UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

SEGURANÇA, DESENHO E CONTROLO DE ACESSOS A CONTEÚDOS

Daniel Duarte Diogo de Abreu

Mestrado em Engenharia Informática

Especialização em Arquitectura, Sistemas e Redes de Computadores

2009

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática

SEGURANÇA, DESENHO E CONTROLO DE ACESSOS A CONTEÚDOS

Daniel Duarte Diogo de Abreu

ESTÁGIO

Trabalho orientado pelo Prof. Doutor Hans Peter Reiser
e co-orientado pelo Eng. Paulo José Simões Torres

Mestrado em Engenharia Informática

Especialização em Arquitectura, Sistemas e Redes de Computadores

2009

Resumo

Este relatório insere-se no âmbito do Projecto de Engenharia Informática com vista à obtenção do grau de Mestre em Engenharia Informática com especialização em Arquitectura, Sistemas e Redes de Computadores na Faculdade de Ciências da Universidade de Lisboa.

Em qualquer organização a informação é vital, essa informação que reside em bases de dados está muitas vezes sujeita ao acesso de técnicos cujo objectivo é o de manter o sistema funcional, mas que devido à posição que ocupam têm acesso a informação à qual, de outro modo, não teriam autorização para manipular.

A nível prático, este relatório tem por objectivo melhorar a segurança do sistema de gestão documental empresarial *e-doc**link***, ao nível da garantia da integridade e confidencialidade de objectos que circulam no sistema, tendo para tal, sido no decorrer deste estágio, analisada a aplicabilidade de várias ferramentas, entre elas o *Encrypting File System* (EFS) e *Trasparent Data Encryption* (TDE). Foram descritos sucintamente os algoritmos aplicados na solução em conjunto com a análise das necessidades de segurança do sistema, recorrendo a técnicas de modelação e classificação de ameaças, como *Attack Trees*, *STRIDE* e *DREAD*.

Foram propostas melhorias de segurança ao sistema, algumas foram implementadas no sistema como é o caso da Assinatura Digital com garantia de Não Repúdio, outras em protótipo, nomeadamente a Cifra e Assinatura Digital de conteúdos. Foi proposto um sistema de garantia de integridade de dados sensíveis de configuração assim como de reconhecimento de confiança em componentes de software, baseado em Assinaturas Digitais com *Elliptic Curve Digital Signature Algorithm* (ECDSA).

A solução encontrada procura alcançar um *trade-off* entre o impacto negativo no desempenho e usabilidade do sistema vs benefícios de segurança.

Palavras-Chave: Segurança, *e-doc**link***, Confidencialidade, Integridade, *Trade-off*

Abstract

This report was made within the scope of Project in Engineering Computer Science to acquire an Engineering Computer Science MSc degree with Systems Architecture and Computer Networks specialization at Faculty of Sciences of the University of Lisbon.

For any organization the information is vital, that information that is kept in databases is most of the time under the control of technicians whose goal is to keep the system up and running, but because of their job position have access to information that, otherwise, wouldn't be authorized to manipulate.

In practice, this work has the goal of improve the security of the enterprise document management system *e-doc**link***, by improving the integrity and confidentiality of the objects that flow through the system, for that matter, during the course of this stage, it was analyzed the applicability of some tools, among them the *Encrypting File System* (EFS) and *Transparent Data Encryption* (TDE). The algorithms used in the solution are briefly described; the analysis of the system security needs were also evaluated by using some threat modeling tools, such as, Attack Trees, STRIDE and DREAD.

Several security improvements were suggested, some of them were implemented in the system, such as the Digital Signature with No Repudiation guarantee, some in a prototype, such as the Encryption e Digital Signature of contents. It was proposed a scheme of integrity check for sensitive configuration data and system software components, using Digital Signatures with the Elliptic Curve Digital Signature Algorithm (ECDSA).

The solution found tries to reach a trade-off between the negative impact on performance and usability of the system vs. security benefits.

Keywords: Security, *e-doc**link***, Confidentiality, Integrity, *Trade-off*

Agradecimentos

Queria agradecer a todos os aqueles que me ajudaram durante a realização deste trabalho, principalmente aos orientadores, Prof. Dr. Hans Reiser e Eng.º Paulo Torres pelos valiosos contributos para a elaboração deste relatório, assim como a todos os colegas de trabalho, cuja colaboração foi fundamental.

Gostava também de agradecer a todos aqueles que me ajudaram durante o percurso académico que agora chega ao fim, em especial ao Dário, Fábio, Mafalda, Mónica, Sara, Renato e Tiago.

Last but not least, queria agradecer à minha família pelo apoio que sempre me deu. A todos, um muito obrigado.

"O que as vitórias têm de mau é que não são definitivas.
O que as derrotas têm de bom é que também não são definitivas."
José Saramago

Acrónimos

Acrónimo	***Descrição***
ACL	Access Control List
AES	Advanced Encryption Standard
BD	Base de Dados
CA	Certificate Authority
CNG	Cryptography Next Generation
DES	Data Encryption Standard
DRA	Data Recovery Agent
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ECRYPT	European Network of Excellence for Cryptology
EFS	Encrypting File System
FIPS	Federal Information Processing Standards
HTTPS	Hypertext Transfer Protocol Secure
IST	Information Society Technologies
ICT	Information & Communication Technologies
NESSIE	New European Schemes for Signatures, Integrity and Encryption
NIST	National Institute of Standards and Technology
NTFS	New Technology File System
PGP	Pretty Good Privacy
PKCS	Public-Key Cryptography Standards
RNG	Random Number Generator
RSA	Rivest Shamir Adleman
SGBD	Sistema de Gestão de Base de Dados
SHA	Secure Hash Algorithm
SO	Sistema Operativo
TDE	Transparent Data Encryption
XML	Extensible Markup Language

Glossário

A

Access Control List – Lista com permissões de acesso a um objecto.

Advanced Encryption Standard – É um *standard* de cifra simétrica para instituições públicas norte-americanas.

B

Base de Dados – Colecção estruturada de dados residente num sistema computacional.

C

Certificate Authority – Entidade emissora de certificados digitais para uso de terceiros.

Cryptography Next Generation – Interface de Programação de Aplicações, fornece mecanismos de segurança para aplicações baseadas em Windows.

D

Data Encryption Standard - É um *standard* de cifra simétrica, predecessor do AES.

Data Recovery Agent – Entidade lógica correspondente a um par de chaves assimétricas.

Digital Signature Algorithm – Algoritmo de cifra assimétrica que permite gerar assinaturas digitais.

E

Elliptic Curve Digital Signature Algorithm – Algoritmos de cifra assimétrica baseada em curvas elípticas, que permite gerar assinaturas digitais.

Encrypting File System – Tecnologia que permite a cifra de ficheiros em sistemas Windows, se residente em sistema de ficheiros NTFS.

European Network of Excellence for Cryptology – Programa colaborativo de investigadores da União Europeia na área da segurança da informação, especialmente criptografia e digital *watermarking*.

F

Federal Information Processing Standards – Conjunto de standards adoptados pelo NIST de forma a regular a utilização de algoritmos criptográficos em instituições públicas nos Estados Unidos.

H

Hypertext Transfer Protocol Secure - Protocolo utilizado na World Wide Web que permite transferências de conteúdo de forma segura.

I

Information Society Technologies – Programa de pesquisa Europeu na área das tecnologias de informação entre 2002 e 2006.

Information and Communication Technologies - Programa de pesquisa Europeu na área das tecnologias de informação entre 2007 e 2013.

Insider – Colaborador de uma organização que tem acesso a informação privada/confidencial da empresa.

M

Microsoft Office – Plataforma para escritório que contem diversas aplicações, servidores e serviços.

N

New European Schemes for Signatures, Integrity and Encryption – Projecto de pesquisa Europeu entre 2000 e 2003 com objectivo de identificar primitivas criptográficas seguras.

National Institute of Standards and Technology - Instituto norte-americano de standards e tecnologia.

P

Pretty Good Privacy – Software que visa garantir segurança e privacidade a conteúdos.

Public-Key Cryptography Standards – Standards de criptografia assimétrica produzidos pela RSA Laboratories.

R

Random Number Generator – Gerador de números aleatórios.

Rivest Shamir Adleman – Algoritmo de criptografia assimétrica composto pelos nomes dos seus criadores.

S

Sistema de Gestão de Base de Dados – Conjunto de *softwares* responsáveis por gerir uma base de dados.

Secure Hash Algorithm – Conjunto de algoritmos que produzem sínteses criptográficas[1].

Sistema Operativo – Interface entre o hardware e o utilizador, responsável por gerir os recursos de hardware e software da máquina.

U

User-Friendly – Aplicação ou interface de fácil utilização.

T

Transparent Data Encryption – Tecnologia que permite a cifra de uma base de dados, pode ou não ser fornecida pelo SGBD.

[1] Consideradas de síntese criptográfica porque asseguram 3 propriedades que as tornam seguras para utilização num contexto de criptografia. Ver 2.4.3

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1 - Introdução

Este relatório insere-se no âmbito da cadeira de Projecto de Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa. A realização deste trabalho iniciou-se dia 1 de Outubro de 2008 na empresa *Link Consulting SA*, e o seu programa enquadra-se na área da segurança de conteúdos, aplicada ao sistema *e-doc**link***.

1.1 Motivação

Em qualquer organização a informação é vital, essa informação guardada sob a forma de dados em bases de dados está muitas vezes sujeita ao acesso de técnicos cujo objectivo é o de manter o sistema a funcionar, mas que devido à posição que ocupam tem acesso a informação privilegiada. Apesar dos SGBD – "Sistema de Gestão de Base de Dados" actuais já disporem de métodos de protecção desses dados, esses técnicos dispõem dos meios para os anular, assim como de privilégios que permitem ocultar as suas acções.

Sendo o *e-doc**link*** um sistema de gestão documental utilizado para os mais diversos fins e nas mais diversas áreas, a necessidade de proteger a informação nele contida sempre foi uma área fundamental no seu planeamento e construção. De forma a evoluir os mecanismos existentes de garantia de segurança de dados, é pretendido contemplar diversas áreas e tecnologias adequadas a este fim. Uma vertente de particular importância a contemplar é a situação dos técnicos a quem determinada informação deve ser vedada por variados motivos, tais como informações confidenciais. Relativamente a este ponto, o objectivo é aumentar a segurança do sistema, não na perspectiva habitual (utilizadores mal intencionados) mas no sentido de ocultar a informação das pessoas que gerem o sistema subjacente ao *e-doc**link***, tais como administradores de sistemas.

1.2 Organização do documento

Este documento está organizado da seguinte forma:

- Capitulo 2 – Problema
- Capítulo 3 – Desenvolvimento
- Capítulo 4 – Resultados
- Capitulo 5 – Discussão dos Resultados
- Capitulo 6 – Conclusão
- Capitulo 7 – Trabalho Futuro
- Referências
- Índice Remissivo
- Anexos

1.3 Instituição de Acolhimento

A *Link Consulting* nasceu no final de 1999, constituindo uma empresa de características inovadoras, que congrega cerca de duas centenas de colaboradores altamente qualificados e motivados, com um amplo conhecimento do mercado e das empresas. Alia uma experiência de 10 anos de actividade em algumas das maiores organizações nacionais e internacionais à investigação e constante aposta na inovação, posiciona-se em termos de mercado, como uma empresa de Consultoria, Integração de Sistemas, Desenvolvimento de Soluções e Prestação de Serviços, actuando na área dos modernos Sistemas de Informação, Negócio Electrónico e Economia Digital.

1.4 Equipa

A integração foi feita na unidade de Administração Pública, da *Link Consulting*. O sistema *e-doc**link***, ao qual está associado o presente projecto, é um produto desenvolvido, comercializado e mantido por uma equipa de 14 elementos, totalmente pertencentes a esta unidade.

1.5 Contexto

O sistema *e-doc**link*** é uma solução integrada de gestão documental e de suporte a processos de decisão desenvolvida em tecnologia *Microsoft .Net* para ambiente *Web*, possibilitando assim o fácil acesso a partir de qualquer equipamento que possua um browser de Internet. A sua implementação abrange a globalidade duma Organização, o que se traduz num importante ganho de eficiência, resultante da possibilidade de todos os colaboradores comunicarem entre si e terem capacidade de aceder, dentro dos acessos que forem atribuídos, a toda a informação gerida.

O sistema *e-doc**link*** é utilizado num conjunto significativo de organizações nacionais, pertencentes à Administração Pública e outros sectores de actividade. É um sistema que já conta com mais de 10.000 utilizadores em todo o país.

1.6 Metodologia

A metodologia que foi seguida assenta em 4 pontos fundamentais:

O primeiro é a análise dos requisitos, isto é, que segurança existe actualmente no sistema, o que pode ser feito para melhorar essa segurança e que ferramentas existem à disposição para atingir o fim pretendido.

O segundo ponto consiste na análise dos dados obtidos no ponto anterior e avaliar a sua aplicabilidade concreta ao sistema.

O terceiro ponto consiste na implementação das ferramentas, técnicas e mecanismos que, se verificou serem aplicáveis, e que são capazes de trazer uma mais-valia à segurança do sistema.

O quarto e último ponto consiste na discussão sobre a segurança adicional trazida ao sistema pela implementação das medidas de segurança, assim como a reflexão sobre o que poderá e deverá ser ainda melhorado.

A metodologia de desenvolvimento do sistema *e-doc**link*** no qual este estágio se enquadra segue um modelo de desenvolvimento em espiral, uma vez que cada caso necessita ser contemplado especificamente e os requisitos têm de ser revistos a cada etapa. Posteriormente serão implementadas as funcionalidades que se considerem relevantes e que não sacrifiquem o sistema em benefício da segurança (terceiro ponto). Finalmente serão discutidos os resultados obtidos assim como os pontos a melhorar posteriormente (quarto ponto).

1.7 Objectivos

Analisar e caracterizar ameaças potenciais à adulteração de conteúdos em sistemas de informação, de forma a definir e implementar práticas e tecnologias que defendam os próprios conteúdos de acessos indevidos, principalmente provenientes de utilizadores com privilégio de administração desses sistemas.

Este relatório pretende dar resposta às seguintes questões:

Questão 1: Como e onde proteger os conteúdos?

Questão 2: Como garantir não repúdio de conteúdos?

Questão 3: Como garantir a integridade de conteúdos?

Questão 4: Como garantir a integridade de dados sensíveis de configuração?

Questão 5: Como garantir segurança em documentos *Microsoft Office*?

Questão 6: Como verificar integridade de componentes de software do sistema?

Questão 7: Como garantir segurança em trocas de conteúdos entre organizações?

Em geral a questão principal é como defender os conteúdos presentes no sistema de ataques de *insiders*.

<table>
<tr><td rowspan="3">Tipo de Conteúdos do Sistema</td><td colspan="3">Conteúdos ou Dados</td></tr>
<tr><td>Objectos do sistema</td><td colspan="2">Documentos</td><td rowspan="2">Dados sensíveis de configuração</td></tr>
<tr><td>Metadados</td><td>Documentos Office</td><td>Outros</td></tr>
<tr><td>Local de Armazenamento</td><td>Base de Dados</td><td colspan="2">Repositório de Documentos</td><td>Servidor Aplicacional</td></tr>
</table>

Tabela 1 – Tipos de conteúdos e local de armazenamento

1.7.1 Plano de trabalho inicial

Plano de Actividades	Calendarização
Fase 1 – Investigação de base e Ensaios Técnicos <u>Análise e avaliação técnica de mecanismos de garantia de segurança no armazenamento de dados</u> ➔ Objectivos práticos: a) Metadados confidenciais em bases de dados b) Garantia de autenticidade e não repúdio de dados estruturados c) Armazenamento de documentos – confidencialidade/protecção de acesso ➔ Tecnologias, práticas e normas a analisar: a) Práticas e técnicas mais adequadas no planeamento e implementação de SGBDs b) Segurança e meios de cifra no armazenamento de ficheiros c) Evolução de mecanismos existentes em ambientes de produção, em linha com estas directrizes d) Tipos e algoritmos de criptografia e) Estudo de meios de defesa e ataque e sua tipificação f) Verificação de integridade	**2008** Outubro Novembro Dezembro Janeiro

Fase 2 – Implementação sobre *e-doclink*** <u>Garantia de não adulteração de conteúdos aplicada à plataforma *e-doc**link***</u> a) Assinatura digital de conteúdos <u>Planeamento e validações de segurança aplicacional</u> a) Protecção de dados sensíveis de configuração b) Mecanismos de cifra de dados c) Mecanismos de segurança e reconhecimento de confiança em componentes <u>Planeamento e validações de segurança na produção de documentos</u> a) Produção de documentos com mecanismos de autenticação e segurança (ambientes Microsoft / Office) b) Evolução de mecanismos de armazenamento e gestão de ficheiros	**2008** Novembro Dezembro **2009** Janeiro Fevereiro Março Abril
Fase 3 – Segurança inter-aplicacional e inter-organizacional a) Mecanismos de garantia de segurança na intercomunicabilidade de dados b) Garantia de autenticidade e não repúdio na troca de documentos entre organizações c) Análise de standard UBL 2.0 em associação aos mecanismos de intercâmbio seguro de informação (dados e/ou documentos) inter-organizações e inter-aplicações	**2009** Março Abril Maio
Fase 4 – Elaboração do Relatório Final de Estágio	**2009** Junho

Tabela 2 – Planeamento Inicial

Capítulo 2 - Problema

2.1 Introdução

Os repositórios de dados são pontos vitais em qualquer organização. Lá encontra-se armazenada a informação que é necessária para o funcionamento da organização. Os técnicos responsáveis por estes repositórios têm devido à relevância do seu posto, acesso a essa informação, tornando-se eles próprios fundamentais para a organização, uma vez que são eles que cuidam para que o sistema funcione. A informação à qual têm acesso está muitas vezes acima do seu nível de autorização, juntamente com o seu poder de manipular esses dados, torna-os assim potenciais pontos de falha na segurança da organização.

No sentido de dar resposta a este e outros problemas de segurança, têm sido propostas melhorias pela *Common Criteria* (CC) (1), que avalia a segurança com base na norma ISO-15408, estabelecendo um nível de confiança dos produtos submetidos a análise. A *Payment Card Industry Data Security Standard* (PCI-DSS) (2), estabelece um conjunto de regras para aumentar a segurança dos dados de transacções bancárias. As normas *ISO*-17799 e ISO-15408 da *International Organization for Standardization* (ISO) (3) estabelece directivas e princípios gerais para melhorar a segurança dos sistemas de informação duma organização. Regras para certos sectores de actividade, como por exemplo, no caso dos Estados Unidos, a manipulação de dados sobre Saúde está sujeita às regras impostas pelo *Health Insurance Portability and Accountability Act* (HIPAA) (4).

No sentido de assegurar a protecção da informação de acessos indevidos de utilizadores privilegiados do sistema, a *Oracle* produziu o *Oracle Database Vault* (5), um produto que visa dar resposta a estas preocupações e respeitar as várias normas recomendadas/impostas por diversas organizações e leis, entre elas, a PCI-DSS, *Sarbanes-Oxley* (SOX) e a HIPAA.

2.2 Noções

Para ajudar a encontrar soluções para este problema, serão apresentadas, resumidamente, um conjunto de noções que importa ter presentes.

2.2.1 Segurança da Informação

A segurança da informação pode ser expressa através de quatro propriedades fundamentais (6):

- **Autenticidade**
- **Confidencialidade**
- **Integridade**
- **Disponibilidade**

Seja A o emissor e B o destinatário,

A propriedade de **Autenticidade** expressa a medida pela qual um serviço ou excerto de informação é genuíno e protegido contra possível forja, isto é, o que B recebe foi enviado por A, que não o pode negar.

A propriedade de **Confidencialidade** expressa a medida pela qual um serviço ou excerto de informação está protegido contra uma revelação não autorizada, isto é, apenas B lê o que A mandou.

A propriedade de **Integridade** expressa a medida pela qual um serviço ou excerto de informação está protegido contra modificações ilegais/indetectáveis, isto é, o que A enviou não pode ser alterado sem ser detectado.

A propriedade de **Disponibilidade** expressa a medida pela qual um serviço ou excerto de informação está protegida contra a impossibilidade de acesso por pessoas autorizadas.

