7.0
13	6	3.4	43	5	3.4	73	7	6.0	103	2	2.3	133	2	3.6
14	7	4.0	44	6	4.1	74	7	6.1	104	6	7.1	134	1	1.8
15	11	6.3	45	5	3.5	75	4	3.5	105	10	11.9	135	4	7.4
16	10	5.8	46	3	2.1	76	3	2.7	106	4	4.8	136	3	5.7
17	6	3.5	47	7	4.9	77	1	0.9	107	3	3.7	137	0	0.0
18	2	1.2	48	2	1.4	78	9	8.1	108	3	3.7	138	2	3.9
19	8	4.7	49	10	7.1	79	8	7.3	109	2	2.5	139	4	8.0
20	23	13.6	50	10	7.2	80	7	6.4	110	9	11.4	140	2	4.1
21	4	2.4	51	10	7.2	81	5	4.6	111	2	2.6	141	3	6.2
22	3	1.8	52	4	2.9	82	6	5.6	112	4	5.2	142	1	2.1
23	7	4.2	53	3	2.2	83	3	2.8	113	3	3.9	143	3	6.5
24	9	5.5	54	6	4.4	84	2	1.9	114	5	6.7	144	4	8.9
25	12	7.3	55	16	11.9	85	8	7.7	115	8	10.8	145	7	15.9
26	6	3.7	56	3	2.3	86	6	5.8	116	4	5.5	146	2	4.7
27	6	3.7	57	2	1.5	87	4	3.9	117	3	4.2	147	1	2.4
28	6	3.7	58	2	1.5	88	2	2.0	118	2	2.8	148	0	0.0
29	7	4.4	59	5	3.8	89	5	5.0	119	3	4.3	149	0	0.0
30	9	5.7	60	7	5.4	90	9	9.1	120	3	4.3	150	1	2.6

Img 6.8– Exemplo de indice de coincidencia usando todo o poema do exemplo

• Máquinas de Rotores

- Concretizam cifras poli-alfabéticas complexas
- Cada rotor efetua uma permutação do alfabeto
- A posição do rotor concertiza um alfabeto de substituição
- A rotação do rotor concretiza uma cifra poli-alfabetica
- Acumulando varios rotores em sequencia e rodando-os de forma diferenciada

consegue-se uma cifra poli-alfabetica complexa

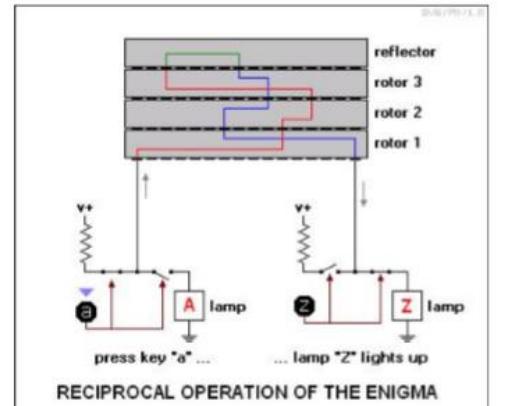
- **A chave da cifra é:**

- O conjunto de rotores usado
- A ordem relativa de cada rotor
- A posição de avanço do rotor seguinte
- A posição original dos rotores

- Rotores simétricos permitem decifrar usando cifras-duplas

- **Operação recíproca com um refletor**

- O operador emissor carrega em "A" (o texto em claro) e obtém "Z" como criptograma, o qual é transmitido
- O operador receptor carrega em "Z" (o criptograma) e obtém "A" como texto em claro
- Uma letra nunca pode ser cifrada para si própria!



Img 6.9– Operação oposta com um refletor

Máquina de rotores usada pelos Alemães na 2^a GG

Originalmente apresentada em 1919

- Enigma I, com 3 rotores

Foram usadas diversas variantes

- Com diferentes números de rotores
- Com cablagem para permutar alfabetos

Seleções de chaves distribuídas em livros de códigos



Img 6.10– Inigma – Maquina de rotores usada na WWII

VII. Criptografia – Aproximações Teóricas

Espaço de Texto – M

- Número de combinações de textos diferentes

Espaço de Criptograma – C

- Número de combinações de criptograma diferentes

Espaço Das Chaves – K

- Número de chaves diferentes para um algoritmo de cifra

Cifra Perfeita

- Dada combinação \mathbf{C}_j pertencente ao espaço do criptograma \mathbf{C}
- $H(M | C)$
 - Entropia condicional de M dado C
- $H(M)$
 - Entropia de M
- $H(M | C) = H(M)$

Cifra de Vernam – One Time Pad

VIII. Criptografia – Aproximações Práticas

Teóricamente Seguras vs Seguras na Prática

- Uso teórico é diferente da exploração prática
- Práticas incorretas podem comprometer boas cifras
- Um exemplo de uma prática incorreta seria a reutilização de One-Time Pads

Cifras Seguras na Prática

- A segurança é assegurada pela dificuldade computacional de realizar criptanálise
 - Usando Brute-Force
- Têm uma segurança baseada em limites razoáveis
 - Custos de uma solução técnica de criptanalise
 - Infraestrutura reservada para a criptanalise
 - Tempo útil de criptanalise

IX. Criptografia – Aproximações Práticas: 5 Critérios de Shannon

1. Quantidade de secretismo oferecida

- a. E.g O comprimento da chave

2. Complexidade na escolha das chaves

- a.** E.g Geração da chave, deteção de chaves fracas

3. Simplicidade da realização

4. Propagação de erros

- a.** Relevante em ambientes onde surgem erros (e.g canais de comunicação ruidosos)

5. Dimensão do criptograma

- a.** Relativamente aos respetivos textos originais

X. Criptografia – Aproximações

Práticas: Confusão vs Difusão

Confusão

- Complexidade na relação entre o texto, a chave e o criptograma
- Os bits resultantes (criptograma) devem depender dos bits de entrada (texto e chave) de forma complexa

Difusão

- Alteração de grandes porções do criptograma em função de uma pequena alteração do texto
- Se um bit de texto se alterar então o criptograma deverá mudar substancialmente de forma imprevisível e pseudoaleatória

- Efeito Avalanche

XI. Criptografia – Aproximações Práticas: Assumir sempre o pior caso

O criptoanalista conhece o algoritmo

- A segurança deve então estar na chave

O criptoanalista possui grande número de criptogramas gerados com um algoritmo e chave

- Os criptogramas não são secretos

O criptoanalista conhece parte dos textos originais

- É normal haver noção do texto original
- Ataques com texto conhecido ou escolhido

XII. Robustez Criptográfica

A robustez dos algoritmos e a sua resistencia a ataques

- Ninguem consegue avaliar a robustez de forma precisa
- Podem especular ou demonstrar usando outras suposições

- São robustos até que alguém os quebre
- Existem orientações públicas sobre o que deve/não deve ser usado
- Devemos antecipar problemas futuros

Algoritmos públicos, sem ataques conhecidos, supostamente são mais robustos

- Mais investigadores à procura de fraquezas

Algoritmos com chaves maiores são tendencialmente mais robustos

- Mas frequentemente também são mais lentos

1997: NIST lançou desafio para o próximo Advanced Encryption Protocol

- de conhecimento e utilização públicos, simétrico, chaves de 128, 192 e 256 bits

1998: 15 candidatos apresentados por investigadores

- CAST-256, Crypton, DEAL, DFC, Frog, HPC, LOKI97, Magenta, MARS, RC6, Rijndael, Safer+, Serpent, Twofish
- Comunidade tentou encontrar problemas nos candidatos

1999: 5 propostas demonstraram ser seguras

- MARS, RC6, Rijndael, Twofish
- Novamente a comunidade tentou encontrar problemas e avaliar a performance

2001: Rijndael selecionado como o vencedor

- Versões reduzidas do MARS foram quebradas, RC6 e Twofish são seguros

2002: Publicado como FIPS PUB 197 e largamente utilizado

Img 12.1– Exemplo da robustez do AES

XIII. Cifras Contínuas - Stream

Mistura de uma chave contínua (keystream) com um texto ou criptograma

Podemos ter dois tipos de chaves:

- **Chave contínua Aleatória**
 - E.g Cifra de Vernam com One time pad
- **Chave contínua Pseudoaleatória**
 - Produzida por gerador

Função de mistura invertível

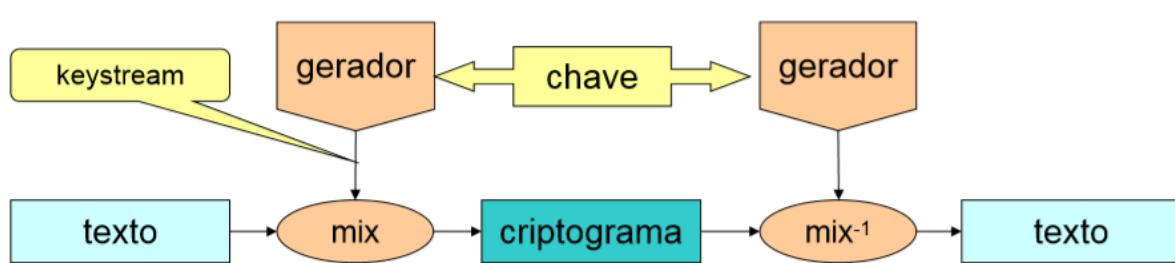
- E.g XOR bit a bit

$$C = P \oplus ks \quad P = C \oplus ks$$

Img 13.1– Criptograma = Operação XOR bit a bit do texto com o keystream ; Texto igual a operação XOR bit a bit do criptograma com o keystream (por isso mesmo se diz q a função de mistura é invertível)

Cifra Poli-Alfabética

- Cada símbolo da chave contínua define um alfabeto



Img 13.2– Funcionamento de uma Cifra continua

○ Keystream pode ser infinito, mas possui um período (que depende do gerador)

Questões/Medidas práticas de segurança:

- Cada keystream só pode ser usado **uma vez**
 - Caso contrario a soma dos criptogramas forneceria a soma dos textos

$$C_1 = P_1 \oplus K_s, \quad C_2 = P_2 \oplus K_s \rightarrow C_1 \oplus C_2 = P_1 \oplus P_2$$

Img 13.3– Porque que um KS só se deve usar uma vez

- Dimensão do texto **tem de ser menor que o período**
 - Exposição da keystream é total com textos escolhidos/conhecidos
 - Período permite aos analistas conhecerem partes do text
- Controlo de integridade é **mandatório**
 - **Não existe difusão, apenas confusão**
 - Criptogramas podem ser manipulados livremente

XIV. Cifras Modernos - Tipos

Cifras modernas podem ser caracterizadas:

- **Quanto à Operação**
 - Por Blocos (mono-alfabéticas)

- Continuas (poli-alfabeticas)

- **Quanto ao tipo de chave**
 - Simétricas
 - Chave Secreta ou Segredo partilhado
 - Potencialmente sujeitas a caução
 - Assimetricas
 - Chave pública

	Cifras Por Blocos	Cifras Contínuas
Cifras Simétricas		
Cifras Assimétricas		

Img 14.1–Combinatória do tipo de cifras modernas

XV. Cifras Simétricas

Chave Secreta – Partilhada por 2 ou mais interlocutores

Permitem

- Confidencialidade para todos os condecedores da chave
- Autenticação de mensagens (Quando se usam cifra por blocos)

Vantagens

- Desempenho (normalmente muito eficientes)

Desvantagens

- N interlocutores
- 2 a 2 secretamente $\rightarrow N \times (N-1) / 2$ Chaves

Problemas

- Distribuição de chaves

XV.I Cifras Simétricas por Blocos

Aproximações usadas:

- Blocos de grande dimensão
 - 64, 128, 256, ...

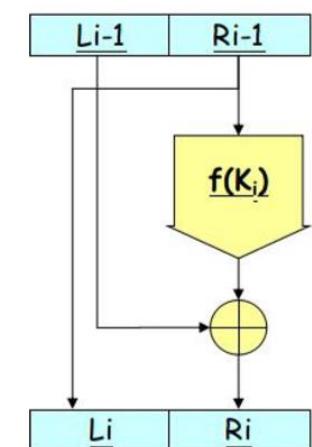
Difusão, Confusão

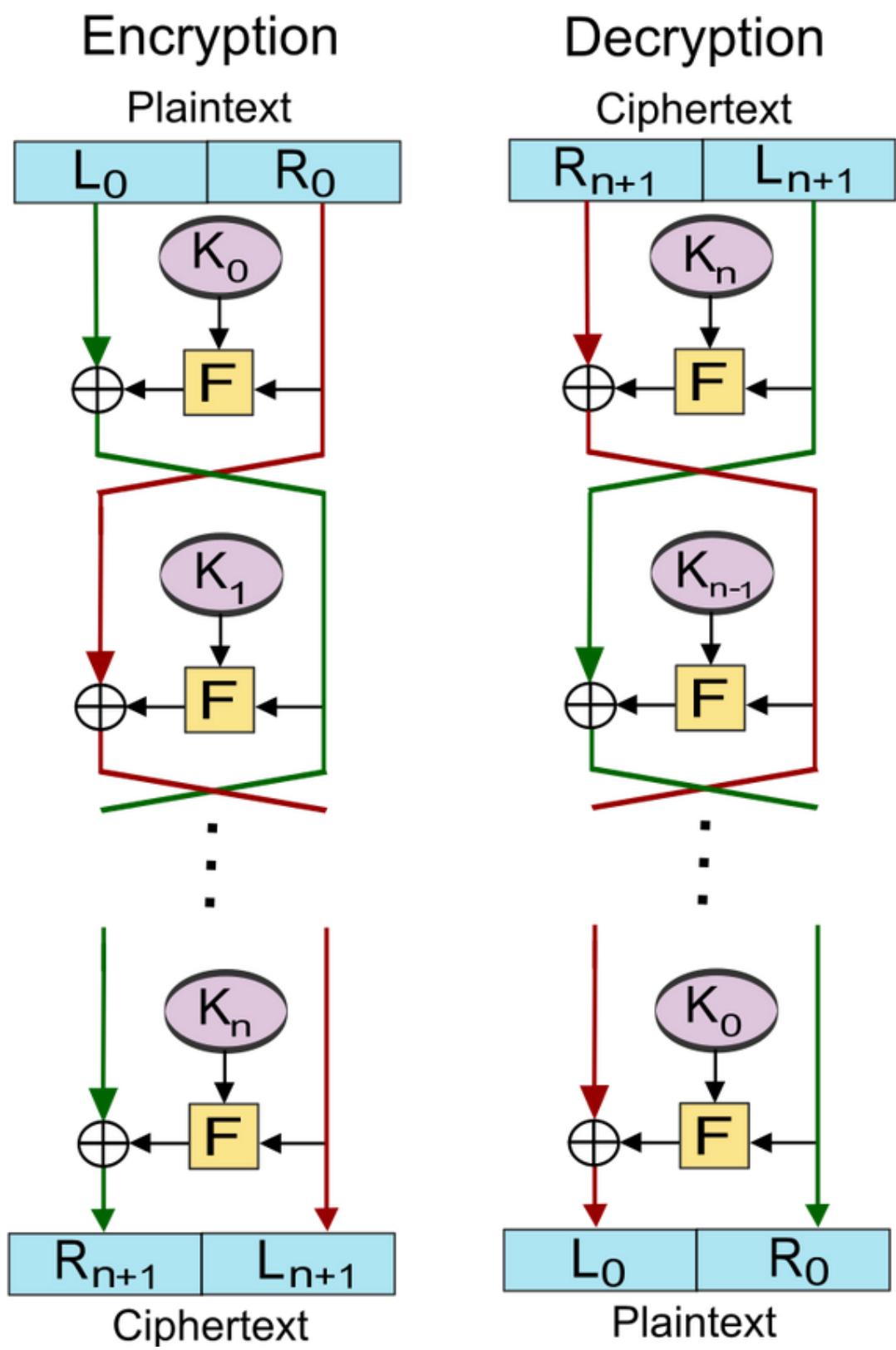
- Permutação, substituição, expansão, compressão
- **Redes de Feistel** com múltiplas iterações
- Ou redes de substituição-permutação

Exemplos de algoritmos incluem **DES** ($D=64, K=56$), **IDEA** ($D=64, K=128$), **AES** ($D=128, K=128, 192, 256$)

Redes de Feistel

- Seja f a função de rotação e K_0, K_1, \dots, K_n as subchaves para as rodadas $0, 1, \dots, n$, então a operação básica é a seguinte:
 - Divide-se o bloco de texto em duas partes com o mesmo tamanho (L_0, R_0)
 - Para cada rotação $i = 0, 1, \dots, n$ então:
 - $L_{i+1} = R_i$
 - $R_{i+1} = L_i \text{ XOR } f(R_i, K_i)$
 - Alternativamente temos:
 - $L_i = R_{i-1}$
 - $R_i = L_{i-1} \text{ XOR } f(R_{i-1}, K_i)$
 - O criptograma é então (R_{n+1}, L_{n+1})
- No que toca a **decriptação** do criptograma temos:
 - Para $i = n, n-1, \dots, 0$ $L_i = R_{i-1} \quad R_i = L_{i-1} \text{ XOR } f(R_{i-1} \oplus , K_i)$
 - $R_i = L_{i+1}$
 - $L_i = R_{i+1} \text{ XOR } f(L_{i+1}, K_i)$





Img 15.1—Encriptação e decriptação usando redes de Feistel

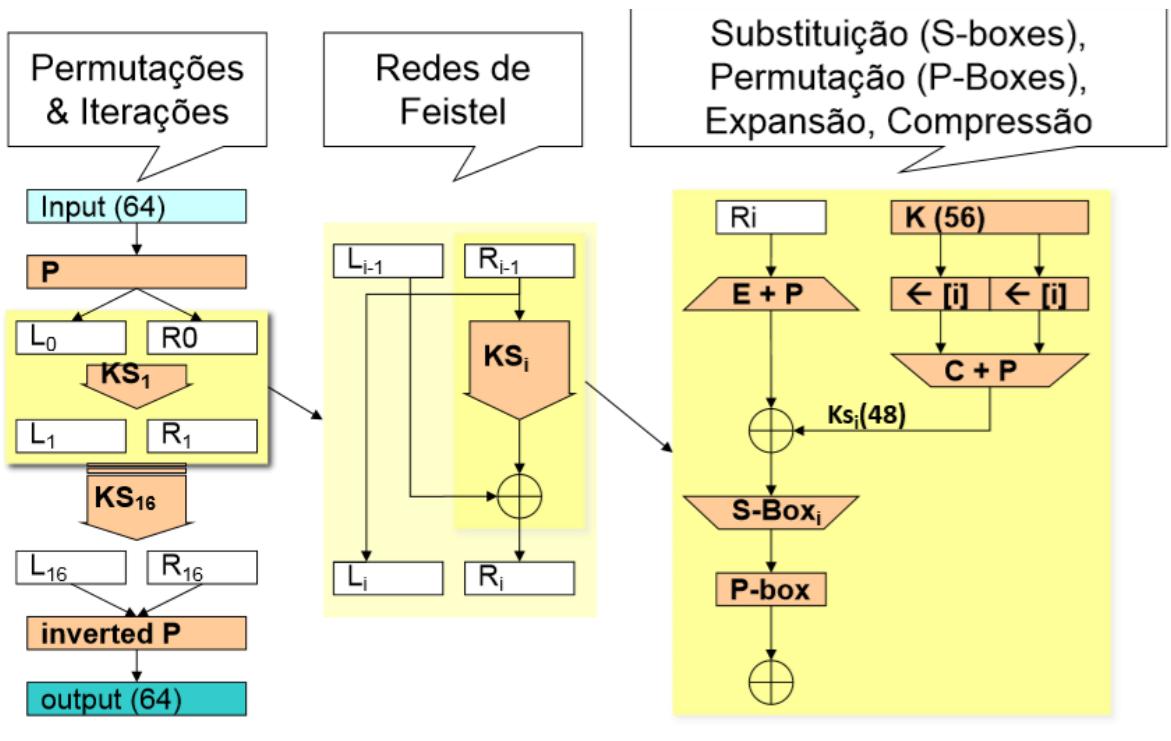
- Comparativamente a uma Rede Substituição-Permutação, uma vantagem das Redes de Feistel é o facto da função de rotação, **f**, **NÃO TEM INVERSA**

