UNIVERSIDADE DE ÉVORA

Mestrado em Engenharia Informática

Um Sistema de Troca Segura de Mensagens com Garantias Fortes de Auditabilidade

Miguel Reis Calisto da Silva

orientador: *Prof. Doutor António Eduardo Dias*

Setembro de 2004

Esta dissertação não inclui as críticas e sugestões feitas pelo júri.

UNIVERSIDADE DE ÉVORA

Mestrado em Engenharia Informática

Um Sistema de Troca Segura de Mensagens com Garantias Fortes de Auditabilidade

Miguel Reis Calisto da Silva

orientador: *Prof. Doutor António Eduardo Dias*

Setembro de 2004

Esta dissertação não inclui as críticas e sugestões feitas pelo júri.

Prefácio

Este documento contém uma dissertação intitulada *"Um Sistema de Troca Segura de Mensagens com Garantias Fortes de Auditabilidade"*, um trabalho do aluno *Miguel Reis Calisto da Silva*[1], estudante de Mestrado em Engenharia Informática na Universidade de Évora.

O orientador deste trabalho é o Professor Doutor António Eduardo Dias[2], do Departamento de Informática da Universidade de Évora.

O autor do trabalho é licenciado em Engenharia Informática, pela Universidade de Évora. A presente dissertação foi entregue em Setembro de 2004.

[1] mreis@isp.novis.pt
[2] aed@di.uevora.pt

Agradecimentos

Durante a dissertação apresentada neste documento muitos foram os momentos em que necessitei de um elevado grau de concentração e isolamento. Devido a esses factores, não raras foram as vezes em que deixei as pessoas que fazem parte da minha vida para segundo plano. É para essas pessoas a mensagem que deixo de seguida.

Carla, foste sem dúvida a maior prejudicada pela minha sede de investigação e conhecimento. No entanto sei que compreendes que não podemos negar aquilo que somos e o que queremos. Embora as palavras sejam sempre pequenas demais para fazerem justiça, agradeço-te de todo o coração.

Mãe, Pai, Tita e Avó, muito obrigado pela compreenção e apoio que me deram deste o primeiro momento. Não foi possível passar mais tempo com vocês, mas não se preocupem, pois o Alentejo fica já ali...

Artur, acima de tudo, este trabalho também é teu! Obrigado pela tua profunda compreensão, apoio e conhecimento.

Professor Eduardo Dias, obrigado pela disponibilidade que demonstrou ao aceitar ser meu orientador. José Carlos e Nuno, obrigado por terem a paciência de reverem este trabalho. A todos vocês, ao Saias, ao David e ao Rui, obrigado pelo companheirismo académico e muitos anos de amizade. Um agradecimento também ao pessoal dos Trusted Services.

Sumário

Os sistemas de troca de mensagens tradicionais, em particular o correio electrónico, não possuem garantias de segurança suficientes para satisfazer grande parte dos requisitos colocados em alguns âmbitos mais exigentes, como o militar ou o comércio electrónico entre empresas. Mesmo o chamado "correio electrónico seguro"não é suficiente, uma vez que proporciona apenas as garantias de autenticação, integridade, confidencialidade e não-repúdio de origem.

Ficam a faltar garantias como o não-repúdio de submissão e de recepção, bem como a auditabilidade confiável dos sistemas que asseguram o transporte e entrega das mensagens. Adicionalmente, são necessárias funcionalidades que assegurem a efectiva entrega das mensagens, ou avisos relativos à impossibilidade desta entrega, e ainda o arquivo fiável e seguro (e.g., confidencial) das mensagens para efeitos legais e/ou de disponibilidade por prazos alargados.

O trabalho a efectuar no âmbito desta tese visa o desenvolvimento e implementação de um sistema de troca de mensagens com todas as garantias mencionadas acima. Terá como base os sistemas e protocolos teóricos em áreas relevantes (em particular, a criptografia) e a tecnologia já existente, de forma a construir um sistema completo e coerente, com as características pretendidas, a partir de um conjunto de elementos dispersos na área da segurança.

A Secure Messaging System With Strong Guarantees of Auditability

ABSTRACT

Common messaging systems, particularly electronic mail, do not possess enough security guarantees to satisfy most of the requirements on security demanding areas, such as military or business to business electronic commerce. Even the so-called "secure electronic mail"is not enough, since it only satisfies some requirements, such as authentication, integrity, confidentiality and non-repudiation of origin.

Stronger security requirements, like non-repudiation of submission and non-repudiation of receipt, together with trusted auditability from the message transportation and delivery systems, are not guaranteed at all. Furthermore, it is fundamental to assure the effective message delivery, or some warning about the impossibility of delivery, as well as reliable and secure (e.g., confidential) message archiving, needed for legal effects and long term availability.

The work described in this thesis includes the development and implementation of a messaging system that provides the security guarantees presented above. It will be based on systems and theoretic protocols in relevant areas (in particular, cryptography) and in widely available technology, in order to build a complete and coherent system meeting the intended guarantees, starting from a wide set of elements related to the security area.

Conteúdo

Prefácio i

Agradecimentos iii

Sumário v

Abstract vii

Conteúdo ix

Lista de Figuras xiii

Lista de Tabelas xv

1 Introdução 1
- 1.1 Enquadramento e Motivação . . . 1
- 1.2 Objectivos . . . 3

2 Criptografia e Conceitos Relacionados 5
- 2.1 Criptografia . . . 5
  - 2.1.1 Criptografia de Chave Simétrica . . . 5
  - 2.1.2 Algoritmos de Sumário . . . 6
  - 2.1.3 Criptografia de Chave Pública . . . 8
  - 2.1.4 Assinaturas Digitais . . . 9
  - 2.1.5 Certificados Digitais . . . 9
  - 2.1.6 Autoridade de Certificação . . . 11
- 2.2 Validação Cronológica . . . 14
- 2.3 Árvores de Merkle . . . 16
- 2.4 Garantias de Segurança . . . 17
  - 2.4.1 Integridade . . . 17
  - 2.4.2 Autenticação . . . 17
  - 2.4.3 Confidencialidade . . . 18

2.4.4 Não-repúdio 18
2.4.5 Auditabilidade 20

3 Trabalho Relacionado **23**
3.1 Não-repúdio 23
3.1.1 Participação Total da Entidade de Confiança 25
3.1.2 Participação Parcial da Entidade de Confiança 27
3.1.3 Ausência de Participação da Entidade de Confiança 29
3.2 Auditabilidade 33
3.2.1 Comercio Electrónico 33
3.2.2 Nomes Seguros 35
3.2.3 Dinheiro Electrónico 37
3.3 Garantias de Segurança em Sistemas de Troca de Mensagens 39

4 Um Novo Esquema de Auditabilidade **43**
4.1 Requisitos de Segurança 43
4.2 Esquema de Auditabilidade 44
4.2.1 Notação 44
4.2.2 Constituição do Repositório 45
4.2.3 Verificação da Integridade de Registos 48
4.3 Partição Hierárquica dos Registos 49
4.3.1 Definições 49
4.3.2 Verificação da Integridade de Registos 50
4.4 Registos de Topo 51
4.5 Análise da Segurança 52
4.6 Eficiência 54
4.7 Comparação com Árvores de Merkle 55
4.8 Um Sistema Completo 56

5 Implementação **59**
5.1 Protolocos, Estruturas e Algoritmos 59
5.2 Tecnologia Utilizada 61
5.3 Arquitectura 62
5.3.1 Módulos 62
5.3.2 Repositório 64
5.3.3 Manutenção dos registos de topo 66
5.3.4 Sistema Criptográfico 67
5.3.5 Interacção com a Autoridade de Certificação 68
5.3.6 Gestão de Chaves e Certificados Digitais 71
5.4 Inserção e Validação de Dados 72
5.4.1 Processamento de Mensagens 72

5.4.2 Processo de Validação do Sistema de Auditabilidade 77
5.5 Interacção com Clientes 79
5.5.1 Protocolos de Comunicação 82
5.5.2 Definição de Etiquetas 83
5.5.3 Formato das Mensagens 84
5.6 Desempenho do Sistema 86

6 Caso de Estudo **89**
6.1 Requisitos Legais 89
6.2 Apresentação do Sistema 90
6.3 Arquitectura 93
6.4 Integração com o Protocolo de Não-Repúdio 95
6.5 Aplicação do Esquema de Auditabilidade 96

7 Conclusões **99**
7.1 Objectivos Alcançados 99
7.2 Discussão e Comparação com Trabalhos Relacionados 100
7.3 Limitações 102
7.4 Trabalho Futuro 102

A Interfaces **105**

Referências **109**

Lista de Figuras

2.1 Representação da estrutura hierárquica dos certificados digitais . . . 13
2.2 Cadeia de certificação do certificado da entidade $CN = C, O = CC$ 13
2.3 Entidades intervenientes na emissão de um selo temporal 15
2.4 Representação dos nós de uma Árvore de Merkle 16
2.5 Caminho de confiança 17
2.6 Emissores e receptores das diferentes provas de não-repúdio 20

4.1 Repositório com número não fixo de registos R_m por época 47
4.2 Repositório baseado num esquema hierárquico 50
4.3 Conjunto de registos de topo existentes no repositório num determinado momento 52
4.4 Violação do registo $R_{m_{<3>}}$ 53

5.1 Arquitectura do Esquema 64
5.2 Inicialização do Módulo de Gestão Criptográfica 68
5.3 Extensões dos Certificados 70
5.4 Diagrama de sequência do procedimento de inserção de um registo R_m (continua) 75
5.5 Diagrama de sequência do procedimento de inserção de um registo R_m (continuação) 76
5.6 Diagrama de sequência do procedimento de validação de um registo R_m (continua) 80
5.7 Diagrama de sequência do procedimento de validação de um registo R_m (continuação) 81

6.1 Módulos do Serviço 94
6.2 Interacções entre intervenientes no protocolo de não-repúdio e inserções de mensagens no Esquema 96

Lista de Tabelas

5.1 Número de elementos que compõem a cadeia de validação de $R_{m_{<n>}}$ quando o número médio de registos por sub-conjunto é 5 86
5.2 Número de elementos que compõem a cadeia de validação de $R_{m_{<n>}}$ quando o número médio de registos por sub-conjunto é 9 87

Capítulo 1

Introdução

Este primeiro capítulo apresenta uma introdução sobre as áreas abrangidas pelo trabalho desenvolvido nesta dissertação. Mais concretamente, na secção 1.1 é apresentado o enquadramento do trabalho proposto e descrita a motivação que levou à realização do mesmo. Na secção 1.2 são descritos os objectivos definidos para o trabalho.