É importante ter em conta estas 4 propriedades, pois são a base que qualquer sistema que pretenda ser seguro deve seguir. Como se verá posteriormente, um dos objectivos deste trabalho é tentar garantir as três primeiras propriedades sem pôr em causa a quarta. Na verdade o que acontece na maior parte dos casos é um *trade-off* entre manter um nível aceitável de segurança sem reduzir o sistema a algo inutilizável. O maior ênfase neste relatório será dado à segunda propriedade, uma vez que é aquela que neste momento tem menos peso no sistema.

Diferentes autores consideram propriedades adicionais (7), tais como

- **Não Repúdio**
- **Autorização**

A propriedade de **Não repúdio** expressa a medida pela qual um utilizador não pode negar que tenha executado uma acção.

A propriedade de **Autorização** expressa a medida pela qual um utilizador está expressamente autorizado ou impedido de aceder a um recurso.

2.2.2 Princípios de Desenho de Saltzer e Schroeder (8)

Apesar da data de publicação remontar a 1975, os princípios de desenho descritos por Saltzer e Schroeder continuam actuais e qualquer aplicação deve segui-los, de modo a minimizar as vulnerabilidades.

- **Mecanismos simples e pequenos**

Quanto maiores e mais complexos forem os mecanismos maior será também a possibilidade de uma vulnerabilidade passar despercebida.

- **Basear os acessos em *Whitelists***

Existem 4 atitudes em relação à definição de políticas de acesso, entre elas temos: Paranóica (ninguém têm acesso), Prudente (alguns têm acesso), Permissiva (alguns não têm acesso) e Promíscua (todos têm acesso). Segundo Saltzer e Schroeder a politica de prudência é a mais adequada.

- **Mediação completa**

Este princípio de desenho expressa a necessidade de validar sempre a permissão de acesso a objectos, isto é, não devem existir mecanismos para contornar ou evitar essa validação.

- **Desenho público**

A segurança não deve ser atingida através da obscuridade, o código deve poder ser conhecido sem que isso ponha em causa a segurança do sistema. Apostar em tal sistema só irá atrasar o problema.

- **Separação de privilégios**

Num sistema não deve existir uma entidade com poder total, os poderes devem estar divididos pelo maior número de pessoas possível, de forma a minimizar os riscos no caso de uma entidade ser comprometida, limitando assim os eventuais danos que se possam causar.

- **Menor privilégio**

Qualquer aplicação e utilizador devem utilizar apenas o mínimo necessário para executar uma qualquer operação, de forma a minimizar os danos em caso de erro ou ataque.

- **Mínimo de mecanismos em comum**

O uso de recursos partilhados deve ser limitado ao mínimo necessário, de forma a limitar danos causado por mau uso desses recursos.

- **Adequação Psicológica**

Este último principio expressa a necessidade de manter o sistema utilizável para que o utilizador não tente boicotar ou ultrapassar mecanismos de protecção pouco *user-friendly*, caso contrário o utilizador poderá mesmo boicotar ou abandonar o sistema.

2.2.3 Perspectiva Microsoft

- **SD3 + C** (8) **e PD3 + C** (9)

Estes são princípios aplicados pela Microsoft nos seus produtos, contudo são princípios genéricos e aplicáveis a outras aplicações:

SD3 + C

- *Secure by Design*
- *Secure by Default*
- *Secure in Deployment*
- *Communications*

PD3 + C

- *Privacy by Desing*
- *Privacy by Default*
- *Privacy in Deployment*
- *Communications*

Estes princípios expressam a necessidade de planear a segurança e a privacidade durante o ciclo de vida do produto, durante a fase de desenho com um planeamento e modelação adequadas. *Out-of-the-box*, deve ser seguro e manter a privacidade por omissão, e sem qualquer configuração adicional. Deve ser gerível e actualizável quando em funcionamento, garantindo a segurança e a privacidade durante o processo. As comunicações relacionadas com a aplicação devem sempre que possível garantir a segurança e privacidade dos dados durante a comunicação.

- **STRIDE e DREAD** (10)

STRIDE é um processo desenvolvido pela *Microsoft Application Consulting and Engineering Team*, representa os vários meios pelos quais um adversário pode comprometer o sistema:

Threat	Security Property
Spoofing	Authentication
Tampering	Integrity
Repudiation	Non-repudiation
Information disclosure	Confidentiality
Denial of service	Availability
Elevation of privilege	Authorization

Tabela 3 – STRIDE, Ameaças e Propriedades de Segurança

- *Spoofing*: Com este ataque o adversário toma a identidade de um utilizador legítimo do sistema.
- *Tampering*: Com este ataque o adversário altera dados do sistema.
- *Repudiation*: Com este ataque o adversário altera dados do sistema de forma indetectável, isto é, não há forma de identificar inequivocamente o autor.
- *Information Disclosure*: Com este ataque o adversário tem acesso a dados, aos quais não deveria ter, por não possuir autorização.
- *Denial of Service*: Com este ataque o adversário inutiliza o sistema temporária ou permanentemente de tal forma que utilizadores legítimos ficam sem acesso.
- *Elevation of Privilege*: Com este ataque o adversário aumenta o seu nível de permissões de acesso no sistema, facilitando a execução dos ataques anteriores.

Threat	Countermeasures
Spoofing user identity	Use strong authentication. Do not store secrets (for example, passwords) in plaintext. Do not pass credentials in plaintext over the wire. Protect authentication cookies with Secure Sockets Layer (SSL).
Tampering with data	Use data hashing and signing. Use digital signatures. Use strong authorization. Use tamper-resistant protocols across communication links. Secure communication links with protocols that provide message integrity.
Repudiation	Create secure audit trails. Use digital signatures.
Information disclosure	Use strong authorization. Use strong encryption. Secure communication links with protocols that provide message confidentiality. Do not store secrets (for example, passwords) in plaintext.
Denial of service	Use resource and bandwidth throttling techniques. Validate and filter input.
Elevation of privilege	Follow the principle of least privilege and use least privileged service accounts to run processes and access resources.

Tabela 4 – Ameaças e Mecanismos de Segurança

DREAD é um processo que permite quantificar, comparar e prioritizar os riscos da exploração de vulnerabilidades. Define 5 características:

- *Damage Potential*: Classifica numericamente o dano causado pelo atacante caso o ataque seja bem sucedido.
  - 0 = Nenhum
  - 5 = Apenas dados individuais são afectados
  - 10 = Destruição completa dos dados do sistema
- *Reproducibility*: Classifica numericamente a facilidade com que o ataque pode ser replicado.
  - 0 = Muito difícil ou impossível, mesmo para administradores da aplicação.
  - 5 = São necessários poucos passos, pode ser necessário autorização de acesso ao sistema.
  - 10 = Apenas um browser, sem necessidade de autorização de acesso ao sistema.
  -

- *Exploitability*: Classifica numericamente a dificuldade de execução do ataque.
    - 0 = Conhecimento avançado de redes e programação, com necessidade de ferramentas feitas à medida para explorar a falha.
    - 5 = Ferramentas para explorar a falha já estão disponíveis na Web.
    - 10 = Apenas um browser.
- *Affected Users*: Classifica numericamente a quantidade de utilizadores afectados, caso o ataque seja bem sucedido.
    - 0 = Nenhum
    - 5 = Alguns, mas não todos
    - 10 = Todos
- *Discoverability*: Classifica numericamente a facilidade e rapidez em descobrir a existência da vulnerabilidade.
    - 0 = Muito difícil, requer acesso ao código fonte e privilégios de administração
    - 5 = Pode ser detectada com monitorização de actividade.
    - 9 = Detalhes de vulnerabilidades como esta são de domínio público e podem ser descobertas apenas utilizando um motor de busca.
    - 10 = A vulnerabilidade é visível através do browser.

Embora seja bastante subjectivo, o DREAD é um sistema que permite ter uma ideia do impacto que um ataque bem sucedido terá no sistema. O algoritmo utilizado para comparar riscos neste modelo é dado pela seguinte equação: Risco = (DAMAGE + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5

2.2.4 Os 6 princípios de Kerckhoff

Apesar de antigos, os 6 princípios propostos por Auguste Kerckhoff para comunicações militares podem aplicar-se ainda hoje. Um dos seus princípios mais conhecido, nomeadamente o 2º que diz resumidamente que "a única coisa que deve ser secreta num sistema deve ser a chave" (12) (13), ou seja, a segurança deve residir na chave e não no algoritmo de cifra, podendo o ultimo ser conhecido sem pôr em causa a segurança. Expressa que a segurança não deve ser alcançada através do secretismo do algoritmo.

2.3 Algoritmos

2.3.1 AES

Advanced Encryption Standard, é a designação do padrão utilizado pelo governo dos Estados Unidos para cifras simétricas que veio substituir o antigo DES, após um escrutínio promovido pelo NIST saiu eleito de entre 5 finalistas o algoritmo Rijndael (14). Apesar de já terem sido analisados diversos ataques (15) (16) (17) a este algoritmo, o mesmo continua a ser seguro para utilização até prova em contrário. No contexto de utilização fornece confidencialidade aos dados que protege. A chave utilizada neste algoritmo deve ser protegida, sendo que para tal deve ser garantida a sua confidencialidade e integridade.

2.3.2 RSA

É um algoritmo de criptografia assimétrica (18), que permite não só cifrar e decifrar como também produzir assinaturas digitais. Mesmo após longos anos sobre a publicação do trabalho acima referido, continua a ser o algoritmo mais utilizado para cifra assimétrica. Existem mais algumas possibilidades, entre elas a mais conhecida é o ElGamal (19) usado pelo PGP.

Sendo um algoritmo de criptografia assimétrica, as suas chaves são bastante longas e as suas operações demoradas, portanto, a utilidade deste algoritmo é direccionada para quantidades pequenas de dados. No contexto de utilização fornece confidencialidade dos dados que protege.

2.3.3 SHA

Secure Hash Algorithm, é uma família de algoritmos de *hash*, isto é, funções de síntese, a partir de um input de tamanho variável produzem um output de tamanho fixo,

por exemplo, o SHA-1 produz um output de 160 bits, enquanto o SHA-512 produz um output fixo de 512 bits. As funções criptográficas de síntese devem garantir 3 propriedades fundamentais (20):

- Dado hm deve ser "impossível" recuperar m tal que hm = H(m) (*Pre-image Resistance*)
- Dado M deve ser "impossível" encontrar uma M' tal que H(M) = H(M') (Second *Pre-image Resistance*)
- Deve ser "impossível" encontrar um par (M,M') tal que H(M)=H(M') (*Collision Resistance*)

2.3.4 RNG

Random Number Generator, este algoritmo permite obter números de forma aleatória, são bastante úteis em criptografia, introduzindo incerteza e aleatoriedade. Existem dois métodos de gerar números aleatórios, geradores pseudo-aleatórios (PRNG) que se baseiam puramente em computações deterministas e os geradores verdadeiramente aleatórios (TRNG) que não se baseiam apenas em computações deterministas mas também em fenómenos físicos que se esperam aleatórios. Os *cryptographically secure pseudorandom number generators* (CSPRNG), são utilizados para fornecer números aleatórios de forma mais "segura" do que usando apenas os PNRG.

2.3.5 ECDSA

É uma variante da criptografia assimétrica, baseada em curvas elípticas (21) (22) reduz de modo significativo o tamanho das chaves em relação a outros algoritmos, tais como o RSA, fornecendo um nível de segurança equivalente com chaves de dimensão inferior. A partir do SO Windows Vista e com a introdução do módulo *Cyptographic*

Next Generation (CNG) está disponível a opção de utilizar o algoritmo ECDSA, isto é, utilizar o algoritmo DSA utilizando chaves de curva elíptica para produzir assinaturas.

2.3.6 Algoritmos – Notas Finais

Existe uma grande variedade de algoritmos à disposição para atingir o mesmo objectivo, alguns mais seguros, outros mais rápidos, outros que tentam alcançar o meio-termo. Algumas instituições como o *National Institute of Standards and Technology* (NIST) e os projectos europeus *Information Society Technologies* (IST) e *Information & Communication Technologies* (ICT) dedicam-se entre outras coisas a estudar e a analisar estes algoritmos, produzindo recomendações sobre a sua utilização, tais como o FIPS, NESSIE e ECRYPT (23).

Bits of security	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
80	2TDEA[19]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

Tabela 5 – Equivalência de Forças Criptográficas de Algoritmos

No quadro acima podemos verificar que para atingir uma "força criptográfica" equivalente ao AES de 256 bits é necessário um tamanho de 15360 bits no caso das chaves RSA e DSA, enquanto o tamanho da chave do ECDSA com 512 bits é equiparável ao AES-256.

Algorithm security lifetimes	Symmetric key algorithms (Encryption & MAC)	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
Through 2010 (min. of 80 bits of strength)	2TDEA[23] 3TDEA AES-128 AES-192 AES-256	Min.: $L = 1024$; $N = 160$	Min.: $k=1024$	Min.: $f=160$
Through 2030 (min. of 112 bits of strength)	3TDEA AES-128 AES-192 AES-256	Min.: $L = 2048$ $N = 224$	Min.: $k=2048$	Min.: $f=224$
Beyond 2030 (min. of 128 bits of strength)	AES-128 AES-192 AES-256	Min.: $L = 3072$ $N = 256$	Min.: $k=3072$	Min.: $f=256$

Tabela 6 – Tempo de vida útil de alguns Algoritmos

No quadro anterior podemos verificar que é expectável que algoritmo *Triple Data Encryption* (3TDEA) dê garantias de segurança até 2030. E que até 2030 é expectável que as chaves do RSA de 2048 bits sejam suficientes.

Os quadros acima referidos foram retirados de recomendações do NIST (24), apesar do acima referido, podem ser encontrados ataques e produzidas novas recomendações com base em novos ataques aos algoritmos que tornem inválidas algumas da recomendações feitas.

2.4 Tecnologias – Protecção de dados

2.4.1 Oracle Database Vault (5)

Esta ferramenta tem precisamente o objectivo de defender os conteúdos da base de dados dos utilizadores com privilégios de administração. Contudo a componente

prática deste trabalho não permite a utilização desta ferramenta, já que o SGBD do sistema é o SQL Server, concorrente da Oracle.

2.4.2 Encrypting File System (25)

Esta tecnologia surgiu no SO *Microsoft Windows 2000* com o intuito de proteger ficheiros e directórios confidenciais, de forma fácil para o utilizador. A principal vantagem do EFS é que funciona independentemente do Sistema Operativo estar ou não a correr, uma vez que os dados cifrados são guardados dessa forma no sistema de ficheiros.

O EFS utiliza um sistema de criptografia híbrida, isto é, uma chave secreta é usada para cifrar os ficheiros confidenciais e posteriormente é utilizada uma chave pública para cifrar a chave secreta anteriormente usada. Para abrir um documento cifrado utilizando este sistema, basta utilizar a chave privada para decifrar a chave secreta cifrada anteriormente pela chave pública. A chave privada é protegida pela *palavra-chave* da conta do utilizador.

Este sistema sofria originalmente de um "problema" de segurança: a obrigatoriedade de existência de um *Data Recovery Agent* (DRA) que, por omissão, é a conta de administrador do *PC* ou do domínio em que este se encontra. Contudo, a partir da introdução do SO *Microsoft Windows XP* esta falha foi corrigida e não há necessidade de existência obrigatória de um *DRA*.

2.4.3 Transparent Data Encryption (26)

Esta tecnologia surgiu com o SGBD *Microsoft SQL Server 2008*, tendo um objectivo similar ao EFS mas funcionando a um nível diferente: ao nível da base de dados. Esta ferramenta permite cifrar a base de dados que se pretenda, de modo a que caso o respectivo ficheiro da base de dados seja comprometido, o atacante não possa aceder ao conteúdo da mesma. À semelhança do EFS, também o TDE usa um sistema de criptografia híbrida, protegendo uma chave secreta *Database Encryption Key* (DEK) com uma chave pública, que apenas pode ser recuperada com a chave privada correspondente. Esta ultima, é também ela protegida por uma chave secreta, a *Database*

Master Key (DMK), que por sua vez é protegida por outra chave simétrica criada na altura da instalação do servidor, a *Service Master Key* (SMK).

2.4.4 Tecnologias – Notas Finais

Estas tecnologias têm por objectivo facilitar e automatizar diversos procedimentos que de outra forma necessitariam de um maior esforço a nível aplicacional. Foram analisadas, com ensaios práticos, diversas tecnologias durante o decurso do estágio, com destaque para o sistema de protecção de ficheiros EFS e o sistema de cifra transparente TDE. Deve também ser feita uma menção ao *BitLocker Drive Encryption* (BDE) (27), que se verificou não ser apropriado à aplicação prática pretendida, uma vez que a sua utilização é transparente para os utilizadores com quaisquer permissões de acesso ao sistema.

Capítulo 3 – Desenvolvimento

3.1 Introdução

Esta secção tem por objectivo apresentar a implementação prática sobre o sistema de gestão documental empresarial *e-doc**link***, recorrendo a vários mecanismos e ferramentas descritos anteriormente.

É feita a análise dos requisitos funcionais e não funcionais que guiaram a construção de um protótipo funcional, o levantamento dos requisitos foi feito ao longo de vários meses, foram produzidos requisitos funcionais com o objectivo de sistematizar quais as novas funcionalidades que o protótipo deve garantir e requisitos não funcionais que demonstram a forma como alguns requisitos funcionais devem ser concretizados.

Apresenta os motivos pelos quais certas decisões de implementação foram tomadas, assim como o desenho da infra-estrutura necessária ao seu funcionamento, este protótipo foi concebido para garantir a confidencialidade e integridade dos conteúdos em descanso, isto é, na base de dados e no repositório de documentos onde ficam armazenados.

São também apresentadas soluções para os vários problemas aos quais o protótipo não dá resposta, uma vez que o seu âmbito de actuação é limitado. Estas soluções visam constituir um sistema mais seguro, escalável e sem comprometer grandemente a facilidade de utilização e manutenção actuais.

3.2 Análise e Desenho

3.2.1 O Sistema

O sistema *e-doc**link*** é neste momento usado nas mais diversas áreas de actividade e tipos de organização. Porém, para todas sem excepção, o problema da segurança dos dados é um ponto fundamental para a garantia de que o sistema oferece um adequado nível de segurança em relação a documentos e metadados presentes no sistema, de modo a que pessoal não autorizado não tenha acesso a essa informação.

Antes do inicio deste estágio o sistema já dispunha de vários mecanismos de segurança, entre os quais se destaca o mecanismo (protocolo) de autenticação de utilizadores *Kerberos* (28). Uma ACL desenvolvida para o efeito de controlar o acesso de utilizadores a objectos do sistema *e-doc**link*** e um mecanismo de assinaturas digitais em textos de despacho (informação especialmente sensível para a organização). A comunicação entre browser e o servidor pode ser protegida por HTTPS, sendo opcional a sua utilização.

Apesar destes mecanismos, era possível o acesso de escrita e leitura, de utilizadores privilegiados, a dados sensíveis geridos pelo sistema, apesar de no caso de textos de despacho, os mesmos não poderem ser realizados de forma indetectável, quando realizada assinatura digital sobre o texto, apesar disto, pode ocorrer a consulta e posterior revelação de dados a terceiros.

Outro ponto a ter em conta é o impacto causado pelo incremento de segurança no desempenho do sistema relativo, nomeadamente, rapidez de resposta, manutenção, disponibilidade, escalabilidade, assim como na facilidade de utilização por parte dos utilizadores do sistema. Deve ser alcançado um equilíbrio entre as soluções encontradas para adicionar segurança ao sistema e simultaneamente garantir que os utilizadores não sofrem uma redução substancial de funcionalidade ou qualidade de utilização.