Redes de Substituição-Permutação

- **S-Box**
 - Substituição baseada num bit de entrada
 - Troca de bits da saída
 - Substituição não é direta (i.e 1 para 1)
 - **Ideal:** A alteração de um bit provoca a alteração de todos os bits
 - **Prática:** A alteração de um bit provoca a alteração de pelo menos metade dos bits
- **P-Box**
 - Permutação da posição de bits entre entrada e saída
 - **Ideal:** Permuta a posição de todos os bits

A operação de ambas depende da chave

DES: Data Encryption Standard



Escolha de chaves

- Chaves fracas, semi-fracas e quasi-fracas
- Fáceis de identificar

Ataques conhecidos

- Pesquisa exaustiva

Dimensão das chaves: 56 bits são atualmente insuficientes

- A pesquisa exaustiva é técnica e economicamente viável

Solução: cifra múltipla

- Cifra dupla não é completamente segura (teoricamente ...)
- Cifra tripla: 3DES (Triple-DES)
 - Com duas ou três chaves
 - Chaves equivalentes de 112 ou 168 bits

Img 15.2 – DES: Algoritmos e Robustez

XV.II Cifras Simétricas Contínuas

Aproximações usadas:

- Desenho de geradores pseudo-aleatórios seguros
 - Baseados em LFSRs
 - Baseados em Cifras por blocos (ironic)
- Outras aproximações – Famílias de funções, ...
- Normalmente sem sincronização
- Normalmente sem possibilidade de acesso aleatório rápido

Exemplos de algoritmos incluem **A5/1, RC4, ChaCha20, Salsa20, ...**

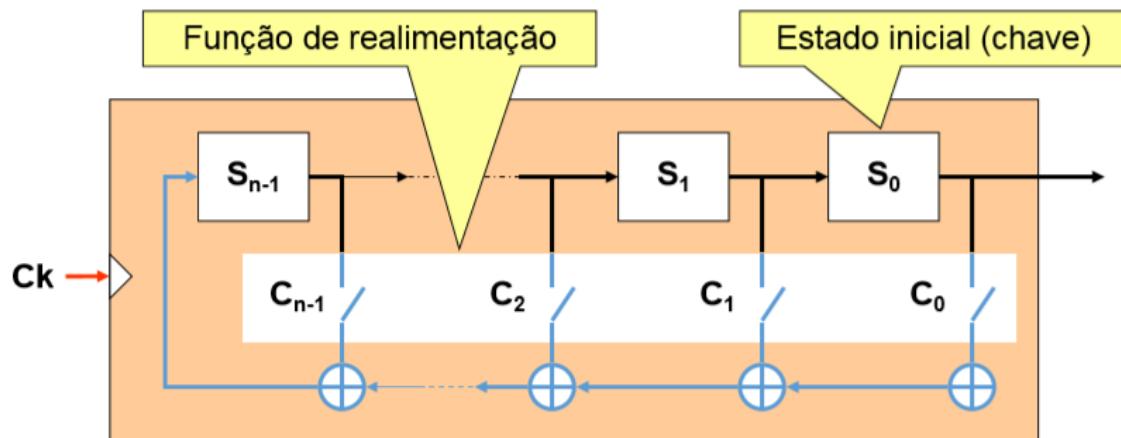
Linear Feedback Shift Register - LSFR

$2^n - 1$ sequencias não nulas

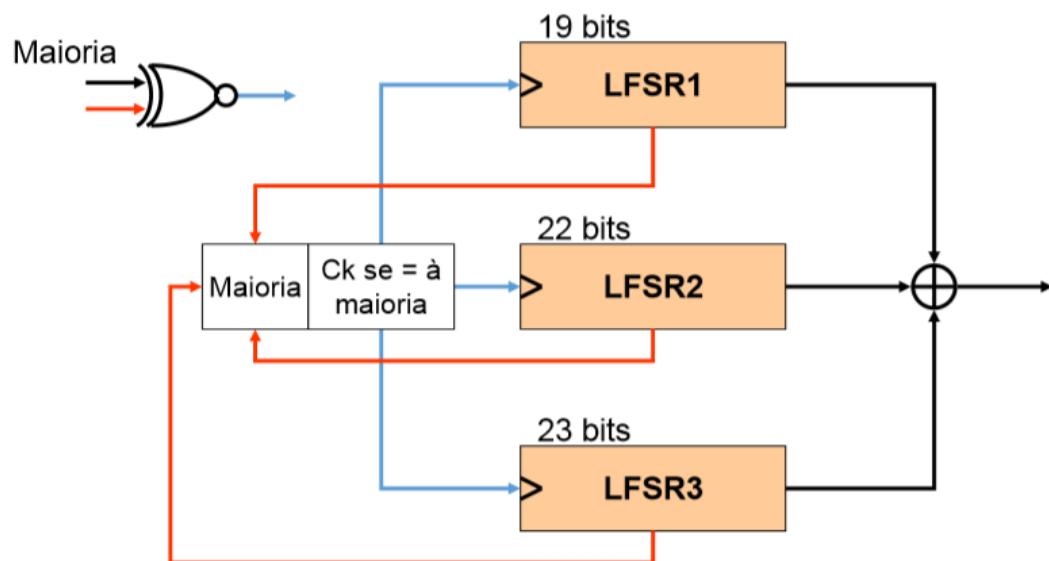
- Se uma delas possuir um período $2^n - 1$ todas o têm

Funções de realimentação primitivas

- Todas as sequências não nulas tem comprimento $2^n - 1$



Img 15.3 – Funcionamento do LFSR



Img 15.4 – Geradores com composições de LFSR – A5/1

XVI. Cifras Assimétricas por Blocos

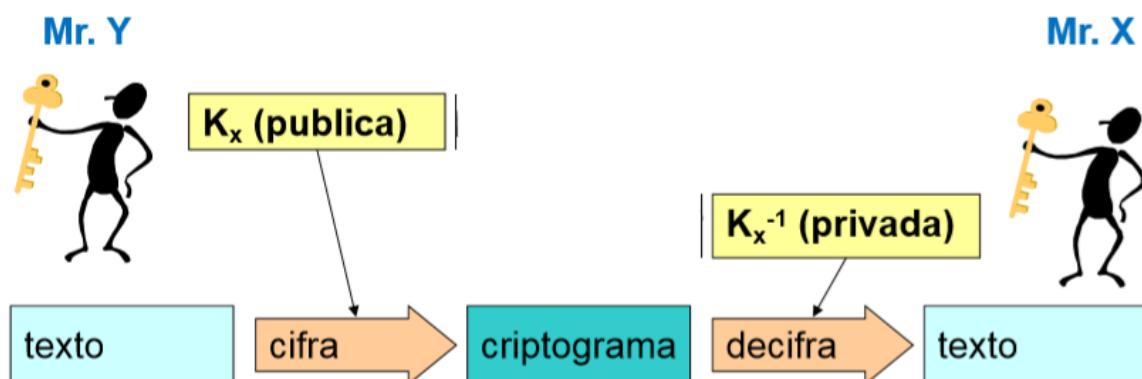
Aproximações:

- Complexidade Matemática
- Calculo de logaritmos discretos
- Fatorização de grandes numeros
- Problema da mochila (Knapsack)

Algoritmos mais usados incluem RSA, ElGamal e Curvas Elípticas

Outra técnica com Chave Pública: **Diffie-Hellman** (negociação de chaves)

XVI.I Confidencialidade e Autenticidade com Cifras Assimétricas



Img 16.1 – Chaves Assimétricas

Em termos de Confidencialidade:

Temos Menos Chaves:

- $C = E(K, P)$
 - Criptograma = Encriptação do texto P com a chave K
- $P = D(K^{-1}, C)$
 - Texto P = Decriptação do texto do criptograma C com o inverso da Chave usada na encriptação

Não há autenticação de origem:

- **X** não sabe quem produziu o criptograma
- Se **Kx** for pública, qualquer pessoa pode te-lo produzido

Em termos de **Autenticidade**:

O criptograma não pode ser alterado:

- $C = E(K^{-1}, P)$
- $P = D(K, C)$
- Só **X** conhece a chave K^{-1} com que foi gerado o criptograma

Não há confidencialidade:

- Quem conhecer **Kx** consegue decifrar o criptograma

- Se K_x for pública, qualquer um o pode decifrar

XVI.II Exemplo de algoritmo – RSA

Complexidade matemática

- Dificuldade de Fatorização de grandes números
- Dificuldade de cálculo de logaritmos discretos

Operações e chaves

- $K = (e, n) \quad K^{-1} = (d, n)$
- $C = P^e \text{ mod } n \quad P = C^d \text{ mod } n$
- $C = P^d \text{ mod } n \quad P = C^e \text{ mod } n$

Escolha dos valores das chaves

- n de grande dimensão (centenas ou milhares de bits)
- $n = p \times q$ p e q primos, de grande dimensão
- Escolher e coprimo de $(p-1) \times (q-1)$
- Procurar um d tal que $e \times d \equiv 1 \pmod{(p-1)(q-1)}$
- Não se consegue deduzir d a partir de e ou de n

Img 16.2 – Funcionamento e considerações do RSA

$$p = 5 \quad q = 11 \quad (\text{pequenos números primos})$$

- $n = p \times q = 55$
- $(p-1)(q-1) = 40$

$$e = 3$$

- Coprimo de 40

$$d = 27$$

- $e \times d \equiv 1 \pmod{40}$

$$P = 26 \quad (\text{note que } P, C \in [0, n-1])$$

- $C = P^e \text{ mod } n = 26^3 \text{ mod } 55 = 31$
- $P = C^d \text{ mod } n = 31^{27} \text{ mod } 55 = 26$

Img 16.3 – Exemplo do RSA

XVI.III Exemplo de algoritmo – ElGamal

Semelhante ao RSA

- Baseado apenas na dificuldade de cálculo de logaritmos discretos

Uma variante é muito usada em assinaturas digitais

- DSA (*Digital Signature Algorithm*)
- US *Digital Signature Standard* (DSS)

Operações e chaves (para assinaturas)

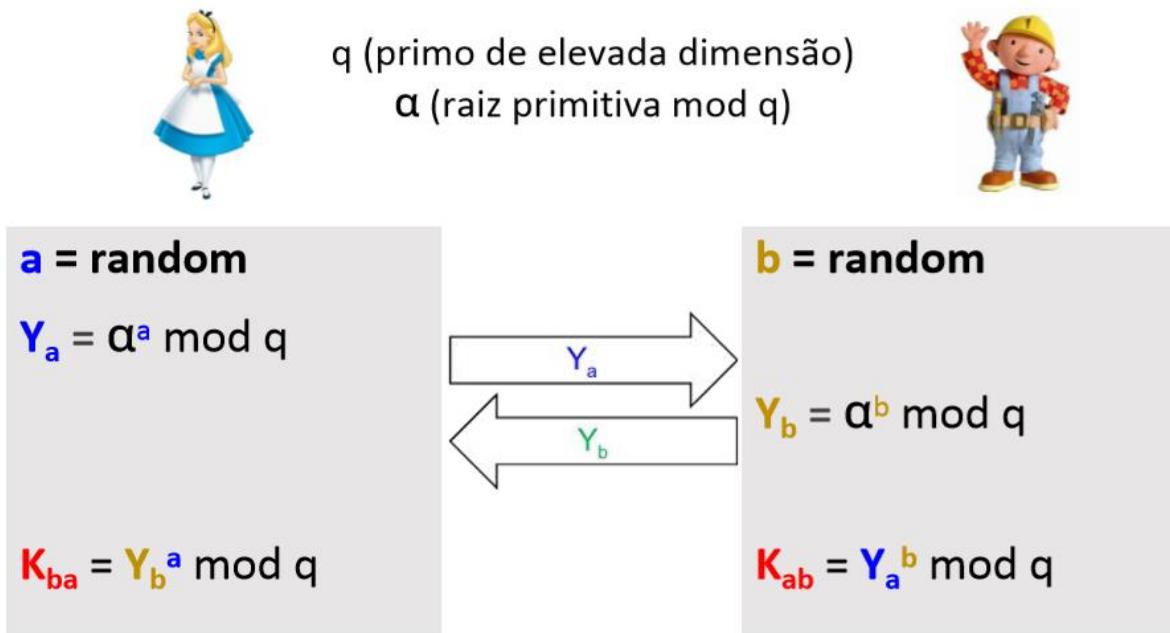
- $\beta = \alpha^x \text{ mod } p$ $K = (\beta, \alpha, p)$ $K^{-1} = (x, \alpha, p)$
- k aleatório, $k \times k^{-1} \equiv 1 \text{ mod } (p-1)$
- Assinatura de M : (γ, δ) $\gamma = \alpha^k \text{ mod } p$ $\delta = k^{-1} (M - x\gamma) \text{ mod } (p-1)$
- Validação da assinatura de M : $\beta^\gamma \gamma^\delta \equiv \alpha^M \text{ (mod } p)$

Problema

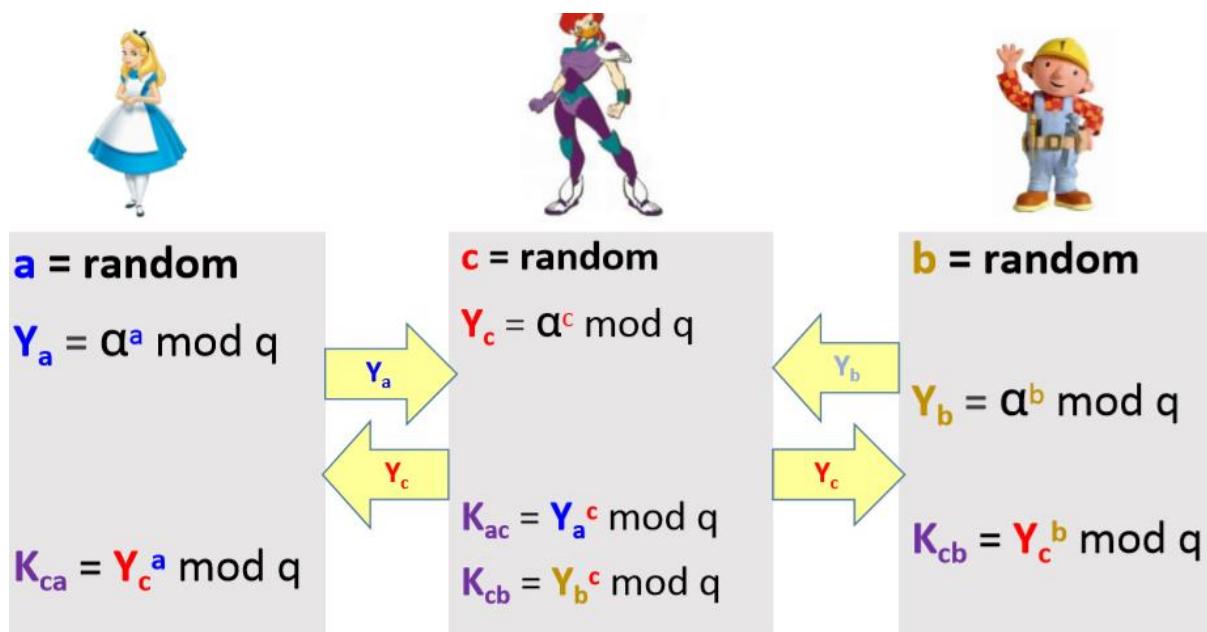
- O valor de k precisa de ser secreto
- O seu conhecimento revela x !

Img 16.4 – Funcionamento e considerações do ElGamal

XVI.IV Diffie-Hellman



Img 16.5 – Diffie Hellman – Técnica de chave pública



Img 16.6 – Diffie Hellman – Ataque por MitM (Meet in the Middle)

XVII. Randomização de Cifras com Chave Pública

O resultado de uma cifra com **chave pública não deverá ser determinístico** (i.e previsível)

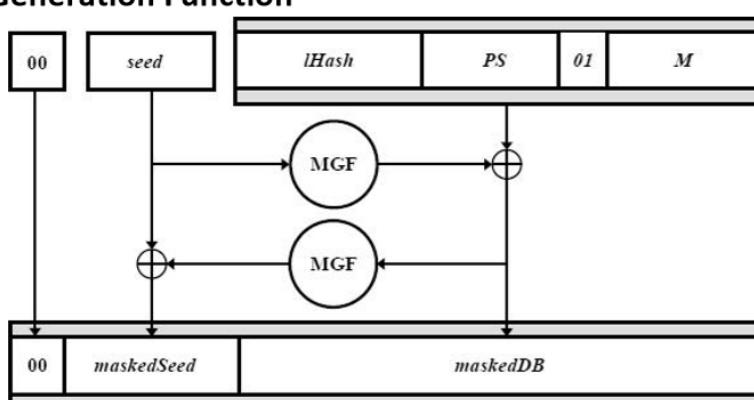
N Cifras do mesmo valor, **com a mesma chave**, devem produzir **N resultados diferentes**

O objetivo é **impedir a descoberta de valores cifrados por tentativa e erro**

Técnicas:

- Concatenação do valor a cifrar com dois valores
 - Um **Fixo**
 - Para controlo de erros
 - Um **Aleatório**
 - Para randomização
- IHash: Digest sobre Label
• seed: Random
• PS: zeros
• M: Texto
• MGF: Mask Generation Function

Img 17.1 –
Randomização
de Cifras com
chave publica
com OAP



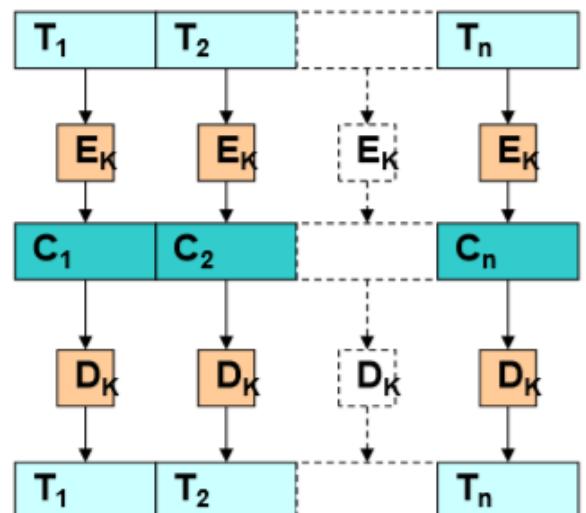
XVIII. Utilização de Cifras por Blocos – Modos

Metodologia inicialmente proposta para o DES

Modos podem ser usados com outras cifras (pelo menos em teoria)

Os principais são:

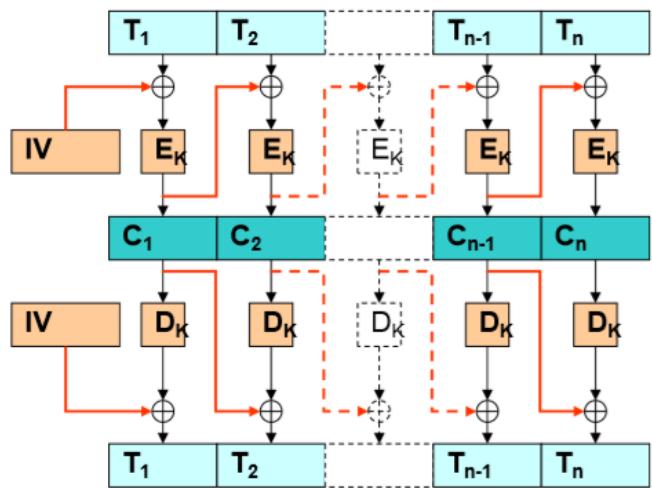
- **ECB – Electronic Code Block**
 - Cifra direta de cada bloco T_i
 - $C_i = E_k(T_i)$
 - Decifra direta de cada bloco
 - $T_i = D_k(C_i)$
 - Blocos **independentes**
 - Sem Feedback
 - Problema:
 - Se $T_1 = T_2$ então $C_1 = C_2$