1.1 Enquadramento e Motivação

Cada vez mais a tecnologia está presente em todos os sectores da sociedade. E cada vez mais esta disseminação é encarada como algo que se encontra perfeitamente integrado na vida de cada um de nós. Desde o simples acto de levantar dinheiro até à negociação de empréstimos, pagamento de bens ou entrega de declarações de impostos ao Estado, todas estas operações podem ser realizadas de forma electrónica.

A Internet tornou-se, devido à sua cada vez maior ubiquidade, o meio por excelência para a realização de todo o tipo de operações electrónicas. O aparecimento de lojas electrónicas destinadas ao consumidor final, bem como de portais que permitem a realização de negócios entre empresas, veio esbater diferenças económicas e geográficas e possibilitar a cada um a escolha do negócio mais vantajoso para si.

Todos estes avanços tecnológicos aumentaram a qualidade de vida das pessoas, tornando operações outrora morosas em actos quase imediatos. No entanto, mudaram também o paradigma de segurança subjacente a cada operação que envolva informação confidencial. Esta questão é particularmente notória em transacções na área do comércio electrónico. Cada vez mais os utilizadores de serviços desta área

estão conscientes dos possíveis problemas de segurança que transacções electrónicas podem gerar. Estes serviços devem conseguir responder a questões como:

- É possível ter garantias fortes que dados críticos apenas possam ser do conhecimento de quem de direito?
- É possível verificar que uma entidade realmente é quem afirma ser?
- O que acontece se um utilizador comprar um produto e a loja electrónica não lhe entregar uma prova de compra? E quem protege a loja no caso de se tratar de um comprador fraudulento?
- É possível reconstituir uma transacção electrónica de modo a que um juiz possa tomar uma decisão sobre uma disputa na posse de todos os dados relativos à mesma?
- Existe uma forma confiável de uma entidade poder guardar informação de modo a que a integridade desta seja garantida a longo prazo?

O desconhecimento da resposta a este tipo de questões, por parte de entidades que operam na área de comércio electrónico, leva à desconfiança e receio de potenciais utilizadores em recorrer a serviços que necessitem de manipular dados críticos. Esta desconfiança limita frequentemente a consolidação da Internet como meio de excelência para realização de negócios.

As observações acima apresentadas tornam crucial a definição e disponibilização de estruturas que permitam fornecer garantias fortes de segurança, não só a sistemas de comércio electrónico, mas a sistemas de troca de mensagens em geral. Estas estruturas devem conseguir adaptar-se aos mais diversos ambientes electrónicos e ser suficientemente transparentes para que os pressupostos de segurança que lhes servem de base não possam ser colocados em causa.

Como será apresentado no capítulo 3, não existem actualmente estruturas ou esquemas que forneçam garantias de segurança fortes a sistemas de trocas de mensagens de forma genérica e transparente. Não obstante, existe muito trabalho desenvolvido e exaustivamente analisado que fornece um sub-conjunto das garantias que aqui se procuram implementar. Existem também trabalhos que respondem a todas as questões acima referidas, sendo no entanto direccionados para segmentos ou serviços em particular.

1.2 Objectivos

O objectivo do trabalho é definir e implementar um sistema de troca de mensagens que forneça as garantias de segurança necessárias para possibilitar a sua aplicação mesmo nas áreas mais exigentes ao nível de segurança (e.g., área militar, comércio electrónico seguro).

O sistema a definir deverá poder ser genericamente utilizado em conjunto com um sistema de troca de mensagens. Assim, o sistema não pode depender do formato de dados transaccionados nem do protocolo de troca de mensagens utilizado. Para além disso, é importante que o mesmo permita relacionar as mensagens de uma mesma transacção[1] de forma simples e eficaz. Este mecanismo deve ser parte integrante do sistema a definir. Por outro lado, deve ser definido um mecanismo que valide o sistema. Este será responsável por verificar se, em determinado momento, o sistema consegue cumprir com todas as garantias de segurança propostas. Deve ainda ser assegurada a possibilidade de dissociar este mecanismo do sistema em si (i.e., permitir que este mecanismo de validação seja executado por uma entidade independente do sistema e confiável por todos os utilizadores deste).

Para além da definição do sistema é apresentada uma implementação do mesmo. Esta implementação recorre a estruturas, algoritmos e protocolos cuja segurança e robustez se encontra perfeitamente demonstrada actualmente. Pretendeu-se implementar um sistema modular, isto é, um sistema baseado em componentes que executem funções independentemente uns dos outros e onde a comunicação entre componentes obedeça a um protocolo perfeitamente definido. Deste modo torna-se fácil substituir um componente (e.g., devido a avanços tecnológicos que detectem uma quebra de segurança na tecnologia utilizada) sem que tal implique uma reformulação profunda no sistema. O sistema implementado coloca ao dispor de cada utilizador um modo de realizar pesquisas complexas sobre conjuntos de dados relativamente aos quais este possui permissão para realizar operações de leitura.

O capítulo seguinte apresenta os conceitos relacionados com as áreas nas quais este trabalho se insere, nomeadamente conceitos de criptografia e definição de garantias de segurança. O capítulo 3 apresenta trabalhos relacionados com o trabalho apresentado nesta dissertação. O capítulo 4 define e apresenta um esquema que fornece garantias fortes de manutenção a longo prazo da auditabilidade de dados. No final do capítulo é apresentada uma solução para a manutenção do conjunto de garantias de segurança identificado como essencial para os objectivos

[1] Por transacção entende-se um conjunto de passos (neste caso de troca de mensagens) tratados como uma única actividade de modo a obter um determinado resultado.

propostos. O capítulo 5 apresenta uma implementação do esquema proposto no capítulo 4 obedecendo aos objectivos acima listados. O capítulo 6 apresenta um caso de estudo onde o esquema definido no capítulo 4 foi utilizado. Por último, a discussão do trabalho realizado é apresentada no capítulo 7.

Capítulo 2

Criptografia e Conceitos Relacionados

2.1 Criptografia

> *"The art and science of keeping messages secure is cryptography, and it is practiced by cryptographers."*
>
> Bruce Schneier [68]

A citação anterior apresenta uma sucinta descrição do conceito de criptografia. Este conceito é explorado em detalhe nas próximas secções.

2.1.1 Criptografia de Chave Simétrica

Critpografia de chave simétrica pressupõe a utilização de algoritmos onde a chave (o segredo) utilizada para decifrar um conjunto de dados pode ser calculada a partir da chave utilizada para cifrar os mesmos (e vice-versa). Na maior parte dos algoritmos simétricos ambas as chaves são iguais. A segurança deste tipo de algoritmos reside na manutenção da privacidade da chave utilizada, uma vez que qualquer entidade com conhecimento da mesma pode cifrar e decifrar dados.

A utilização de algoritmos de chave simétrica implica que emissor e receptor possuam a chave utilizada antes dos dados serem cifrados. Este problema pode ser resolvido, por exemplo, através da utilização de um protocolo de troca de chaves[31]. Um outro problema está relacionado com a necessidade de autenticar as chaves trocadas (i.e., ter a certeza que uma chave pertence à entidade à qual é suposto pertencer). Uma solução para este problema passa pela utilização de assinaturas e certificados digitais (como definido nas secções 2.1.4 e 2.1.5).

Um dos algoritmos de chave simétrica mais recentes dá pelo nome de AES[52] (*Advanced Encryption Standard*). Este algoritmo[1] venceu o concurso público apresentado pelo NIST (*National Institute of Standards and Technology*) para o desenvolvimento de um novo algoritmo de chave simétrica. O objectivo do concurso foi encontrar um substituto para o algoritmo DES[51]. O NIST propôs recentemente[57] a invalidação dos documentos nos quais é definido o algoritmo DES por considerar que este já não oferece as garantias de segurança necessárias. O NIST propõe a utilização do algoritmo DES somente como componente do algoritmo Triple DES[15] e recomenda o abandono gradual deste último em detrimento da utilização do algoritmo AES.

O algoritmo AES funciona com chaves de 128, 192 e 256 *bits*, necessitando de poucos recursos (quer de processador ou memória). Todas as operações realizadas sobre o conjunto de dados a cifrar são simples operações matriciais, nas quais cada passo pode ser facilmente reversível, desde que se possua a chave.

2.1.2 Algoritmos de Sumário

Um algoritmo de sumário[2] é uma função que, recebendo como valor de entrada dados de tamanho arbitrário, produz como resultado um valor de tamanho fixo. Para além disso, este tipo de funções tem de obedecer a um conjunto de propriedades:

- Facilidade no cálculo do resultado da função com base nos dados de entrada.
- Dificuldade no cálculo dos dados de entrada com base no resultado da função.
- Dificuldade em encontrar dois conjuntos de dados de entrada distintos que, aplicados ao algoritmo de sumário, produzam o mesmo resultado.

Com base nas propriedades acima definidas consegue-se obter uma identificação inequívoca de qualquer conjunto de dados. Um algoritmo de sumário é público (i.e., não existe qualquer tipo de segredo no processo). A segurança do mesmo baseia-se no conjunto de propriedades acima definido (ou seja na sua não-invertibilidade e na independência que o sumário obtido possui dos dados de entrada). A alteração de um *bit* nos dados de entrada produz, com uma alta probabilidade, um resultado distinto do produzido pelos dados iniciais. Desta forma, uma entidade pode sempre verificar se um determinado sumário corresponde a um conjunto de dados.

[1] Cujo nome original é *Rijndael*.

[2] Também conhecido por *hash* ou *message digest*.

Devido às suas propriedades, este tipo de algoritmos é bastante utilizado em conjunto com algoritmos de assinatura digital (secção 2.1.4). Assim, em vez de ser assinado um conjunto de tamanho arbitrário de dados, é assinado o sumário deste, com todas as vantagens em termos de eficiência do processo que daí advêm.

Os algoritmos de sumário SHA-1[56] e MD5[63] são os mais estudados[68] e utilizados actualmente. O funcionamento de ambos é similar. São definidas (e inicializadas com valores pré-definidos) variáveis de 32 bits (5 no caso do SHA-1 e 4 no caso do MD5). Ambos os algoritmos dividem o processamento dos dados em blocos de 512 bits. Em cada ronda do algoritmo as variáveis são actualizadas de acordo com os próprios valores no momento, com o bloco da mensagem que está a ser processado e com um conjunto de funções não-lineares pré-definidas. O resultado final é fornecido pela concatenação dos valores das variáveis. O algoritmo SHA-1 gera um bloco de 160 bits como resultado enquanto que o algoritmo MD5 gera um bloco de 128 bits.