3.2.2 Arquitectura e Modelo de Dados

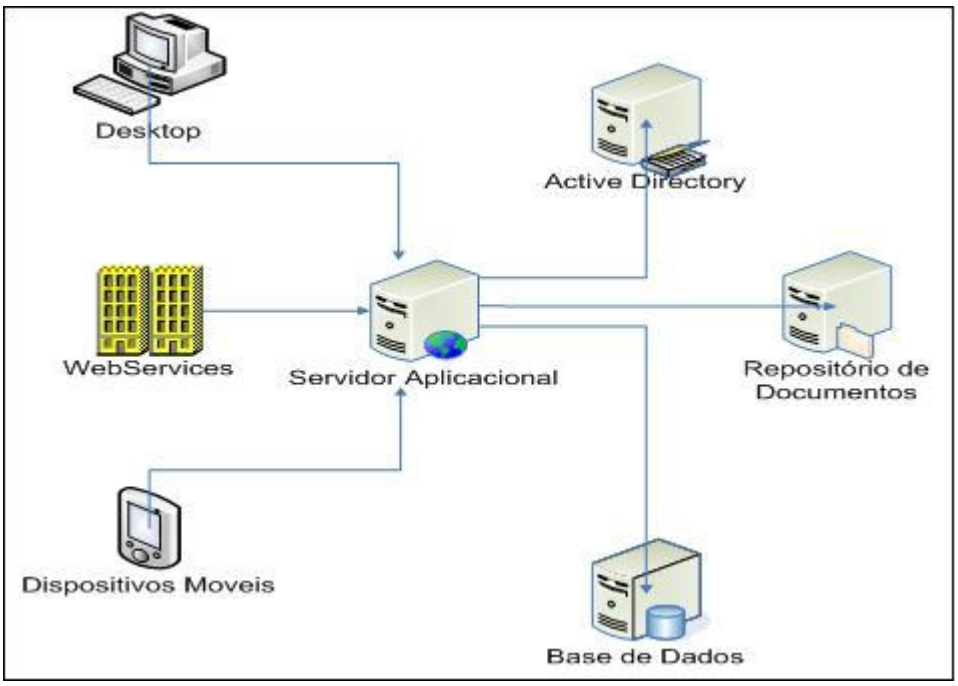

Figura 1 – Arquitectura do Sistema

O sistema gere diversos tipos de Conteúdos, tais como metadados do sistema (i.e. *records* de base de dados), tais como:

- Distribuições
- Etapas
- Registos

O sistema gere também Conteúdos do tipo Documento (i.e. ficheiros) independentemente do formato, estes últimos funcionam como "*attachments*" aos metadados do sistema, permitindo criar, por exemplo, uma Distribuição com várias Etapas em que quer as Etapas quer a Distribuição podem ter Documentos associados. O armazenamento físico de Documentos é estruturado de forma totalmente distinta dos restantes metadados, sendo armazenados em repositórios específicos, que podem ser implementados com base em *Windows SharePoint Services* ou *WebDav*.

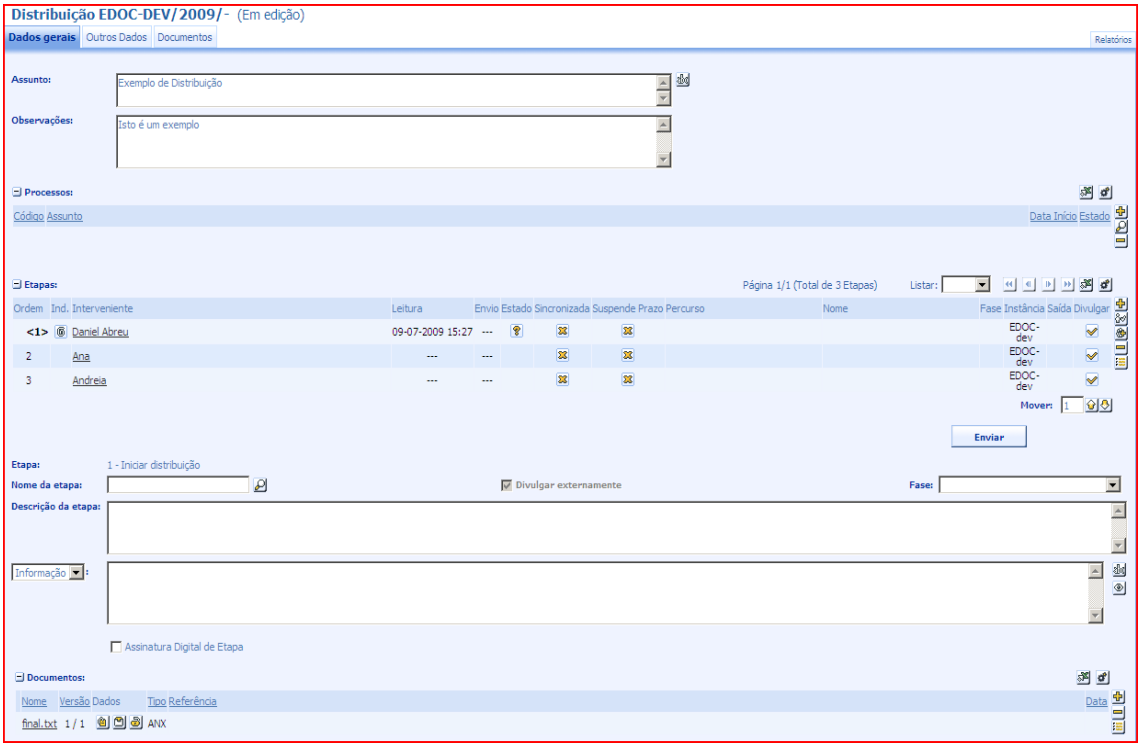

Figura 2 – Exemplo de uma Distribuição com três Etapas e um Documento

Tipo de Conteúdos do Sistema	Conteúdos ou Dados			
	Objectos do sistema	Documentos		Dados sensíveis de configuração
	Metadados	Documentos Office	Outros	
Local de Armazenamento	Base de Dados	Repositório de Documentos		Servidor Aplicacional

Tabela 7 – Tipos de conteúdos e local de armazenamento

3.2.3 Enquadramento dos objectivos

Partindo do plano de trabalhos apresentado anteriormente, podemos expressar os objectivos, divididos pelas propriedades que visam conferir:

- **Autenticidade**
  - Garantia de autenticidade e não repúdio de dados
- **Confidencialidade**
  - Metadados confidenciais em bases de dados
  - Armazenamento de documentos – confidencialidade/protecção de acesso
  - Mecanismos de segurança e reconhecimento de confiança em componentes de software do sistema
  - Mecanismos de cifra de dados
  - Mecanismos de segurança na intercomunicabilidade dos dados
- **Integridade**
  - Garantia de autenticidade e não repúdio de dados
  - Assinatura digital de conteúdos
  - Protecção de dados sensíveis de configuração
  - Mecanismos de segurança e reconhecimento de confiança em componentes
  - Garantia de autenticidade e não repúdio na troca de conteúdos entre organizações
- **Disponibilidade**

Tabela 8 – Objectivos por Propriedades de Segurança

Verifica-se, que as maiores necessidades de segurança se encontraram em garantir confidencialidade, integridade dos dados e não repúdio. Neste sentido, serão apresentados, resumidamente, alguns algoritmos que fornecem garantias destas propriedades. Importa também referir, que os algoritmos por si só não garantem qualquer propriedade, a sua aplicação na prática, quando integrada com os princípios anteriormente descritos e com políticas de utilização correctas são fundamentais para fornecer, correctamente, as propriedades a que se destinam.

3.2.4 Caracterização de ameaças

Tendo em conta que o sistema corre maioritariamente em *intranets*, os eventuais ataques aos quais o sistema poderá ser exposto partem fundamentalmente de *insiders* (28), isto é, pessoas com permissões limitadas de acesso ao sistema via *front-end* (*browser*), pessoas com permissões de administração via *back-end* (administradores de sistemas e bases de dados), assim como pessoas com acesso físico aos diversos

componentes do sistema. Apesar de tudo os administradores são necessários para o correcto funcionamento do serviço o que os leva a ter acesso a informação que de outro modo não teriam. O problema reside no facto de, se por um lado se pretende proteger a informação, por outro também se pretende que este acréscimo de segurança não ponha em causa a usabilidade, desempenho ou manutenção do próprio sistema.

Contudo, o sistema disponibiliza um conjunto de *webservices*, que por sua vez são utilizados por aplicações de clientes, não raras vezes estas aplicações estão expostas na internet, o que traz um perigos adicionais, tais como a utilização de *SQL Injection*.

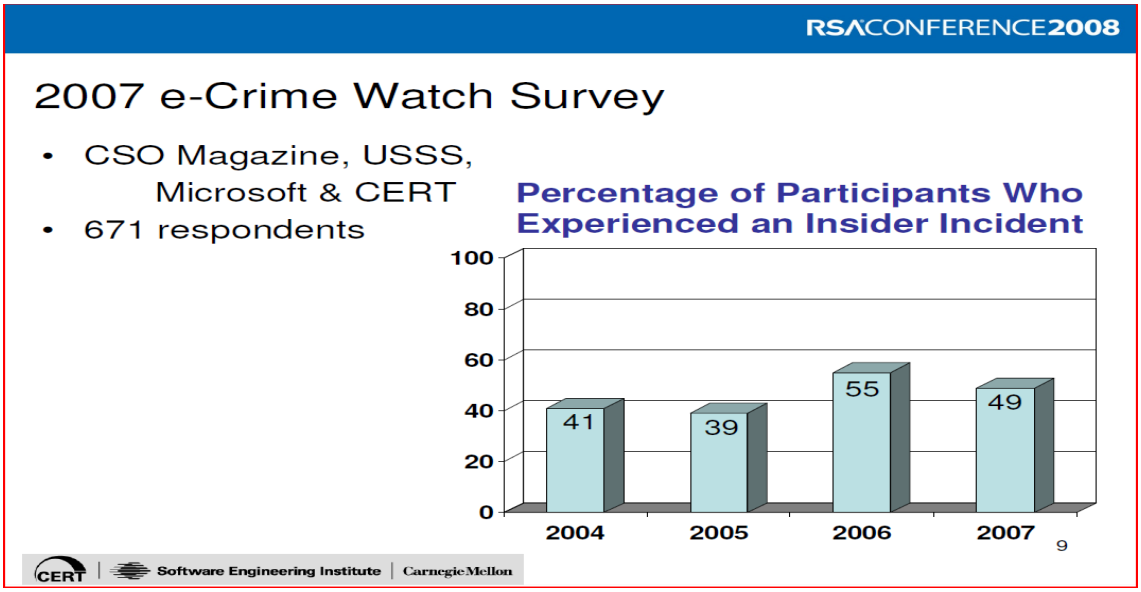

Figura 3 – Incidentes com *insiders*

3.2.5 Árvores de ataque (*Attack Trees*)

De modo a não cair no erro de depender apenas na análise das falhas de engenharia *a posteriori* para melhorar o sistema e sabendo ainda que habitualmente esta informação é ocultada do público (30), procedeu-se à modelação de árvores de ataque (31) (32) de modo a sistematizar os meios de entrada no sistema que um atacante pode utilizar.

Objectivo: Alterar despacho

1. Convencer o autor a alterar o conteúdo
    - 1.1 Subornar o autor
    - 1.2 Chantagear o autor
    - 1.3 Ameaçar o autor
    - 1.4 Enganar o autor
2. Alterar o despacho quando este é escrito no computador
    - 2.1 Alterar o despacho quando o autor não esteja presente
3. Alterar o despacho enquanto este passa pela rede
    - 3.1 Usar o ataque Homem-no-meio
4. Alterar o despacho quando este está guardado na BD
    - 4.1 Subornar o Administrador
    - 4.2 Chantagear o Administrador
    - 4.3 Ameaçar o Administrador
    - 4.4 Enganar o Administrador
    - 4.5 Ter privilégios de administração
    - 4.6 Instalar um virus/worm/keylogger no computador
    - 4.7 Quebrar a palavra-chave de administração
    - 4.8 Aceder aos ficheiros da BD
    - 4.9 Utilizar SQL Injection
5. Convencer alguém com acesso ao despacho a alterar o conteúdo
    - 5.1 Subornar o utilizador
    - 5.2 Chantagear o utilizador
    - 5.3 Ameaçar o utilizador
    - 5.4 Enganar o utilizador

Tabela 9 – Arvore de Ataque, objectivo alterar despacho

Objectivo: Visualizar um documento

1. Convencer a autor a revelar o documento
    - 1.1 Subornar o autor
    - 1.2 Chantagear o autor
    - 1.3 Ameaçar o autor
    - 1.4 Enganar o autor
2. Convencer utilizador com acesso ao documento a revelá-lo
    - 2.1 Subornar o utilizador
    - 2.2 Chantagear o utilizador
    - 2.3 Ameaçar o utilizador
    - 2.4 Enganar o utilizador
3. Visualizar o documento quando este está guardado na BD
    - 3.1 Obter privilégios de Administração
        - 3.1.1 Com ajuda do Administrador
            - 3.1.1.1 Subornar o Administrador
            - 3.1.1.2 Chantagear o Administrador
            - 3.1.1.3 Ameaçar o Administrador
            - 3.1.1.4 Enganar o Administrador
            - 3.1.1.5 Ter privilégios de administração
        - 3.1.2 Sem ajuda do Administrador
            - 3.1.2.1 Atacar Active Directory
    - 3.2 Obter privilégios de Consulta
        - 3.2.1 Com ajuda do Administrador
            - 3.2.1.1 Subornar o Administrador
            - 3.2.1.2 Chantagear o Administrador

3.2.1.3 Ameaçar o Administrador 3.2.1.4 Enganar o Administrador 3.2.1.5 Ter privilégios de administração 3.2.2 Sem ajuda do Administrador 3.2.2.1 Atacar Active Directory 3.2.2.2 Atacar repositório de ficheiros 3.2.2.2.1 Atacar Sistema Operativo do repositório 3.2.2.2.2 Usar um live SO

Tabela 10 – Arvore de Ataque, objectivo visualizar documento

Objectivo: Alterar um documento 1. Convencer a autor a revelar o documento 1.1 Subornar o autor 1.2 Chantagear o autor 1.3 Ameaçar o autor 1.4 Enganar o autor 2. Convencer utilizador com acesso ao documento a revelá-lo 2.1 Subornar o utilizador 2.2 Chantagear o utilizador 2.3 Ameaçar o utilizador 2.4 Enganar o utilizador 2. Alterar o despacho quando este está guardado na BD 2.1 Obter privilégios de Administração 2.1.1 Subornar o Administrador 2.1.2 Chantagear o Administrador 2.1.3 Ameaçar o Administrador 2.1.4 Enganar o Administrador 2.1.5 Ter privilégios de administração

Tabela 11 – Arvore de Ataque, objectivo alterar documento

Apesar de poderem estar incompletas, estas árvores de ataque servem para dar uma boa ideia dos meios de ataque que podem ser utilizados, quer para ler um despacho, quer para alterar um despacho assinado e não assinado, quer no caso do objectivo ser ler e ou alterar um documento.

Podemos verificar pelas árvores acima, que o facto de se adquirir privilégio de administração confere permissões não desejáveis e não necessárias à realização da função de administração, violando um dos princípios da segurança o do menor privilégio, que neste caso pode ser traduzido na regra militar do *need-to-know*, isto é, apesar de o administrador ter privilégio de consulta e manipulação dos dados, não deve poder consultá-los ou manipulá-los apenas porque tem poder para tal.

3.2.6 STRIDE e DREAD

A tabela STRIDE traduz os ataques que são necessários realizar para atingir determinado objectivo:

ID	Descrição	Spoofing	Tampering	Repudiation	Info Disclosure	Denial of Service	Elevation
1	Ver conteúdos	X		X	X		X
2	Alterar conteúdos	X	X	X[2]			X
3	Apagar conteúdos	X	X	X			X

Tabela 12 – Avaliação STRIDE das ameaças

Apesar dos valores das tabelas terem um grande dose de subjectividade, permite comparar e verificar quais os riscos aos quais se deve dar mais prioridade. Por exemplo, se a entrada no sistema se der através da base de dados, a tabela DREAD com escala de 0 a 10 será:

ID	Descrição	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total
1	Ver conteúdos	2	10	10	10	10	8.4
2	Alterar conteúdos	9	10	10[2]	10	10	9.8
3	Apagar conteúdos	7	10	10	10	10	9.4

Tabela 13 – Avaliação DREAD, via Base de Dados

O conteúdo da tabela foi preenchido supondo que o atacante é o Administrador da Base de Dados, que apenas precisa de utilizar o SQL Management Studio.

[2] Só é possível se o conteúdo não estiver assinado digitalmente (apenas se aplicava a texto de despacho)

Se a entrada se der através interface de utilizador a tabela será:

ID	Descrição	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total
1	Ver conteúdos	2	5	3	10	2	4,4
2	Alterar conteúdos	9	5	3^2	10	2	5,8
3	Apagar conteúdos	7	5	3	10	2	5,4

Tabela 14 – Avaliação DREAD, via Interface Utilizador

O conteúdo da tabela foi preenchido supondo que o atacante é um utilizador normal da aplicação, que tem de escalar privilégios para Administrador da aplicação.

3.2.7 Desenho – Metadados confidenciais em BD

No sistema *e-doc**link*** existem diversos metadados relativos a informação inserida no sistema, que devido à sua importância podem necessitar de ser ocultadas, contudo, devido ao número elevado destes metadados e à importância de alguns deles nas optimizações feitas pela BD torna-se difícil obter a confidencialidade sem prejudicar a disponibilidade do sistema. Para alcançar este objectivo, deverá optar-se por utilizar os meios disponibilizados pelo SGBD de forma a cifrar colunas inteiras, sem deixar de ter presente, que irá existir um decréscimo de desempenho que poderá ser minimizado pela escolha do tipo de cifra.

Importa referir que no caso de metadados especialmente sensíveis e devido à sua relevância para a organização, é essencial garantir que a cifra dos mesmos é realizada pelo servidor aplicacional, para que esses metadados sejam guardados na base de dados de forma opaca, mas que a chave com o qual foram cifrados lá não se encontre, nem nunca por lá passe (o que não se verifica com os meios disponibilizados pelo SGBD).

3.2.8 Desenho – Garantia de autenticidade e não repúdio

Tal como foi dito anteriormente o sistema já dispunha de um método para execução de assinaturas de dados especialmente sensíveis, nomeadamente textos de despacho. O contexto em que determinada informação é produzida pode ser tão importante quanto a própria informação. Assim, os mecanismos de garantia de

autenticidade têm obrigatoriamente de abranger, como um bloco, não só a informação sensível mas o seu contexto, de forma a não permitir deturpações desse conjunto.

O processo de assinatura, deverá ser feito com um certificado reconhecido por um CA de modo a garantir o não repúdio da assinatura. Pode, por exemplo, ser usado o cartão do cidadão para assinar esta informação. De modo a dar a conhecer a chave pública do assinante, é utilizado o *Cryptographic Message Syntax Standard* (PKCS#7) (33), que encapsula a(s) assinatura(s), a(s) chave(s) pública(s) do(s) assinante(s) e o caminho de certificação, permitindo verificar a validade da assinatura.

3.2.9 Desenho – Armazenamento de Documentos

No caso do sistema *e-doc**link,*** os repositórios de documentos que o sistema suporta variam entre *Windows SharePoint Services*, *WebDav*, Base de Dados, entre outros e devido a esta variedade de localizações deverá optar-se por cifrar os documentos a quando do upload para o servidor aplicacional e antes de serem enviados para o repositório de documentos. Garantindo assim, à semelhança do que acontece com os textos de despacho, que a chave fica separada do conteúdo que protege (13). Os documentos serão então armazenados de forma opaca, garantindo a confidencialidade dos mesmos. De modo a possibilitar a garantia de integridade, deve também ser efectuada uma assinatura pelo servidor, com o objectivo de garantir a integridade dos dados que se encontram no repositório, assim como a possibilidade de o utilizador assinar o documento para garantir o não repúdio, sendo esta uma possibilidade opcional.

3.2.10 Protecção de dados sensíveis de configuração - Integridade

Para alcançar a protecção dos dados sensíveis de configuração optou-se por fazer verificações de integridade dos mesmos, uma vez que estes dados são acedidos e alterados pela aplicação com alguma frequência. A sua adulteração não deve provocar de imediato um *Denial Of Service* (DOS) para os utilizadores, o que poderia acontecer se estes dados estivessem cifrados. Adições, remoções e alterações devem forçar um novo cálculo de checksum para o valor modificado e assinatura, enquanto as consultas apenas devem validar o checksum existente. De forma a perceber o que foi adulterado,

deve existir um binómio formado por ((chave de configuração ,valor(es) de configuração) , checksum), sendo que deve ser gerado um checksum a partir dos *checksums* referidos anteriormente. Este ultimo deverá ser assinado.