• CBC – Cipher Block Chaining

- Cifra de cada bloco T_i com feedback de C_{i-1}
 - $C_i = E_K(T_i \oplus C_{i-1})$
- Decifra de cada bloco C_i com feedback de C_{i-1}
 - $T_i = D_K(C_i) \oplus C_{i-1}$

- Bloco inicial usa IV
 - Initialization Vector
 - Valor aleatório
 - Pode estar em claro



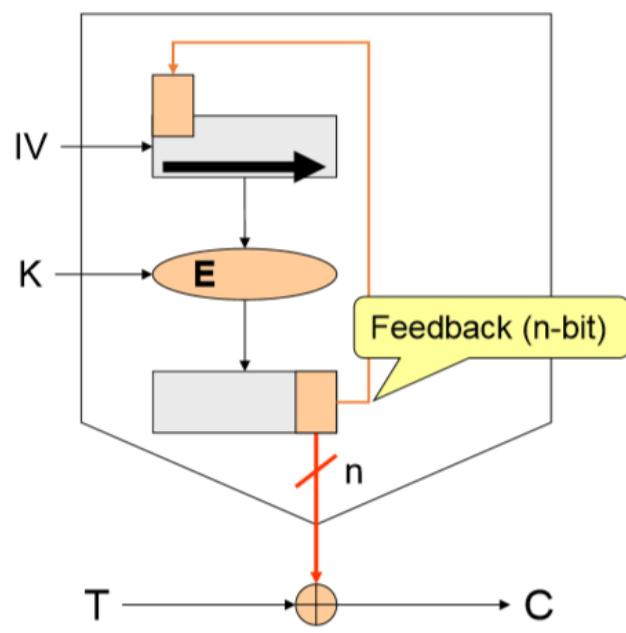
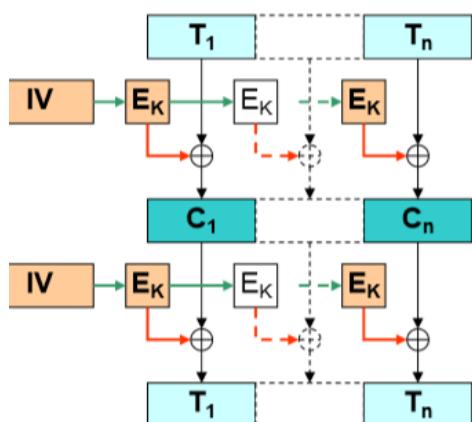
• OFB – Output Feedback Mode

$$C_i = T_i \oplus E_K(S_i)$$

$$T_i = C_i \oplus E_K(S_i)$$

$$S_i = f(S_{i-1}, E_K(S_{i-1}))$$

$$S_0 = IV$$



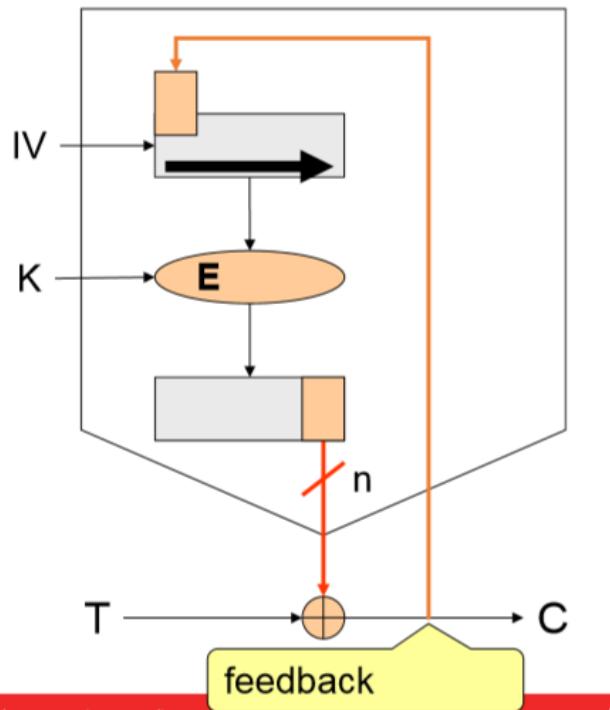
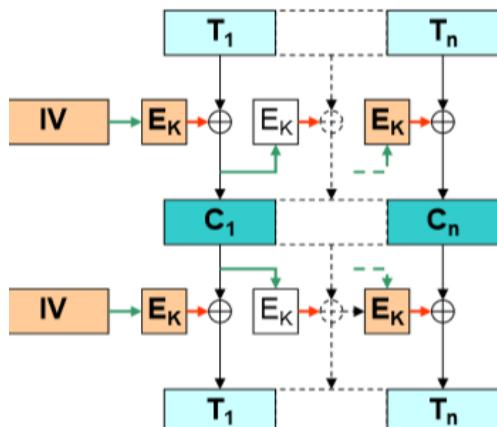
- CFB – Cipher Feedback Mode

$$C_i = T_i \oplus E_K(S_i)$$

$$T_i = C_i \oplus E_K(S_i)$$

$$S_i = f(S_{i-1}, C_i)$$

$$S_0 = IV$$



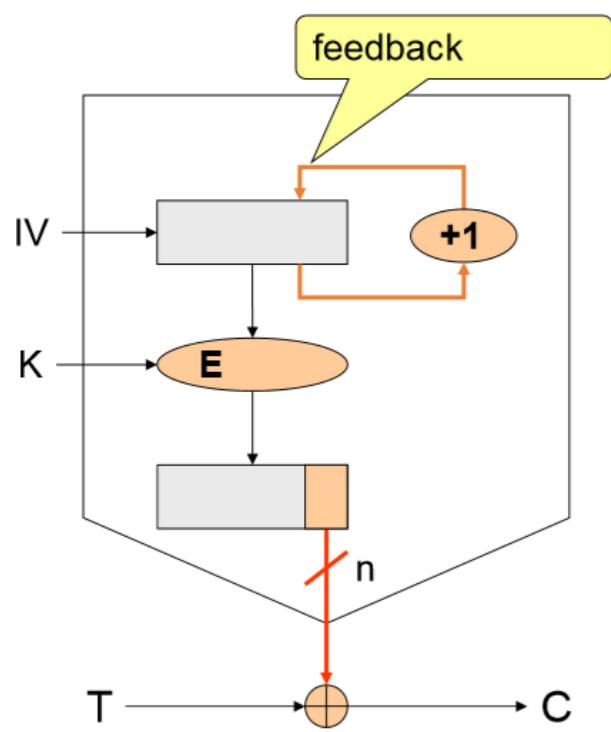
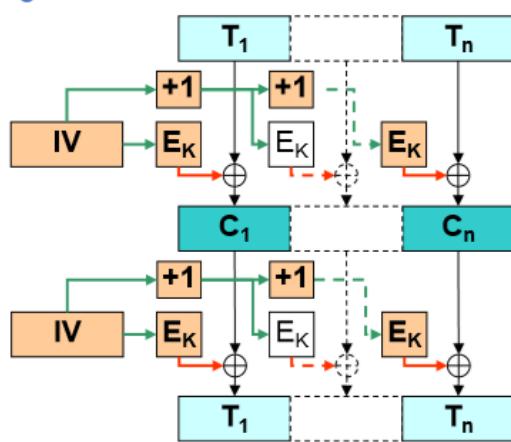
- CTR – Counter Mode

$$C_i = T_i \oplus E_K(S_i)$$

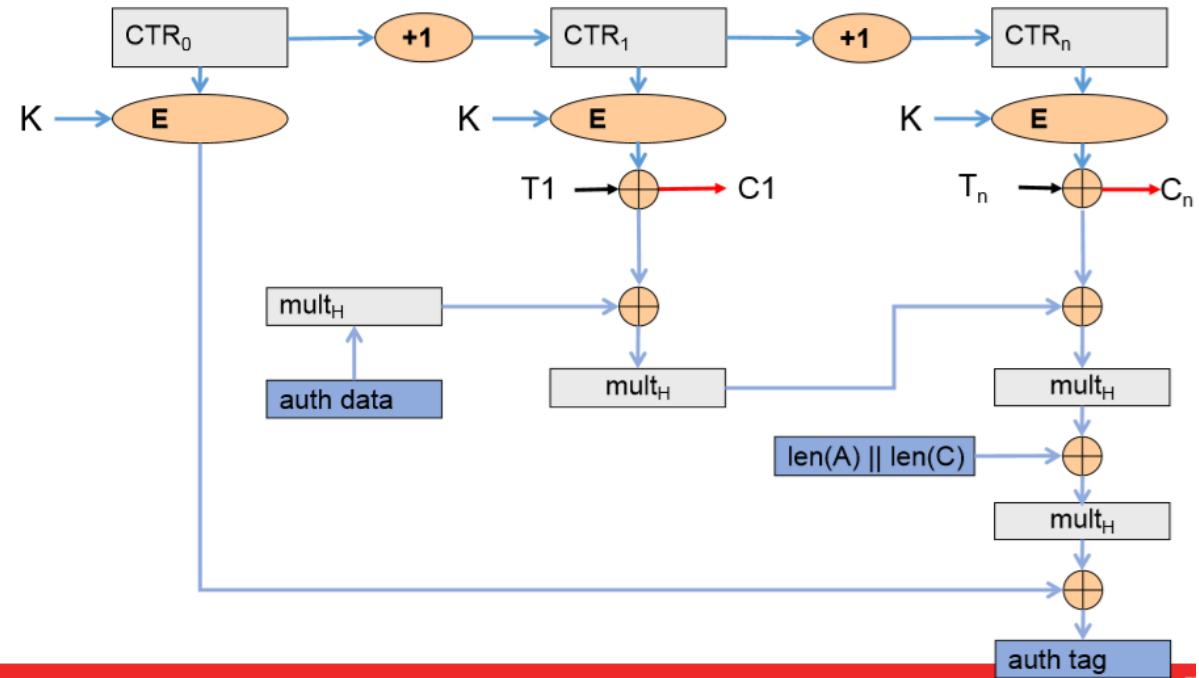
$$T_i = C_i \oplus E_K(S_i)$$

$$S_i = S_{i-1} + 1$$

$$S_0 = IV$$



• GCM – Galois with Counter Mode



XIX. Comparação dos Modos

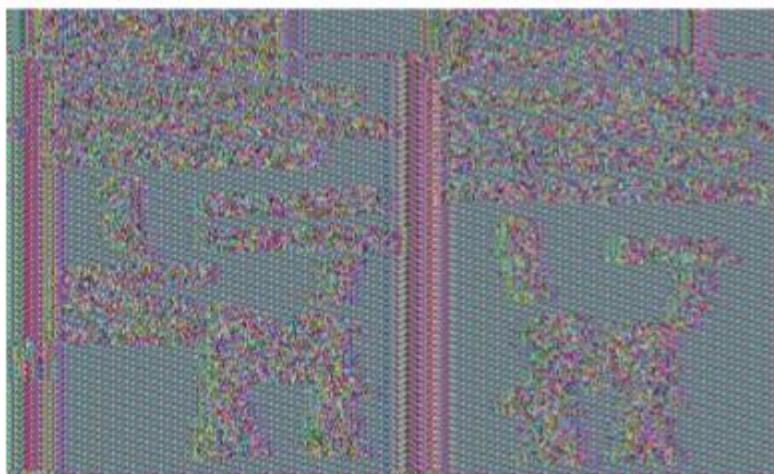
	Bloco		Contínua			
	ECB	CBC	OFB	CFB	CTR	GCM
Ocultação de padrões no texto		✓	✓	✓	✓	✓
Confusão na entrada da cifra		✓		✓	Contador Secreto	Contador Secreto
Mesma chave para mensagens diferentes	✓	✓	Outro IV	Outro IV	Outro IV	Outro IV
Dificuldade de alteração	✓	✓ (...)				✓
Pré-processamento			✓		✓	✓
Paralelização	✓	decrifra	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

Img 19.1 – Comparação entre os modos

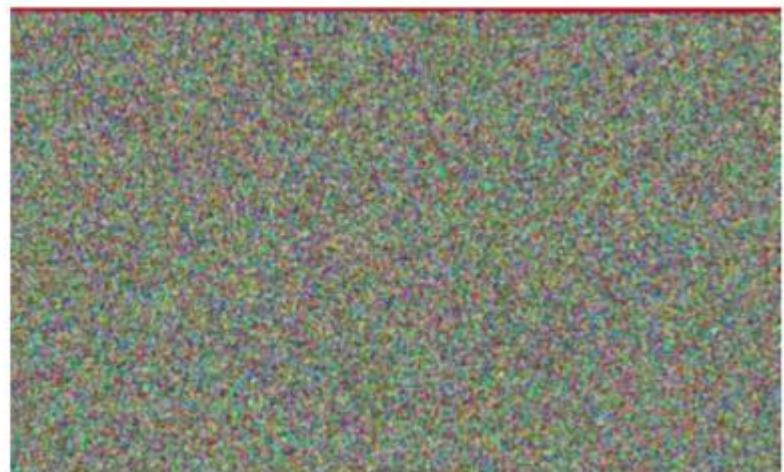
XX. Modos: ECB vs CBC – Propagação de Padrões

Analizando os padrões produzidos por ECB contra os produzidos pelo CBC é notável a diferença do quanto mais aleatoriedade o CBC possui

ECB



CBC



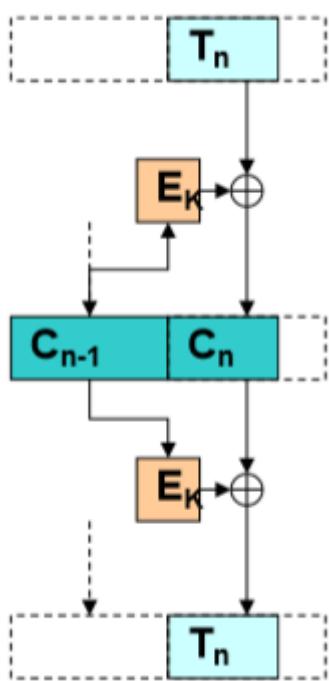
Img 20.1 – Comparação entre os padrões produzidos por ECB vs os produzidos por CBC

XXI. Modos: ECB/CBC – Problemas de Alinhamento

Os modos ECB e CBC necessitam de textos com dimensão múltipla da dimensão do bloco. Isto acontece devido ao facto da cifra ser aplicada por blocos de texto..

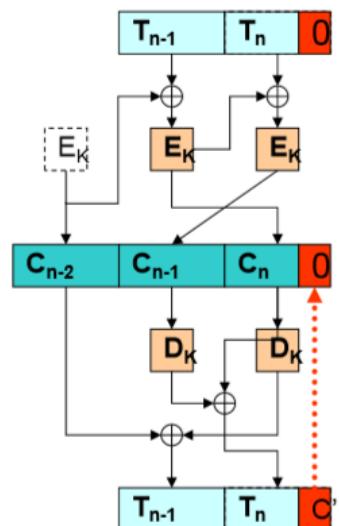
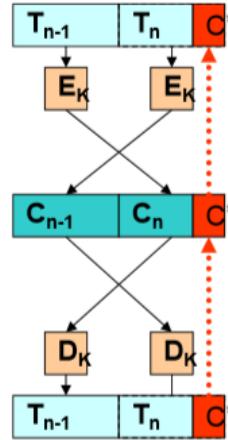
Blocos Incompletos (i.e o ultimo) precisam de ter um tratamento diferente, tanto na cifra como na decifra.

Podemos usar um processo semelhante a uma cifra contínua



Ciphertext Stealing

- Troca ordem de cifra/decifra dos dois últimos blocos
- a) Usa parte do criptograma do penúltimo para preencher último
- b) Usa excipiente fixo e cifra contínua antes de cifra por blocos



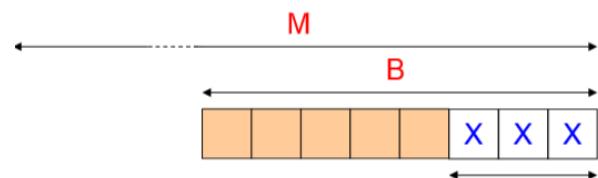
Img 21.1 – Cifragem/Decifragem especial para o ultimo bloco

O resultado da aplicação dos modos é um bloco, sendo que o criptograma pode chegar a ser maior do que o texto

Como alternativa à cifragem/decifragem especial para o ultimo bloco podemos usar **Excipiente (Padding)**

PKCS#7

- $X = B - (M \bmod B)$
 - M – Tamanho da mensagem
 - B – Tamanho dos blocos
- X bytes extra com valor X
- Se $M \bmod B = 0$, adiocinamos um bloco inteiro com o valor B



XXII. Modos: Reforços de Segurança

Cifra Múltipla

Cifra Dupla

- Violável por intromissão em 2^n+1 tentativas
 - Com 2 ou mais blocos de texto conhecido

- Usando 2^n blocos de memória
- Não é teoricamente muito mais segura :/

Cifra Tripla - EDE

- $C_i = E_{K1}(D_{K2}(E_{K3}(T_i)))$
 - C – Criptograma
 - EK1 – Encriptação usando chave 1
 - Dk2 – Encriptação usando chave 2
 - Ek3 – Encriptação usando chave 3
 - Ti – Bloco i
- $P_i = D_{K3}(E_{K2}(D_{K1}(C_i)))$
 - C – Criptograma do bloco i
 - Dk3 – Encritação usando chave 3
 - Ek2 – Encriptação usando chave 2
 - Dk1 – Decriptação usando chave 1
 - Pi – Texto do bloco i
- Normalmente usa-se $K1 = K3$
- Se $K1 = K2 = K3$ voltamos a ter uma cifra simples...no fucking shit sherlock

Ataque Meet in the Middle – Cifra Dupla

Cifra Dupla com chaves Ka e Kb

- $C = E_b(K_b, E_a(K_a, T))$
- $T = D_a(K_a, D_b(K_b, C))$

Logo: $\mathbf{Db(Kb, C) = Ea(Ka, T)}$

Se C e T forem conhecidos podem-se calcular

- Todos os valores $\mathbf{Db(Kb, C)}$ variando o \mathbf{Kb}
- Todos os valores $\mathbf{Ea(Ka, T)}$ variando \mathbf{Ka}

Chaves encontradas quando se verificar a igualdade

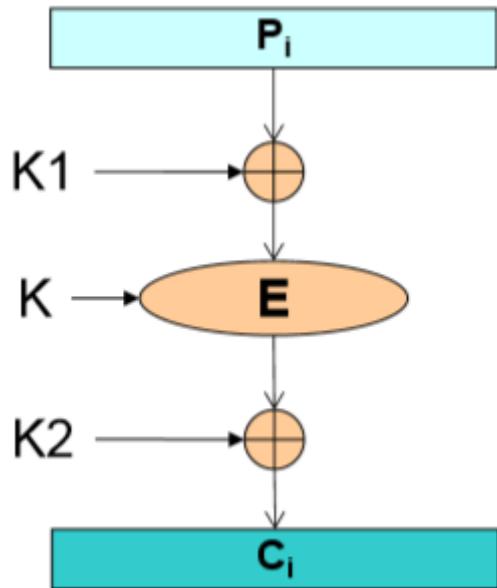
- Complexidade esperada: $2^{\text{len}(Ka)+\text{len}(Kb)}$
- Complexidade real: $2^{\text{len}(Ka)} + 2^{\text{len}(Kb)}$