Não são conhecidos actualmente ataques que quebrem a segurança do algoritmo SHA-1. Existe uma falha na função de compressão[29] utilizada no algoritmo MD5 mas que não tem impacto na segurança do próprio algoritmo[66]. Recentemente foi publicado um artigo[70] que apresenta colisões nas funções de sumário quer do algoritmo MD5, quer do algoritmo SHA-0[55].

A utilização de um algoritmo de sumário com cifra garante a autenticação dos dados de entrada no mesmo. Este tipo de algoritmos garante que apenas quem possuir um segredo (e.g., uma chave simétrica) pode gerar sumários válidos ou verificar a validade dos mesmos. O sumário resultante é chamado de *message authentication code* (MAC). Este esquema pode ser utilizado, por exemplo, para autenticação de mensagens (que não necessitem de confidencialidade) entre utilizadores. Desta forma, e uma vez que apenas estes têm conhecimento do segredo utilizado, conseguem-se evitar ataques de substituição das mensagens originais.

Um dos algoritmos de sumário com cifra mais simples e utilizados é o algoritmo HMAC[23]. Se o algoritmo de sumário subjacente for definido pela função F, os dados de entrada forem representados por x, k for uma chave secreta de l *bits* e *ipad* e *opad* constantes, a função de HMAC é definida como:

$$HMAC_k(x) = F(\bar{k} \oplus opad, F(\bar{k} \oplus ipad, x))$$

Supondo que a função de sumário subjacente processa blocos de b *bits* (i.e., o tamanho de x é b *bits*), então $\bar{k}$ representa k aumentado de $b - l$ *bits* (com o valor zero). As duas constantes *opad* e *ipad* (de b *bits* cada) estão preenchidas respectivamente pelo byte x'36' e pelo byte x'5c'.

2.1.3 Criptografia de Chave Pública

A criptografia de chave pública[31] (ou assimétrica) pressupõe a existência de um par de chaves complementares, sendo uma pública e outra privada. A chave pública pode ser divulgada, já que esta é utilizada apenas para cifrar dados, enquanto que a privada se deve manter secreta. Para a criptografia de chave pública, é essencial que a chave privada seja conhecida apenas por quem de direito, uma vez que esta é utilizada para decifrar os dados.

Os sistemas de criptografia de chave pública utilizados actualmente exploram as propriedades da aritmética através da utilização de grandes grupos finitos. Para a maioria dos métodos, como RSA[64, 65], ElGamal[33, 32] ou DSS[50], a segurança do algoritmo depende da dificuldade de realização de duas operações de grupo:

Exponenciação vs. Logaritmos discretos

A operação de exponenciação deve ser fácil de realizar, em comparação com a dificuldade de realizar a operação inversa, o logaritmo discreto.

O algoritmo de chave pública RSA foi um dos primeiros a surgir e é actualmente um dos mais estudados e utilizados em todo o mundo. O par de chaves utilizado e operações relacionadas são descritos da seguinte forma:

1. Chave pública : n (produto de dois números primos, p e q) e e (não deve ter factores em comum com $(p-1)(q-1)$)
2. Chave privada : $d = e^{-1} mod((p-1)(q-1))$
3. Operação de cifrar m : $c = m^e \ mod \ n$
4. Operação de decifrar : $m = c^d \ mod \ n$

Após a obtenção da chave privada d os números primos p e q já não são necessários e devem ser eliminados. A segurança do algoritmo depende apenas da dificuldade de factorização de grandes números, ou seja, da dificuldade de factorizar n de modo a obter p e q e poder assim reconstruir a chave privada.

2.1.4 Assinaturas Digitais

Uma assinatura *manual* é utilizada por uma entidade para certificar por escrito que determinado texto é do conhecimento da própria. O propósito de uma assinatura digital é o mesmo que o de uma assinatura manual. A diferença está em que, enquanto esta última faz uso de papel e caneta, a primeira utiliza chaves digitais e criptografia de chave pública. Tal como no método manual, uma assinatura digital liga inequivocamente uma entidade com um documento específico.

Ao contrário do que se passa na utilização de assinaturas manuais, é extremamente difícil forjar uma assinatura digital, já que esta última é válida apenas para a informação existente no documento original (como apresentado na secção 2.1.3). Isto significa que, sempre que um documento assinado é alterado, a assinatura digital deixa de ser válida, sendo necessário re-assinar o mesmo. É no entanto possível que uma entidade assine documentos de forma válida em nome de outra, bastando para tal que possua a chave privada desta última. Esta é a razão pela qual se torna essencial manter a chave privada conhecida apenas por quem de direito. Se, mesmo assim, esta for comprometida, caindo em mãos erradas, existem métodos (como apresentado na secção 2.1.6) que possibilitam à entidade detentora da chave invalidar a mesma.

Embora uma assinatura digital se baseie em criptografia de chave pública, a utilização do par de chaves é contrária à apresentada na secção 2.1.3. Neste caso, é a chave privada que é utilizada para produzir uma cifra (uma assinatura) dos dados, enquanto que a chave pública é utilizada para decifrar os mesmos (verificar a assinatura).

2.1.5 Certificados Digitais

Um certificado digital[16, 37] (daqui por diante designado simplesmente por "certificado") é um documento electrónico que liga inequivocamente uma entidade, representada por um identificador único, a uma chave pública. Cada certificado tem um período temporal durante o qual é considerado válido[3] e é assinado digitalmente por uma entidade designada por Autoridade de Certificação[4] (AC). A validação de um certificado pode ser feita por qualquer entidade que tenha acesso à chave pública da AC. Os elementos básicos constituintes de um certificado são os seguintes:

[3]Salvo numa situação de revogação do certificado.

[4]Para mais detalhes consultar a secção 2.1.6.

- Versão do formato do certificado.
- Número de série do certificado. Este é único entre os certificados emitidos pela AC.
- Algoritmo utilizado pela AC para assinar digitalmente o certificado.
- Nome identificativo da AC responsável pela emissão do certificado (*Issuer DN*).
- Período de validade do certificado.
- Nome identificativo da entidade detentora do certificado (*Subject DN*).
- Chave pública da entidade detentora do certificado e identificação do algoritmo criptográfico com o qual esta é utilizada.
- Assinatura do certificado pela chave privada da AC.

O nome identificativo da AC e o número de série identificam inequivocamente um certificado. A versão do formato do certificado pode ser 1, 2 ou 3. Se a versão for 1 apenas os elementos acima referidos podem ser utilizados; se a versão for 2 ou 3 também podem ser o utilizados os seguintes elementos:

- Identificador único da entidade detentora do certificado.
- Identificador único da AC responsável pela emissão do certificado.

Estes elementos existem para propósito de re-utilização de nomes quer da AC, quer da entidade detentora do certificado. No entanto, e segundo a última revisão da definição da infra-estrutura X.509[37], a utilização destes elementos não é aconselhada.

Se a versão do certificado for 3, então pode ser utilizado um elemento composto por um conjunto de extensões. Estas tanto podem ser definidas universalmente, como ser definidas para um âmbito particular. Por exemplo, um certificado utilizado somente para assinar *e-mail* deverá ter a extensão identificada pelo nome *keyUsage* com o valor *digitalSignature*. Um serviço utilizado no âmbito restrito de *e-mail* interno a uma determinada empresa, poderá ter uma extensão identificada com um nome interno à empresa cujo valor reflicta as operações que um determinado utilizador pode realizar sobre uma conta de *e-mail*.

O texto incluído em baixo representa a informação presente num certificado (utilizando a aplicação *openssl*[13]).

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            d5:32:13:17:27:b0:9b:8e:0d:76:13:d3:f5:92:fa:2b
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=PT, O=XCOM, CN=XCOM Personal
        Validity
            Not Before: Mar  5 17:35:50 2004 GMT
            Not After : Mar  5 17:35:50 2005 GMT
        Subject: emailAddress=miguel.reis@isp.novis.pt, C=PT, O=Novis Telecom SA,
                 OU=Trusted Services, CN=Miguel Reis
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:bf:8e:56:69:bc:d8:27:f3:0c:e0:29:35:84:85:
                    59:5b:82:3b:84:ab:9c:19:16:be:13:e3:5f:94:e9:
                    (...)
                    15:74:2c:46:b9:a0:eb:44:48:8b:75:6c:e4:6c:9b:
                    02:8b:24:be:fa:f6:a7:74:bf
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints: critical
            CA:FALSE
            X509v3 Certificate Policies:
            Policy: 1.3.6.1.4.1.15994.1.100.1
              CPS: https://www.xcom.pt/doc/CP_XCOM_Personal.pdf

            X509v3 CRL Distribution Points:
            URI:https://www.xcom.pt/crl/CRL_XCOM_Personal.crl

            X509v3 Key Usage: critical
            Digital Signature, Key Encipherment, Data Encipherment
            X509v3 Authority Key Identifier:
            keyid:EA:33:03:E9:AE:44:4D:C0:AF:8C:EC:0F:1E:13:D4:AA:9E:8F:6C:98