Desta forma é possível garantir que o documento não é adulterado de forma imperceptível, devendo ficar registado a chave e valor de configuração alterados.

Funcionamento:

Trigger: Alterar valor de configuração

1. É verificado o *checksum* do binómio (chave,valor) pretendido.
2. É verificada a assinatura, com base nos *checksums* existentes, à excepção do que se pretende alterar, ao invés é utilizado o *checksum* calculado em 1. (falha)
   2.1. É registado o erro na verificação da assinatura
   2.2. São verificados todos os checksums individuais
   2.3. É registada a(s) chave(s) alterada(s) (falha)
      2.3.1. É registado que a assinatura foi alterada
3. É alterado o valor de uma chave
4. É gerado um checksum a partir de (chave,valor)
5. É gerada a assinatura, baseada no *checksum* dos *checksums* anteriores.

Trigger: Ler valor de configuração

1. É verificado o *checksum* do binómio (chave,valor) pretendido.
2. É verificada a assinatura, com base nos *checksums* existentes, à excepção do que se pretende alterar, ao invés é utilizado o *checksum* calculado em 1. (falha)
   2.1. É registado o erro na verificação da assinatura
   2.2. São verificados todos os *checksums* individuais
   2.3. É registada a(s) chave(s) alterada(s) (falha)
      2.3.1. É registado que a assinatura foi alterada

3.2.11 Mecanismos de segurança e reconhecimento de confiança em componentes

A forma de resolver este problema é recorrendo à criptografia de chave pública, isto é, quando um componente do sistema é produzido, deve ser assinado. Esta assinatura é verificada com o auxilio da chave pública a quando da inicialização do componente, de forma a verificar se o componente foi ou não adulterado. Caso tenha sido adulterado, deve ser registado que módulo que foi sujeito a alteração.

3.2.12 Documentos Microsoft Office com mecanismos de autenticação e segurança

Para documentos produzidos com a plataforma Microsoft Office é possível de forma simples integrada adicionar assinaturas digitais e cifrar conteúdos de documentos como o Word, Excel e PowerPoint. Apresentam-se duas formas de cifrar conteúdos e assinar digitalmente estes documentos, recorrendo ao mecanismo descrito acima, apenas disponível para aqueles 3 tipos de documentos, uma vez todos os outros não dispõem da funcionalidade integrada. Para outros tipos de documentos, que não suportem esta funcionalidade é necessário utilizar uma aplicação que actue ao nível do ficheiro, de forma a cifrá-lo e ou assinar digitalmente a síntese desse ficheiro, à semelhança do alcançado pelo protótipo, que será descrito com mais detalhe posteriormente.

3.2.13 Mecanismos de autenticidade, segurança e não repúdio na troca de conteúdos entre organizações

Para conteúdos transaccionados entre organizações devem ser utilizados envelopes seguros, isto é, o conteúdo deve ser cifrado e assinado com um certificado emitido por um *Certificate authority* (CA). Devido ao tamanho variável dos conteúdos, estes devem ser cifrados com uma chave simétrica, chave essa que deve ser cifrada com uma chave privada. As chaves públicas que permitem decifrar as chaves simétricas devem, preferencialmente, ser transmitidas e instaladas manualmente, de forma a verificar a sua autenticidade.

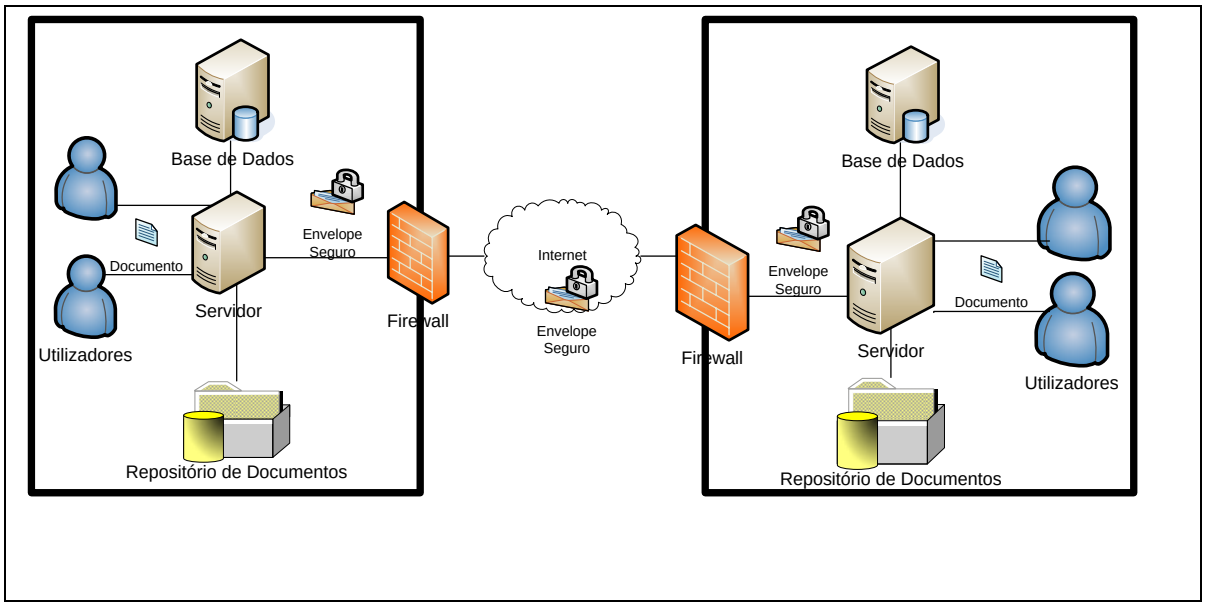

Figura 4 – Exemplo de utilização de Envelopes Seguros (6)

3.2.14 Requisitos Funcionais e Não Funcionais do protótipo

Estes requisitos expressam a necessidade de conferir confidencialidade e integridade a metadados e documentos.

Os requisitos funcionais do protótipo são os seguintes:

1. O protótipo deve poder receber um texto legível (*plaintext*) e transformá-lo num texto ininteligível (*ciphertext*).
2. O protótipo deve poder receber um ficheiro inteligível e transformá-lo num ficheiro ininteligível.
3. O protótipo deve poder receber um texto ininteligível e transformá-lo num texto inteligível, se o texto inicial tiver sido produzido pelo mesmo protótipo.
4. O protótipo deve poder receber um ficheiro não legível e não executável e transformá-lo num ficheiro legível e ou executável, se o ficheiro inicial tiver sido produzido pelo mesmo protótipo.

5. O protótipo deve poder verificar se um texto que tenha sido previamente cifrado pelo protótipo foi ou não alterado, quando for requisitado o acesso ao texto.
6. O protótipo deve poder verificar se um ficheiro que tenha sido previamente cifrado pelo protótipo foi ou não alterado, quando for requisitado o acesso ao ficheiro.

Os requisitos não funcionais do protótipo são os seguintes:

1. Quer a cifra de ficheiros, quer a de textos, não deve estar dependente da introdução de uma palavra-chave ou outra semelhante por parte do utilizador

3.2.15 Desenho da Solução implementada no protótipo

Como?

Devido às limitações das criptografias simétrica e assimétrica, optou-se pela utilização de criptografia híbrida. Chegou-se a dois cenários genéricos distintos de aplicação concreta, contudo, entre vantagens e desvantagens, apenas um foi implementado no protótipo. Os dois métodos distintos estão descritos abaixo:

Seja Sk uma chave secreta, i.e. uma chave simétrica.
Seja Puk uma chave pública, i.e. uma chave assimétrica pública.
Seja Prk uma chave privada, i.e. uma chave assimétrica privada.

Método 1:

Trigger_1: Utilizador pretende guardar informação no sistema de forma confidencial

1. A partir da interface de utilizador, selecciona quais os utilizadores que podem ter acesso à informação.
2. Selecciona a informação a inserir no sistema.

3. O cliente (browser) gera uma chave secreta Sk, com a qual cifra a informação seleccionada em 2.
4. O cliente cifra Sk com a Puk do próprio, e cópias de Sk com as Puks de todos os utilizadores escolhidos em 1.
5. O cliente envia os dados dos pontos 3 e 4 para o servidor, que os processa e guarda na base de dados.

Trigger_2: Utilizador pretender aceder a informação confidencial no sistema

1. O cliente indica ao servidor que o utilizador requisitou acesso a dados confidenciais.
2. O servidor verifica se o utilizador tem acesso aos dados, isto é, verifica se existe uma cópia de Sk cifrada com a Puk do utilizador a requisitar o acesso.
3. São enviados para o cliente os dados cifrados juntamente com Sk cifrada com a Puk do utilizador requerente.
4. O cliente requer acesso à Prk do utilizador, ao utilizador.
5. O cliente decifra Sk com a Prk do utilizador.
6. O cliente decifra os dados com Sk.
7. O cliente revela ao utilizador a informação confidencial através da interface de utilizador.

Em relação ao método 1 existem vários problemas:

- O utilizador teria concretizar o ponto 1 do trigger_1 imperativamente.
- Recuperação de acessos em caso de perda de chave privada, muito difícil.
- Carga adicional sobre o computador do utilizador.
- Distribuição das chaves públicas pelos clientes.
- Conflitos de interesse (social)
- O ponto 4 do trigger_2 requer interacção do utilizador
- Multiplicação da informação guardada na BD

Os benefícios mais destacáveis são os seguintes:

- Segurança *end-to-end*
- Contenção de danos em caso de roubo de chave privada

Método 2:

Trigger_1: Utilizador pretende guardar informação no sistema de forma confidencial

1. Selecciona a informação a inserir no sistema.
2. O cliente (browser) envia a informação para o servidor.
3. O servidor gera uma chave Sk, com a qual cifra os dados recebidos em 2.
4. O servidor utiliza a Puk da aplicação para cifrar Sk.
5. O servidor envia os dados de 3 e 4 para a base de dados.

Trigger_2: Utilizador pretender aceder a informação confidencial no sistema

1. O cliente indica ao servidor que o utilizador requisitou acesso a dados confidenciais.
2. O servidor verifica se o utilizador tem acesso aos dados, isto é, verifica se na ACL do sistema o utilizador requerente tem acesso aos dados.
3. O servidor decifra Sk fazendo uso da sua Prk.
4. O servidor decifra os dados usando Sk de 3.
5. O servidor envia os dados resultantes de 4 ao cliente.
6. O cliente mostra a informação ao utilizador.

Em relação ao método 2 existem também vários problemas:

- A chave secreta é crítica
- A segurança não é *end-to-end*
- É difícil conter os estragos em caso de roubo de chave privada
- Carga adicional no servidor aplicacional

Os benefícios mais destacáveis são os seguintes:

- Gestão de chaves facilitada (vs metodo1)
- Independente dos clientes e de utilizadores
- Utilização a ACL existente
- Maior rapidez de execução (vs método1)
- Sem necessidade de duplicar informação na BD (vs metodo1)

Após ponderação sobre ambos os métodos, optou-se pelo método 2. Em primeiro lugar para esta escolha está o facto de não estar dependente da interacção com o utilizador, ou seja, não requer um esforço adicional ao utilizador. Por outro lado, e visto que só existe um par de chaves, a gestão dessa chave é facilitada, assim como a sua salvaguarda e recuperação em caso de falha. Outro factor de decisão deve-se a que com o método 2 não existe execução de código extra do lado do cliente. Por outro lado, aceita-se o facto de que o par de chaves é crítico e deve ser protegido a todo o custo, assim como as consequências da sua perda ou comprometimento.

Onde?

Advêm da decisão tomada anteriormente que a execução da cifra seja feita ao nível da aplicação, desta forma evitando um acréscimo de carga à BD e evitando o poder do administrador de bases de dados, uma vez que os dados estão sempre em estado cifrado na base de dados. A cifra na aplicação também permite uma maior flexibilidade em termos de algoritmos e métodos de cifra, flexibilidade que o SGBD não permite.

Quando?

A opção de cifrar ou não uma informação introduzida no sistema é responsabilidade do utilizador, uma vez que apenas esse pode considerar se será relevante tornar confidencial a informação introduzida. O utilizador poderá notar um acréscimo no tempo de espera, o que o alertará para o facto de que apenas informações relevantes devem ser tornadas confidenciais.

Esta decisão propicia o risco de "marcar", para um atacante, os dados cifrados como importantes e potencialmente desejáveis pelo seu conteúdo.

3.2.16 Analise e Desenho - Notas Finais

Esta fase teve por objectivo analisar e orientar o desenho da solução. Optou-se por um sistema de criptografia híbrida de modo a garantir os benefícios e reduzir os problemas individuais de cada uma das criptografias que lhe dão origem. Das duas soluções que se apresentavam foi escolhida a que, sem pôr em causa a segurança do sistema, menos impacto directo têm no utilizador final, facilitando assim a sua função mas conferindo grande importância ao par de chaves utilizado. O método escolhido sofreu ainda algumas alterações de modo a garantir maior segurança e fiabilidade.

3.3 Implementação

3.3.1 Introdução

Nesta fase descreve-se como foi aplicado o método apresentado na secção anterior, assim como em que medida são cumpridas as várias recomendações para um design seguro e fiável, apresentadas no Capitulo 1.

Este protótipo foi desenvolvido na linguagem C#, na Framework .Net 3.5 que conta com a implementação de todos os algoritmos utilizados no protótipo.

3.3.2 Especificações da solução do protótipo

A fim de melhorar a solução encontrada na secção anterior, introduziram-se algumas modificações com o objectivo de a tornar mais robusta.

Chave Simétrica - Sk

Por cada conteúdo a ser cifrado, é gerada uma nova chave simétrica. A geração desta chave é influenciada por 5 factores, três deles são fixos, nomeadamente o tamanho

da chave simétrica, que se fixa em 256 bits, o número de iterações feitas para derivar a chave simétrica e finalmente, o vector de inicialização. Os outros dois factores, são gerados no momento em que é necessária a chave simétrica, os factores são um *salt* e uma *passphrase*.

Salt e Passphrase

Estes dois elementos servem para introduzir factores aleatórios na geração da chave simétrica. A *passphrase* pode ainda ser utilizada para receber informações de referência, de modo a identificar o contexto onde o conteúdo se insere, tais como número de registo, data do registo ou *ids* vários.

Assinatura

Após a cifra de uma informação é gerada uma assinatura a partir da sua síntese. Desta forma garante-se a integridade dos dados e também se evita a carga adicional de decifrar dados corrompidos. Esta assinatura não pretende garantir o não repúdio e pode ser gerada a partir de um par de chaves de um certificado *self-signed*.

Importa também referir que o protótipo permite gerar dois tipos de assinaturas, ECDSA – P521 se funcionar em sistemas Windows Vista / Windows Server 2008 ou RSA com SHA1 se anteriores.

Esta assinatura não tem qualquer relação com a assinatura previamente existente, na qual é usado um certificado pessoal de quem assina, garantindo não repúdio e que se aplica exclusivamente a textos.

Certificado

A chave pública para cifra de dados está contida num certificado, sendo necessária uma correspondência entre a chave pública e privada, esta ultima é protegida pelo Windows e pelas credenciais da conta onde o sistema corre. Os certificados devem ser criados *a priori*, o que for utilizado para cifra, deve ser emitido por uma CA, com vista

a proteger de possíveis forjas. O certificado utilizado para assinaturas, pode ser *self-signed* e caso não exista pode ser criado pelo sistema, contudo, deve respeitar a norma X509v3. Este certificado deve ser configurado para utilização no ficheiro XML de configuração do protótipo.

3.3.3 Descrição da solução implementada no protótipo

O protótipo conta com dois módulos[3] base, o primeiro é responsável pelas operações criptográficas (cifra, decifra, assinaturas e certificados). O segundo módulo funciona como um "*stub"* entre a componente de criptografia e outras componentes que queriam usufruir dos serviços do primeiro módulo, é também responsável por manter o ficheiro de registo do protótipo. O protótipo conta ainda com uma terceira componente, gráfica, que comunica com o segundo módulo. Devido à forma modular como foi concebido é possível fornecer os serviços criptográficos, não apenas através de uma interface de utilizador mas também através de *webservices*.

- Seja FR a função *random*, gerador de números aleatórios
- Seja FS a função *store*, que guarda dados
- Seja FRS a função que recupera dados da *store*
- Seja FD a função de derivação de Sk
- Seja FC a função de cifrar dados
- Seja FRC a função de decifrar os dados
- Seja FA a função de assinar dados
- Seja FVA a função de verificar a assinatura dos dados
- Seja FH a função de produzir uma síntese
- Seja Sk a chave simétrica
- Seja Puk a chave pública
- Seja Prk a chave privada
- Seja S o *salt*
- Seja P a *passphrase*
- Seja I o número de iterações a fazer por FD

[3] Diagramas de classe em anexo

- Seja M o conteúdo em claro (*plaintext*)
- Seja Mc o conteúdo cifrado (*ciphertext*)
- Seja PPuk a P cifrada com Puk
- Seja Hash a síntese produzida por FH
- Seja A a assinatura

Pré-requisitos:

- Deverá ser instalado no servidor aplicacional, um certificado com uma chave privada associada.
- O certificado deve ser instalado com a mesma conta Windows da aplicação.
- O certificado deve ter uma cifra do tipo RSA com SHA-256 ou ECDSA-521 se o SO for Windows Vista.
- O certificado instalado não deve permitir que a chave privada seja exportada.
- No caso de ser Windows Vista e usar ECDSA-521, deve ser instalado um certificado desse tipo. È possível criar este tipo de certificado utilizando o protótipo.
- Deve ser configurado no ficheiro XML de configuração do protótipo, qual o identificador do certificado a ser utilizado, em conjunto com o identificador da chave privada.

O método de cifra utilizado será concretizado recorrendo a criptografia híbrida. Para cifrar um conteúdo o protótipo executa os seguintes passos:

1. São gerados 2 números aleatórios, recorrendo a um CSPRNG que fornece dois *arrays* de 32 bytes, sendo um deles o *salt* e sendo o outro a *passphrase*.
2. Com os dois números gerados anteriormente, é derivada uma chave simétrica com 256 bits, sendo o método de derivação baseado no *Password-Based Cryptography Standard* (PKCS#5) (34).

3. Após gerada a chave de 256 bits, é utilizada para cifrar o conteúdo recorrendo ao algoritmo AES/*Rijndael*.
4. Recorrendo ao certificado previamente instalado no sistema e à sua chave pública, a *passphrase* é cifrada utilizando o algoritmo RSA.
5. É então utilizado o algoritmo SHA1 ou ECDSA-P521 para gerar uma síntese do que foi gerado no ponto 4.
6. A síntese gerada no ponto 5 é assinada com a chave privada do certificado.
7. São guardados na BD os seguintes dados, o texto cifrado com a chave simétrica, o *salt* utilizado para gerar a chave simétrica, a chave simétrica cifrada com a chave pública e a assinatura.