Branqueamento / Whitening

Técnica simples e eficiente de introdução de confusão

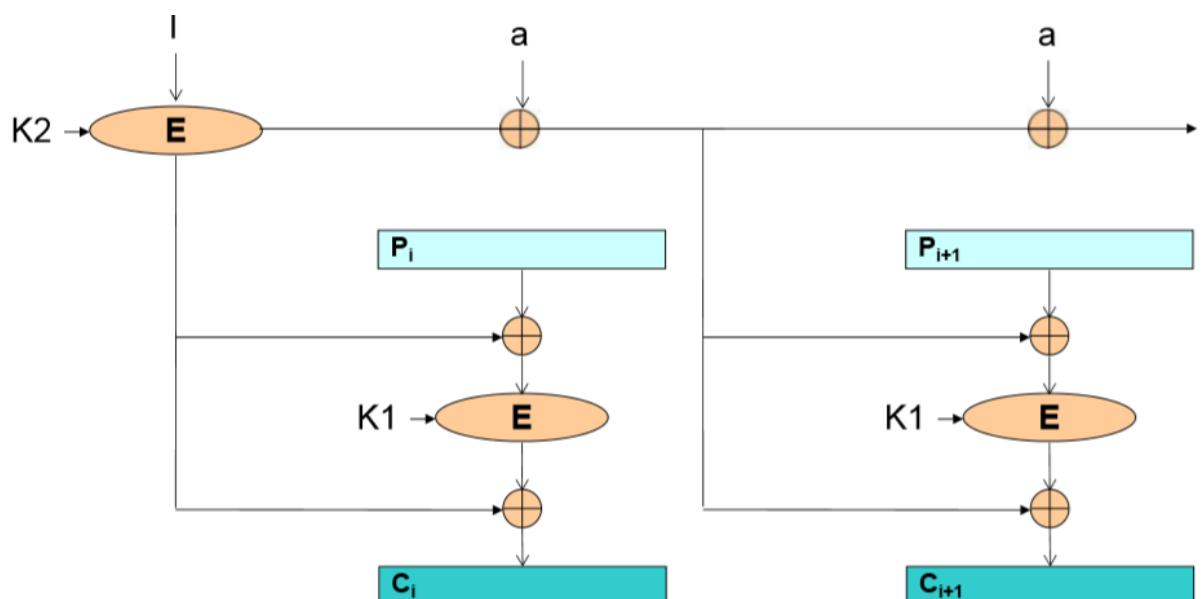
- $\mathbf{Ci = Ek(K1 XOR Ti) XOR K2}$
 - Ci – Criptograma do bloco i
 - Ek – Encriptação por chave K
 - K1 – Chave 1
 - Ti – Bloco i
 - K2 – Chave 2
- $\mathbf{Ti = K1 XOR Dk(K2 XOR Ci)}$
 - Ti – Texto do bloco i
 - K1 – Chave 1
 - Dk – Decriptação por chave K
 - K2 – Chave 2

- C_i – Criptograma do bloco i



XOR-Encrypt-XOR (XEX)

XTS = XEX + Ciphertext Stealing



XXIII. Cifra Híbrida – Aumento de Performance

Combinação de Cifra Assimétrica com Simétrica.

Vamos buscar o melhor de ambas:

- **Cifra Assimétrica – Uso de chaves públicas**
 - Mas Lenta
- **Cifra Simétrica – Rápida**
 - Mas com fraca troca de chaves

Aproximação:

1. Obter Kpublica do destinatário
2. Gerar Ks de forma aleatória
3. Calcular C1 = Esim(Ks, T)
4. Calcular C2 = Eassim(Kpub, Ks)
5. Enviar C1 + C2
 - a. C1 – Texto cifrado com chave simetrica
 - b. C2 – Chave simetrica cifrada com chave publica do destinario (podendo tambem conter um IV)

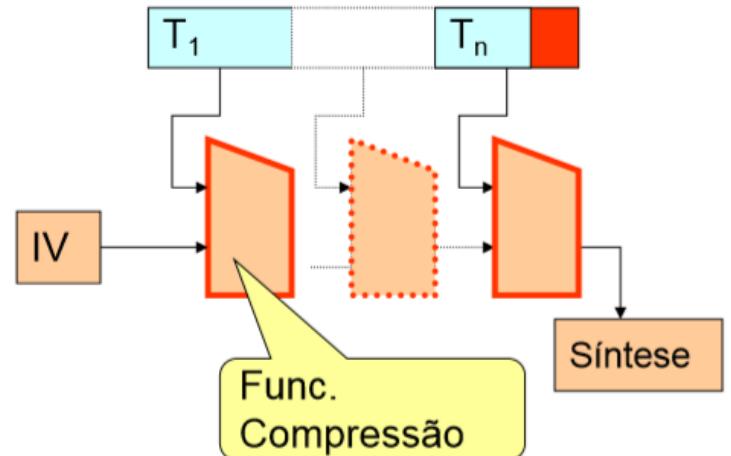
XXIV. Funções de Síntese (Digest)

Resultado de dimensão constante com entradas de dimensão variavel

Especie de impressão digital dos textos

Resultados muito diferentes para entradas similares

(Funções de dispersão criptograficas unidirecionais)



Propriedades:

- **Resistencia a descoberta de um texto**
 - Dada uma síntese é difícil encontrar um texto que o produza
- **Resistencia a descoberta de um 2º texto**
 - Dado um texto, é difícil encontrar um segundo texto com a mesma síntese
- **Resistencia à Colisão**
 - Difícil encontrar dois textos com a mesma síntese
 - Paradoxo do aniversário

Aproximações:

- Difusão e confusão em funções de compressão
- Construção Merkle-Damgard
 - Compressão iterativa
 - Padding com o comprimento

Alguns dos algoritmos mais comuns de funções de síntese incluem MD5 (128 bits), SHA-1 (160 bits), SHA-2 (256 bits) [MD5 e SHA-1 já não são seguros]

XXIV.I Funções de Síntese - Dimensão

Considerando o textos:

- T1: "Hello User_A!", T2: "Hello User_B!", T3: "Hello User_XY!"

Diferentes algoritmos produzem valores de dimensão diferente, mas independente da dimensão do texto

- MD5:
 - T1: 70df836fdaf02e0dfc990f9139762541
 - T3: a08313b553d8bf53ca7457601a361bea
- SHA-1:
 - T1: f591aa1eabcc97fb39c5f422b370ddf8cb880fde
 - T3: c28b0520311e471200b397eaa55f1689c8866f25
- SHA-256:
 - T1: 9649d8c0d25515a239ec8ec94b293c8868e931ad318df4ccd0dff67aff89905
 - T3: 8fc49cde23d15f8b9b1195962e9ba517116f45661916a0f199fcf21cb686d852

Considerando o textos:

- T1: "Hello User_A!", T2: "Hello User_B!", T3: "Hello User_XY!"

Uma pequena alteração no texto produz uma alteração drástica no resultado

- MD5:
 - T1: 70df836fdaf02e0dfc990f9139762541
 - T2: c32e0f62a7c9c815063d373acac80c37
- SHA-1:
 - T1: f591aa1eabcc97fb39c5f422b370ddf8cb880fde
 - T2: bab31eb62f961266758524071a7ad8221bc8700b
- SHA-256:
 - T1: 9649d8c0d25515a239ec8ec94b293c8868e931ad318df4ccd0dff67aff89905
 - T2: e663a01d3bec4f35a470aba4baccece79bf484b5d0bffa88b59a9bb08707758a

XXV. Message Integrity Code - MIC

Fornecem a capacidade de detetar alterações por máquinas

- Erros de comunicação/armazenamento
- De carácter aleatorio ou não controlado

Envio (sendo T = texto e S = Função Sintese):

- Calcular $S(T)$
- Enviar $T + MIC$
 - $MIC = Sintese(T)$

Receção (sendo T = texto e S = Função Sintese):

- Receber dados T'
- Verificar se $S(T') = MIC$
- Calcular $S' = Sintese(T)$
- Validar se $S(T') == MIC$

Não protege contra alterações deliberadas!!

- Atacantes podem manipular T em T' e calcular novo MIC

XXVI. Message Authentication Code - MAC

Síntese gerada com recurso a **uma chave**

Só os conhcedores da chave conseguem gerar/validar o MAC

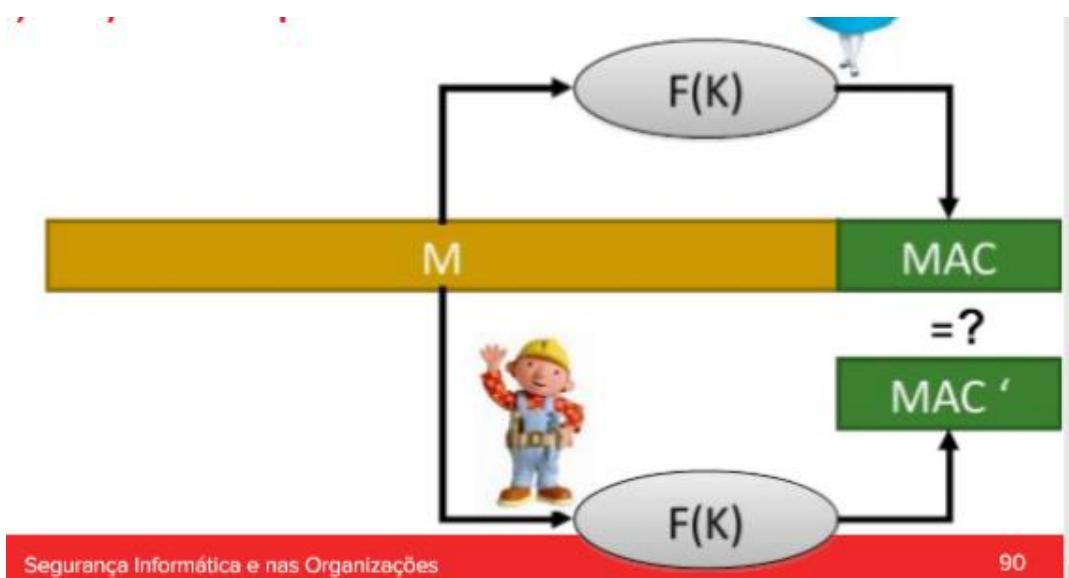
Utilizado para **garantir autenticidade/integridade**

Enviar

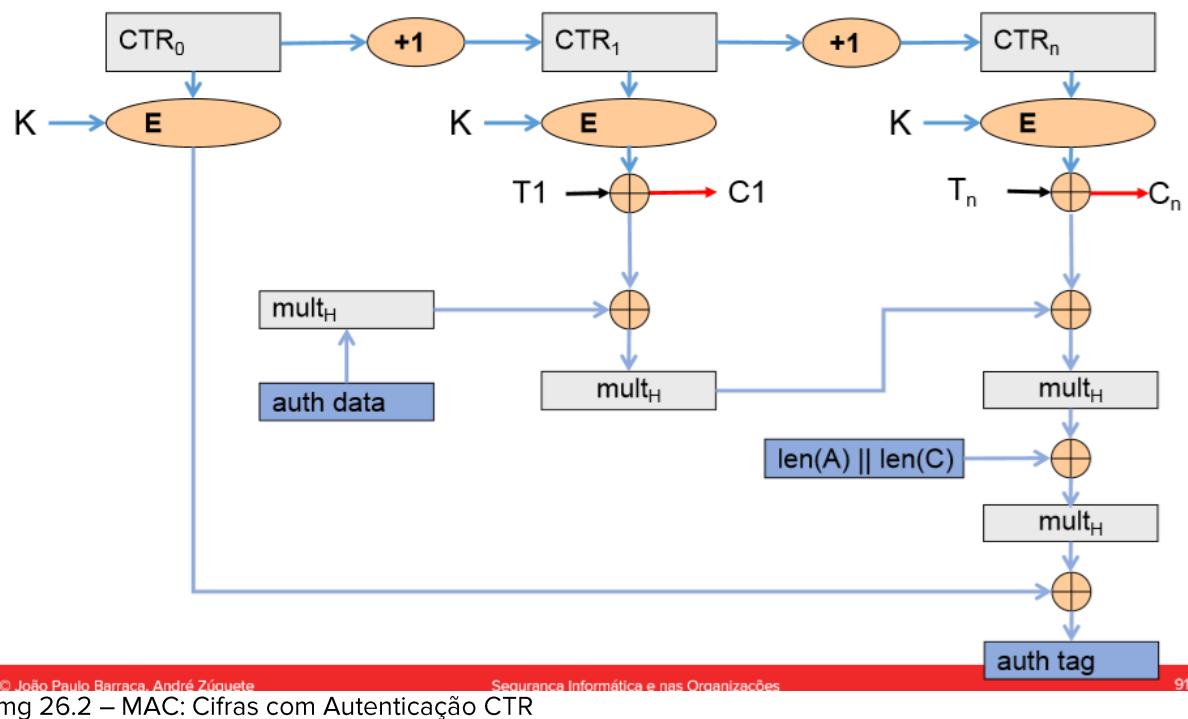
- $M + MAC$
 - $MAC = F(K, M)$

Receber

- Calcular $F(K, M')$
- Comparar com MAC



Img 26.1 – Exemplo do envio e receção MAC



© João Paulo Barraca, André Zúquete

Segurança Informática e nas Organizações

91

Img 26.2 – MAC: Cifras com Autenticação CTR

Aproximações do MAC

- **Cifrando uma síntese normal**
 - P.ex com uma cifra simétrica por blocos
- **Usando uma função chaveada, realimentação e propagação de erros**
 - ANSI X9.9 (ou DES-MAC) com DES CBC (64 bits)
- **Usando uma chave nos parametros da função**
 - Keyed-MD5 (128 bits): MD5(K, keyfill, texto, K, MD5Fill)
- **Construção HMAC**
 - $H(K, opad, H(K, ipad, texto))$
 - Ipad = 0x36 B vezes
 - Opad = 0x5C B vezes

- HMAC-MD5, HMAC-SHA, etc

XXVII. Cifra e Autenticação

Encrypt-then-MAC

- MAC calculado do criptograma
- Permite verificar a integridade antes da decifra

Encrypt-and-MAC

- MAC calculado do texto
- MAC não é cifrado
- Fornece informação acerca do texto original (se for igual a outro)

MAC-then-Encrypt

- MAC é calculado do texto
- MAC é cifrado
- Obriga a decifra completa antes da validação do MAC
- Erros só são detetados após a decifra e validação

XXVIII. Assinaturas Digitais

Autenticam o conteúdo de documentos

- Garantem a sua integridade

Autenticam o autor

- Garantem a identidade do autor/criador

Previnem repudião do conteúdo

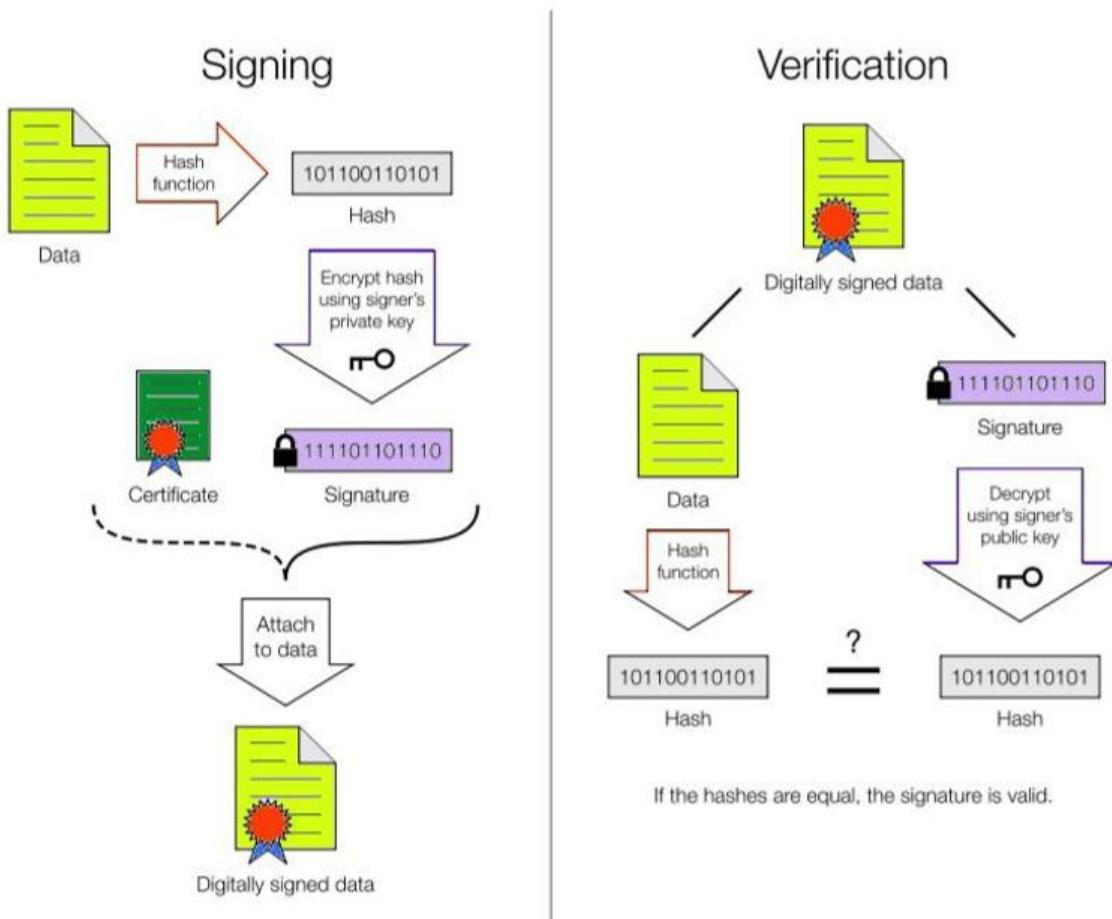
- Autor não pode negar a sua criação
- Só ele tem acesso à chave privada

Aproximações:

- **Cifra Assimétrica sobre Síntese**
 - Síntese usada por questões de desempenho
 - Cifra assimétrica para garantir autenticidade
- **Assinar:**
 - $Ax(doc) = info + E(Kx^{-1}, digest(doc+info))$
 - **Info -> Kx**

- **Verificar**

- $D(K_x, A_x(\text{doc})) = \text{digest}(\text{doc} + \text{info})$



Img 28.1 – Uso de Assinaturas digitais (verificação e signing)

XXIX. Assinaturas Cegas

Assinaturas podem ser efetuadas de forma cega

- Assinante não consegue observar os conteudos assinados
- Semelhante a assinar um envelope com um documento e um papel químico

Servem para garantir o anonimato e a não alteração da informação assinada

- Assinante X sabe quem lhe pede a assinatura (Y)
- X assina T1, mas Y depois recupera a assinatura sobre T2
 - T2 não é qualquer nem ao calhas
 - T2 está relacionado com T1
- O requerente pode apresentar T2 assinado por X
 - Mas não pode alterar T2
 - X não consegue associar T2 ao T1 que viu e assinou

XXX. Derivação de Chaves

Algoritmos requerem chaves de uma dimensão fixa (56, 128, 256, ... bits)

É necessário derivar chaves de várias fontes

- Segredos partilhados
- Passwords geradas por humanos
- Códigos PIN e segredos pequenos
- ...