            X509v3 Subject Alternative Name:
            email:miguel.reis@isp.novis.pt
            X509v3 Subject Key Identifier:
            1D:62:9B:F4:27:60:30:7C:E9:89:E6:A7:D2:91:EC:B5:38:19:5E:4C
    Signature Algorithm: sha1WithRSAEncryption
        03:d3:f0:ba:6f:fb:1c:ed:bb:d7:50:5d:52:7d:e3:b8:20:ff:
        00:28:2d:cb:df:01:6c:51:33:80:3e:c1:15:59:ee:d3:e5:8c:
        (...)
        ba:e8:e5:50:ff:3d:f1:58:02:ba:6d:de:10:7b:7a:92:53:c8:
        c5:b8:13:66
```

2.1.6 Autoridade de Certificação

Uma Autoridade de Certificação é responsável pela gestão do ciclo de vida (i.e., emissão, suspensão, revogação ou re-emissão) de certificados. Para poder executar essa gestão, possui um (ou mais) certificados próprios. A confiança da AC deve ser

globalmente reconhecida e a segurança dos seus processos de actuação deve estar garantida por uma auditoria realizada por uma entidade independente.

Todas as AC possuem um documento (*Certificate Practice Statement* - CPS) disponível publicamente que atesta as práticas seguidas por esta entidade no que concerne à emissão e gestão dos certificados emitidos. Mais concretamente, este documento define os requisitos legais, técnicos e de negócio para a operacionalidade do serviço de certificação digital, bem como as políticas seguidas para a gestão do ciclo de vida dos certificados. Por norma, devem seguir-se as indicações apresentadas em [27] para a elaboração deste documento. A localização do CPS vem, na maior parte dos casos, indicada numa extensão (*Certificate Policies*) do certificado da AC (é esse o caso do certificado acima apresentado).

Entre os processos de actuação da AC encontram-se operações como:

- Protecção das suas chave privadas utilizadas para geração de certificados.
- Garantia de adequação do tamanho da chave privada às capacidades computacionais do período temporal durante o qual um certificado pode permanecer válido.
- Identificação inequívoca da entidade para a qual vai ser emitido um certificado.
- Garantia de inviolabilidade do processo de geração do certificado.
- Garantia de validação da identidade da entidade que requer uma suspensão, revogação ou re-emissão de um certificado.

As AC estão organizadas numa estrutura hierárquica (como ilustrado na figura 2.1), podendo existir AC de "topo"e sub-AC. As AC de topo (ou de "raíz") possuem certificados assinados pela sua própria chave privada. Normalmente, os certificados destas entidades estão incluídos por defeito em programas onde possam existir, por exemplo, operações de autenticação (e.g., *browsers Web*).

As sub-AC possuem certificados próprios assinados por AC hierarquicamente superiores. Estas últimas podem por sua vez ser AC de topo ou possuirem, à semelhança das anteriores, certificados próprios assinados por AC hierarquicamente superiores. É usual que um certificado de um utilizador não seja assinado directamente por uma AC de topo, mas sim por uma sub-AC. O conjunto de certificados apresentado é chamado de "cadeia de certificação"(como ilustrado na

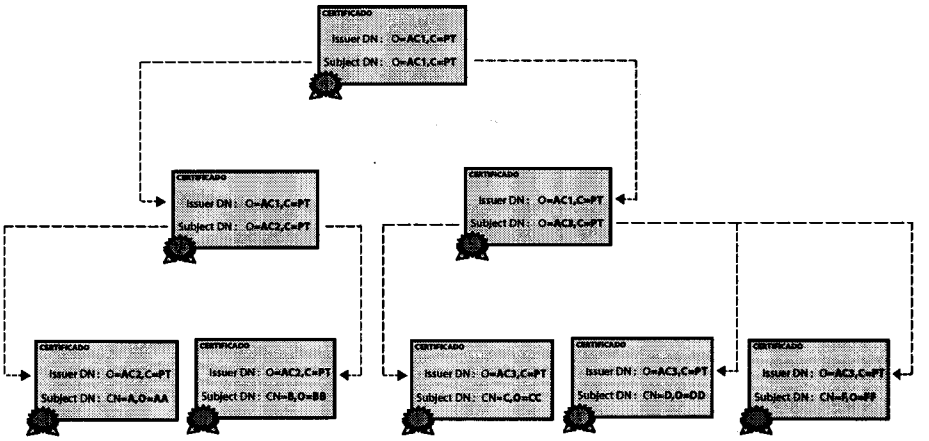

Figura 2.1: Representação da estrutura hierárquica dos certificados digitais. Cada certificado do nível hierárquico n é assinado por um certificado no nível hierárquico $n+1$

Figura 2.2).

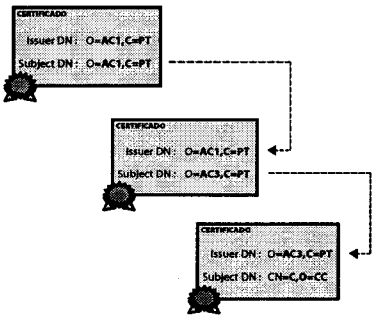

Figura 2.2: Cadeia de certificação do certificado da entidade identificada pelo nome (ou *subject DN*) $CN = C, O = CC$ (segundo a ilustração da Figura 2.1)

Se um utilizador A pretender autenticar um utilizador B, o primeiro necessita de validar a cadeia de certificação do segundo. Para isso é necessário seguir uma série de procedimentos:

- Verificar se o certificado do utilizador B se encontra correctamente assinado por uma sub-AC ou pela própria AC.

- Se existirem sub-AC, verificar que todos os certificados de sub-AC estão correctamente assinados pela sub-AC imediatamente superior na hierarquia e que nenhum se encontra expirado ou inválido (i.e., revogado ou suspenso). O certificado da sub-AC hierarquicamente mais alta deverá estar assinado pela própria AC.
- Verificar que o certificado da AC de topo é da confiança do utilizador A e que não se encontra expirado ou inválido.

A AC presta também um papel de apoio à pesquisa de informação relativa aos certificados por esta emitidos através da disponibilização de um repositório de certificados (usualmente num servidor LDAP[36, 25]). Para além disso, a AC possui um serviço de consulta do estado dos certificados (e.g., activo, revogado, etc.). Este serviço é disponibilizado através de uma Lista de Revogação de Certificados[37] (CRL[5]) e/ou de um serviço OCSP[49] (*On-line Certificate Status Protocol*). A CRL é constituída pelo conjunto de números de série dos certificados revogados pertencentes à AC, juntamente com as respectivas datas de revogação. Para garantir a integridade e autenticação dos dados, a CRL é assinada pela AC. O serviço OCSP recebe pedidos de validação de certificados, entregando respostas assinadas ou re-encaminhando o pedido para outro serviço.

2.2 Validação Cronológica

Entende-se como validação cronológica[19] de mensagens o fornecimento de garantias fortes sobre a existência de determinada informação num momento exacto, através de um selo temporal. Esse selo não é mais que um documento que liga inequivocamente um determinado valor temporal com a informação fornecida, recorrendo para esse efeito a uma assinatura digital.

A entidade responsável pela geração de selos temporais designa-se por Autoridade de Validação Cronológica (TSA[6]). Esta entidade pode obter a informação temporal a partir de fontes próprias ou a partir de outras entidades (e.g., a partir das entidades legalmente autoritárias sobre informação temporal em cada país). Em Portugal, a infra-estrutura da entidade legalmente responsável pela fixação, difusão e fiscalização da Hora Legal, o Observatório Astronómico de Lisboa (OAL), encontra-se inserida numa rede mundial, coordenada pelo *Bureau International des Poids e Mesures*, em colaboração com o *Bureau International de l'Heure*. Esta

[5]CRL - *Certificate Revocation List.*
[6]TSA - *Time Stamping Authority.*

rede é utilizada para definir o Tempo Universal Coordenado (UTC[7]). A estrutura subjacente à disponibilização de selos temporais encontra-se ilustrada na Figura 2.3.

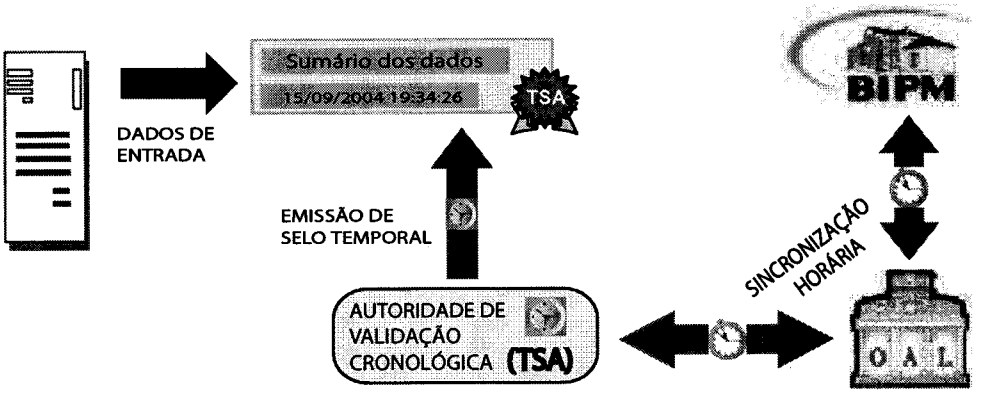

Figura 2.3: Entidades intervenientes na emissão de um selo temporal

O protocolo NTP[47] é o mais utilizado para sincronização de relógios na Internet, estando disseminada uma grande quantidade de servidores que, ligados a relógios de alta precisão (atómicos) ou a receptores GPS (*Global Positioning System*) específicos para o efeito, difundem informação horária segundo o padrão UTC para a Internet. Esta informação é utilizada por aplicações para acerto dos relógios dos computadores onde são executadas, por um lado, e para difusão pelas redes corporativas internas, por outro.

A última especificação oficial do protocolo é o NTPv3[48]. Pese embora o carácter não oficial do NTPv4 (porque não formalizado em RFC), o mesmo é hoje bastante utilizado por entidades que requerem funcionalidades adicionais neste serviço, pelo que se pode dizer que se está em presença de um standard de facto. Uma destas funcionalidades tem que ver com a possibilidade de se poder autenticar a origem da informação horária, assim como a integridade do seu conteúdo, obtendo-se a garantia que se está a obter a hora certa a partir da entidade certa.

[7]UTC - *Universal Coordinated Time.*

2.3 Árvores de Merkle

Uma árvore de Merkle[45] é uma estrutura hierárquica constituída por nós (como ilustrado na figura 2.4). O conjunto de nós existentes no nível hierárquico mais baixo, designados por "folhas", representam toda a informação que a estrutura contem. Todos os restantes nós são de controlo, existindo de modo a assegurar a integridade da informação. Cada nó é uma representação de um algoritmo de sumário aplicado sobre os nós "filhos"(i.e., sobre os nós que se encontram hierarquicamente por baixo deste). Os nós de topo (i.e., os nós da árvore que, em determinado momento, não possuam um nó hierarquicamente superior) são também apresentados como nós "raíz".

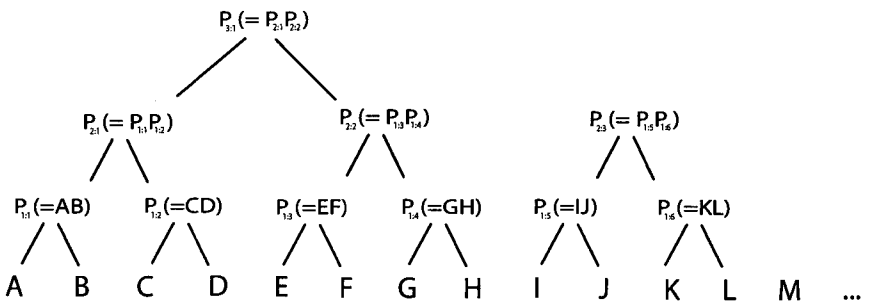

Figura 2.4: Representação dos nós de uma Árvore de Merkle

Este tipo de estrutura encontra aplicação em muitas áreas directa ou indirectamente relacionadas com criptografia, devido à sua simplicidade e generalidade. Uma dessas áreas é a auditabilidade de informação, conseguida através da geração de "caminhos de confiança"constituidos por uma sequência de nós da árvore. Esta sequência é constituída por um ou mais nós folhas, bem como por todos os nós hierarquicamente superiores cujo valor dependa do valor dos primeiros (como ilustrado na Figura 2.5).

Uma vantagem deste tipo de sistemas relativamente a outros com objectivos semelhantes é a sua independência da utilização de criptografia. Toda a segurança subjacente ao sistema assenta na dificuldade de inverter o processo de cálculo de um sumário (como apresentado na secção 2.1.2) e na manutenção da integridade dos nós raíz. Estes últimos devem ser publicados num repositório de acesso público (e.g., jornais, servidores *Web*) e se possível replicados, de modo a que os caminhos de confiança possam ser universalmente verificáveis.

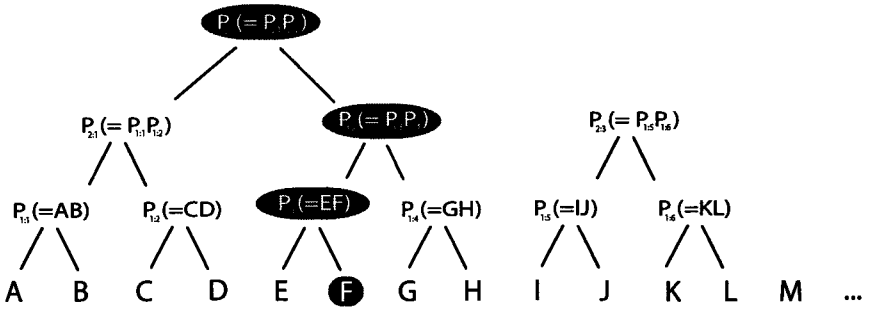

Figura 2.5: Cada caminho de confiança é representado por um conjunto de nós desde um nó folha até um nó raíz

2.4 Garantias de Segurança

Na próxima secção são apresentados os conceitos de segurança que o sistema de troca de mensagens que este trabalho se propõe definir necessita de garantir.

2.4.1 Integridade