Descrevendo os passos anteriores como funções, assim:

1. FR () → S , FR () → P
2. FD (S,P,I) → Sk
3. FC (Sk,M) → Mc
4. FC (Puk, P) → PPuk
5. FH (PPuk) → Hash
6. FA (Hash, Prk) → A
7. FS (Mc,S,PPuk,A)

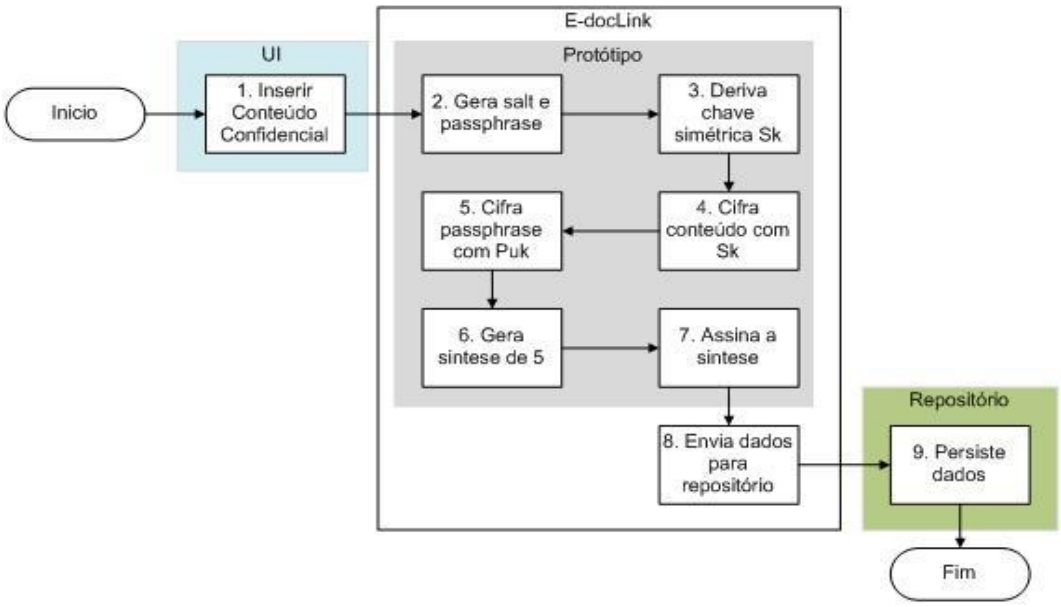

Figura 5 – Diagrama de Fluxo, mecanismo de cifra via Protótipo

Para decifrar um conteúdo cifrado anteriormente, o protótipo segue os seguintes passos:

1. É indicado pelo utilizador qual o conteúdo a decifrar
2. São verificadas as permissões do utilizador na ACL
3. O protótipo procura na BD por um registo do conteúdo.
4. Utilizando a chave pública verifica se a assinatura contida na BD correspondente àquele ficheiro/texto está correcta.
5. Acede ao certificado especificado no ficheiro XML de configuração e utiliza a chave privada RSA correspondente para decifrar a *passphrase*.
6. Utilizando o *salt* e a *passphrase* deriva a chave simétrica, da mesma forma que foi feita para cifrar, ou seja, utilizando o método descrito no Rfc-2898.
7. Tendo a chave simétrica disponível e utilizando o algoritmo AES/*Rijndael* é decifrado o conteúdo.
8. O conteúdo é apresentado ao utilizador

Descrevendo os passos anteriores como funções, assim:

1. FRS () → Mc , S , PPuk , A
2. FH (PPuk) → Hash
3. FVA (Hash, Puk) → True V False
4. FRC (PPuk, Prk) → P
5. FD (S,P,I) → Sk
6. FRC (Sk,Mc) -> M

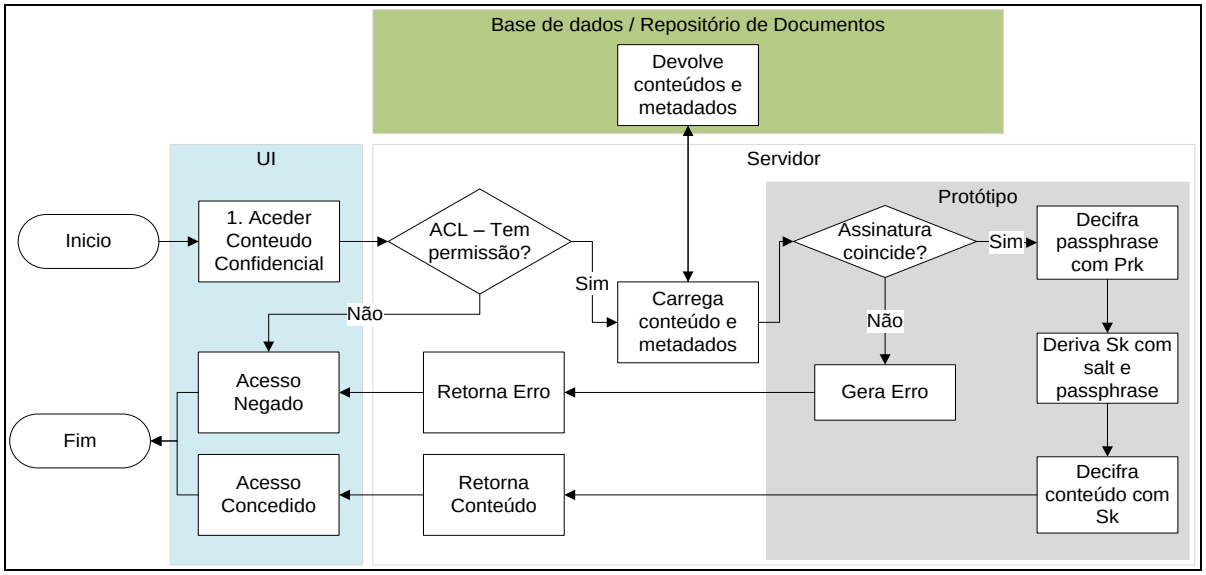

Figura 6 – Diagrama de Fluxo, mecanismo de decifra via Protótipo [4]

3.3.4 Considerações adicionais sobre o protótipo

No protótipo não existe controlo sobre quem requer a decifra do conteúdo, uma vez que a ACL é fornecida pelo sistema e o protótipo tem apenas por objectivo demonstrar a possibilidade de garantir confidencialidade e integridade de documentos e textos. Existe também outra limitação em relação a textos, para decifrar um texto, é necessário introduzir a versão cifrada desse texto no protótipo. Em relação à protecção das chaves privadas utilizadas pelo protótipo, é garantido pelo sistema operativo Windows que apenas o utilizador que instala o certificado tem acesso à chave privada, não sendo possível exportar essa chave. O certificado que serve para instalar na máquina deve ser guardado em caso de necessidade de recuperação da chave privada.

A interface com o utilizador foi feita apenas para demonstração, e está feita de forma independente do módulo de cifra, uma vez que ao ser integrado não necessitará de qualquer interface com o utilizador.

[4] Na imagem, metadados são informações adicionais apenas utilizados pelo protótipo. Não são metadados do sistema.

3.3.5 Especificação da solução de assinaturas digitais

Foi estendido o mecanismo anterior de assinaturas digitais, de modo a ter em conta o contexto da assinatura, neste momento é gerada uma síntese do contexto que envolve o texto de despacho, síntese essa que é adicionada à síntese do texto de despacho, funcionando à semelhança de um *salt*, irá tornar única a síntese daí resultante, por fim essa síntese é assinada com a chave privada de um certificado digital devidamente emitido por um CA, a chave pública desse certificado assim como a assinatura são encapsuladas num objecto, que segue o standard PKCS#7.

Funções de Sintese

Existem duas funções de síntese em utilização no sistema, a SHA512 que é usada para sintetizar identificadores, texto, datas, autores e outras *flags*. É usada também a função SHA1 por motivos de compatibilidade com o sistema actual de funcionamento, que é usada para gerar a síntese que será assinada pelo RSA.

Assinatura e não repúdio

De modo a garantir o não repúdio da assinatura, esta deve pertencer a um certificado emitido por um CA com o objectivo de garantir o não repúdio. A assinatura digital qualificada constante no cartão do cidadão é disso exemplo, estes certificados usam o algoritmo SHA1 com o RSA para produzir uma assinatura digital qualificada (35). Como este processo é feito no browser do cliente é necessário recorrer a um ActiveX que adiciona funcionalidades ao browser do cliente, assim como a AJAX que permite fazer pedidos e receber respostas assincronamente ao servidor, para obter dados necessários à assinatura.

Certificado

A chave pública para cifra de dados está contida num certificado, sendo necessária uma correspondência entre a chave pública e privada, esta ultima é protegida pelo Windows e pelas credenciais da conta onde o sistema corre, ou no caso do cartão do cidadão pelo PIN do utilizador. O certificado deve respeitar a norma X509v3.

3.3.6 Descrição da solução implementada – Assinaturas Digitais

Pré-requisitos:

- O utilizador deverá ter um certificado, cuja *flag* "*Key Usage*" da norma X509v3 indique "*Non Repudiation*" activo, caso contrário a assinatura não terá será válida para conferir não repúdio.

- Seja FS a função *store*, que guarda dados
- Seja FRS a função que recupera dados da *store*
- Seja FA a função de assinar dados
- Seja FVA a função de verificar a assinatura dos dados
- Seja FH a função de produzir uma síntese
- Seja FE a função de encapsular num objecto CMS/PKCS#7
- Seja FC a função de escolher um certificado
- Seja Puk a chave pública
- Seja Prk a chave privada
- Seja M o texto
- Seja Hash a síntese produzida por FH
- Seja A a assinatura
- Seja C o certificado
- Seja D dados em edição
- Seja X dados de contexto
- Seja CMS um objecto da norma PKCS#7

O método de assinatura digital qualificada executa os seguintes passos:

1. O ActiveX procura na *Keystore* da máquina por certificados instalados, e dá ao utilizador a hipótese de escolher o certificado desejado.
2. Após a selecção do certificado, os dados actualmente em edição pelo utilizador são enviados para o servidor, via AJAX de modo a serem guardados na BD.
3. O servidor na posse dos dados mais actuais gera uma síntese com base em diversas informações de contexto usando SHA512, retornando uma síntese desses dados.
4. Quando o ActiveX do cliente recebe a síntese vinda do servidor, efectua por sua vez uma síntese do texto, à qual junta a síntese recebida do servidor.
5. Com a síntese resultante do ponto 4 é iniciado o processo de assinatura, utilizando o algoritmo SHA1 com RSA, durante esta fase o utilizador deve introduzir o PIN que lhe dá acesso à chave privada do cartão do cidadão, com a qual será executada a assinatura.
6. A assinatura resultante do ponto 6, juntamente com a chave pública do certificado é encapsulada num objecto CMS/PKCS #7.
7. Este objecto é enviado para o servidor a fim de ser persistido na BD.

Descrevendo os passos anteriores como funções, assim:

1. FC() → C (cliente)
2. FS(D) (cliente)
3. FH(X) → Hash (servidor)
4. FH(FH(M) + Hash) → Hash (cliente)
5. FA(Hash,Prk) → A (cliente)
6. FE(A,Puk) → CMS (cliente)
7. FS(CMS) (cliente)

O método para fazer a verificação da assinatura segue os seguintes passos:

1. São carregados os dados de contexto relativos a um texto de despacho.
2. É gerada uma síntese com base nesses dados utilizando o SHA512
3. É gerada uma síntese com base no texto do despacho, utilizando SHA512
4. É gerada uma síntese com base nas sínteses dos pontos 2 e 3.
5. Com a síntese do ponto 4, e na posse do objecto CMS é verificada a validade da assinatura

Descrevendo os passos anteriores como funções, assim:

1. FRS() -> CMS,X,M
2. FH(X) -> Hash1
3. FH(M) -> Hash2
4. FH(X,M) -> Hash
5. FVA(Hash, CMS) -> True V False

3.3.7 Considerações adicionais sobre assinaturas digitais

A utilização do algoritmo SHA1 tem sido, nos últimos tempos (desde 2005), posta em causa devido aos "recentes" desenvolvimentos no sentido de quebrar o algoritmo, segundo Schneier conseguiu-se quebrar o algoritmo reduzindo o número de operações até encontrar uma colisão de 2^{80}, recorrendo a um ataque *birthday* (36), para 2^{63} (37) e mais recentemente para 2^{52} (38). Contudo, o método de assinatura mais utilizado é o SHA1 com RSA, também os cartões do cidadão utilizam este algoritmo apesar da data de entrada em vigor ter sido posterior à descoberta do problema.

Apesar do referido anteriormente, estas assinaturas conferem a propriedade de não repúdio e portanto quem assinar é inequivocamente autor da assinatura sem hipótese de contestação.

3.3.8 Documentos MS Office com mecanismos de autenticação e segurança

Devido à simplicidade de cifrar e assinar os documentos Office mais utilizados de forma integrada com a própria aplicação, optou-se por não utilizar uma aplicação que actue ao nível do ficheiro. Apesar de limitar a utilização a três tipos de documentos, esta decisão torna mais fácil e transparente a utilização ao utilizador.

3.3.9 Protecção de dados sensíveis de configuração

Graficamente o algoritmo de verificação de alterações funciona da seguinte forma, para leitura de valores de configuração:

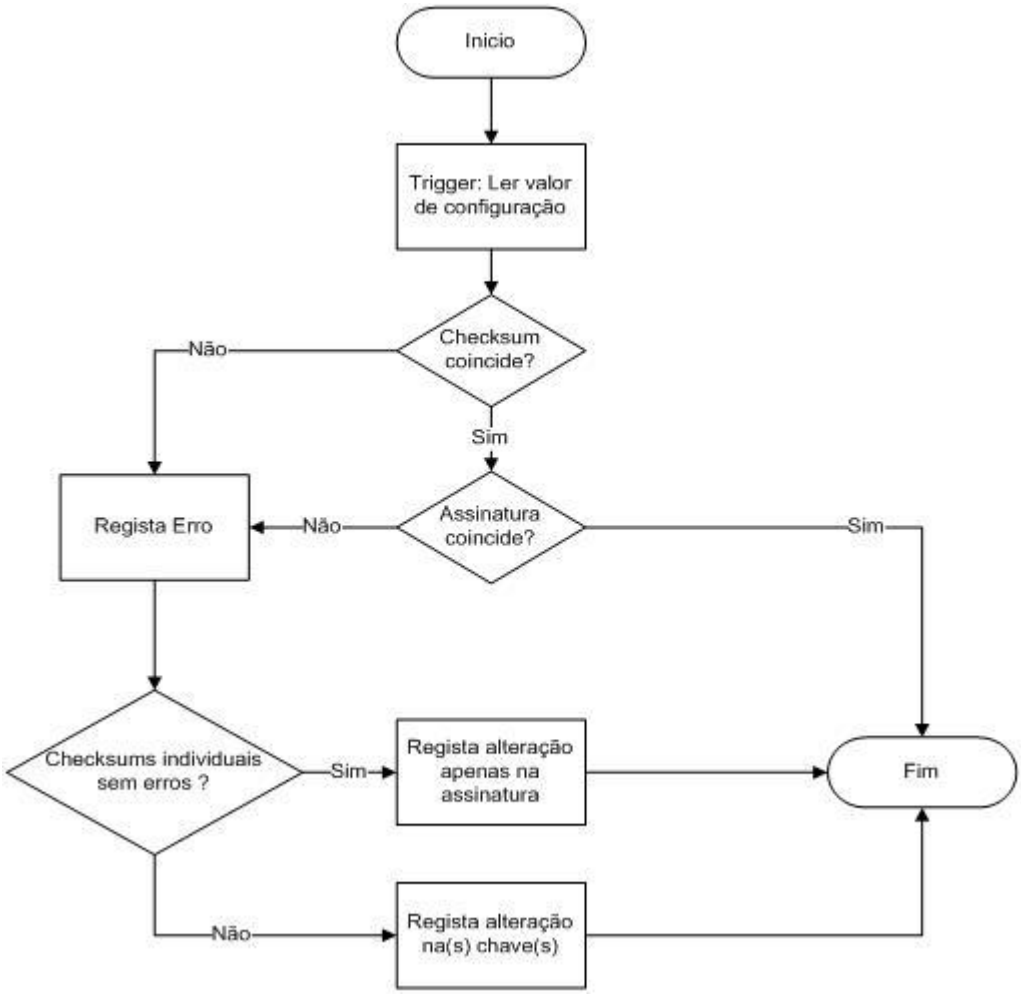

Figura 7 – Leitura de dados sensíveis de configuração

A implementação deste mecanismo possibilitou a utilização do algoritmo ECDSA, uma assinatura recorrendo a criptografia baseada em curvas elípticas, bastante mais rápidas que as tradicionais, factorização de números primos grandes como o RSA e logaritmos discretos como no ElGamal. Á semelhança do que foi utilizado no protótipo, o algoritmo ECDSA-P521 para a assinatura de uma síntese produzida pelo algoritmo SHA512. Os *checksums* dos pares (chave,valor) são feitos utilizando o SHA512. O algoritmo de alteração de valores de configuração funciona da seguinte forma:

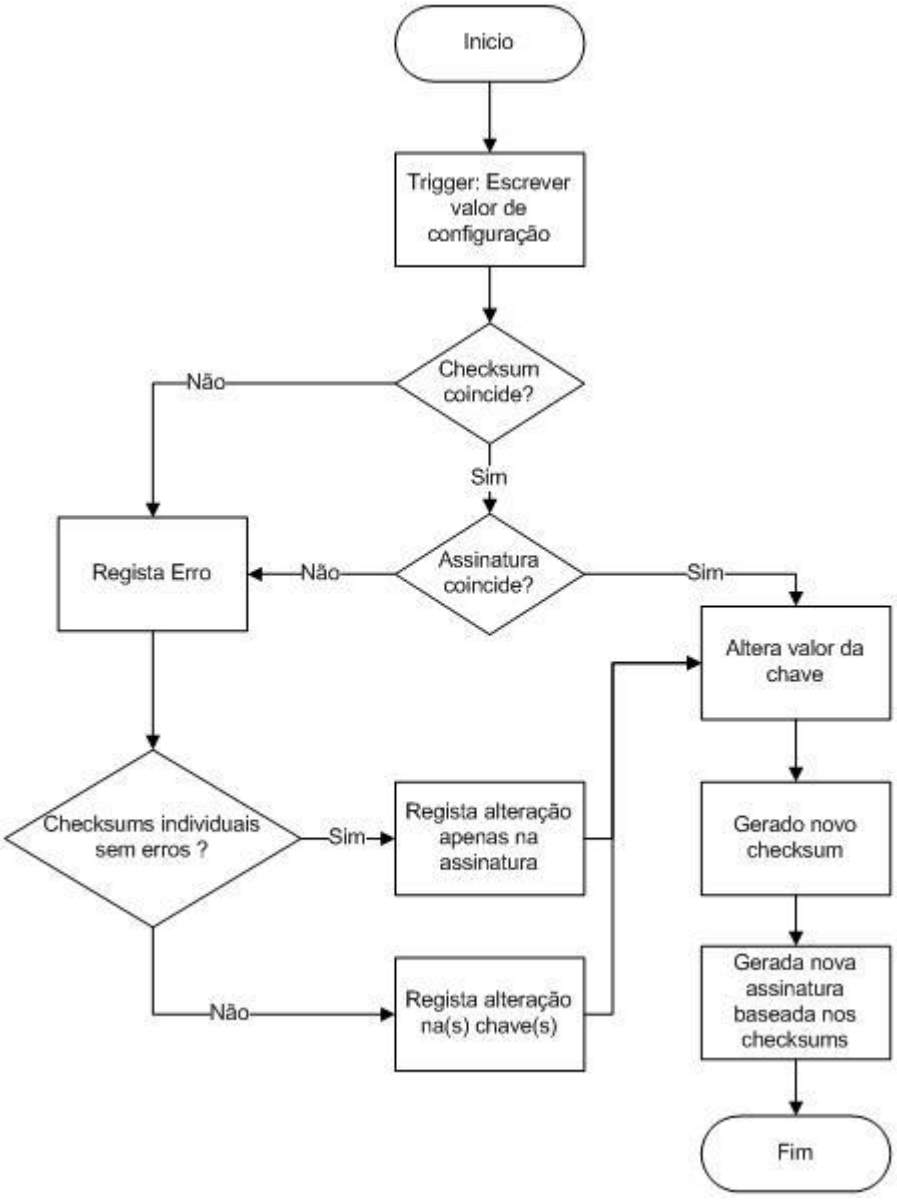

Figura 8 – Escrita de dados sensíveis de configuração

3.3.10 Considerações adicionais sobre protecção de dados de configuração

Esta assinatura é conseguida usando o método de assinatura do protótipo, portanto esta sofre das mesmas limitações.

Capítulo 4 – Resultados

Este capítulo tem por objectivo mostrar os resultados obtidos, na prática, após a utilização do protótipo. Serão feitas comparações entre a análise que foi feita *a priori* e após aplicação do protótipo. É também importante referir que este protótipo utiliza a abordagem do "método 2", descrito no ponto 3.1.8.

Em relação às assinaturas digitais de textos e documentos dentro do sistema *e-doc**link***, o novo método que engloba o contexto é transparente para o utilizador, que não notará diferenças na interface. Quanto aos documentos Office será mostrado a aplicação prática da solução adoptada.