Fonte original pode ter baixa entropia

- Reduz necessidade de um ataque de força bruta
- Necessário existir uma transformação complexa entre fonte e chave

Necessário poder-se chegar a multiplas chaves para a mesma password

- Evitar deduzir a password a partir da chave gerada

Reforço das chaves – Aumento da segurança de uma password

- Tipicamente definida por humanos
- Tornar os ataques por dicionario impraticaveis

Expansão das chaves – Aumento da dimensão de uma password

- Expansão ate ao pretendido para o algoritmo
- Eventualmente tambem a geração de outros valores como chaves para MACs

Derivação de chaves impõe a existencia de

- Um sal (SALT) que torna a geração unica
- Um problema custoso
- Um grau de complexidade

Dificuldades Computacionais

- Transformação requer recursos computacionais relevantes para ser realizada

Dificuldades de armazenamento

- Transformação ocupa recursos de armazenamento relevantes (memória)

XXXI. Derivação de Chaves – PBKDF2

Password Based Key Derivation Function 2

Produz uma chave com um custo computacional pre-definido

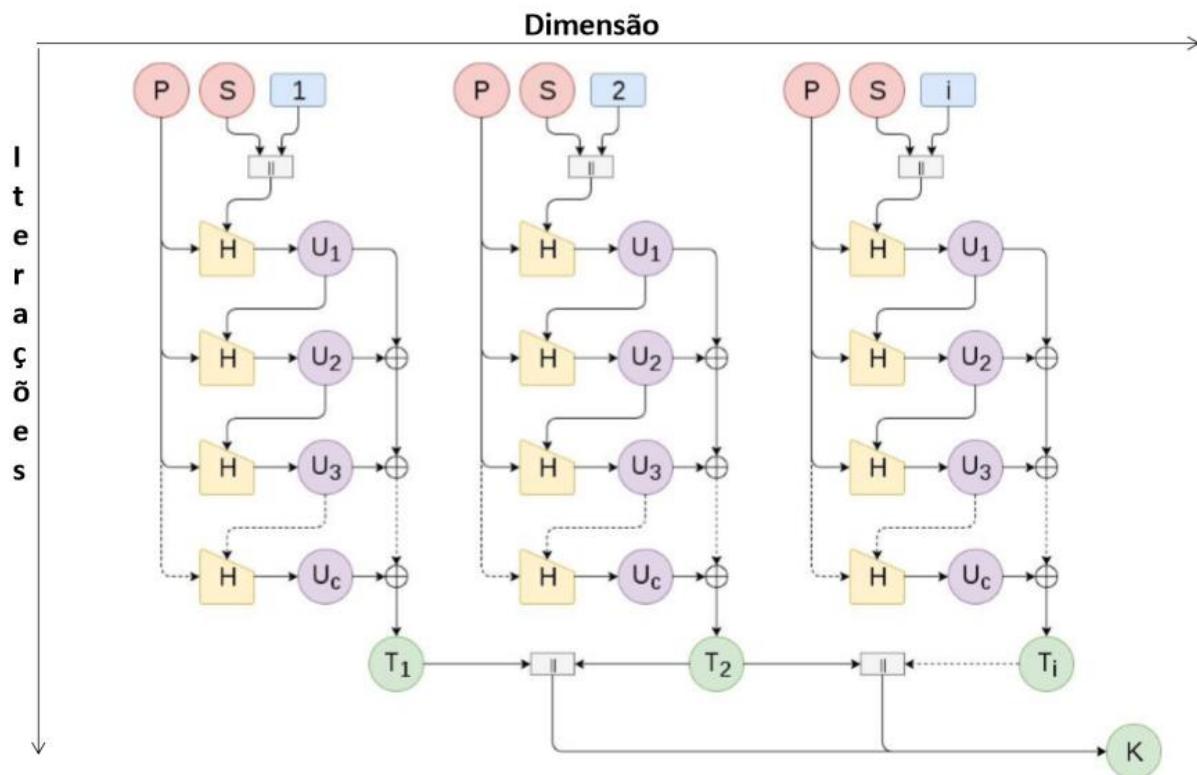
$K = \text{PBKDF2(PRF, Sal, Iterações, Password, dim)}$

- PRF: Pseudo-Random-Function: Uma Sintese

- Sal: Valor aleatório
- Iterações: Custo (valor nas centenas de milhares)
- Password: Segredo
- Dim: Dimensão do resultado pretendido

Operação

- Realiza $N \times \text{dim}$ operações do PRF com base no SAL e password
- Quanto maior o valor de N , maior o custo



Img 31.1 – Funcionamento do PBKDF2

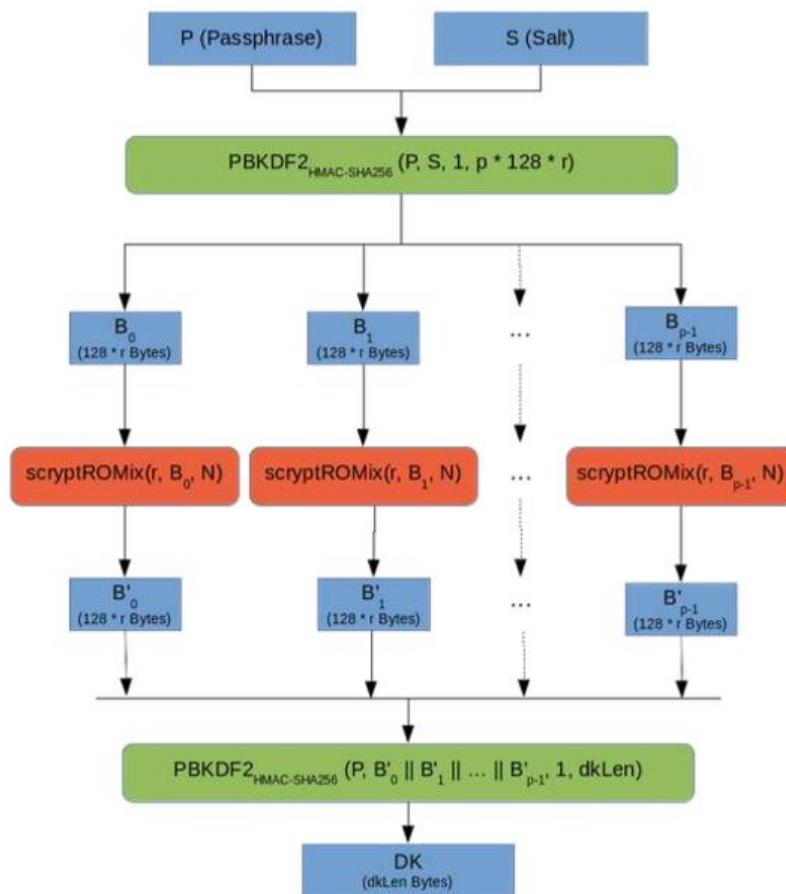
XXXII. Derivação de Chaves – SCRYPT

Produz uma chave com um custo de armazenamento pre-definido

$$K = \text{scrypt}(\text{Password}, \text{Sal}, N, p, \text{dim}, r, hLen, MFLen)$$

- Password: Segredo a expandir
- Sal: Valor aleatorio
- N: Parametro de custo
- P: Parametro de paralelização
 - $p \leq (2^{32}-1) * hLen/MFLen$
- dim: Dimensão da chave a produzir
- r: Tamanho dos blocos a usar (normalmente 8)
- hLen: Dimensão da função de síntese (32 para SHA256)
- MFLen: Octetos na mistura interna (Tipicamente $8 \times r$)

Img 32.1 – Funcionamento do Scrypt



Gestão de Chaves Assimétricas

I. Quais são os Problemas?

Temos que garantir a utilização apropriada dos pares de chaves

Privacidade das Chaves Privadas

- Para garantir autenticidade
- Para prevenir a repudiação das assinaturas

Distribuição correta das Chaves Públlicas

- Para garantir confidencialidade
- Para garantir a validação correta das assinaturas digitais

Evolução temporal do mapeamento entre entidade <-> par de chaves

Lidar com ocorrências catastróficas

- Perda de chave privada

Lidar com requisitos básicos da sua exploração

- Atualizar pares para reduzir riscos de impersonificação

Garantir a geração correta dos pares de chaves

Garantir uma qualidade dos pares de chave

- Aleatoriedade do gerador dos valores secretos
- Evitar que possam ser adivinhados

Melhorias da eficiência sem comprometer a segurança

- Tornar os mecanismos mais úteis
- Aumentar o performance

II. Objetivos e tarefas a cumprir

1. Geração de Pares de Chaves

a. Quando e como devem ser gerados

2. Manuseamento de chaves privadas

a. Como as manter privadas

3. Distribuição de chaves públicas

a. Como devem ser distribuidas para todo o mundo

4. Ciclo de vida dos pares de chaves

- a.** Qual a sua expiração
- b.** Como podem ser utilizadas
- c.** Como verificar a sua obsolência

III. Geração de Chaves - Princípios

Utilizar geradores bons na produção de segredos

Resultado é indistinguível de ruído

- Todos os valores devem possuir uma probabilidade igual
- Não devem existir padrões derivados no número da iteração ou valores anteriores

Um exemplo de um bom gerador é o **Gerador de Bernoulli**

- Gerador sem memória
- $P(b=1) = P(b=0) = 1/2$
- Equivale a atirar uma moeda perfeitamente balançada ao ar

Facilitar os processos sem comprometer a segurança

Chaves públicas eficientes

- Dimensão reduzida (tipicamente valores 2^{k+1} , i.e valores ímpares)
- Acelera operações com chaves públicas
- Não adiciona questões de segurança

A chave privada deve ser GERADA PELO PRÓPRIO

Para assegurar ao maximo a sua privacidade

- Ou seja, apenas o **dono possui a chave**
- Melhor ainda seria que o dono não tivesse nem ele a chave – Deveria ter apenas acesso aos processos com ela

Este princípio pode ser relaxado se não se pretender assinaturas digitais

- Onde não existem questões relacionadas com a não repudiação

Correção

A chave privada representa um sujeito

- Ex. um cidadão
- O risco do seu comprometimento deve ser minimizado
- Devemos considerar cópias de salvaguarda

O caminho de acesso à chave deve ser controlado

- Correção das aplicações que a usam
- Utilização de autenticação nas aplicações
- Cifra da chave privada

Confinamento

Armazenamento da chave numa entidade autónoma segura

- Módulo seguro de hardware interno
- Partição logica segura realizada ao nível do CPU
- Smartcard ou chave externa

Utilização protegida da chave

- Aplicações não devem utilizar a chave
- Invoca-se ao dispositivo a realização de operações

IV. Distribuição de Chaves Públicas

Como é que vamos distribuir a nossa Chave Pública ao mundo?

- Devemos distribuir a **quem pretender enviar informação confidencial**
 - Distribuição manual
 - Distribuição protegida por um segredo partilhado
 - Distribuir de forma AD-Hoc usando certificados digitais
- Devemos distribuir a **quem pretender validar informação autenticada**
 - Distribuição manual
 - Distribuir de forma AD-Hoc usando certificados digitais

Como é que garantimos a correção de uma chave pública?

- **Disseminação confiável de chaves públicas**
 - Usar caminhos ou grafos de relação de confiança
 - “Se **A confia em Kx+**, e **B confia em A**, então **B confia em Kx+**”
- **Hierarquias e grafos de certificação**
 - Expressão clara das relações de confiança entre entidades
 - Certificação é unidirecional

V. Certificados Digitais de Chaves Públicas

Documentos digitais emitidos por uma Entidade Certificadora (EC) (Certification Authority (CA))

Ligam uma chave pública a uma entidade

- Pessoa, sistema ou serviço

São documentos públicos

- Contem apenas informação pública
- Podem conter informação adicional associada a entidade

São seguros por meios criptográficos

- Possuem uma impressão digital para identificação
- São assinados com uma assinatura digital criada pelo emissor (CA)

São usados para distribuir chaves públicas de forma confiável

Os verificadores podem **validar os documentos**

- Validar identificação com o contexto atual
- Validar instantes temporais
- Validar a utilização da chave pública
- Validam a assinatura digital do documento usando a chave pública da CA

Os verificadores **confiam no comportamento das CA**

- Portanto confiam nos documentos que emitem
- Se uma CA associou uma chave pública a A e se o verificador confiar na CA, então irá confiar que a associação de A é correta

Norma X.509v3

- Campos obrigatórios
 - Versão
 - Sujeito (subject)
 - Chave pública
 - Datas (início e expiração)
 - Emissor (issuer)
 - Assinatura
 - ...
- Extensões: definem utilização
 - Críticas ou não Críticas

PKCS #6

- Extended-Certificate Syntax Standard

• Formatos binários

- ASN.1 (Abstract Syntax Notation
 - DER, CER, BER, etc.
- PKCS #7
 - Cryptographic Message Syntax Standard
- PKCS #12
 - Personal Information Exchange Syntax Standard

• Outros formatos

- PEM (Privacy Enhanced Email)
 - Base64

Img 5.1 – Alguns formatos de Certificados

VI. Entidades Certificadoras – CA's

Organizações que gerem certificados de chave pública

- Empresas, entidades sem fins lucrativos, entidades governamentais
- Normalmente possuem a tarefa de validar associações chave-entidade
- Importante que operem corretamente para serem confiáveis

Definem políticas e mecanismos para...

- Emissão de certificados
- Revogação de certificados
- Distribuição de certificados
- Emissão e distribuição das chaves privadas correspondentes

Gerem processos de revogação de certificados

- Listas de identificadores de certificados revogados
- Interfaces para verificação do estado do certificado

VII. Entidades Certificadoras Confiáveis

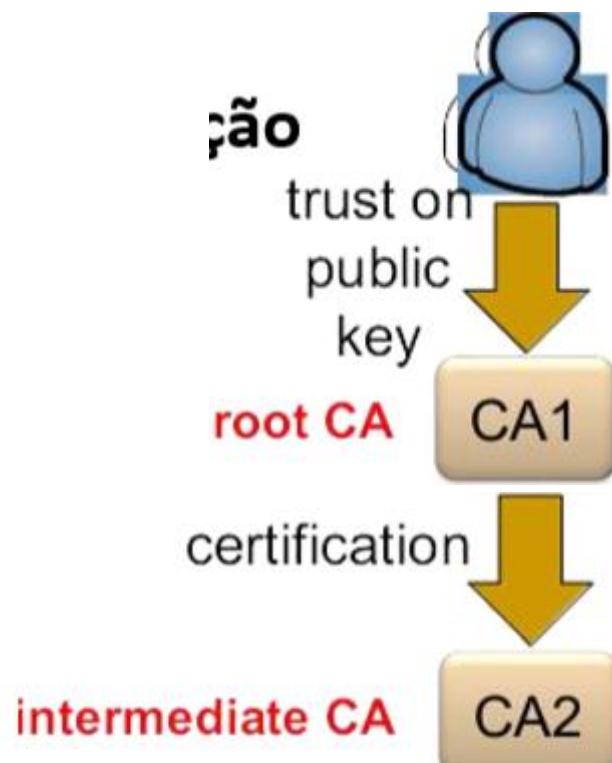
Entidades Certificadoras (CAs) em que um sujeito confia

- Podem ser confiaveis por um grupo restrito ou uma maioria
- Possuem processos de gestão confiaveis (o que implica custos)

Raizes de confiança (ou raizes de certificação)

- Alguem possui e confia numa chave publica

- Certificados das Cas são auto-assinados
 - Podem tambem ser assinados por outras Cas
- Distribuição Manual
 - Nos browsers ou no OS



Img 7.1 – Transmissão de confiança (User confia na CA 1 – Root CA, esta confia na CA 2, logo User confia na CA 2)

Certificate Viewer: "www.ua.pt"

General **Details**

This certificate has been verified for the following uses:

- SSL Client Certificate
- SSL Server Certificate

Issued To

Common Name (CN) www.ua.pt
 Organization (O) Universidade de Aveiro
 Organizational Unit (OU) STIC
 Serial Number 06:B4:17:0C:D7:EF:AC:9F:A3:79:9A:78:0E:7E:5A:8C

Issued By

Common Name (CN) TERENA SSL CA 3
 Organization (O) TERENA
 Organizational Unit (OU) <Not Part Of Certificate>

Period of Validity

Begins On May 27, 2019
 Expires On June 3, 2021

Fingerprints

SHA-256 Fingerprint 6C:BA:BD:A1:7E:A9:8D:EA:7B:18:22:44:EC:71:D5:41:4D:08:D4:A6:FC:48:1B:3C:9B:05:EB:DA:69:A6:A5:EE
 SHA1 Fingerprint 17:79:15:B5:0E:E0:34:51:2D:FA:DE:DF:77:1E:E1:0A:B3:4B:2F:2B

Certificate Viewer: "www.ua.pt"

General **Details**

Certificate Hierarchy

- ▼ DigiCert Assured ID Root CA
 - ▼ TERENA SSL CA 3
 - www.ua.pt

Certificate Fields

- ▼ www.ua.pt
 - ▼ Certificate
 - Version
 - Serial Number
 - Certificate Signature Algorithm
 - Issuer
 - ▶ Validity
 - ▶ Subject
 - ▼ Subject Public Key Info
 - Subject Public Key Algorithm

Field value

CN = www.ua.pt
 OU = sTIC
 O = Universidade de Aveiro
 L = Aveiro
 C = PT

Export...

Certificate Viewer: "TERENA SSL CA 3"

General **Details**

This certificate has been verified for the following uses:

- SSL Certificate Authority

Issued To

Common Name (CN) TERENA SSL CA 3
 Organization (O) TERENA
 Organizational Unit (OU) <Not Part Of Certificate>
 Serial Number 08:70:BC:C5:AF:3F:DB:95:9A:91:CB:6A:EE:EF:E4:65

Issued By

Common Name (CN) DigiCert Assured ID Root CA
 Organization (O) DigiCert Inc
 Organizational Unit (OU) www.digicert.com

Period of Validity

Begins On November 18, 2014
 Expires On November 18, 2024

Fingerprints

SHA-256 Fingerprint BE:B8:EF:E9:B1:A7:3C:84:1B:37:5A:90:E5:FF:F8:04:88:48:E3:
 A2:AF:66:F6:C4:DD:7B:93:8D:6F:E8:C5:D8
 SHA1 Fingerprint 77:B9:98:B2:BD:75:22:E1:7E:C0:99:EA:71:77:51:6F:27:78:7C:AD

Close

CA Intermédia
(Certificado emitido por outra CA)
(Issued To != Issued By)

Img 7.3 – Exemplo de CA Intermediadas e CA Root

Certificate Viewer: "DigiCert Assured ID Root CA"

General **Details**

This certificate has been verified for the following uses:

- SSL Certificate Authority

Issued To

Common Name (CN) DigiCert Assured ID Root CA
 Organization (O) DigiCert Inc
 Organizational Unit (OU) www.digicert.com
 Serial Number 0C:E7:E0:E5:17:D8:46:FE:8F:E5:60:FC:1B:F0:30:39

Issued By

Common Name (CN) DigiCert Assured ID Root CA
 Organization (O) DigiCert Inc
 Organizational Unit (OU) www.digicert.com

Period of Validity

Begins On November 10, 2006
 Expires On November 10, 2031

Fingerprints

SHA-256 Fingerprint 3E:90:99:B5:01:5E:8F:48:6C:00:BC:EA:9D:11:1E:E7:21:FA:BA:
 35:5A:89:BC:F1:DF:69:56:1E:3D:C6:32:5C
 SHA1 Fingerprint 05:63:B8:63:0D:62:D7:5A:BB:C8:AB:1E:4B:DF:B5:A8:99:B2:4D:43

Close

CA Raiz
(Certificado auto-emitido)
(Issued To == Issued By)

VIII. Hierarquias de Certificação – Modelo PEM

PEM: Privacy-enhanced Electronic Mail

Modelo de Monopólio

- Raiz única – IPRA (Internet Policy Registration Authority)
- Várias PCA (Policy Creation Authorities) abaixo da raiz
- Várias Cas abaixo de cada PCA
- Forma uma cadeia de certificação
 - Árvore de raiz única

Modelo nunca foi implementado globalmente :(

Alternativa preferida – **Floresta de Hierarquias em cada CA, SEM IPRA**

- Hierarquias independentes sem uma raiz única
- Oligarquia

Cada **CA Raiz negocia a distribuição da sua chave pública em cada entidade**

- Entidades: Browsers, OS, ...