Integridade de um conjunto de dados implica garantir a não alteração sem detecção deste depois do momento de geração do mesmo. Através desta propriedade um receptor de uma mensagem pode verificar se esta foi ou não (ilegitimamente) modificada em trânsito. A modificação de uma mensagem pode ter origem em dois tipos de acontecimentos:

- Alteração de dados em trânsito devido a falhas no sistema de comunicação subjacente.
- Intercepção e alteração de dados em trânsito por parte de uma entidade com fins maliciosos.

Uma forma comum de garantir integridade é juntar à mensagem um sumário da mesma (como definido na secção 2.1.2). É fundamental a utilização de um sumário com cifra, visto que de outro modo uma entidade maliciosa poderia substituir uma mensagem em trânsito mantendo-a aparentemente legítima, bastando para tal recalcular o sumário associado à mesma.

2.4.2 Autenticação

A integridade de uma mensagem não implica necessariamente que se conheça a identidade do emissor da mesma (embora em grande parte dos casos estas duas

propriedades se utilizem em conjunto). Através da autenticação obtêm-se garantias fortes de que os dados provêm de determinada entidade. Como já foi explicitado, a utilização de assinaturas digitais só por si nem sempre garante a autenticação de dados. Adicionalmente, é necessário que existam provas mais fortes, como as conseguidas através da utilização de certificados digitais emitidos por uma AC idónea e reconhecida.

2.4.3 Confidencialidade

A confidencialidade de uma mensagem electrónica resulta da cifra da mesma, quer através de utilização de criptografia simétrica, quer assimétrica. Actualmente encontra-se generalizada a utilização de sistemas "híbridos". Este tipo de sistemas procura utilizar o melhor de dois mundos,[8] recorrendo a criptografia assimétrica para fornecer confidencialidade a chaves simétricas que serão posteriormente utilizadas para a efectiva troca de informação.

A tecnologia baseada em *TLS* [30] é actualmente uma das mais difundidas e utilizadas no que toca à confidencialidade de comunicações. O protocolo permite requerer autenticação dos intervenientes (baseada na utilização de cadeias de certificação) e utiliza chaves simétricas para a troca de mensagens. Quando se pretende estabelecer uma sessão *TLS* é executado um protocolo de troca de parâmetros (*handshake*). Este protocolo permite aos intervenientes acordarem nos parâmetros e protocolos específicos a utilizar na sessão segura a estabelecer.

2.4.4 Não-repúdio

Num sistema de troca de mensagens, não-repúdio implica garantir que nenhum dos participantes numa determinada transacção possa negar ter participado em parte ou em toda a transacção. Para tal, é essencial garantir que todos os participantes possuam provas suficientes para que, na presença de um juiz, a sua posição seja salvaguardada. Para que estas provas sejam aceites como válidas é necessário que estejam inequivocamente ligadas à entidade que as gerou, sendo esta premissa conseguida, na maior parte dos casos, através da utilização de assinaturas digitais.

O modo como as provas são entregues num protocolo de não-repúdio pode tornar-se factor de injustiça para os participantes. Suponhamos que uma entidade A envia a uma entidade B uma determinada mensagem, juntamente com

[8] O desempenho de uma operação criptográfica realizada utilizando um algoritmo de chave simétrica é substancialmente superior ao de uma operação realizada utilizando um algoritmo de chave assimétrica. Para mais detalhes consultar [68].

uma prova de origem. Se B receber correctamente estes documentos e não enviar a A uma prova de recepção, então B estará em vantagem sobre A perante um juíz. Para que isso não se verifique é necessário garantir que o protocolo é justo[73, 44], assegurando que nenhum dos intervenientes ganha algum tipo de vantagem (e.g., provas) sobre outros.

Uma outra propriedade relevante nos protocolos de não-repúdio é a definição de um limite temporal para a finalização do mesmo. Esta propriedade garante que qualquer um dos intervenientes numa transacção tenha a possibilidade de terminar o protocolo após um período temporal pré-definido e finito. Deste modo, evitam-se situações em que os participantes tenham de manter uma transacção aberta por períodos de tempo potencialmente infinitos, de modo a assegurar que o protocolo permaneça justo.

Podemos identificar diferentes tipos de não-repúdio (como ilustrado na Figura 2.6):

- Não-repúdio de origem

  Serve ao receptor como prova do envio de uma mensagem pelo emissor. Através da utilização de uma assinatura digital, são ligados de forma inequívoca um emissor, uma mensagem e um receptor. A assinatura deve ser feita sobre a mensagem em conjunto com a identidade do receptor.

- Não-repúdio de recepção

  Serve ao emissor como prova de recepção de uma mensagem pelo receptor. Neste caso a assinatura deve ser feita sobre a mensagem em conjunto com a identidade do emissor.

- Não-repúdio de submissão

  Este tipo de prova apenas está presente nos protocolos de não-repúdio que recorram a uma entidade independente e de confiança de ambos os participantes para participar em toda ou em partes de uma transacção. Esta entidade é comummente conhecida como "Entidade de Confiança"(EC). Neste caso, a EC emite uma prova ao emissor comprovando que a mensagem que este lhe enviou foi recebida com sucesso.

A manutenção de provas de não-repúdio a longo termo pode tornar-se numa tarefa difícil de realizar com sucesso. Um dos factores que contribui para essa dificuldade prende-se com a possibilidade de revogação das chaves de assinatura

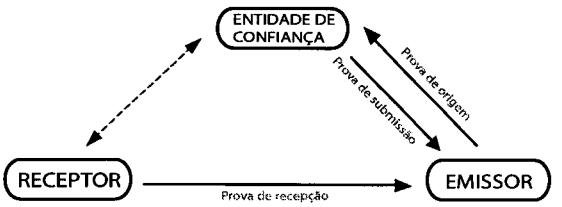

Figura 2.6: Representação dos emissores e receptores das diferentes provas de não-repúdio

utilizadas nas provas. Para a resolução desse problema usualmente recorre-se à utilização de selos temporais (secção 2.2). Um outro problema está relacionado com a possibilidade de quebra das chaves utilizadas quer para a assinatura das provas, quer para a assinatura do selo temporal associado. Esta é uma possibilidade bem real, em especial à medida que avanços criptográficos são realizados. No capítulo 4 é apresentado um esquema que visa resolver, entre outros, este problema.

2.4.5 Auditabilidade

À medida que a Internet se torna um meio cada vez mais importante e crucial de transacções de bens e serviços, também os registos relativos a transacções vão mudando de formato[58]. Cada vez mais se empregam meios electrónicos para assinar contratos, comprar software, bilhetes de avião, realizar aplicações financeiras ou até mesmo adquirir propriedades. Este tipo de transacções pressupõem a existência de um sistema de registo das mesmas que forneça garantias fortes de auditabilidade.

A auditabilidade de um sistema electrónico implica garantir que é extremamente difícil corromper informação sem que tal acção não seja detectada. Um auditor deve poder aceder à informação e determinar inequivocamente se esta é fidedigna (i.e., se esta não está corrompida). A corrupção de informação pode tomar várias formas:

- Alteração do conteúdo da própria informação.
- Alteração cronológica de informação (sem alteração do conteúdo da mesma).
- Eliminação de informação.
- Adição de informação de forma não autorizada.

Em muitas situações os conceitos de auditabilidade e de arquivo confiável e seguro de dados sobrepõem-se e chegam mesmo a confundir-se. Por um lado, os próprios dados arquivados podem fazer parte dos elementos de registos no sistema de auditabilidade. Por outro, os registos auditáveis podem conter elementos que permitam identificar inequivocamente um determinado conjunto de dados (e.g., através da utilização de um sumário dos mesmos).

A auditabilidade de um sistema electrónico está directamente relacionada com a aplicação da maior parte dos conceitos apresentados neste capítulo. A integração de conceitos como sumários, assinaturas e certificados digitais, validação cronológica e estruturas baseadas em árvores de Merkle permite implementar sistemas que fornecem garantias fortes de auditabilidade, como apresentado na secção 3.2.

Capítulo 3

Trabalho Relacionado

O presente capítulo faz uma síntese do trabalho relacionado com a área desta dissertação. O foco é colocado em protocolos que forneçam tanto garantias fortes de não-repúdio de origem, integridade, autenticação e confidencialidade (designadas por garantias de segurança "básicas"), como garantias fortes de auditabilidade e de não-repúdio de submissão e recepção de mensagens (designadas por garantias de segurança "avançadas").[1] O conjunto de protocolos justos consegue fornecer a maior parte das garantias pretendidas. Falta-lhes no entanto fornecer garantias fortes de auditabilidade e de arquivo seguro e confiável de dados (como justificado nas próximas secções).

Na secção 3.1 são apresentadas diferentes abordagens à problemática da garantia de não-repúdio em sistemas de troca de mensagens. Na secção 3.2 é apresentado um conjunto de possíveis soluções para a inclusão de garantias de auditabilidade neste tipo de sistemas.

3.1 Não-repúdio

Os primeiros protocolos a surgirem que tentavam garantir ser justos baseavam-se na troca gradual de partes da informação. É crucial para a manutenção do protocolo como justo, que o esforço computacional requerido para qualquer um dos intervenientes obter a restante informação seja aproximadamente igual em cada passo do protocolo. Isto leva à necessidade de garantir que os intervenientes possuam capacidades computacionais semelhantes, o que não é uma premissa razoável actualmente.

[1] As garantias de segurança avançadas são assim apelidadas porque usualmente apenas se encontram em sistemas que manipulem informação bastante crítica.