4.1 Confidencialidade e integridade de documentos e textos – Protótipo

Para demonstrar os resultados obtidos, será utilizado um ficheiro de texto, simulando um ficheiro plausível de ser introduzido no sistema. Este ficheiro conterá uma informação importante que deve ser ocultada dos olhos de simples técnicos.

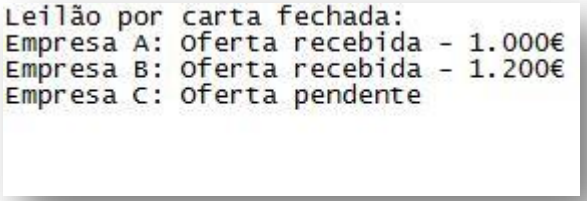

Figura 9 – Ficheiro de texto

Antes da utilização do protótipo, este ficheiro ficaria acessível a quem tivesse permissões ao nível da ACL do SO do Repositório de Documentos para o consultar e alterar. Ou seja, seria possível a um técnico adulterar este ficheiro, por exemplo:

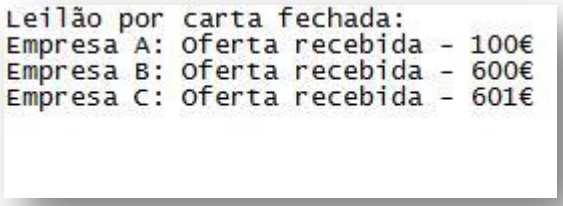

Figura 10 – Ficheiro de texto

Após a recepção desta informação pelo responsável que deve decidir que proposta escolher, este irá tomar uma decisão com base em pressupostos errados e declarar a Empresa C como vencedora, quando na verdade, esta só teve que alterar o ficheiro através de pessoas com permissão para tal.

Com a introdução do protótipo, o ficheiro ficará ininteligível, assim:

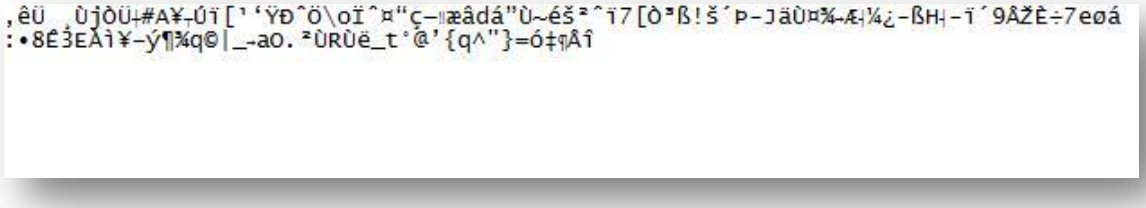

Figura 11 – Ficheiro de texto cifrado

Existe também um ficheiro que guarda um registo dos documentos e textos cifrados, de modo a poder recuperá-los e verificar alterações. Com o ficheiro acima descrito, fica registada a seguinte informação:

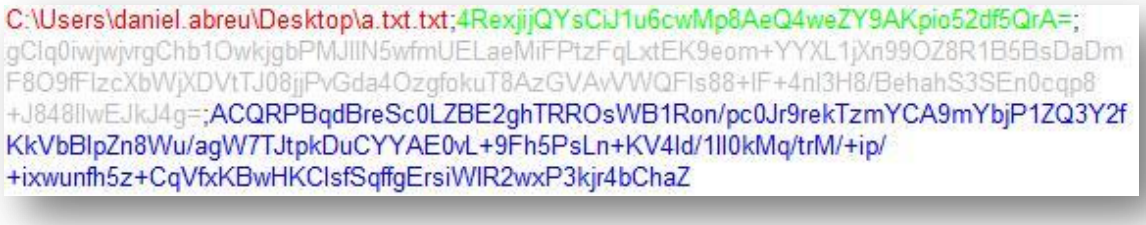

Figura 12 – Ficheiro de registo do Protótipo

Qualquer alteração ao ficheiro anterior resultará em mau funcionamento do sistema e detecção imediata de problemas, causados por alguém que tentou adulterar ou o ficheiro do leilão, ou o ficheiro de registo. Caso se altere o texto a vermelho, resultará na impossibilidade de decifrar o ficheiro. No caso de se alterar o texto a verde irá resultar em "*password* inválida". Caso se altere o texto a cinzento, o resultado será

“Dados adulterados” e se for o texto a azul alterado, o resultado será “Assinatura não coincide”. Caso os dados originais sejam repostos, o protótipo irá decifrar o ficheiro.

No caso de ser um texto, o resultado é semelhante. No caso anterior, na possibilidade de se consultar o texto em aberto, mesmo que assinado digitalmente, existe a possibilidade da Empresa C ao ter conhecimento da maior licitação, poder dar o menor valor possível a seguir ao máximo previamente existente. Com a utilização do protótipo este cenário já não se coloca, uma vez que o administrador não consegue de forma simples aceder ao conteúdo do ficheiro/texto.

No entanto ainda é possível, por parte de administradores, apagar ficheiros confidenciais. Este comportamento é impossível de ser prevenido, uma vez que no SO Windows o administrador é o papel com mais permissões. Não deve fazer parte da responsabilidade do sistema proteger estes ficheiros de serem apagados, uma vez que a mesma é do sistema operativo, que deveria dispor de mecanismos como esse. Se o objectivo for o de impedir a eliminação de ficheiros, deve-se optar por discos físicos de persistência permanente.

Apesar disto, é possível detectar a falta desses ficheiros, através do ficheiro de registo do protótipo que regista os ficheiros cifrados, quando não existe um ficheiro com o nome especificado e no entanto este está presente no registo, significa que este foi apagado ou movido em estado cifrado.

4.2 Autenticidade e Não Repúdio – Assinaturas Digitais Qualificadas

Para demonstrar os resultados obtidos, será mostrado como se processava anteriormente a assinatura, isto é, alterando o contexto da assinatura não fazia com que esta ficasse inválida, isto apenas acontecia se o texto fosse adulterado.

Figura 13 – Assinatura Digital válida

Podemos ver que o despacho foi assinado, aprovando a venda. Quando a informação assinada era alterada,

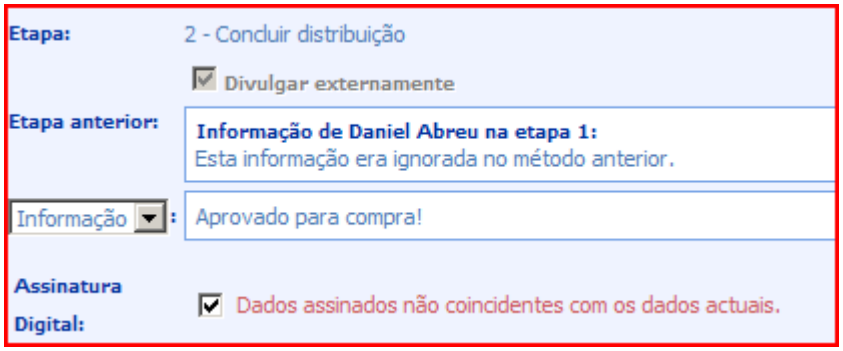

Figura 14 – Assinatura Digital com conteúdo alterado

O erro era detectado e o utilizador informado, contudo, caso se alterasse o fundamento da decisão, isto é, a informação da etapa anterior, o algoritmo não detectava a alteração:

Figura 15 – Assinatura Digital válida com dados alterados

Levando o signatário a assinar um despacho com um fundamento adulterado.

Com o novo método, caso o utilizador assine uma decisão, assim

Figura 16 – Assinatura Digital válida com novo método

Alterando o contexto do despacho resultará numa mensagem de erro, apesar da mensagem do despacho não ter sido alterada:

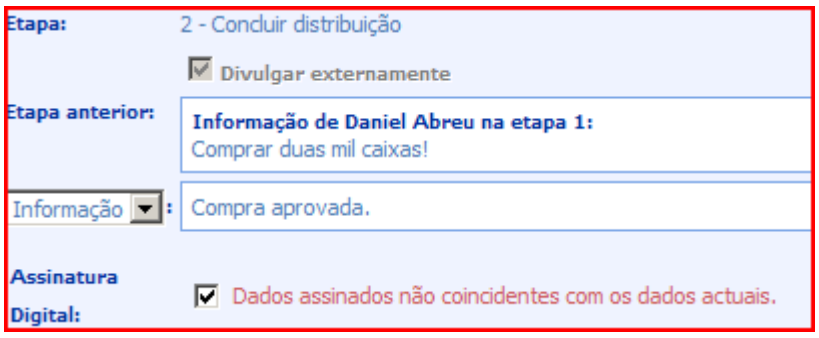

Figura 17 – Assinatura Digital inválida com dados alterados.

4.3 Documentos MS Office com mecanismos de autenticação e segurança

Tomando como exemplo um documento Office Word 2007 é possível verificar a facilidade de aplicação de assinatura digital, assim como a sua verificação.

Para assinar o documento o utilizador deve seguir os passos descritos na secção de ajuda do Office, assim:

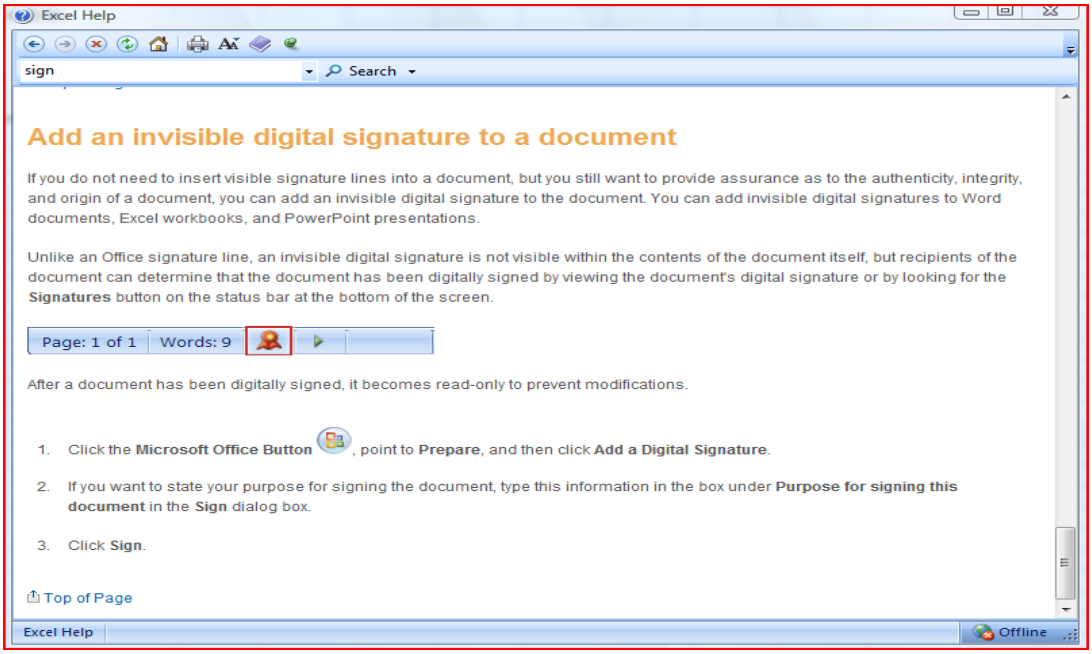

Figura 18 – Como adicionar uma assinatura digital a um documento Office

É possível verificar que assinaturas estão associadas ao documento através da barra lateral:

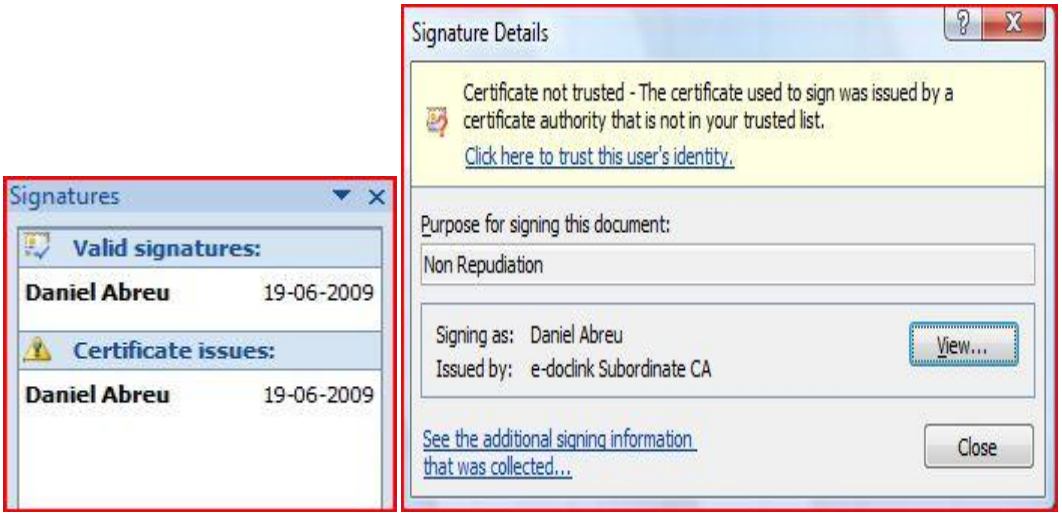

Figura 19 – Documento Office assinado digitalmente

Para cifrar o conteúdo do documento é necessário colocar uma palavra-chave que dará acesso ao conteúdo do ficheiro se introduzida correctamente quando se tenta abrir o mesmo:

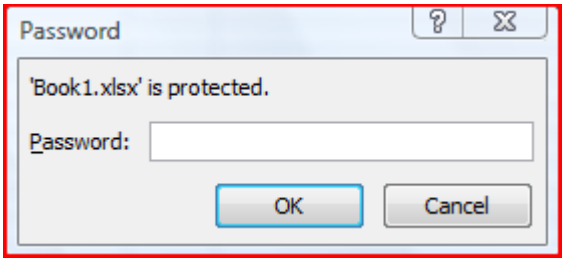

Figura 20 – Documento Office cifrado, pedido da *password*

Capítulo 5 – Discussão dos Resultados

5.1 Introdução

Este capítulo tem por objectivo comparar e discutir os resultados obtidos em comparação com o que existia anteriormente, de modo a verificar os benefícios e as limitações que advieram da implementação do protótipo, assim como da introdução das diversas medidas propostas para incrementar a segurança do sistema.

5.2 Resultados do Protótipo

5.2.1 Protecção das chaves privadas do certificado

O protótipo recorre a um par de chaves para cifrar e assinar os conteúdos, contudo, o ponto fulcral da segurança do sistema reside na chave privada que é utilizada para decifrar a chave simétrica. A quando da instalação de uma instância é essencial definir a propriedade de "não exportável" relativamente à chave privada. Por outro lado esta chave está disponível para o utilizador cuja conta foi utilizada para instalar o certificado, sendo a *password* desta conta a raiz da cadeia de chaves.

Esta limitação poderia ser ultrapassada, se fosse utilizado o "método 1" descrito no ponto 3.1.8, uma vez que a segurança não residiria apenas numa chave mas sim num número de chaves igual ou superior ao número de utilizadores do sistema, embora essa solução trouxesse outros problemas.

5.2.2 Confidencialidade - Método 1 VS Método 2

Como foi visto anteriormente adoptou-se o método 2 para adicionar segurança ao sistema e foram diversas as razões apresentadas anteriormente para esta escolha. Contudo, não está ainda posta de parte a utilização do método 1 em ambiente de produção. O método 2 apesar de concentrar a segurança numa chave única, permite uma utilização transparente para o utilizador, o mesmo não aconteceria se fosse utilizado o outro método. Permite ainda que caso algum utilizador perca a sua chave, o acesso à informação não se encontra permanentemente comprometido, uma vez que o certificado que é usado para instalado no servidor aplicacional deve ser guardado em local seguro e o acesso a esse certificado deve ser controlado.

A manutenção também é um ponto a ter em conta, com apenas um par de chaves torna-se simplificado a renovação do mesmo, podendo e devendo ser renovado com alguma frequência, contudo, é necessário percorrer os dados e voltar a cifrá-los com a nova chave pública. Como se verá posteriormente, a quantidade de dados a ser afectado pela mudança pode variar, caso se opte por um sistema de classificação de documentos, descrito no capítulo "Trabalho Futuro".

5.2.3 Árvores de Ataque

Após a introdução do protótipo, verificaram-se diversas alterações nas árvores de ataque:

Objectivo: Alterar despacho

1. Convencer o autor a alterar o conteúdo
    - 1.1 Subornar o autor
    - 1.2 Chantagear o autor
    - 1.3 Ameaçar o autor
    - 1.4 Enganar o autor
2. Alterar o despacho quando este é escrito no computador
    - 2.1 Alterar o despacho quando o autor não esteja presente
3. Alterar o despacho enquanto este passa pela rede
    - 3.1 Usar o ataque Homem-no-meio
4. Alterar o despacho quando este está guardado na BD
    - ~~4.1 Subornar o Administrador~~
    - ~~4.2 Chantagear o Administrador~~

~~4.3 Ameaçar o Administrador~~
~~4.4 Enganar o Administrador~~
~~4.5 Ter privilégios de administração~~
~~4.6 Instalar um virus/worm/keylogger no computador~~
~~4.7 Quebrar a palavra-chave de administração~~
~~4.8 Aceder aos ficheiros da BD~~
4.9 Utilizar SQL Injection
5. Convencer alguém com acesso ao despacho a alterar o conteúdo
5.1 Subornar o utilizador
5.2 Chantagear o utilizador
5.3 Ameaçar o utilizador
5.4 Enganar o utilizador

Tabela 15 – Arvore de Ataque, objectivo alterar despacho (após alteração de funcionamento)

Objectivo: Visualizar um documento
1. Convencer a autor a revelar o documento
1.1 Subornar o autor
1.2 Chantagear o autor
1.3 Ameaçar o autor
1.4 Enganar o autor
2. Convencer utilizador com acesso ao documento a revelá-lo
2.1 Subornar o utilizador
2.2 Chantagear o utilizador
2.3 Ameaçar o utilizador
2.4 Enganar o utilizador
3. ~~Visualizar o documento quando este está guardado na BD~~
~~3.1 Obter privilégios de Administração~~
~~3.1.1 Com ajuda do Administrador~~
~~3.1.1.1 Subornar o Administrador~~
~~3.1.1.2 Chantagear o Administrador~~
~~3.1.1.3 Ameaçar o Administrador~~
~~3.1.1.4 Enganar o Administrador~~
~~3.1.1.5 Ter privilégios de administração~~
~~3.1.2 Sem ajuda do Administrador~~
3.1.2.1 Atacar Active Directory
~~3.2 Obter privilégios de Consulta~~
~~3.2.1 Com ajuda do Administrador~~
~~3.2.1.1 Subornar o Administrador~~
~~3.2.1.2 Chantagear o Administrador~~
~~3.2.1.3 Ameaçar o Administrador~~
~~3.2.1.4 Enganar o Administrador~~
~~3.2.1.5 Ter privilégios de administração~~
~~3.2.2 Sem ajuda do Administrador~~
3.2.2.1 Atacar Active Directory
~~3.2.2.2 Atacar repositório de ficheiros~~
~~3.2.2.2.1 Atacar Sistema Operativo do repositório~~
~~3.2.2.2.2 Usar um live SO~~

Tabela 16 – Arvore de Ataque, objectivo visualizar um documento (após alteração de funcionamento)

Objectivo: Alterar um documento

1. Convencer a autor a revelar o documento
    - 1.1 Subornar o autor
    - 1.2 Chantagear o autor
    - 1.3 Ameaçar o autor
    - 1.4 Enganar o autor
2. Convencer utilizador com acesso ao documento a revelá-lo
    - 2.1 Subornar o utilizador
    - 2.2 Chantagear o utilizador
    - 2.3 Ameaçar o utilizador
    - 2.4 Enganar o utilizador
3. ~~Visualizar o documento quando este está guardado na BD~~
    - ~~3.1 Obter privilégios de Administração~~
        - ~~3.1.1 Com ajuda do Administrador~~
            - ~~3.1.1.1 Subornar o Administrador~~
            - ~~3.1.1.2 Chantagear o Administrador~~
            - ~~3.1.1.3 Ameaçar o Administrador~~
            - ~~3.1.1.4 Enganar o Administrador~~
            - ~~3.1.1.5 Ter privilégios de administração~~
        - ~~3.1.2 Sem ajuda do Administrador~~
            - 3.1.2.1 Atacar Active Directory
    - ~~3.2 Obter privilégios de Consulta~~
        - ~~3.2.1 Com ajuda do Administrador~~
            - ~~3.2.1.1 Subornar o Administrador~~
            - ~~3.2.1.2 Chantagear o Administrador~~
            - ~~3.2.1.3 Ameaçar o Administrador~~
            - ~~3.2.1.4 Enganar o Administrador~~
            - ~~3.2.1.5 Ter privilégios de administração~~
        - ~~3.2.2 Sem ajuda do Administrador~~
            - 3.2.2.1 Atacar Active Directory
            - ~~3.2.2.2 Atacar repositório de ficheiros~~
                - ~~3.2.2.2.1 Atacar Sistema Operativo do repositório~~
                - ~~3.2.2.2.2 Usar um live SO~~

Tabela 17 – Arvore de Ataque, objecto alterar um documento (após alteração de funcionamento)

Com a aplicação do protótipo verifica-se que os conteúdos em "descanso", isto é, na BD estão protegidos contra ataques com vista à adulteração dos mesmos. Visto que a chave se encontra no servidor aplicacional e não na base de dados, assim como o facto de essa mesma chave não sair dessa máquina impossibilita o facto de poder ser utilizada na base de dados para decifrar os dados. O único método para aceder ao conteúdo dos dados cifrados na base de dados sem utilizar a chave é o de quebrar os algoritmos de cifra.