IX. Hierarquias de Certificação – Modelo PGP

PGP: Pretty Good Privacy

Segue o modelo baseado numa **rede de confiança**

- E não baseado em arvore

Sem qualquer **autoridade central de confiança**

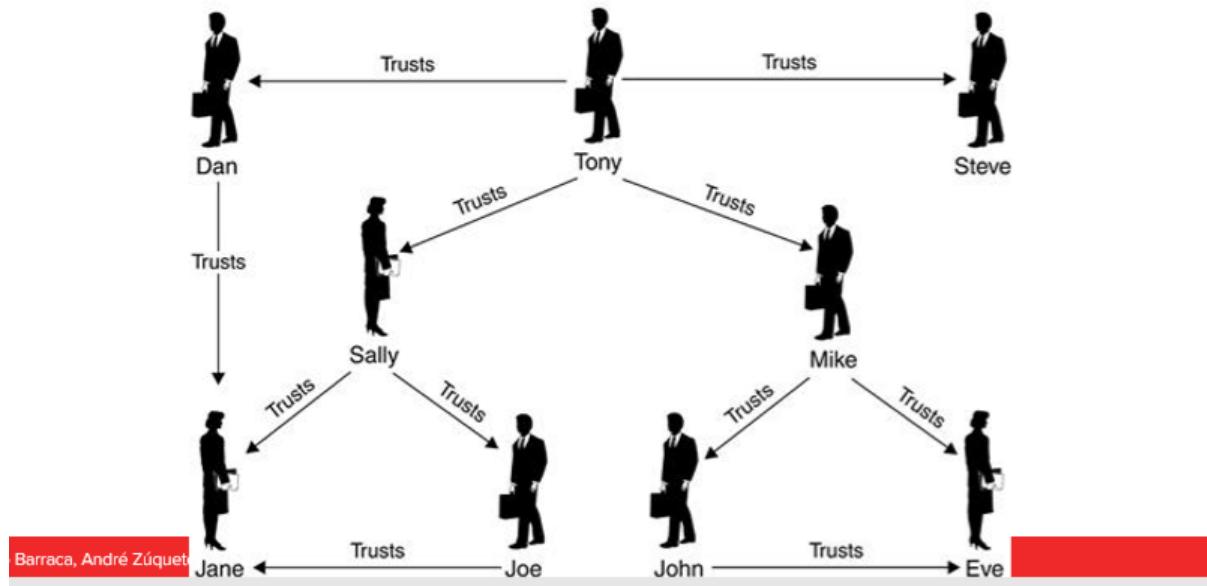
- Qualquer pessoa/entidade é um potencia certificador
- Qualquer pessoa/entidade pode certificar uma chave publica e publicar a assinatura para os outros

Pessoas usam **dois tipos de confiança**

- Confiança nas **chaves que conhecem**
 - Validadas diretamente por qualquer meio (presencial, telefone, ...)
- Confiança no **comportamento de outros certificadores**
 - Assuminda que estes verificam as chaves que certificam

Confiança Transitiva

**SE Mike confia que o John é um certificador correto,
E John certificou a chave pública da Eve,
ENTÃO Mike confia na chave pública da Eve**



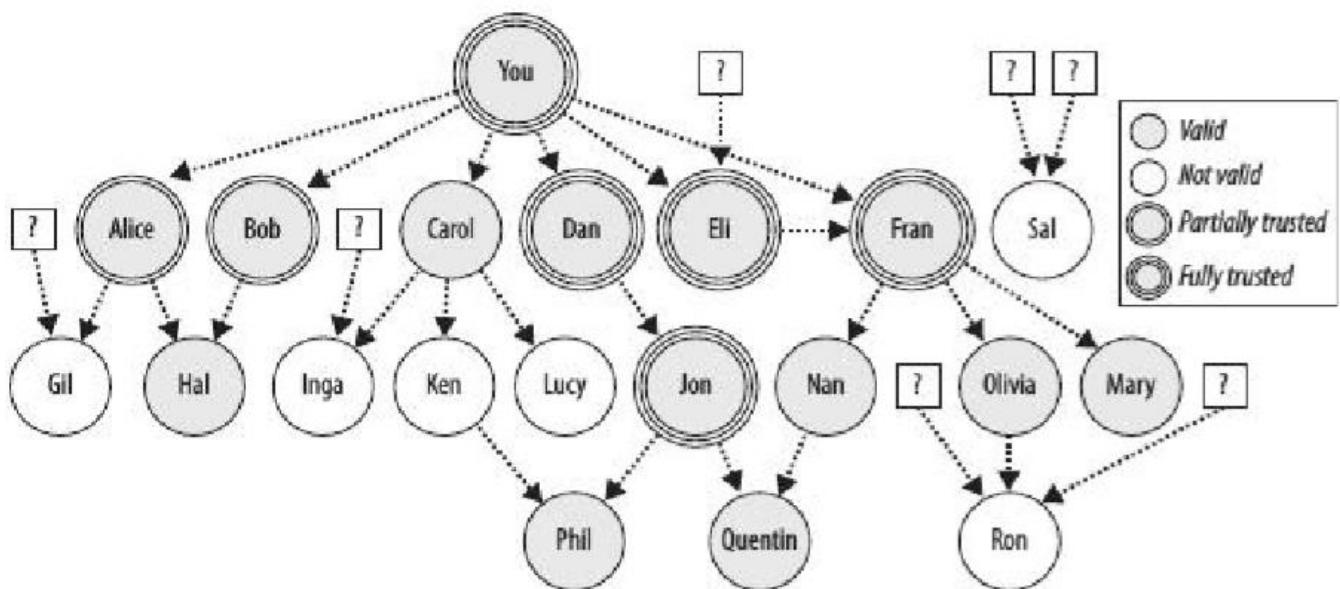
Img 9.1 – Exemplo de rede de confiança e de confiança transitiva

Confiança – Quando uma pessoa confia noutra pessoa

- É unidirecional, pessoal e subjetiva
- Níveis
 - Ultimate
 - Chaves próprias das quais se tem a chave privada
 - Complete
 - Marginal
 - NoTrust
 - Ou Untrusted

Validade – Quanta verificação a chave possui

- Valida:
 - A confia completamente em B
 - Ou A confia marginalmente em C e D
 - e D ou B em conjunto com C assinaram a chave de E
- Marginalmente Valida
 - A confia marginalmente em B e B assinou a chave de E
- Invalida
 - Sem um caminho



Img 9.2 – Exemplos de Validade dentro de uma rede de confiança

X. Reminder sobre Chaves Assimétricas

Pares de chaves devem ter uma validade limitada

- Porque as chaves privadas podem ser perdidas ou descobertas
- Para implementar mecanismos de atualização periódicos

Problemas

- Os certificados podem ser copiados e distribuídos livremente
- O universo de possuidores de certificados é desconhecido
 - Não é viável contactar todos os possuidores de certificados para eliminar certificados específicos

Soluções

- Certificados com validade temporal definida
- Listas de Revogação de Certificados (CRL)
 - Para permitir revogar certificados antes que expirem

XI. Listas de Revogação de Certificados

- CRL

Listas assinadas com identificadores de certificados revogados prematuramente

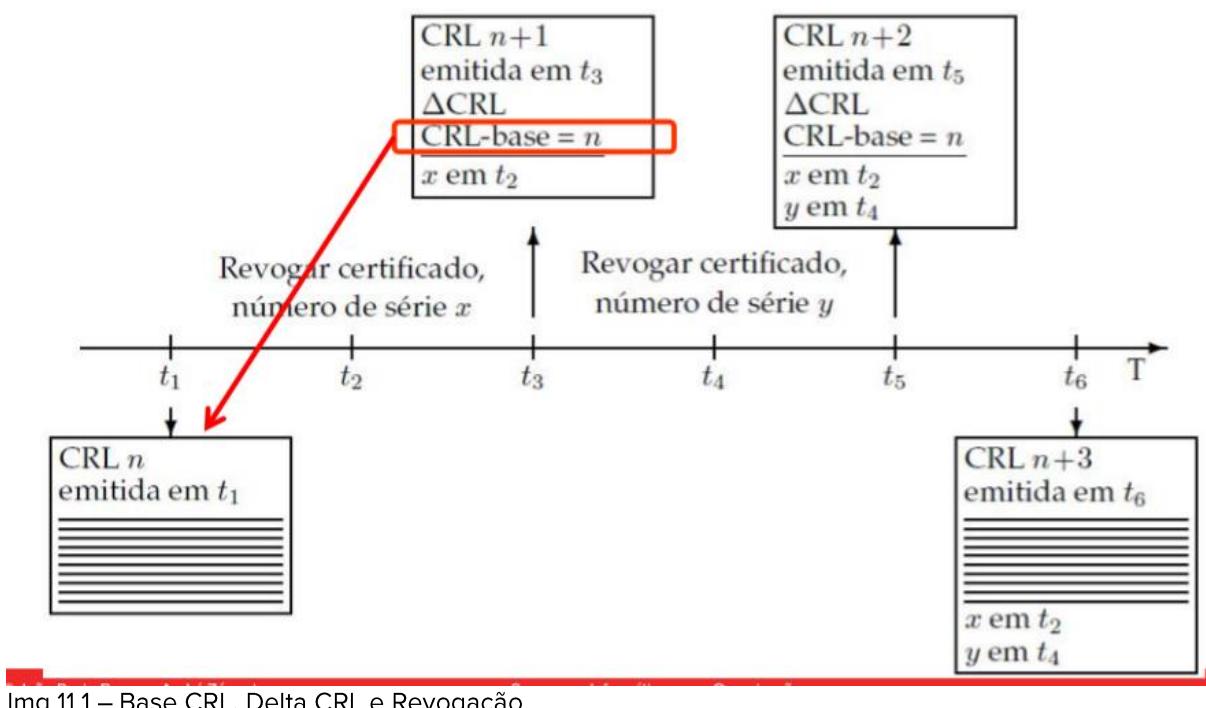
- Devem ser consultadas periodicamente pelos verificadores
- Entradas podem conter a razão

Publicação e distribuição de CRLs

- Cada CA possui a sua CRL
- São de Acesso publico
- Cas trocam CRLs para facilitar distribuição

Varios formatos disponíveis

- Base CRL
 - Lista completa com todos os certificados revogados
- Delta CRL
 - Lista com as diferenças desde a ultima Base CRL
- OCSP
 - API para verificação individual de cada certificado



Img 11.1 – Base CRL, Delta CRL e Revogação

XII. Online Certificate Status Protocol – OCSP

Protocolo baseado em HTTP para **verificar a revogação de certificados**

- Pedido inclui o numero de serie do certificado
- Resposta assinada afirma qual o estado
- Uma verificação por certificado

Reduz a largura de banda usada pelos clientes

- Um pedido por certificado, em vez de toda a lista (Base CRL)

Pode envolver maior largura de banda para as Cas

- Se clientes validarem sempre os certificados
- Pode comprometer a privacidade
 - CA sabe quando um sistema acede a um serviço

OCSP Stapling

- Inclui um instante temporal assinado na resposta
- Clientes podem guardar respostas durante a sua validade

XIII. Distribuição de Certificados de Chave Pública

Transparente e integrado nos sistemas e aplicações

- Sistema de Diretórios
- Online
 - Incluido nos protocolos
 - Assinaturas digitais de correio com MIME ou em documentos
 - Comunicações seguras usando TLS
- Pre-Distribuição
 - Incluido nas aplicações / OS

XIV. Public Key Infrastructure - PKI

Infraestrutura de apoio ao uso de pares de chaves e certificados. Permite:

Criação segura de chaves assimétricas

- Políticas de subscrição
- Políticas de geração de pares de chaves

Criação segura de certificados de chaves públicas

- Políticas de subscrição
- Definição de atributos do certificado

Definição e uso de cadeias de certificação

- Inserção numa hierarquia de certificação
- Certificação de outras Cas

Atualização, publicação e consulta de listas de certificados revogados

- Políticas para revogar certificados
- Distribuição permanente de CRLs
- Serviço OCSP

Uso de estruturas de dados e protocolos que permitem a interoperabilidade entre componentes

- **Subscrição**
 - Em locais próprios, pessoal
- **Vários pares de chaves por pessoa**
 - Um para autenticação
 - Uma para assinaturas qualificadas
 - Gerados no cartão, não exportáveis
 - Requerem um PIN em cada operação
- **Uso autorizado dos certificados**
 - Autenticação
 - SSL Client Certificate, Email (Netscape cert. type)
 - Signing, Key Agreement (key usage)
- **Assinatura**
 - Email (**Netscape cert. type**)
 - Non-repudiation (**key usage**)
- **Caminho de certificação**
 - Raiz bem conhecida e divulgada
 - GTE Cyber Trust Global Root
 - Baltimore CyberTrust Root
 - MULTICERT Root Certification Authority 01
 - CA raiz PT debaixo da GTE/Baltimore/Multicert
 - CA raiz CC debaixo de CA raiz PT
 - CAs Autenticação CC e Assinatura CC debaixo CA raiz CC
- **CRLs**
 - Certific. de assinatura pré-revogados
 - Removida se o dono explicitamente requerer o uso de assinaturas
 - Todos os certificados são removidos a pedido do dono
 - Mediante a apresentação de um PIN de revogação
 - Pontos de distribuição das CRL indicados em cada certificado

Img 14.1 – Exemplo PKI – Políticas do Cartão de Cidadão

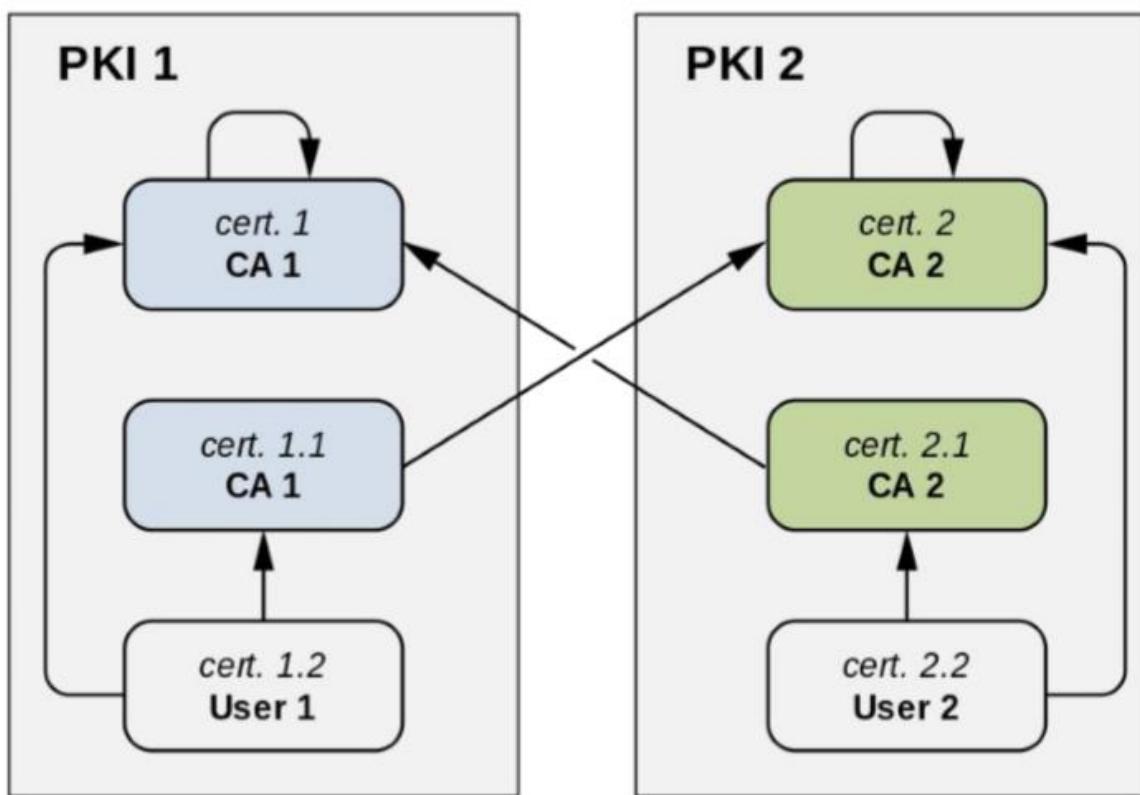
XV. PKI – Relações de Confiança

Um PKI estabelece **relações de confiança** de duas formas:

- **Emitindo certificados de chaves públicas de outras Cas**
 - Abaixo na hierarquia ou...
 - Não relacionadas hierarquicamente
- **Requerendo a certificação da sua chave pública a outras Cas**
 - Acima da hierarquia ou...
 - Não relacionadas hierarquicamente

Relações de confiança caracteristicas:

- Hierarquicas
- Cruzadas
 - A certifica B e vice-versa
- Ad-Hoc (Meshed)
 - Grafos mais ou menos complexos de certificação



Img 15.1 – PKI – Certificação hierarquica e cruzada

XVI. Fixação de Certificados - Pinning

Se um atacante possui acesso a uma raiz de confiança, ele pode emitir qualquer certificado para qualquer entidade D:

- Pode manipular a CA para que ela emita um certificado
 - Dificil
- Pode Injetar raizes adicionais nos sistemas da vitima
 - Facil

Certificate Pinning – Adicionar uma impressão digital da chave publica ao codigo

- Impressão digital usa SHA256
- Associada a um pedido HTTP especifico

Processo de validação normal + verificação de impressão digital

- Certificado tem de ser assinado por uma raiz de confiança
- Certificado tem de ter uma chave publica com impressão digital especificada

XVII. Transparência de Certificação

Problemas:

- Cas podem ser comprometidas
 - Por atacantes maliciosos
 - Por governos
 - ...
- Comprometimento é difícil de detetar
 - Resulta na alteração das regras de funcionamento da PKI
 - Dono legítimo dificilmente saberá

Definição – Sistema que regista todos os certificados públicos emitidos

- Garante que só são publicados certificados que levam a raízes legítimas
- Armazena toda a cadeia de certificação de cada certificado
- Apresenta esta informação para auditoria
 - Organização ou ad-hoc pelos utilizadores

Smartcards & Cartão de Cidadão

I. Smartcards

São dispositivos físicos para armazenamento de chaves e operações sobre as mesmas

Inviolaveis, resistentes a ataques por canais paralelos ou virus

Objetivo – Permitir a utilização de chaves sem o seu compromisso

- Titular pode utilizar chave para realizar operações criptograficas – Simetricas E Assimetricas
 - Ex. Autenticar o titular, Gerar assinaturas de documentos, Gerar repostas a desafios, Armazenar Valores, ...