Uma evolução importante no campo do não-repúdio foi a introdução de protocolos probabilísticos[24]. À semelhança dos anteriores, estes também não são totalmente justos e baseam-se na troca gradual de informação. A grande diferença reside no facto de estes últimos garantirem que, em cada momento da execução do protocolo, a probabilidade de um dos intervenientes ser privilegiado relativamente a outro não ultrapassa um limite pré-definido.

Embora existam actualmente diferentes abordagens à problemática da garantia de não-repúdio[42, 72], é de realçar o facto de todas elas tornarem indispensável a inclusão de uma EC no protocolo.[2] O grau de interacção dos intervenientes com a EC acaba mesmo por ser o principal factor de diferença entre tipos de protocolos. Nas seguintes secções são apresentadas diferentes abordagens à definição de protocolos de não-repúdio. Para a apresentação formal dos protocolos é utilizada a seguinte notação:

A e B : identificadores do emissor e do receptor

EC : Entidade de Confiança do emissor e do receptor

TSA : Autoridade de Validação Cronológica

M : mensagem a ser trocada entre os intervenientes

$h(M)$: algoritmo de sumário aplicado à mensagem M

T : valor temporal

$M1, M2$: concatenação das mensagens $M1$ e $M2$

C_A e S_A : chave de cifra (pública) e de assinatura (privada) da entidade A

$s_k(M)$: assinatura digital de M através da utilização da chave privada k

$c_k(M)$: cifra de M através da utilização da chave pública k

$S \rightarrow R : M$: a entidade S envia a mensagem M à entidade R

[2]Não se considera aqui a utilização de protocolos de não-repúdio com base na troca gradual de segredos ou probabilísticos, uma vez que estes não são totalmente justos.

$S \leftrightarrow R : M$: a entidade S obtém a mensagem M a partir da entidade R

3.1.1 Participação Total da Entidade de Confiança

Este tipo de protocolo de não-repúdio implica a actuação de uma EC como entidade agregadora, participando em todas as trocas de mensagens a realizar durante uma transacção. São normalmente protocolos que não primam pela eficiência, dado o número elevado de interacções com a EC que necessitam de realizar, tornando-a um ponto de estrangulamento do fluxo de informação, bem como um ponto central de falha. Estes conseguem, no entanto, fornecer garantias que outros protocolos, devido à sua arquitectura, se encontram impossibilitados de oferecer, como a possibilidade de colocação de selos temporais em todas as mensagens ou ainda a manutenção num único repositório de todas as mensagens de forma auditável e confidencial.

A primeira utilização de protocolos em que existe uma EC pela qual passam todas as trocas de mensagens surgiu no contexto do correio electrónico certificado[22]. A maior parte dos autores não faz qualquer tipo de distinção entre um protocolo de correio electrónico certificado e um protocolo de não-repúdio. Em [41] essa distinção é feita através da introdução de uma nova propriedade definida como *author-based selective receipt*. Esta propriedade define que, após a entidade do emissor se tornar conhecida pelo receptor, seja obrigatória a entrega de um recibo ao emissor (bem como a mensagem ao receptor). Esta propriedade é específica a protocolos de correio electrónico certificado, não sendo obrigatória em protocolos de não-repúdio.

De seguida é apresentado um protocolo, definido por Coffey et al. em 1996[28], que é ilustrativo do tipo de protocolos de não-repúdio em que todas as trocas de mensagens passam obrigatoriamente por uma terceira entidade de confiança de ambas as partes. De modo a apresentar o protocolo é utilizada a seguinte notação adicional:

L_1 e L_2: etiquetas de prova de origem (PO) e de prova de recepção (PR)

n_1, n_2 : identificadores de transacção gerados pela EC

$ppo = s_{S_A}(L_1, A, B, M)$: prova parcial de origem

$PO = s_{S_{TSA}}(ppo, TSA, T_1)$: prova de origem

$ppr = (L_2, B, A, h(PO))$: prova parcial de recepção

$s\_ppr = s_{S_B}(ppr)$: prova parcial de recepção assinada

$PR = s_{S_{TSA}}(s\_ppr, TSA, T_2)$: prova de recepção

N_R_Req : pedido de identificador de transacção

As interacções entre os intervenientes no protocolo são realizadas através do seguinte modelo:

$A \rightarrow TSA : c_{C_{TSA}}(ppo)$

$TSA \rightarrow A : c_{C_A}(PO)$

$A \rightarrow EC$: N_R_Req

$EC \rightarrow A : c_{C_A}(n_1)$

$A \rightarrow EC : c_{C_{EC}}(s_{S_A}(n_1, PO, ppr))$

$EC \rightarrow B : c_{C_B}(s_{S_{EC}}(n_2, ppr))$

$B \rightarrow TSA : c_{C_{TSA}}(s\_ppr)$

$TSA \rightarrow B : c_{C_B}(PR)$

$B \rightarrow EC : c_{C_{EC}}(s_{S_B}(n_2, PR))$

$EC \rightarrow B : c_{C_B}(PO)$

$EC \rightarrow A : c_{C_A}(PR)$

O emissor começa por submeter à TSA uma prova parcial de origem assinada, recebendo como resposta um selo temporal aplicado sobre esta última, e representando a prova de origem PO. O emissor requer de seguida à EC um identificador

de transacção. Após a obtenção do identificador, o emissor envia à *EC* a prova de origem juntamente com uma prova de recepção parcial *ppr*. De seguida a *EC* envia ao receptor a prova parcial de recepção, juntamente com um identificador de transacção. O receptor assina a prova parcial de recepção e envia-a à *TSA*, recebendo como resposta a prova de recepção *PR*. Este última é posteriormente enviada à *EC*, juntamente com o identificador de transacção recebido. Por fim a entidade *EC* envia a prova de recepção ao emissor e a prova de origem ao receptor. Pressupõe-se que, em todas as transacções descritas, a entidade receptora de uma mensagem faz a verificação quer do formato desta, quer da validade das assinaturas digitais presentes na mesma.

No caso de disputas um juiz requer à *EC* ou aos restantes intervenientes as provas de recepção e origem produzidas, verificando a validade das mesmas e obtendo os selos temporais associados. Deste modo é possível verificar se uma mensagem foi efectivamente enviada pelo emissor e/ou recebida pelo receptor.

Como é apresentado pelo modelo acima descrito, todas as trocas de mensagens passam obrigatoriamente pela *EC*. O protocolo utiliza provas parciais de origem e recepção de modo a manter-se justo ao longo das diferentes transacções. Deste modo em nenhum momento (excepto na transacção final originada na *EC*) o receptor se encontra na posse da prova de origem sem que o emissor possua também a prova de recepção respectiva (e vice-versa). Para além disso, é também assegurada a confidencialidade de todas as mensagens trocadas (e logo dos próprios intervenientes) através da cifra das mesmas.

Os identificadores de transacção são desnecessários tal como apresentados. Se se pretender distinguir diferentes transmissões de uma mesma mensagem, então o identificador da transacção deveria estar presente em *ppo* e *ppr*. A cifra de todas as mensagens transmitidas limita fortemente o desempenho do protocolo. Mesmo com esta cifra, a privacidade da mensagem pode não estar assegurada, uma vez que esta última tem de ser apresentada à EC. De referir também que é necessário, de modo a que o protocolo permaneça justo, que a EC continue a enviar as últimas duas mensagens até que estas sejam entregues.

3.1.2 Participação Parcial da Entidade de Confiança

O tipo de protocolo apresentado nesta secção necessita de uma EC que participe activamente em cada transacção. No entanto, e ao contrário do tipo de protocolo apresentado na secção 3.1.1, a EC não participa em todas as trocas de mensagens existentes durante uma transacção.

Em 1996 Zhou e Gollmann [73] apresentaram um protocolo de não-repúdio no qual tentam reduzir ao máximo o trabalho da EC. Para além disso, se uma mensagem incorrecta é transmitida ou se uma mensagem não é entregue, existe a possibilidade de abortar o protocolo. De modo a apresentar o protocolo é utilizada a seguinte notação adicional:

k : chave simétrica definida por A

C : mensagem M cifrada pela chave k

f_{PO}, f_{PR}, f_{Sub}, f_{Con} : identificadores de tipo de mensagem

L : identificador de transacção

$PO = s_{S_A}(f_{PO}, B, L, T, C)$: prova de origem

$PR = s_{S_B}(f_{PR}, A, L, T, C)$: prova de recepção

$sub\_k = s_{S_A}(f_{Sub}, B, L, T, k)$: prova de submissão da chave k

$con\_k = s_{S_{EC}}(f_{Con}, A, B, L, T, T_0, k)$: confirmação de submissão da chave k

O seguinte modelo representa as trocas de mensagens entre os diferentes intervenientes no protocolo:

$A \rightarrow B : f_{PO}, B, L, T, C, PO$

$B \rightarrow A : f_{PR}, A, L, PR$

$A \rightarrow EC : f_{Sub}, B, L, T, k, sub\_k$

$B \leftrightarrow EC : f_{Con}, A, B, L, T_0, k, con\_k$

$A \leftrightarrow EC : f_{Con}, A, B, L, T_0, k, con\_k$

O emissor começa por enviar ao receptor a mensagem cifrada pela chave de sessão k, bem como a respectiva prova de origem. Para além disso, é também transmitido um valor temporal T antes do qual a chave k tem de ser enviada à

EC e após o qual deixa de estar disponível nesta última. Se o receptor concordar com o valor temporal transmitido no passo anterior, envia ao emissor uma prova de recepção da mensagem cifrada. De seguida o emissor envia para a *EC* a chave de sessão k. A *EC*, após verificar que a mensagem foi enviada antes do valor temporal acordado, disponibiliza a chave de sessão k, bem como uma prova de recepção desta mesma. O valor temporal presente em T_0 representa o momento a partir do qual a *EC* disponibilizou a informação referida. Note-se que a chave de sessão k estará apenas disponível até ser atingido o valor temporal definido em T. Por fim o receptor obtem, a partir da *EC*, tanto a chave de sessão k como a prova de submissão desta à *EC*. De modo similar, o emissor recorre à *EC* para obter esta mesma prova de submissão.