5.2.4 DREAD

Após a implementação do protótipo e caso a entrada se dê a partir da base de dados, a tabela DREAD com escala de 1 a 10, passou a ser seguinte:

ID	Descrição	Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability	Total
1	Ver conteúdos	2	0	1	5	1	1,8
2	Alterar conteúdos	9	0	1	5	1	3,2
3	Apagar conteúdos	7	10	10	10	10	9,4

Tabela 18 – Avaliação DREAD, via Base de Dados, com utilização do protótipo

O conteúdo da tabela foi preenchido supondo que o atacante é o Administrador da Base de Dados, ou que têm privilégio de administrador.

5.2.5 Escalabilidade e Manutenção

A solução foi aplicada a pensar também na facilidade de escalar e manter a solução, com a existência de apenas 1 certificado evita-se a dificuldade de efectuar revogações de certificados expirados ou comprometidos, por outro lado, não existe o problema que se poria caso os utilizadores perdessem ou se esquecessem das *passwords* dos certificados. Fazer uma cópia de segurança de apenas 1 certificado é bastante mais fácil, assim como a recuperação do mesmo, a alteração de chaves pode também ocorrer com maior frequência.

Do ponto de vista da manutenção, a existência de diversos certificados ao longo do tempo levanta várias questões, tal como, se a chave privada for comprometida que procedimentos devem ser tomados? Ou como proceder se o certificado expirar?

Em relação à primeira pergunta deve ser utilizada a cópia de segurança do par de chaves de modo a recuperar os dados e voltar a cifrá-los com um novo par de chaves, de modo a evitar que sejam re-cifrados milhares ou milhões de conteúdos existe a possibilidade de coexistirem diversos certificados em utilização simultânea, limitando assim o impacto de uma renovação

Em relação à segunda, deve ser pedido periodicamente uma renovação do certificado ao CA, de modo a não deixar expirar o certificado.

5.3 Resultados das Assinaturas Digitais Qualificadas

5.3.1 Não Repúdio

Com a aplicação de assinaturas digitais qualificadas passou a ser possível garantir o não repúdio por parte do assinante. A alteração do método de funcionamento anterior, que só tinha em conta o próprio despacho, para um método que entra em conta com o contexto em que se insere o despacho, adicionando informação sobre os factos e pressupostos que levaram o utilizador a tomar uma decisão que posteriormente assinou.

5.3.2 Impacto no utilizador

O impacto no utilizador limita-se a um tempo de resposta que aumentou ligeiramente, devido às alterações introduzidas e às implicações que teve no sistema. No entanto, em termos de interface a alteração é transparente para o utilizador.

5.3.3 Escalabilidade e Manutenção

Com a utilização do CMS/PKCS#7, a escalabilidade deste método está assegurada, uma vez que a assinatura, chave pública e o caminho de certificação são guardadas em conjunto com os dados assinados, tornando desnecessária a existência de uma *Public Key Infrastructure* (PKI) para difusão de chaves públicas.

5.4 Mitigação de Riscos

Risco	Mitigação
Perda do certificado responsável pela cifra das chaves secretas do protótipo.	Guardar uma cópia do certificado e da chave privada correspondente, em local seguro.
Roubo da chave privada do certificado	Deve ser feito, o mais rapidamente possível, uma nova cifra de todas as chaves secretas que estavam protegidas pela chave pública deste par. O certificado anterior deve ser revogado.
Remoção intencional de conteúdos da base de dados.	Deve existir um processo de *backup* para discos externos com regularidade, de modo a mitigar esta situação.
Roubo dos ficheiros de base de dados	Não é necessário mitigar este risco, uma vez que os conteúdos mais importantes estão cifrados.
Reverse-Engeneering, Vírus, *Keyloggers* e outros, no servidor aplicacional	Em relação ao *Reverse-Engeneering*, pode optar-se por utilizar ferramentas de ofuscação de código (39), em relação aos restantes, é responsabilidade do sistema operativo garantir a separação de recursos utilizados por processos diferentes.

Tabela 19 – Mitigação de Riscos

5.5 Testes realizados

5.5.1 Especificações da máquina de testes

Os testes de desempenho foram realizados em duas máquinas, uma máquina virtual, com Windows Server 2003 EE SP1 32bits, com 2 processadores Intel Core2Duo E8400 3GHZ e com 2GB de RAM a que será chamada de VM, e numa segunda máquina *host* da 1ª com Windows Vista Business SP1 64bits, com 2 processadores Intel Core2Duo E8400 3GHZ e com 4GB de RAM a que será chamada de Host.

Importa referir que entre testes os testes foram efectuados sucessivamente, de forma a simular uma situação real de utilização.

5.5.2 Testes do Protótipo

Foram realizados vários testes relativamente à duração da computação da cifra, uma vez que apesar dos benefícios que trazidos em termos de segurança, é necessário ter em conta a disponibilidade e usabilidade.

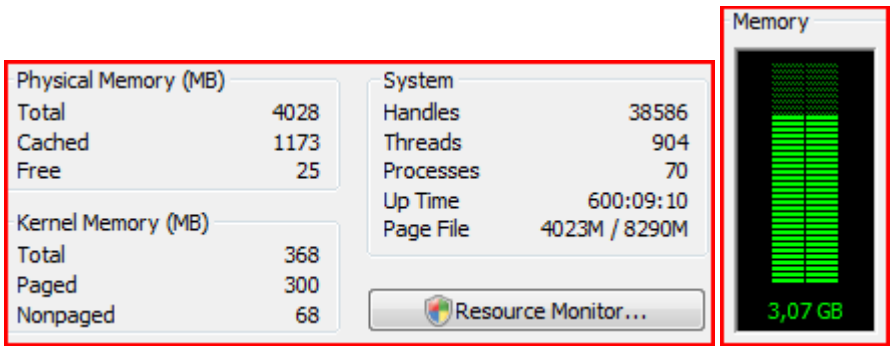

Figura 21 – Utilização da memória no HOST, antes da realização do teste.

Verificou-se experimentalmente que a cifra de documentos de diferentes tamanhos na Host demorou o seguinte:

Tamanho do Ficheiro (KB)	Tempo médio para cifrar (ms)	Tempo médio para decifrar (ms)
559	96	65
4.225	271	225
9.187	443	453
11.451	524	543
27.279	1445	1247
30.666	1736	1309
58.272	3396	3103

Figura 22 – Tempo médio de cifra com utilização do protótipo

Verifica-se, naturalmente, que com o aumento da dimensão dos documentos, aumenta também o tempo necessário para os cifrar, este problema é reduzido pela utilização de um algoritmo de chave simétrica na cifra do documento, contudo, e no caso de serem armazenados documentos de grandes dimensões o sistema poderá tornar-se lento.

Surgem por isso diversos métodos para solucionar este problema, tais como cifrar o documento assincronamente, ou classificar documentos em função da sua

confidencialidade de modo a reduzir ou aumentar a força dos algoritmos de modo a reduzir os tempos de computação das cifras. Estas soluções serão novamente abordadas no capítulo 7 - "Trabalho futuro".

5.5.3 Testes de Assinaturas Digitais Qualificadas

Foram realizados testes comparativos para verificar o impacto no desempenho com a nova versão das assinaturas digitais qualificadas no sistema, contudo o processo na sua globalidade tem diversos pontos de interacção com o utilizador, por isso, os tempos considerados compreendem o tempo decorrido entre a escolha do certificado até à interface estar novamente disponível para utilização. Estes testes foram realizados na VM.

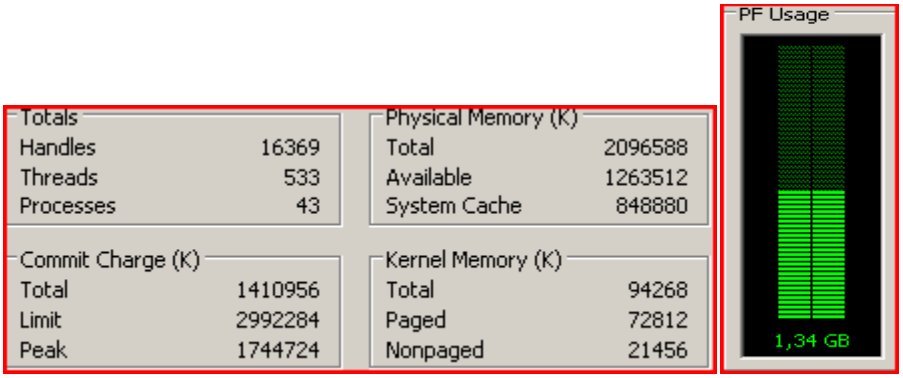

Figura 23 – Utilização da memória na VM, antes da realização do teste

A latência entre o browser e o servidor aplicacional é desprezível, uma vez que ambos se encontram na VM, a latência entre o servidor aplicacional e o servidor SQL foi, em média, inferior a 1ms.

Antes do novo método ser aplicado em média o tempo decorrido entre a escolha do certificado e a interface estar de novo disponível era de 2873 ms, com a alteração efectuada a média de tempo passou a ser de 5024 ms. A diferença de valores deve-se não apenas à alteração do algoritmo, mas também ao local onde se processa a assinatura, anteriormente, a assinatura era colocada sem ser necessário fazer avançar o processo de *workflow*, isto é, assinar não implicava que o processo avançasse para a etapa seguinte. Com a alteração efectuada a assinatura só é calculada quando o processo

avança para a próxima etapa, o que causa a execução de um maior número de operações não relacionadas com a assinatura.

5.5.4 Testes com EFS

Foram também realizados alguns testes sobre o funcionamento do EFS, nomeadamente os mecanismos de recuperação utilizando o DRA. Para tal foi realizada uma recuperação de um ficheiro de base de dados que foi protegido por esta tecnologia.

Este teste revelou que um administrador de sistema consegue facilmente recuperar um ficheiro cifrado com EFS, não sendo viável a sua utilização devido à facilidade com que a protecção pode ser ultrapassada, quer através da utilização do DRA quer pelo *reset* da *password* utilizador.

5.5.5 Testes com TDE

A realização de testes com o TDE tinha por objectivo verificar a sua adequação aos objectivos de proteger os dados do administrador, contudo, verificou-se experimentalmente que o administrador podia aceder aos dados de forma transparente, a mais-valia trazida pelo TDE verifica-se ao nível dos ficheiros da base de dados, que podem ser roubados e transportados para outra SGBD, contudo, não será possível ver o conteúdo uma vez que o TDE continua a proteger os dados. Por outro lado, tendo um impacto negativo no desempenho da base de dados na ordem dos 3~5% em média e 28% no pior cenário (40), contudo, é muito mais eficiente que o EFS que sendo genérico não está optimizado para estes ficheiros.

5.6 O que não foi feito

Fazia parte do plano inicial deste estágio o estudo da norma UBL 2.0, contudo devido a diversos desenvolvimentos, a sua utilização no contexto do sistema foi abandonada e portanto não se deu seguimento ao estudo da norma. Não houve também

disponibilidade física e temporal para se efectuarem testes às trocas de documentos entre instâncias de modo seguro, isto é, não se implementou o que ficou estipulado na fase de desenho em relação à intercomunicabilidade dos dados entre instituições.

Capítulo 6 – Conclusões

Na secção "Objectivos" foram fixadas várias perguntas às quais este relatório se propunha dar resposta:

Em resposta à primeiro pergunta – "Como e onde proteger os conteúdos?" a resposta que foi encontrada foi a de cifrar os conteúdos utilizando criptografia híbrida, sendo estes cifrados no servidor aplicacional. A segunda questão levantava a dúvida sobre "Como garantir não repúdio de conteúdos?", sendo que a resposta encontrada foi a de utilizar assinaturas digitais qualificadas sobre os conteúdos. A terceira questão – "Como garantir integridade de conteúdos?" foi respondida com recurso à utilização de assinaturas digitais e *checksums* com o objectivo de verificar apenas a integridade dos dados. A quarta questão era sobre "Como proteger dados sensíveis de configuração", também como resposta a esta pergunta encontramos as assinaturas digitais, com objectivo de detectar alterações que não foram feitas através da interface. "Como garantir segurança em documentos Office?", era a 5ª questão, a resposta dada foi a de recorrer aos mecanismos oferecidos pelo Office que permite a cifra e assinatura desses conteúdos, sem ter de recorrer a programas externos. Em resposta à 6ª questão – "Como verificar integridade de componentes do sistema" encontra-se novamente as assinaturas digitais, de forma a verificar se a componente foi ou não adulterada. Por fim a 7ª questão – "Como garantir segurança em trocas de conteúdos entre organizações", sugere-se a utilização de envelopes seguros, que recorrem a criptografia híbrida para garantir confidencialidade, integridade e não repúdio.

No decorrer deste relatório de estágio foram analisadas várias ferramentas, com o objectivo de adicionar segurança ao sistema de estudo prático *e-doc**link***. Começou-se por definir um plano de trabalhos onde se procurou contemplar várias vertentes da segurança de um sistema de gestão documental, foram analisados os alicerces de um sistema seguro, tais como as propriedades fundamentais da segurança, os princípios de

desenho de Saltzer e Schroeder, assim como a perspectiva da Microsoft relativamente à segurança e como nota histórica, o segundo princípio de Kerckhoff. Foram descritos sucintamente os algoritmos AES, RSA, SHA, RNG e ECDSA, que foram aplicados na solução. As tecnologias EFS, TDE e BDE foram também analisadas e alvos de ensaios técnicos, que revelaram a sua não aplicabilidade no contexto prático deste trabalho, devido à transparência de utilização para utilizadores com quaisquer privilégios, como o TDE e BDE.

Foi realizada a análise das necessidades de segurança do sistema com recurso à caracterização das ameaças, análise do plano inicial e ferramentas de modelação de ameaças, como a elaboração de Arvores de Ataque e tabelas STRIDE e DREAD. Foram propostas diversas melhorias de segurança ao sistema que garantem mais confidencialidade e integridade, algumas foram implementadas no sistema, como é o caso da Assinatura Digital com garantia de Não Repúdio, outras em protótipo, como é o caso da Cifra de conteúdos com SHA-RSA, Assinatura Digital de conteúdos com ECDSA, Mecanismo de Integridade para dados sensíveis de Configuração com recurso a Assinatura Digital com ECDSA. Apesar de não implementada, foi proposta a utilização de Envelopes Seguros para transferências de conteúdos entre Organizações.

A implementação das várias medidas está também dependente da necessidade dos clientes, assim como a capacidade de aceitar o *trade-off* entre o impacto no desempenho e benefícios na segurança.

Efectivamente, apesar das medidas de segurança implementadas ou recomendadas, um atacante cujo papel seja o de administrador, poderá eventualmente mais cedo ou mais tarde alcançar o objectivo que pretende, tudo dependendo do esforço que esteja disposto a despender.

Law #1: If a bad guy can persuade you to run his program on your computer, it's not your computer anymore
Law #2: If a bad guy can alter the operating system on your computer, it's not your computer anymore
Law #3: If a bad guy has unrestricted physical access to your computer, it's not your computer anymore
Law #4: If you allow a bad guy to upload programs to your website, it's not your website any more
Law #5: Weak passwords trump strong security
Law #6: A computer is only as secure as the administrator is trustworthy
Law #7: Encrypted data is only as secure as the decryption key
Law #8: An out of date virus scanner is only marginally better than no virus scanner at all
Law #9: Absolute anonymity isn't practical, in real life or on the Web
Law #10: Technology is not a panacea

Figura 24 - 10 immutable laws of security by Microsoft (41)

Não obstante estas medidas são essenciais, uma vez que obrigam a aumentar o esforço do atacante para conseguir atingir o objectivo. Muitas vezes optará certamente por não tentar vencer a criptografia, e tentará usar outras formas de ataque, mais baratas e mais rápidas. Foram por isso seguidos alguns princípios de desenho que visam limitar os pontos vulneráveis por onde o atacante possa contornar a criptografia. Em suma optou-se por utilizar uma estratégia de defesa em profundidade, para que o atacante tenha que vencer não apenas um mecanismo defensivo mas vários.

Capítulo 7 – Trabalho Futuro

Um dos pontos que mais penaliza o desempenho é o facto da cifra de documentos não distinguir os documentos que devem obrigatoriamente ser tornados confidenciais e os documentos cuja confidencialidade não é importante. Neste sentido deveria ser criado um meio automático ou manual de classificação de documentos, e dependendo do nível de confidencialidade requerido assim se processaria a cifra. Por exemplo, um sistema à semelhança do sistema de classificação militar, variável entre não confidencial até altamente secreto, sendo que no primeiro caso não existiria qualquer tipo de cifra, apenas verificação de integridade e sendo o sendo o caso em que existiria o maior nível de confidencialidade e integridade.

Outro ponto importante é a questão da interface ficar parada enquanto é feito o *upload* e posteriormente a cifra. Poderia optar-se por fazer esta transferência/cifra em *background*, de forma a não bloquear o utilizador, podendo existir um local onde este tenha a informação sobre o estado do upload/cifra do documento.

Do ponto de vista dos algoritmos, o ponto de maior interesse futuro é o da substituição de algoritmos mais antigos e mais lentos por mais recentes e rápidos, um desses caso é a substituição do velhinho RSA por um algoritmo que utilize curvas elípticas. Num sentido diferente é a substituição do SHA1, cuja segurança aparenta estar comprometida por outro da mesma "família" mas mais seguro, como o SHA512. No sentido de encontrar os algoritmos da próxima geração devem ser seguidos os estudos quer do NIST quer da União Europeia com o ICT, principalmente o projecto ECRYPT.

Referências

1. **CC.** Common Criteria. *CC*. [Online] http://www.commoncriteriaportal.org/.

2. **PCI.** PCI Security Standards Council. *PCI DSS*. [Online] 15 de 12 de 2004. [Citação: 8 de 7 de 2009.] https://www.pcisecuritystandards.org/.

3. **ISO.** Internation Organization for Standardization. *ISO*. [Online] http://www.iso.org/iso/home.htm.

4. Health Insurance Portability and Accountability Act. *HIPAA*. [Online] 1996. [Citação: 8 de 7 de 2009.] http://www.hipaa.org/.

5. **Oracle.** Oracle Database Vault. [Online] Oracle. [Citação: 8 de 7 de 2009.] http://www.oracle.com/technology/deploy/security/database-security/database-vault/index.html.

6. **Veríssimo, P. e Rodrigues, L.** *Distributed Systems for System Architects*. s.l. : Kluwer Academic Publishers, 2001.

7. **Zuquete, A.** *Segurança em Redes*. s.l. : FCA, 2006.

8. *The Protection of Information in Computer Systems*. **Saltzer, J. e Schroeder, M.** s.l. : IEEE, Setembro 1975. Proceedings of the IEEE. Vol. 63 No 9.