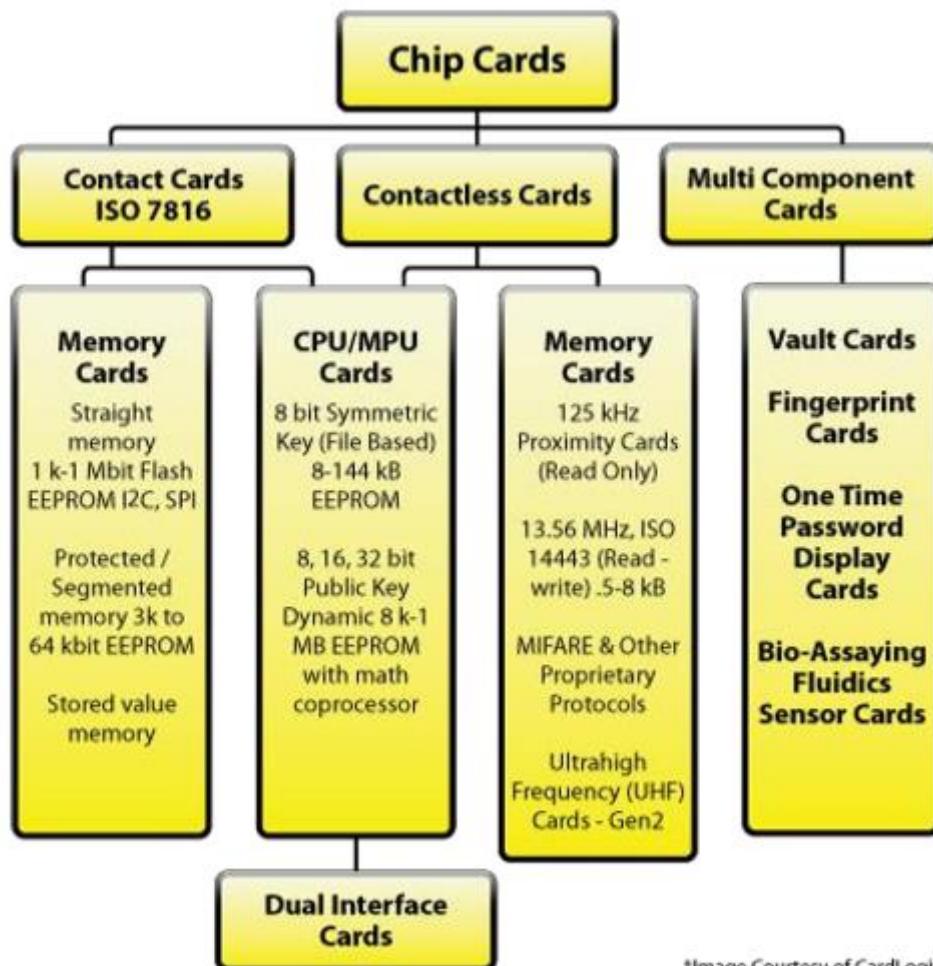
Exemplos de Smartcards incuem o Cartão de Cidadao, Cartões bancarios, Transportes, SIM, ...

São cartões com capacidade de computação

- CPU
- ROM
- EEPROM
- RAM

Podem ter uma **Interface**:

- **Com contactos**
- **Sem contactos**

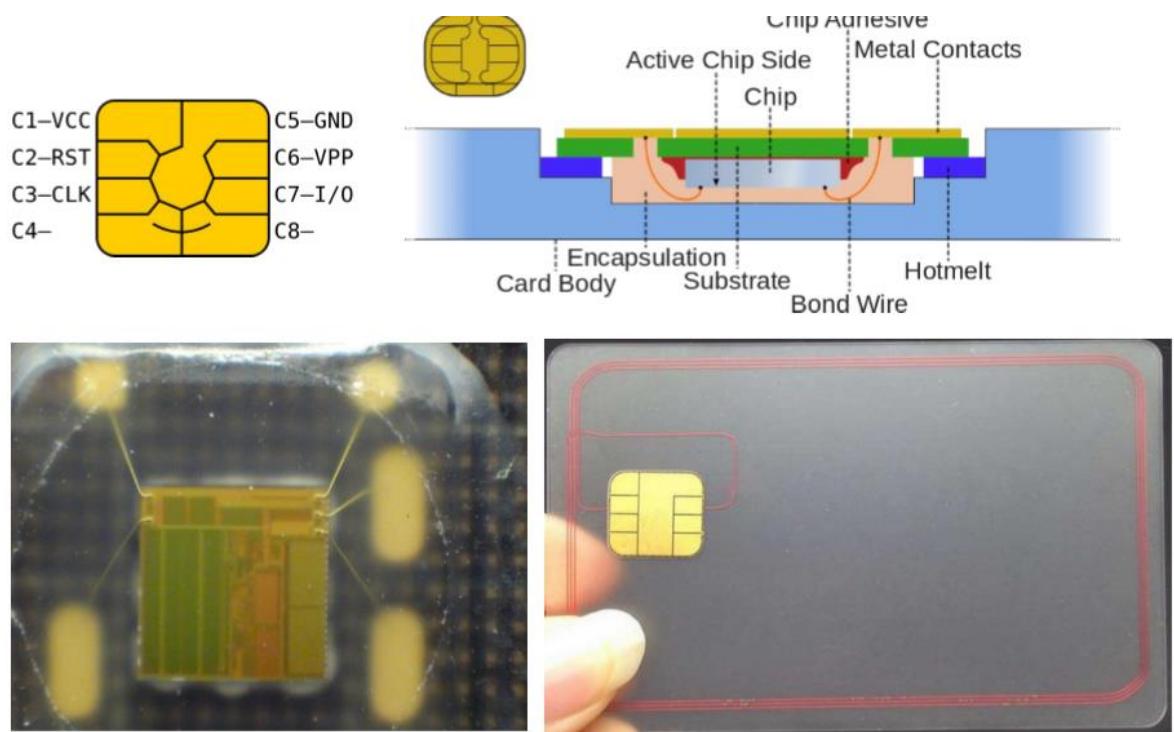


*Image Courtesy of CardLogix

Img 1.1 – Tipos de Smartcards

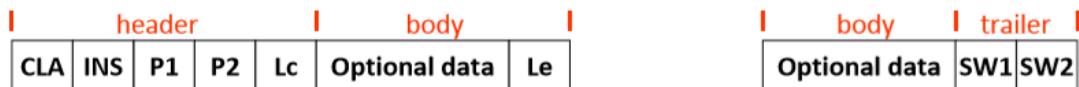
- **CPU**
 - 8/16 bit
 - Crypto-coprocessor (opt.)
- **ROM**
 - Sistema Operativo
 - Comunicação
 - Algoritmos criptográficos
- **EEPROM**
 - Sistema de Ficheiros
 - Programas / aplicações
 - Chaves/ passwords
- **RAM**
 - Dados temporários
 - Apagados quando cartão é desligado
- **Contactos Mecânicos**
 - ISO 7816-2
 - Power
 - Soft reset
 - Clock
 - Half duplex I/O
- **Segurança Física**
 - Resistente a acessos físicos diretos
 - Resistente a ataques por canais paralelos

Img 1.2 – Componentes dos SmartCards



Img 1.3 – Chips dos SmartCards

II. Interação com Smartcards – APDU



- **APDU de Comando**

- CLA (1 octeto)
 - Classe da instrução
- INS (1 octeto)
 - Comando
- P1 e P2 (2 octetos)
 - Parâmetros específicos do comando
- Lc
 - Comprimento dos dados opcionais
- Le
 - Comprimento dos dados esperados na resposta
 - Zero (0) significa todos os dados disponíveis

- **APDU de Resposta**

- SW1 e SW2 (2 octetos)
 - Octeto de estado
 - 0x9000 significa SUCESSO

Img 2.1 – Modo de interação com smartcards

III. Interação com Smartcards – Protocolos de Baixo-nível (T=0 e T=1)

T = 0

- Enviado um octeto de cada vez
- Mais lento

T = 1

- Octetos transmitidos em blocos
- Mais rápido

ATR

- Resposta à operação de RESET
- Reporta o protocolo esperado pelo cartão

IV. Codificação de Objetos nos Smartcards – TLV e ASN.1 BER

TLV – Tag-Length-Value

- Tag: Tipo de objeto
- Length: Tamanho do objeto
- Value: Dados do objeto

Cada TLV é codificado através das regras **ASN.1 BER – Abstract Syntax Notation, Basic Encoding Rules**

Dados de um objeto podem conter outros TLV (é uma estrutura recursiva)

Permite ignorar objetos desconhecidos

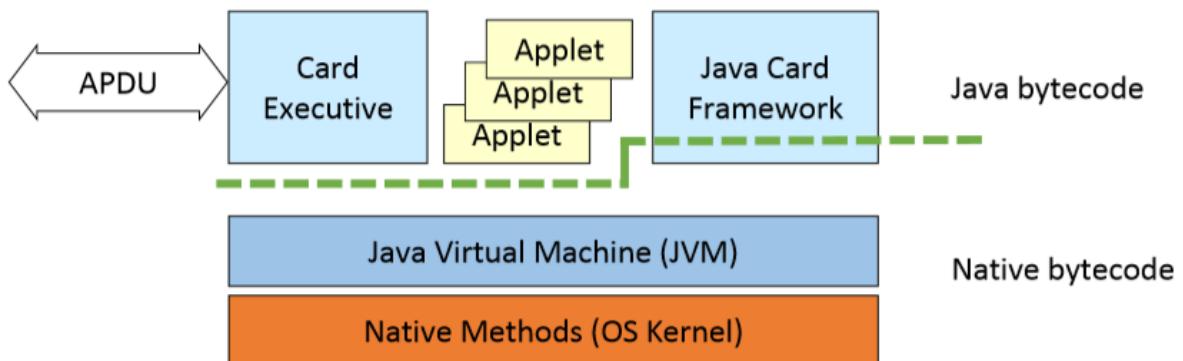
V. Modelo de Computação do Smartcard – Cartões Java

Smartcards executam **Java Applets**

- Utilizam o **Java Card Runtime Environment**

O JCRE executa em cima (no topo) do **OS Nativo**

- Java Virtual Machine
- Card Executive
 - Gestão do Cartão
 - Comunicações
- Java Card Framework
 - Bibliotecas de funções



Img 5.1 – Arquitetura dos Smartcards

VI. Cartão de Cidadão

É um cartão de identificação com as dimensões de um cartão de crédito

Contém vários métodos de fornecer informação de identidade

1. Informática

- a. Interção com o smartcard

2. Visual (legível por humanos)

- a. Fotografia, números e nomes

3. Visual (legível para dispositivos)

- a. MRZ – Machine Readable Zone



Img 6.1 – Estrutura dos CC

VII. Cartão de Cidadão - Atributos Visuais

Legíveis por Humanos

- **Nome**
 - Sobrenome
 - Nome Proprio
 - Pais
- **Atributos fisicos**
 - Sexo
 - Altura
- **Outros**
 - Data de nascimento
 - Nacionalidade
 - Fotografia
 - Assinatura caligrafica
- **Numeros**
 - Numero de Identificação Civil (e checksum)
 - Numero de identificação Fiscal
 - Numero do Sistema Nacional de Saude
 - Numero da Segurança Social
 - Numero do documento
 - Validade
- **Versão do Cartão**

Legíveis por Dispositivos

- Nome
 - Sobrenome
 - Nome Proprio
 - Nomes adicionais
 - Numero de Nomes
- Atributos Fisicos
 - Sexo
- Outros
 - Data de Nascimento
 - Nacionalidade
- Numeros
 - Identificação Civil (e Checksum)
 - Do Documento (e Checksum)
 - De Documentos Emitidos
- Validade



Img 7.1 – MRZ (por ordem temos): 000000000 – NIF/Documento ; 0 – checksum Nif ; 4 – checksum do documento ; 810810 – Data de Nascimento ; F – Sexo ; 120115 – Validade ; PRT – Nacionalidade ; 1 – Numero de Nomes ; Avila Paula Andreia Conceicao – Sobrenome, Nome Proprio, Nomes adicionais

De Segurança



Img 7.2 – Legenda dos atributos visuais de segurança

VIII. Proteção por PIN

Posuir o cartão é **insuficiente para**:

- Obter a morada
 - Exceto nos CC mais recentes
 - Forças policiais podem obter a morada sem o PIN
- Obter ou usar a chave privada de autenticação
- Obter ou usar a chave privada de assinatura
- Obter ou usar a chave secreta de EMV-CAP

Operações estão protegidas por um PIN

- PIN de 4 números
- PIN é bloqueado após 3 tentativas incorretas

IX. Certificados nos SmartCards

Possibilita autenticar o dono do cartão

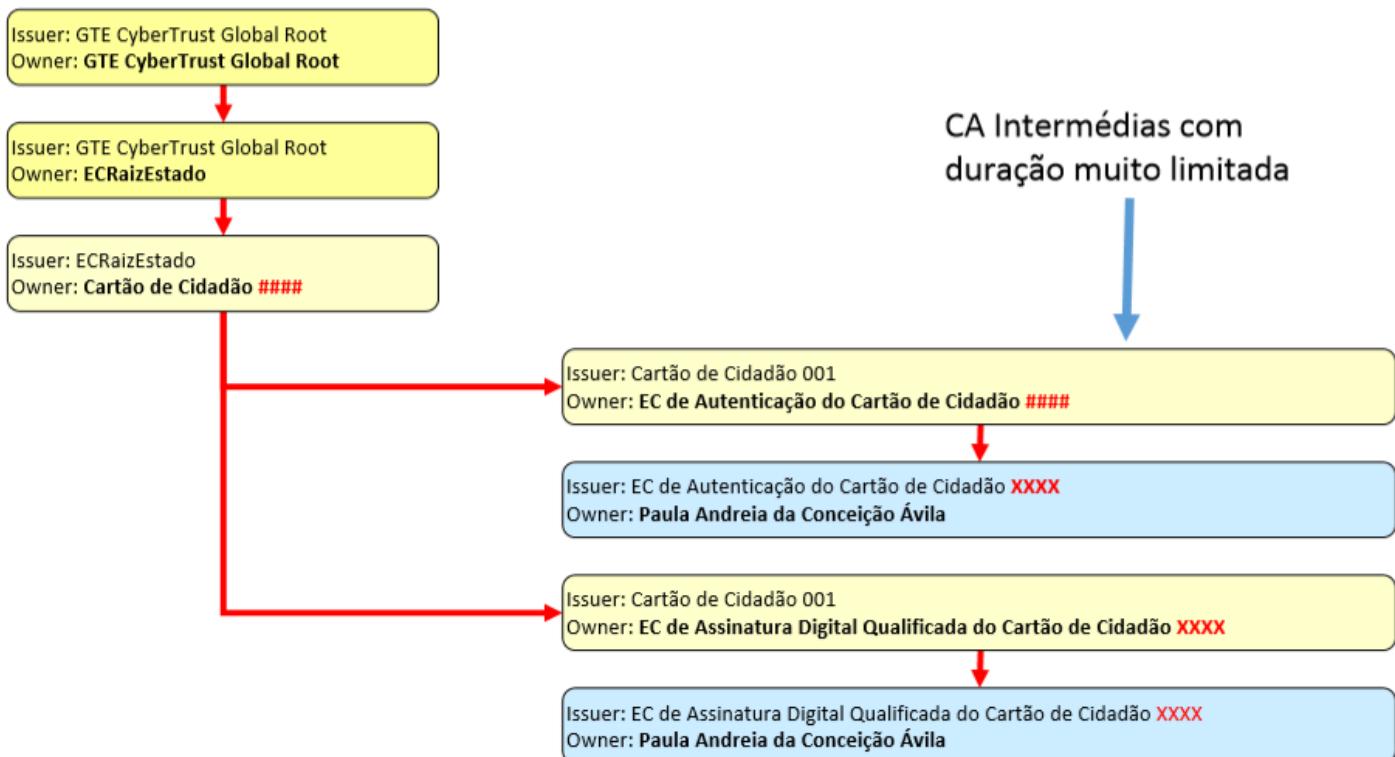
- O dono pode distribuir o seu certificado para outras pessoas/serviços que passam a poder verificar a sua identidade

Possibilita o dono autenticar outras pessoas com cartões semelhantes

- Cadeia de certificações presente no cartão

Possibilita o cartão autenticar clientes com certificados semelhantes

- Algumas operações podem ser pedidas ao cartão com certificados “especiais” que o cartão valida



Img 9.1 – Certificados no Smartcard

X. Certificados nos SmartCards – Interoperações com outras aplicações

Aplicações de Watchdog detetam inserção e remoção

Inserção:

- Aplicações obtêm certificados e inserem-nos nos repositórios dos navegadores
- Utilização das chaves respetivas é condicionada pelos PIN

Remoção:

- Aplicações removem certificados dos repositórios dos navegadores

XI. Aplicações em Smartcards – Aplicações no CC

IAS Classic V3

- Autenticação e Assinatura digital
- Utilização de pares de chaves assimétricas

EMV-CAP

- Geração de One-Time-Passwords para canais alternativos (telefone, FAX, etc...)
- Retirado em 2016

Precise Biometric BIO Match On Card

- Validação de impressões digitais

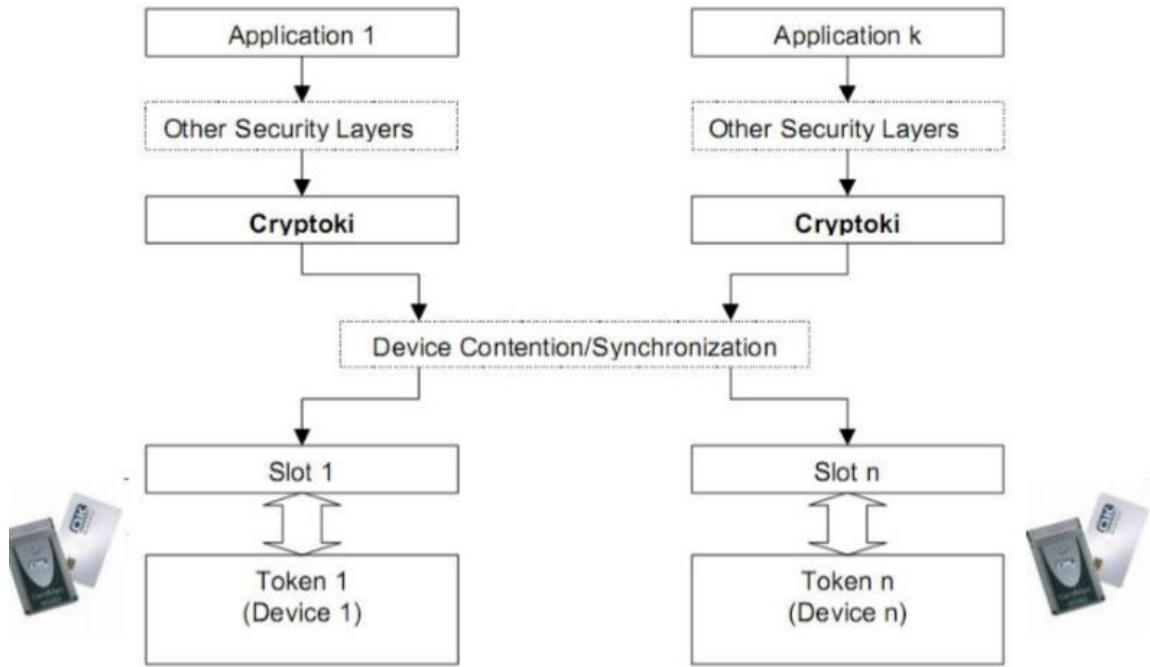
XII. Serviços Criptográficos do SmartCard - Middleware

Bibliotecas que servem como ponte entre as funcionalidades dos smartcards e as aplicações de mais alto nível