9. **Microsoft.** Building a Secure Platform for Trustworthy Computing. *Whitepaper*. Redmond : Microsoft, 2002.

10. —. SDL Introduction. *MSDN*. [Online] [Citação: 8 de 7 de 2009.] http://msdn.microsoft.com/en-us/library/cc307406.aspx.

11. **OWASP.** Threat Risk Modeling. *OWASP*. [Online] [Citação: 8 de 7 de 2009.] http://www.owasp.org/index.php/Threat_Risk_Modeling.

12. *La Cryptographie Militaire*. **Kerckhoff, A.** s.l. : Journal des Sciences Militaires, 1883.

13. **Daemen, J. e Rijmen, V.** *The Design of Rijndael*. s.l. : Springer-Verlag, 2002.

14. **Bernstein, D.** *Cache-timing attacks on AES*. Chicago : The University of Illinois, 2005.

15. *Related-Key Impossible Differential Attacks on Reduced-Round AES-256.* **Zhang, W., Wu, W. e Zhang, L.** s.l. : Journal of Software, 2007.

16. **Oswald, E., Daemen, J. e Rijmen, V.** *AES - The State of the Art of Rijndael's Security.* 2002.

17. **Rivest, R., Shamir, A. e Adleman, L.** A method for obtaining Digital Signatures and Publick Key Cryptosystems. *Communications of the ACM.* 1978.

18. *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.* **ElGamal, T.** Santa Barbara : Springer-Verlag, 1985. Proceedings of CRYPTO 84 on Advances in Cryptology.

19. **Cochran, M.** *Notes on the Wang et al. 2^63 SHA-1 differential path.* s.l. : University of Colorado, 2007. http://eprint.iacr.org/2007/.

20. *Elliptic Curve Cryptosystems.* **Koblitz, N.** s.l. : Mathmatics of Computation, 1987, Vol. 48.

21. *Use of elliptic curves in cryptography.* **Miller, V.** s.l. : Springer-Verlag, 1986. Advances in Cryptology - CRYPTO 85.

22. **IST.** *ECRYPT - Yearly Report on Algorithms and Keysizes.* s.l. : IST, 2008. http://www.ecrypt.eu.org/ecrypt1/documents/D.SPA.28-1.1.pdf.

23. **NIST.** *Recommendation for Key Management Part 1: General.* s.l. : NIST, 2007. http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part1.pdf.

24. **Microsoft.** Encrypting File System. *Technet.* [Online] [Citação: 8 de 7 de 2009.] http://technet.microsoft.com/en-us/library/cc721923(WS.10).aspx.

25. —. Transparent Data Encryption. *MSDN.* [Online] [Citação: 8 de 7 de 2009.] http://msdn.microsoft.com/en-us/library/cc278098.aspx.

26. —. BitLocker Drive Encryption. *Technet.* [Online] [Citação: 8 de 7 de 2009.] http://technet.microsoft.com/en-us/library/cc731549(WS.10).aspx.

27. —. Kerberos. *Technet.* [Online] [Citação: 8 de 7 de 2009.] http://technet.microsoft.com/en-us/library/cc753173(WS.10).aspx.

28. *Risk Mitigation Strategies: Lessons Learned from Actual Attacks.* **RSA Security.** s.l. : RSA Security Inc., 2008. RSA Conference. http://www.cert.org/insider_threat/.

29. *Why Cryptosystems Fail.* **Anderson, R.** 1993. Conference on Computer and Communications Security.

30. *Attack Trees: Modeling Security Threats.* **Schneier, B.** s.l. : Dr. Dobb's Journal, Dezembro 1999.

31. **Moore, A., Ellison, R. e Linger, R.** *Attack Modelling for Information Security and Survivability*. s.l. : Software Engeneering Institute - Carnegie Mellon, 2001.

32. **RSA.** PKCS #7: Cryptographic Message Syntax Standard. *RSA Laboratories*. [Online] [Citação: 8 de 7 de 2009.] http://www.rsa.com/rsalabs/node.asp?id=2129. Rfc 3852.

33. **RSA Security.** *Securing Data at Rest: Developing a Database Encryption Strategy*. s.l. : RSA Security Inc, 2007. Whitepaper. http://www.rsa.com/products/bsafe/whitepapers/DDES_WP_0702.pdf.

34. **Microsoft.** 10 Immutable Laws of Security. *Technet*. [Online] 7 de 2009. [Citação: 8 de 7 de 2009.] http://technet.microsoft.com/en-us/library/cc722487.aspx.

35. **Schneier, B.** Secrecy, Security and Obscurity. *Crypto-Gram Newsletter*. 2002.

36. **Microsoft.** Database Encryption in SQL Server 2008 EE - 4. Impact on the Database. *Technet*. [Online] [Citação: 8 de 7 de 2009.] http://msdn.microsoft.com/en-us/library/cc278098.aspx.

37. **RSA Security.** PKCS #5: Password-Based Cryptography Standard. *RSA Laboratories*. [Online] [Citação: 8 de 7 de 2009.] http://www.rsa.com/rsalabs/node.asp?id=2127. Rfc 2898.

38. *Watermarking, Tamper-Proofing and Obfuscation - Tools for Software Protection*. **Collberg, C. e Thomborson, C.** 8, 2002 : IEEE Transactions on Software Engineering, Vol. 28.

39. **Miranda, José Pina.** Cartão do Cidadão. [Online] 17 de 8 de 2007. [Citação: 8 de 7 de 2009.] http://pki.cartaodecidadao.pt/publico/politicas/MULTICERT_PJ.CC_24.1.2_0009_pt_AsC.pdf.

40. **Schneier, B.** Ever Better Cryptanalytic Results Against SHA-1. *Schneier on Security*. [Online] 6 de 2009. [Citação: 8 de 7 de 2009.] http://www.schneier.com/blog/archives/2009/06/ever_better_cry.html.

41. —. New Cryptanalytic Results Against SHA-1. *Schneier on Security*. [Online] 17 de 8 de 2005. [Citação: 8 de 7 de 2009.] http://www.schneier.com/blog/archives/2005/08/new_cryptanalyt.html.

42. **Yuval, G.** How to Swindle Rabin. *Cryptologia*. 1979, Vol. 3.

Índice Remissivo

E

I

M

N

P

R

S

T

U

Anexos

- Diagrama de Classes do Protótipo
- Mapa de Gaant

Diagrama de Classes do Protótipo

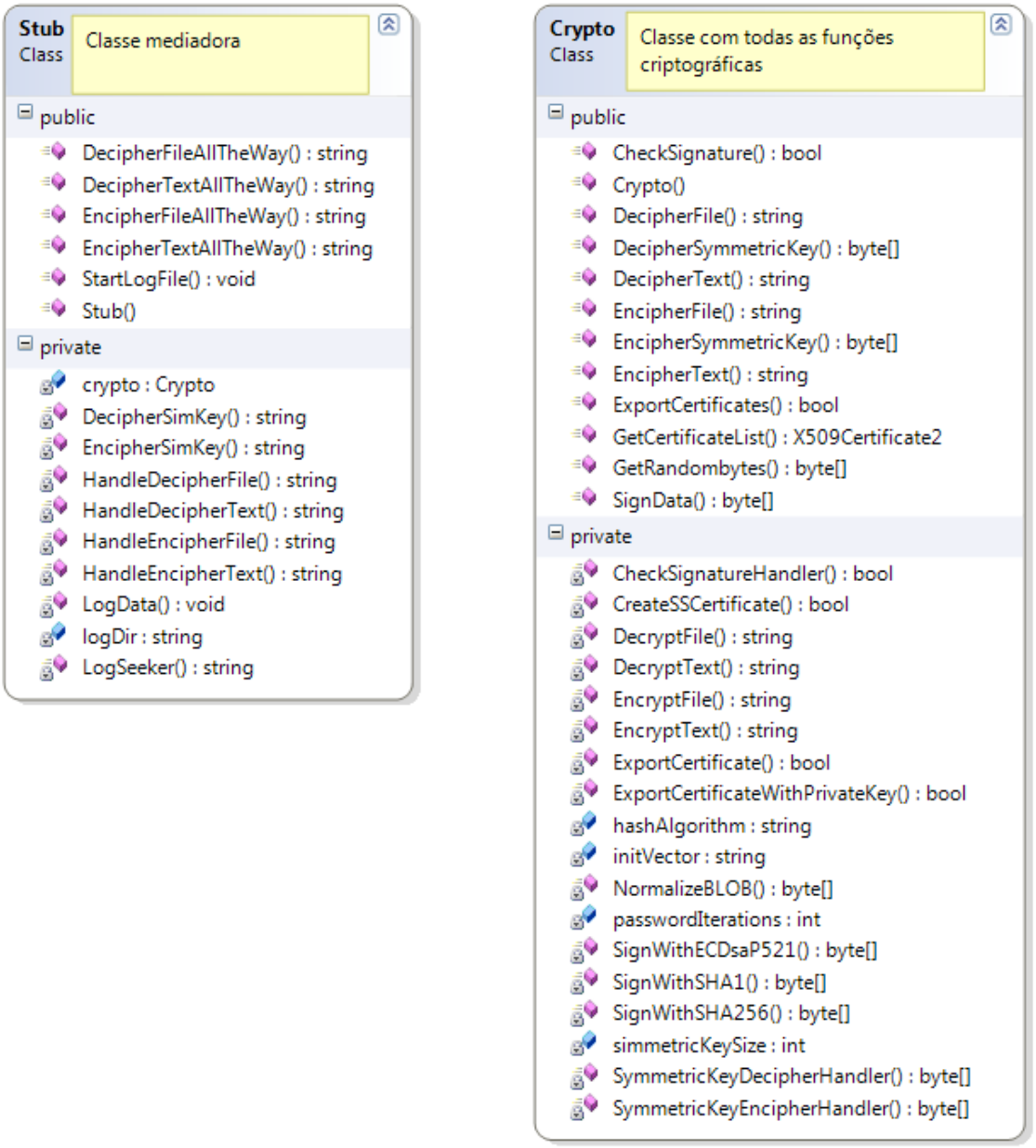

Figura 25 – Diagrama de classes

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="RSACertificate" value="RSAtest" />
    <add key="ECDSACertificate" value="ECDSAtest"/>
    <add key="keyPairName" value="myKeyPair"/>
  </appSettings>
</configuration>
```

Figura 26 – Ficheiro XML de configuração

Mapa de Gaant

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Fase 1 - Investigação e Ensaios Técnicos**	**1168 hrs?**	**Wed 01-10-08**	**Wed 22-04-09**		
2	**Objectivos Práticos**	**704 hrs**	**Wed 01-10-08**	**Fri 30-01-09**		
3	Metadados confidenciais em bases de dados	117,33 days	Wed 01-10-08	Fri 30-01-09		
4	Garantia de autenticidade e não repúdio de dados estruturados	96 days	Thu 23-10-08	Fri 30-01-09		
5	Armanezamento de documentos - confidencialidade/protecção de acesso	84 days	Wed 05-11-08	Fri 30-01-09		
6	**Tecnologias, Práticas e Normas a analisar**	**1168 hrs?**	**Wed 01-10-08**	**Wed 22-04-09**		
7	**Práticas e técnicas mais adequadas no planeamento e implementação de SGBDs**	**704 hrs?**	**Wed 01-10-08**	**Fri 30-01-09**		
8	Oracle Vault	4 days?	Fri 03-10-08	Tue 07-10-08		
9	SQL Server 2008 - Segurança	54,67 days?	Fri 03-10-08	Sun 30-11-08		
10	SQL Server 2005 - Segurança	76 days?	Wed 01-10-08	Thu 18-12-08		
11	Ofuscação	16 days?	Thu 13-11-08	Fri 28-11-08		
12	Vários artigos e documentos	114,67 days?	Fri 03-10-08	Fri 30-01-09		
13	**Segurança e meios de cifra no armazenamento de ficheiros**	**504 hrs?**	**Fri 03-10-08**	**Tue 30-12-08**		
14	SQL Server 2008 - FileStreams	53,33 days?	Wed 05-11-08	Tue 30-12-08		
15	Bitlocker	8 days?	Fri 03-10-08	Fri 10-10-08		
16	EFS	76 days?	Mon 13-10-08	Tue 30-12-08		
17	SYSKEY	18,67 days?	Tue 18-11-08	Sat 06-12-08		
18	Vários artigos e documentos	84 days?	Fri 03-10-08	Tue 30-12-08		
19	**Evolução de mecanismos existentes em ambientes de produção**	**520 hrs?**	**Mon 03-11-08**	**Fri 30-01-09**		
20	Árvores de ataque	28 days?	Fri 02-01-09	Fri 30-01-09		
21	Análise dos mecanismos existentes	86,67 days?	Mon 03-11-08	Fri 30-01-09		
22	**Tipos e algoritmos de criptografia**	**704 hrs?**	**Wed 01-10-08**	**Fri 30-01-09**		
23	Criptografia Simétrica	114,67 days?	Fri 03-10-08	Fri 30-01-09		
24	Criptografia Assimétrica	114,67 days?	Fri 03-10-08	Fri 30-01-09		
25	Criptografia Híbrida	68 days?	Fri 21-11-08	Fri 30-01-09		
26	SHA	6,67 days?	Mon 01-12-08	Fri 05-12-08		
27	Criptografia de Curva Elíptica	9,33 days?	Thu 01-01-09	Fri 09-01-09		
28	RSA	4 days?	Mon 08-12-08	Wed 10-12-08		
29	MD5	4 days?	Wed 10-12-08	Fri 12-12-08		
30	AES - Rijndael	13,33 days?	Mon 20-10-08	Fri 31-10-08		
31	Outros - Blowfish, Twofish, Elgamal, IDEA, etc	10,67 days?	Mon 03-11-08	Wed 12-11-08		
32	Shared Secret	8 days?	Wed 12-11-08	Wed 19-11-08		
33	Steganography	9,33 days?	Thu 20-11-08	Fri 28-11-08		
34	Vários artigos e documentos	117,33 days?	Wed 01-10-08	Fri 30-01-09		
35	**Estudo de meios de defesa e ataque e sua tipificação**	**176 hrs?**	**Thu 01-01-09**	**Fri 30-01-09**		
36	Arvores de ataque	28 days?	Fri 02-01-09	Fri 30-01-09		
37	Ataques mais utlizados	29,33 days?	Thu 01-01-09	Fri 30-01-09		
38	Vários artigos e documentos	29,33 days?	Thu 01-01-09	Fri 30-01-09		
39	**Verificação de integridade**	**360 hrs?**	**Mon 01-12-08**	**Fri 30-01-09**		
40	**Assinaturas Digitais**	**240 hrs?**	**Mon 01-12-08**	**Fri 09-01-09**		
41	ECDSA	6,67 days?	Mon 05-01-09	Fri 09-01-09		

ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
42	✓	RSA-SHA1	10,67 days?	Mon 01-12-08	Wed 10-12-08		
43	✓	**Funções de Sintese**	**64 hrs?**	**Mon 05-01-09**	**Wed 14-01-09**		
44	✓	SHA-1	4 days?	Mon 05-01-09	Wed 07-01-09		
45	✓	MD5	4 days?	Wed 07-01-09	Fri 09-01-09		
46	✓	SHA-2	4 days?	Mon 12-01-09	Wed 14-01-09		
47	✓	**Checksum Criptográfico**	**48 hrs?**	**Wed 14-01-09**	**Wed 21-01-09**		
48	✓	MAC	4 days?	Wed 14-01-09	Fri 16-01-09		
49	✓	MIC	4 days?	Mon 19-01-09	Wed 21-01-09		
50	✓	**Registo de Eventos**	**104 hrs?**	**Wed 14-01-09**	**Fri 30-01-09**		
51	✓	Vários Papers	17,33 days?	Wed 14-01-09	Fri 30-01-09		
52	✓	**Implementação**	**272 hrs?**	**Tue 16-12-08**	**Fri 30-01-09**		
53	✓	Implementação prototipo (Adler + HMAC) em C	45,33 days?	Tue 16-12-08	Fri 30-01-09		
54	✓	**Normas**	**584 hrs?**	**Mon 12-01-09**	**Wed 22-04-09**		
55	✓	Common Criteria	6,67 days?	Mon 12-01-09	Sat 17-01-09		
56	✓	PCI DSS - Payment Card Industry Data Security Standard	6,67 days?	Mon 12-01-09	Sat 17-01-09		
57	✓	SOX - Sarbanes-Oxley security measures	2,67 days?	Mon 20-04-09	Tue 21-04-09		
58	✓	ISO/IEC 15408	2,67 days?	Tue 21-04-09	Wed 22-04-09		
59	✓	NIST - Recommendation for Key Management	12 days?	Tue 20-01-09	Fri 30-01-09		
60	✓	**Relatório Preliminar**	**240 hrs?**	**Wed 22-10-08**	**Tue 02-12-08**		
61	✓	Elaboração do relatório preliminar	40 days?	Wed 22-10-08	Tue 02-12-08		
62	✓	Outras	48 hrs?	Wed 26-11-08	Wed 03-12-08		
63	✓	**Fase 2 – Implementação**	**992 hrs?**	**Mon 03-11-08**	**Thu 23-04-09**		
64	✓	**Planeamento e validações de segurança aplicacional**	**864 hrs?**	**Mon 03-11-08**	**Wed 01-04-09**		
65	✓	**Assinatura digital de conteúdos**	**464 hrs?**	**Tue 09-12-08**	**Thu 26-02-09**		
66	✓	Protótipo com ECDSA	50,67 days?	Tue 09-12-08	Thu 29-01-09		
67	✓	Protoripo com RSA-SHA1	28 days?	Thu 29-01-09	Thu 26-02-09		
68	✓	Protecção de dados sensíveis de configuração	144 days	Mon 03-11-08	Wed 01-04-09		
69	✓	**Mecanismos de cifra de dados**	**200 hrs?**	**Thu 20-11-08**	**Wed 24-12-08**		
70	✓	Prótotipo com AES e RSA	33,33 days?	Thu 20-11-08	Wed 24-12-08		
71	✓	Mecanismos de segurança e reconhecimento de confiança em componentes	28 days	Thu 18-12-08	Thu 15-01-09		
72	✓	**Planeamento e validações de segurança na produção de documentos**	**488 hrs?**	**Thu 20-11-08**	**Thu 12-02-09**		
73	✓	Produção de documentos com mecanismos de autenticação e segurança (ambientes Microsoft / Office)	12 days	Mon 02-02-09	Thu 12-02-09		
74	✓	**Evolução de mecanismos de armazenamento e gestão de ficheiros**	**408 hrs?**	**Thu 20-11-08**	**Thu 29-01-09**		
75	✓	Prototipo com AES, RSA-SHA1	68 days?	Thu 20-11-08	Thu 29-01-09		
76	✓	**Relatório Final**	**192 hrs?**	**Mon 02-02-09**	**Thu 05-03-09**		
77	✓	Elaboração do Relatório Final - Draft	32 days?	Mon 02-02-09	Thu 05-03-09		
78	✓	Outras	48 days	Thu 05-03-09	Thu 23-04-09		
79		**Fase 3 – Segurança inter-aplicacional e inter-organizacional**	**392 hrs?**	**Thu 23-04-09**	**Tue 30-06-09**		
80	✓	Mecanismos de garantia de segurança na intercomunicabilidade de dados	36 days	Thu 23-04-09	Fri 29-05-09		
81	✓	Garantia de autenticidade e não repúdio na troca de documentos entre organizações	36 days	Thu 23-04-09	Fri 29-05-09		
82		Análise de standard UBL 2.0	0 days	Thu 23-04-09	Thu 23-04-09		

ID		Task Name	Duration	Start	Finish	Predecessors	Resource Names
83	✓	**Relatório Final**	**392 hrs?**	**Thu 23-04-09**	**Tue 30-06-09**		
84	✓	Elaboração do Relatório Final	65,33 days?	Thu 23-04-09	Tue 30-06-09		
85	✓	Outras	36 days	Thu 23-04-09	Fri 29-05-09		
86	✓	**Fase 4 – Relatório Final de Estágio**	**240 hrs?**	**Mon 01-06-09**	**Fri 10-07-09**		
87	✓	Conclusão da elaboração do relatório	40 days?	Mon 01-06-09	Fri 10-07-09		