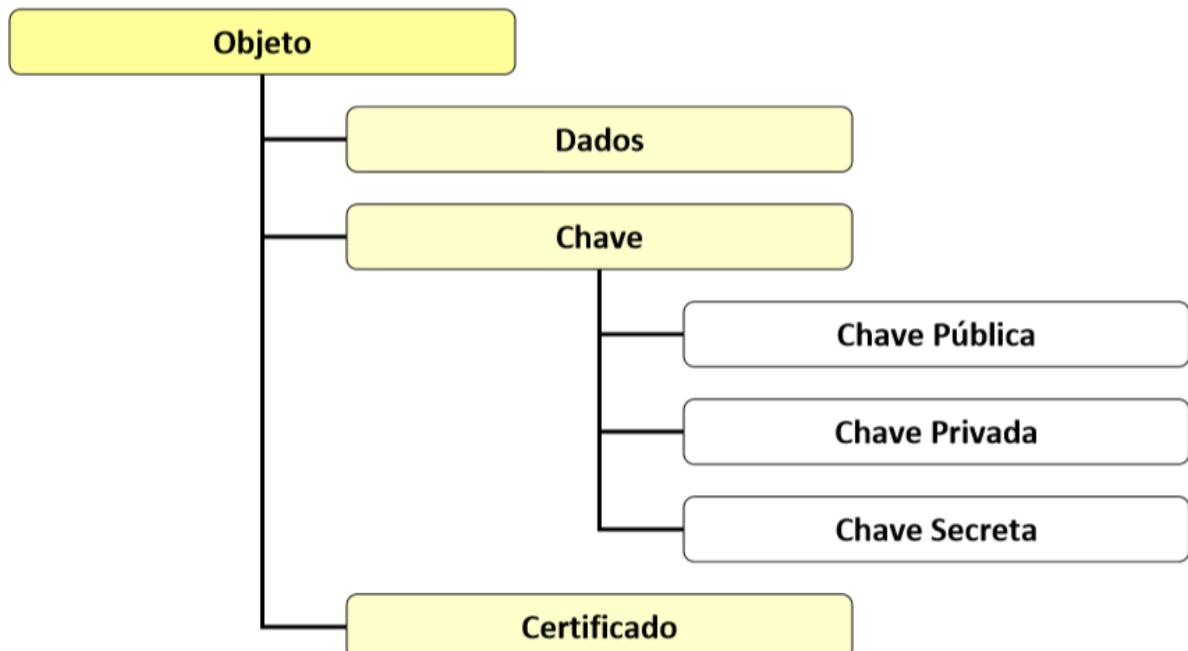
Baseados em soluções normalizadas:

- **PKCS#11**
 - Cryptographic Token Interface Standard
 - Cryptoki
- **PKCS#15**
 - Cryptographic Token Information Format Standard
- **CAPI CSP**
 - CryptoAPI Cryptographic Service Provider
- **PC/SC**
 - Personal Computer/SmartCard
 - Plataforma para acesso a smartcards em Windows/Linux

XIII. PKCS#11



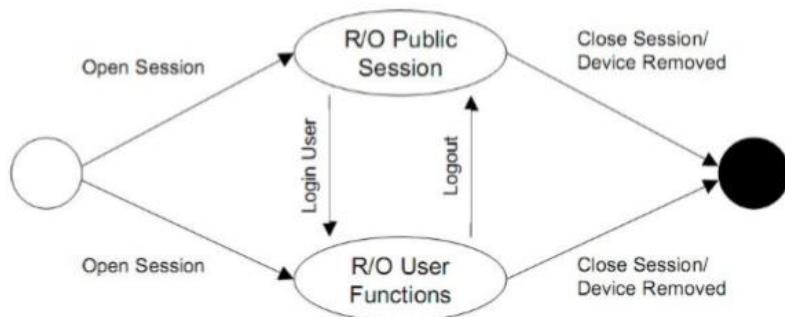
Img 13.1 – Integração do Middleware Cryptoki



Img 13.2 – Hierarquia de Objetos

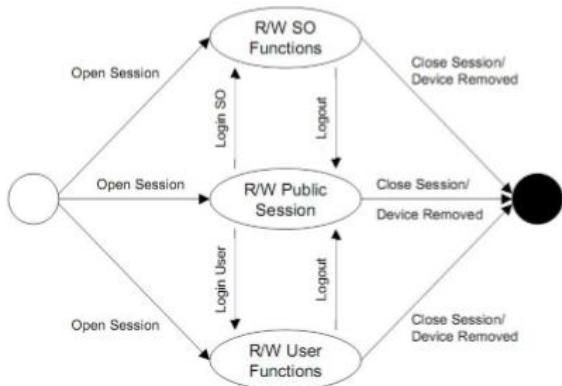
Sessões do Cryptoki

- **Ligações logicas entre aplicações e cartões (tokens)**
 - Sessões de Leitura
 - Sessões de Leitura e Escrita
- **Operações em Sessões Ativas**
 - Administrativas
 - Login/Logout
 - Gestão de Objetos
 - Criar ou destruir um objeto no cartão
 - Criptograficas
- **Objetos de sessão**
 - Objetos temporarios criados e validados durante a sessão
- **Tempo de Vida das Sessões**
 - Normalmente apenas para uma unica operação



- **Sessão pública de Leitura**
 - Acesso de leitura aos objetos públicos
 - Acesso de leitura/escrita aos objetos de sessão públicos
- **Funções de leitura do utilizador**
 - Acesso de leitura a todos os objetos do cartão (públicos ou privados)
 - Acesso de leitura/escrita a todos os objetos de sessão (públicos ou privados)

Img 13.3 – Cryptoki – Sessão de Leitura



- **Sessão pública e Leitura e Escrita**
 - Ler e escrever todos os objetos públicos
- **Funções do SO de Leitura e Escrita**
 - Ler/escrever objetos públicos
 - Não os objetos privados
 - O SO pode definir o PIN dos utilizadores
 - SO = Security Officer
- **Funções do utilizador de Leitura e Escrita**
 - Ler e escrever todos os objetos

Img 13.4 – Cryptoki – Sessão de Leitura e escrita

XIV. PKCS#11 – Conceitos usados pelo CC

PIN de Autenticação

- PIN do utilizador no PKCS#11

PIN de Assinatura

- Não exposto pelo interface PKCS#11

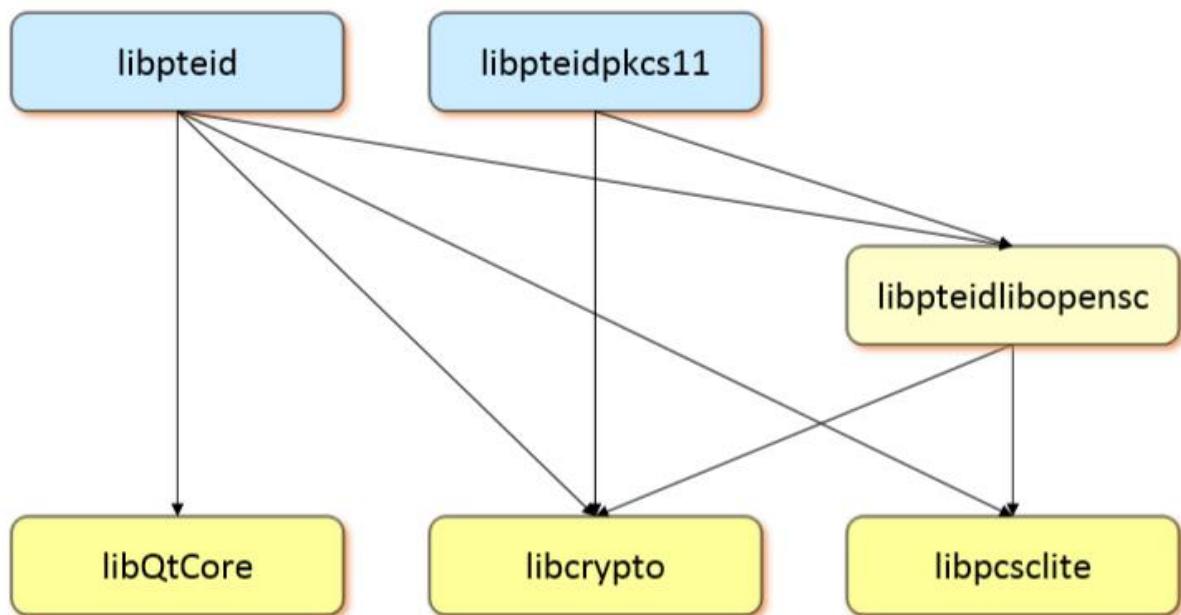
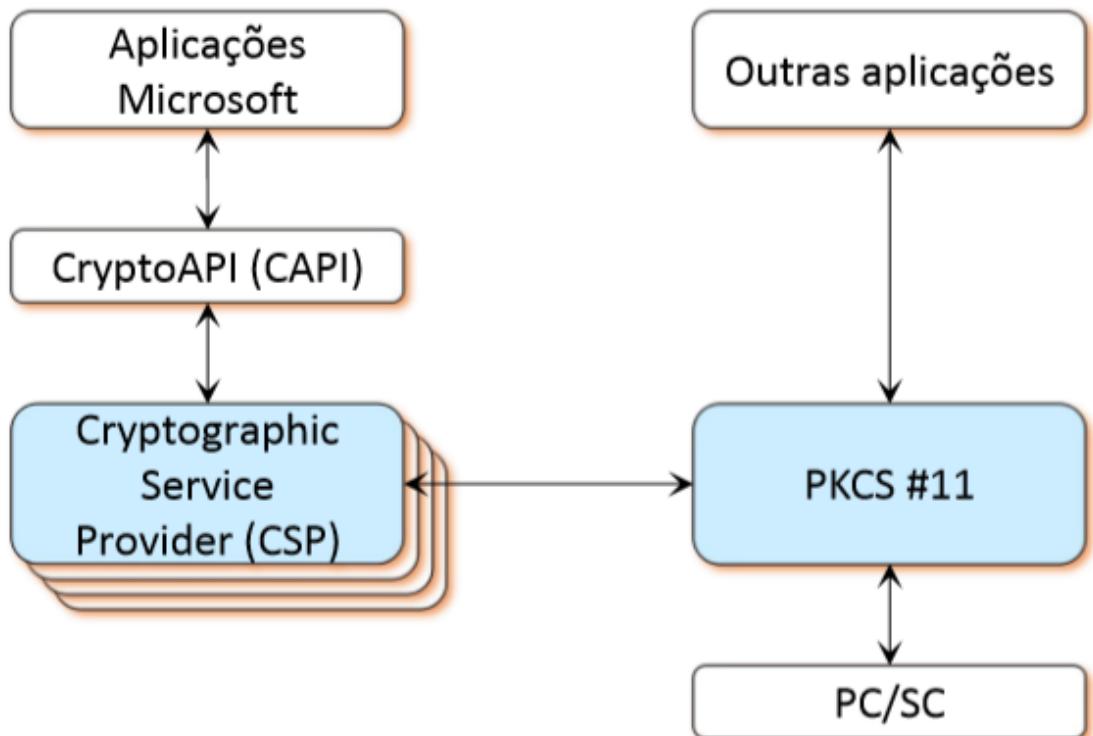
PIN de Morada

- Não exposto pelo interface PKCS#11
- 0000 por omissão nos cartões recentes

PKCS#11 SO PIN

- Não utilizado pelo titulare do cartão

XV. Middleware PTEID & SDK



Img 15.1 – PTEID Middleware para Windows vs para Unix

API Adicional para interagir com o CC

- Fornecida pela biblioteca libpteid.so

Permite acesso aos dado relativos ao cidadão

- Nome, fotografia, etc...

Objetos PTEID são armazenados como ficheiros

- 3f000003 – Trace
- 3f005f00ef02 – Citizen Data
- 3f005f00ef05 – Citizen Address Data (PIN Protected)
- 3f005f00ef06 – SOd (Security Object Data)
- 3f005f00ef07 – Citizen Notepad

XVI. Assinatura de Documentos

O CC permite a geração de assinaturas

Estas **podem ser inseridas em objetos** (e.g Email, Documentos PDF, ...)

Assinatura Digital substitui Assinatura Presencial

- Importante no contexto legal ou de Administração Pública
- Nativamente suportada em alguns formatos

Utiliza chave privada e Selo Temporal da PKI

- Selo Temporal é vital para garantir instante da assinatura

Demonstração da assinatura do CC
Assinado por : JOÃO PAULO SILVA BARRACA
Num. de Identificação: BI115785728
Data: 2019.11.03 23:53:38 +0000
Localização: Aveiro



Img 16.1 – Exemplo de Assinatura Digital

XVII. Autenticação com o CC

Envolve enviar um **NONCE** ao CC para ser cifrado com a chave privada

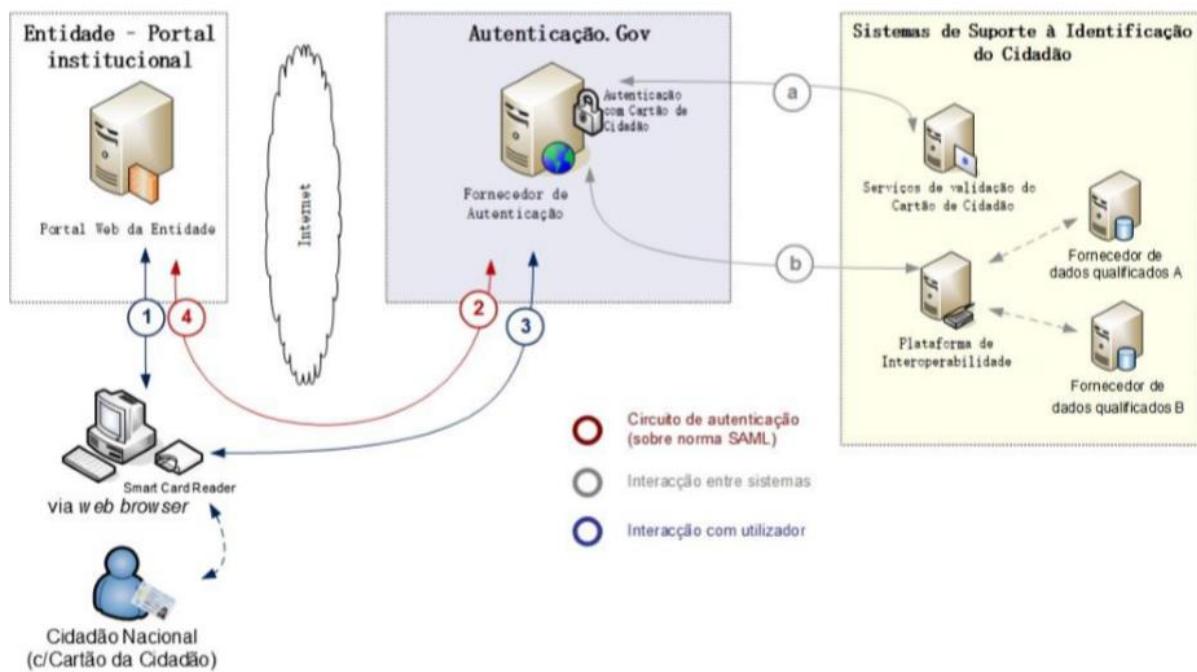
Problema: **Browsers não possuem acesso ao cartão**

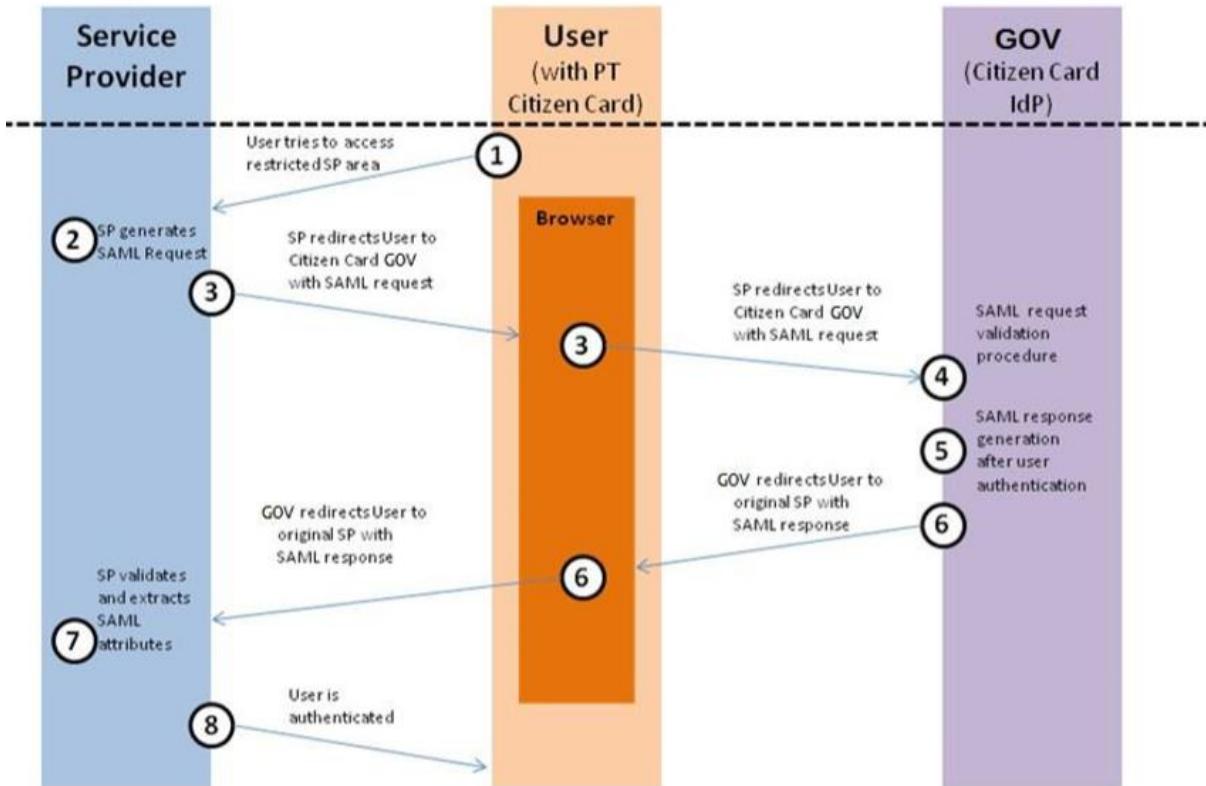
- Possível configurar libpteidpkcs11.so, mas só para acesso via API PKCS#11
- Possível usar um Java Applet – Obsoleto

Solução: Utilizar um plugin no computador do utente

- Expõe servidor web no localhost
- Permite acesso ao cartão através do servidor web
 - Apenas a pedidos autenticados pela infraestrutura do CC
- Necessita de aprovação previa para cada nova integração

XVIII. Plugin Autenticação.gov





Img 18.1 – Autenticação usando o plugin

XIX. Chave Móvel Digital – CMD

Objetivo: **Possibilitar a autenticação / assinatura mesmo sem o cc presente MAS com um nível de segurança “semelhante”**

Princípios de funcionamento:

- Necessite de um CC para autenticar o pedido de uma CMD
- Utentes podem autenticar-se/assinar documentos usando a CMD
- Não necessita de plugin instalado

- Não necessita de cartão para utilização futura
- Utiliza 2FA (2 factor authentication)
 - PIN no site + Código por outro canal (p.ex SMS)

Processo baseado na **criação de um par de chaves armazenado remotamente:**

1. **Cidadão usa CC para pedir uma CMD**
 - a. Especifica uma senha/PIN
 - b. Especifica um canal de autenticação
2. **É gerado um par de chaves**
3. **Chave publica enviada para geração de certificados**
4. **Chaves e certificado armazenados em ambiente seguro**
 - a. Protegido pela senha do utilizador
5. **Permitidas operações a quem validar a autenticidade**



Faça a sua autenticação com :

CARTÃO DE CIDADÃO

CHAVE MÓVEL DIGITAL

Universidade de Aveiro solicitou alguns dos seus dados para realizar o serviço *online* pretendido [i](#)

- Nome Próprio
- Nome Completo
- Nacionalidade
- Identificação Fiscal
- Identificação Civil

RECUSAR

AUTORIZAR

Img 19.1 – Autenticação usando uma Chave Móvel Digital