No caso de o emissor prentender demonstrar que enviou uma determinada mensagem ao receptor terá de apresentar, perante um juiz, a mensagem enviada, bem como uma completa prova de origem. Esta última é constituída pela prova de submissão da mensagem cifrada (que lhe foi entregue pelo receptor) e pela prova de submissão da chave utilizada para a cifra (que foi disponibilizada pela *EC*). Se for o receptor a pretender demonstrar que uma mensagem lhe foi entregue, este terá de apresentar ao juiz quer a própria mensagem, quer a prova de recepção completa. Esta é formada pela prova de origem da mensagem cifrada (gerada pelo emissor), em conjunto com a prova de origem da chave de sessão (gerada pela *EC*).

O trabalho desenvolvido em [39] corrige falhas do protocolo apresentado. É demontrado que o protocolo se pode tornar injusto no caso em que o emissor envie a chave k apenas instantes antes do tempo limite T ser atingido e de seguida impeça o receptor de receber esta prova (e.g., através da interrupção do canal de comunicação subjacente à transacção) até T ser ultrapassado. Por outro lado, o protocolo, como apresentado, não confere confidencialidade às mensagens trocadas. Qualquer entidade pode obter a mensagem cifrada em trânsito e mais tarde ler a chave de cifra a partir da EC.[3]

3.1.3 Ausência de Participação da Entidade de Confiança

Os primeiros trabalhos publicados sobre este tipo de protocolos de não-repúdio surgiram em 1997[20, 46]. A principal característica destes é o facto de a *EC* não participar activamente numa transacção que termine sem que ocorra nenhum problema. Esta apenas é chamada a intervir no caso de ocorrer alguma anomalia (e.g., um problema de rede ou um comportamento incorrecto por parte de um dos intervenientes). Se tal situação ocorrer, a *EC* possui permissão para gerar

[3] O trabalho apresentado em [39] também resolve este problema.

mensagens que permitam abortar o protocolo ou levá-lo a terminar com sucesso, garantindo sempre que este se mantem justo para ambos os participantes.

O protocolo apresentado por Kremer e Markowitch em 2000[40] é representativo do tipo de protocolo apresentado nesta secção. É definido um protocolo principal, onde a EC não interfere, bem como sub-protocolos que permitem aos intervenientes abortar ou recuperar a normal execução do primeiro. Este modelo é baseado no trabalho realizado por Asokan et al.[21]. De modo a apresentar o protocolo, é utilizada a seguinte notação adicional:

f_{PO}, f_{PR}, f_{Sub}, f_{PO_k}, f_{PR_k}, f_{Rec_X}, f_{Conf_k}, f_{Abort}, f_{Conf_a} : identificadores de tipo de mensagem

k : chave simétrica definida por A utilizada para cifrar M

$l = h(M, k)$: identificador de transacção

c : mensagem M cifrada pela chave k

$PO = s_{S_A}(f_{PO}, B, l, h(c))$: prova de origem da mensagem cifrada c

$PR = s_{S_B}(f_{PR}, A, l, h(c))$: prova de recepção da mensagem cifrada c

$Sub = s_{S_A}(f_{Sub}, B, l, c_{C_{EC}}(k))$: prova de submissão da chave k

$PO_k = s_{S_A}(f_{PO_k}, B, l, k)$: prova de origem da chave k

$PR_k = s_{S_B}(f_{PR_k}, A, l, k)$: prova de recepção da chave k

$Rec_X = s_{S_X}(f_{Rec_X}, Y, l)$: pedido de execução do protocolo *recuperar* à EC

$Conf_k = s_{S_{EC}}(f_{Conf_k}, A, B, l, k)$: prova de confirmação de submissão de k

$Abort = s_{S_A}(f_{Abort}, B, l)$: pedido execução do protocolo *abortar* à EC

$Conf_a = s_{S_{EC}}(f_{Conf_a}, A, B, l)$: confirmação de *abortar* pela EC

$T_l => Prot$: se T_l for excedido executar o sub-protocolo $Prot$

O protocolo principal é composto pelas seguintes trocas de mensagens:

$A \rightarrow B : f_{PO}, f_{Sub}, B, l, c, c_{C_{EC}}(k), PO, Sub$

$B \rightarrow A : f_{PR}, A, l, PR$ (T_{l_1} => *abortar*; FIM)

$A \rightarrow B : f_{PO_k}, B, l, k, PO_k$ (T_{l_2} => *recuperar* (com $X = B$ e $Y = A$); FIM)

$B \rightarrow A : f_{PR_k}, A, l, PR_k$ (T_{l_3} => *recuperar* (com $X = A$ e $Y = B$); FIM)

O emissor começa por enviar ao receptor a mensagem cifrada c bem como a chave de sessão k utilizada para a cifrar. Esta última encontra-se, por sua vez, cifrada para a entidade EC. As assinaturas do emissor sobre estas cifras constituem uma prova de origem das mesmas. A resposta por parte do receptor é constituida por uma assinatura sobre o sumário da mensagem cifrada, que serve de prova de recepção de c. Se esta não for enviada antes de um determinado limite temporal, o emissor inicia o sub-protocolo *abortar*. Caso contrário, o emissor envia ao receptor a chave k assinada. Esta assinatura, juntamente com a prova de origem de c, constituem, para o receptor, a prova de origem de M. Se o receptor não receber esta última assinatura antes de um determinado limite temporal, inicia o sub-protocolo *recuperar*. Caso contrário, envia ao emissor uma prova de recepção da chave k. De modo similar, se esta última mensagem for entregue constitui, juntamente com a prova de recepção de c, prova de recepção da mensagem M para o emissor. Se por outro lado, não for entregue antes de um determinado limite temporal, o emissor inicia o sub-protocolo *recuperar*.

O sub-protocolo *abortar* apenas é executado se o próprio ou o sub-protocolo *recuperar* ainda não foram executados durante a transacção. Este é composto pelas seguintes trocas de mensagens:

$A \rightarrow EC : f_{Abort}, l, B, Abort$

$EC \rightarrow A : f_{Conf_a}, A, B, l, Conf_a$

$EC \rightarrow B : f_{Conf_a}, A, B, l, Conf_a$

Se a entidade EC aceitar o pedido do emissor, envia tanto ao emissor como ao receptor uma mensagem assinada que confirma que o protocolo foi abortado.

Se o sub-protocolo *abortar* já foi executado durante a transacção, o sub-protocolo *recuperar* devolve à entidade X a mesma resposta que o primeiro. O sub-protocolo *recuperar* apenas é executado se o próprio ainda não foi executado durante a transacção. Este é composto pelas seguintes trocas de mensagens:

$$X \rightarrow EC : f_{Rec_X}, f_{Sub}, Y, l, h(c), c_{C_{EC}}(k), Rec_X, Sub, PR, PO$$

$$EC \rightarrow A : f_{Conf_k}, A, B, l, k, Conf_k, PR$$

$$EC \rightarrow B : f_{Conf_k}, A, B, l, k, Conf_k$$

A entidade EC, para aceitar o pedido de execução do sub-protocolo *recuperar*, necessita de dois tipos de provas:

- Provas de origem da mensagem cifrada c e da chave de sessão k; e
- Prova de recepção destes mesmos items.

Após validar estas provas, a entidade EC envia para ambos os intervenientes uma prova assinada de confirmação de submissão da chave k. Para além disso, envia para o emissor a prova de recepção PR, já que o receptor pode ter iniciado este sub-protocolo imediatamente após a recepção da mensagem cifrada c. Pressupõe-se que, em todas as transacções descritas, a entidade receptora de uma mensagem faz a verificação quer do formato desta, quer da validade das assinaturas digitais presentes na mesma.

No caso de disputas ambos os intervenientes deverão, de modo a defenderem a sua posição, apresentar perante um juiz as respectivas provas de origem ou de recepção de M. Caso o sub-protocolo *recuperar* tenha sido executado, deverá ser apresentada a prova de confirmação de submissão da chave k assinada pela EC, juntamente com a mensagem cifrada c, a mensagem M, a chave k e a etiqueta identificativa da transacção l.

3.2 Auditabilidade

Os protocolos apresentados na secção 3.1 são omissos no que toca à questão da auditabilidade de uma transacção a longo prazo. Embora realcem a necessidade de garantir que os diferentes tipos de provas apresentados durante a troca de mensagens se mantenham na posse dos intervenientes, não apresentam ou sugerem um modo que garanta que estas provas não sejam corrompidas ou simplesmente eliminadas. Existem, no entanto, outro tipo de protocolos cujo foco se centra na área da auditabilidade, como apresentado de seguida.

3.2.1 Comercio Electrónico

Em 1999 Peha[58] desenvolveu um sistema que procura fornecer garantias fortes sobre a auditabilidade e confidencialidade da informação trocada entre intervenientes num sistema de comércio electrónico. Para isso define três tipos de entidades de confiança, que funcionam de forma independente entre si:

- **Gestor de autenticação**: Responsável pela validação, disponibilização e gestão dos dados pessoais dos intervenientes numa transacção.
- **Notário**: Responsável pela manutenção da auditabilidade da informação associada a uma transacção.
- **Auditor**: Responsável pela verificação da integridade dos registos de cada transacção.

De modo a poder realizar transacções no sistema, um utilizador tem de se registar perante o gestor de autenticação, obtendo assim uma nova credencial que o passará a representar. A partir desse momento, e sempre que existir uma transacção, a informação relativa a esta mesma tem de ser entregue ao notário. Quando se pretende verificar a integridade dos registos (onde se mantém a informação sobre transacções), recorre-se aos serviços do auditor. Este tem conhecimento sobre todas as credenciais que o utilizador possui.

Os dados fornecidos por cada utilizador ao gestor de autenticação podem conter informações a disponibilizar publicamente (e.g., a chave pública, o número de conta), bem como informações que devem permanecer confidenciais (e.g., número de bilhete de identidade, nome). As informações públicas representam inequivocamente o utilizador, constituindo a credencial deste perante os outros intervenientes numa transacção. Deste modo, a entidade real de um utilizador pode ser mantida secreta dos restantes, para além de permitir que este possua várias credenciais

(e.g., para utilização em serviços que tenham diferentes requisitos no que concerne à necessidade de disponibilização de informação de forma pública).

A informação representativa de cada transacção, associada a uma credencial de um utilizador, é processada por um (e só um) notário, ficando esta entidade responsável pela manutenção da auditabilidade da mesma. A informação contem assinaturas de cada um dos intervenientes na transacção sobre o conjunto de dados que identificam a mesma. Entre os dados, encontram-se as credenciais dos intervenientes na transacção, bem como a data da mesma. Cada registo presente no repositório do notário é constituido por vários elementos:

- Sumário da informação representativa de uma transacção, assinado pela chave pública associada às credenciais do utilizador;
- Selo temporal aplicado sobre o sumário acima referido;
- Índice utilizado para garantir a ordem dos registos, bem como a impossibilidade de inserção ou remoção de registos sem detecção por parte de um auditor; e
- Sumário do conjunto de informação constituido pelos restantes elementos do registo, assinado pelo notário.

A verificação da integridade de uma determinada transacção é realizada pelo auditor recorrendo aos registos do notário em conjunto com a informação representativa da própria transacção. Esta última é obtida a partir de um dos utilizadores que tomou parte na mesma. Como o auditor tem conhecimento das credenciais que cada utilizador possui, pode sempre requerer a informação necessária ao utilizador em questão. Na posse desta informação o auditor verifica:

- A validade das assinaturas presentes no registo;
- Se a informação apresentada pelo utilizador se encontra coerente com a informação presente no registo; e
- Se a data do selo temporal presente no registo se encontra coerente com a data presente na informação representativa da transacção (fornecida pelo utilizador).

Esta abordagem apresenta falhas no que toca à manutenção a longo prazo da auditabilidade da informação mantida pelo notário. Mesmo que as chaves de assinatura sejam trocadas com regularidade, acompanhando eventuais avanços criptográficos, as transacções mais antigas tornar-se-ão cada vez mais vulneráveis a tentativas de quebra de integridade da informação. Uma possível solução pode passar por re-assinar periodicamente as transacções que ultrapassem um determinado período de vida. Este procedimento revela-se, no entanto, impraticável à medida que o número de registos cresce.

3.2.2 Nomes Seguros

Em 1997 Haber e Stornetta[35] apresentaram um trabalho no qual se que garante que documentos digitais possam ser referidos de forma não ambígua. Desta forma, um utilizador do sistema tem a certeza que um documento é realmente aquele referido por um nome específico. A base do sistema está assente na utilização de algoritmos de sumário e árvores de Merkle. É apresentada a garantia de que os nomes gerados possam persistir a longo prazo, independentemente de eventuais avanços criptográficos que possam quebrar a segurança dos algoritmos de sumário utilizados.

O foco principal deste sistema é colocado na garantia de integridade do conteúdo de um documento digital, através da geração de um nome não ambíguo para este. Para isso, foram consideradas duas formas de actuação. A primeira compreende a aplicação de uma assinatura digital sobre o conteúdo de um documento e, a partir do valor obtido, a geração de um nome não ambíguo. Esta solução tem como desvantagem a necessidade de interacção com uma infra-estrutura de chave pública (i.e., de utilização de criptografia assimétrica). Para além disso, a segurança de um determinado nome passa a estar dependente da manutenção da inviolabilidade da chave privada utilizada na assinatura.

Uma segunda forma de actuação passa pela geração de um nome pela aplicação de um algoritmo de sumário ao conteúdo do documento. Esta forma foi a escolhida para a definição deste sistema. Embora este método não necessite da manutenção de chaves secretas, foram detectadas algumas limitações que necessitavam de ser ultrapassadas:

- Avanços criptográficos poderiam obrigar à actualização dos nomes gerados;
- Os valores produzidos pelo algoritmo de sumário são demasiado longos e complexos para serem referenciados por humanos; e

- O autor de um documento não tem controlo sobre a forma do nome do mesmo.

O principal componente deste serviço é o módulo de geração de nomes. Este é composto por um repositório construído com base numa Árvore de Merkle, onde os nós folhas representam os nomes dos documentos. Após receber um documento, o serviço calcula o seu sumário com base numa função bem conhecida (e.g., $SHA-1$). O sumário é de seguida inserido no repositório de nomes.

O repositório não gera nomes instantaneamente, mas apenas no fim de intervalos temporais. Quando um intervalo termina, são gerados identificadores dos documentos cujo sumário foi inserido na árvore durante esse mesmo intervalo. Para isso é utilizado o nó que constitui o actual topo da árvore de Merkle, identificado inequivocamente por uma etiqueta t. Cada identificador c é constituído pelos seguintes elementos:

$$c = \{t, (z_1, b_1), ..., (z_l, b_l)\}$$

Cada elemento z_i (com $i = 1, ..., l$) representa o valor de um nó da Árvore de Merkle no nível hierarquico i. Estes são necessários para calcular o elemento identificado por t a partir do documento original (este último é necessário para validar o identificador). Cada elemento b_i indica em que ordem (i.e., como prefixo ou sufixo) o respectivo valor z_i deve ser adicionado ao elemento z_{i-1} de modo verificar a integridade do identificador. Por fim, o nome n do documento é obtido inequivocamente a partir do identificador c:

$$n = \{t, b_1, ..., b_l\}$$

O processo de verificação da integridade de um nome necessita que sejam fornecidos como parâmetros de entrada um documento x e o respectivo nome n e identificador $c = \{t', (z_1, b_1'), ..., (z_l, b_l')\}$. Os passos de verificação são os seguintes:

- Verificar se $t = t'$ e se cada $b_i = b_i'$ (com $i = 1, ..., l$);
- Calcular $y_1 = h(x)$, valor representativo da aplicação do algoritmo de sumário ao documento x;
- Calcular (para cada $i = 1, ..., l$) o valor de $y_{i+1} = h(w_i)$, onde:

- $w_i = z_i, y_i$ se $b_i = L_{esquerda}$
- $w_i = y_i, z_i$ se $b_i = L_{direita}$; e

- Verificar se y_{l+1} é igual ao valor do nó representado pela etiqueta t.

A segurança do sistema assenta na segurança do algoritmo de sumário subjacente, bem como na integridade dos nós raíz de cada sub-árvore, representados pelas etiquetas t. A manutenção da integridade deste tipo de nós é baseada no facto de estes serem disponibilizados publicamente e de forma replicada. Este facto não constitui no entanto, por si só, uma garantia de auditabilidade da árvore de nomes.

Sem a devida protecção, os nós raíz poderão facilmente ser manipulados. Se uma entidade com fins maliciosos pretender, por exemplo, associar um nome inválido com um determinado documento, sem que tal fraude seja detectada pelo sistema, apenas necessitará de colocar o documento no fim da Árvore de Merkle e associá-lo com um novo nó de topo e uma nova etiqueta t.

3.2.3 Dinheiro Electrónico

O protocolo apresentado por Sander et al. em 1998[67] utiliza árvores de Merkle de modo a garantir a auditabilidade de um sistema de emissão de dinheiro electrónico. Quando um utilizador do sistema efectua um levantamento de dinheiro de uma determinada conta, são geradas moedas electrónicas. Cada moeda é representada de forma única num nó folha da Árvore de Merkle definida como repositório. As moedas são utilizadas como forma de pagamento entre entidades. Para que uma entidade deposite moedas recebidas na sua conta basta apresentá-las perante o sistema. A partir desse momento, e após correcta validação das mesmas, as moedas são marcadas no repositório (i.e., na Árvore de Merkle) como já utilizadas.

De modo a descrever o protocolo vamos assumir que existe uma entidade A que possui uma conta no sistema bancário aqui descrito. Esta entidade faz levantamentos de moedas electrónicas, que utiliza posteriormente para executar pagamentos a um comerciante M. O sistema é descrito através das seguintes operações:

- Inicialização

Durante esta fase os intervenientes no protocolo definem o número máximo de moedas electrónicas (i.e., o número máximo de nós folha) que a Árvore de Merkle representativa do repositório (aqui designada por RT) pode manter.

- Abertura de Conta

  É através deste procedimento que a entidade A obtem uma identificação inequívoca perante o sistema bancário.

- Levantamento de moedas

  A entidade A gera uma nova moeda electrónica $z = h(x, r)$. A função h possui as mesmas propriedades de um algoritmo de sumário. O valor x representa o número de série da moeda e o valor r é um número aleatório. De seguida a entidade A autentica-se perante o sistema bancário sem nunca revelar os dados necessários para a geração da moeda (i.e., sem nunca revelar os valores x e r). Caso a moeda z não tenha sido levantada é colocada numa das folhas de RT. Por fim, é entregue à entidade A (pelo sistema bancário) a cadeia de nós que ligam a moeda z a um dos nós de topo de RT nesse período temporal. Devido às propriedades da função h, apenas quem conhecer os valores x e r poderá utilizar a moeda gerada.

- Pagamento

  Quando a entidade A pretender entregar a um comerciante M uma moeda electrónica w, fornece-lhe uma prova *zero-knowledge* não-interactiva[54] de que conhece um par de valores x e r que representam inequivocamente uma moeda válida, bem como uma cadeia de nós desde w até ao conjunto de nós de topo reconhecidos pelo comerciante. Deste modo, a entidade A prova que conhece uma moeda válida sem nunca revelar o valor de w, impossibilitando o relacionamento entre moedas geradas e moedas utilizadas. Após correcta validação dos elementos acima referidos, o comerciante M aceita como válida a moeda electrónica w.

- Depósito

  Quando o comerciante M deposita a moeda w (identificando-a através dos dados fornecidos pela entidade A na operação de pagamento) no sistema bancário é verificado primeiramente se ainda não foi feito um depósito com a moeda, sendo de seguida verificada a validade da cadeia de nós apresentada. Se a moeda ainda não foi utilizada, a conta do comerciante M é creditada e a moeda é marcada como já depositada. Caso contrário, o banco consegue obter a identidade de A com base nos dados da moeda apresentados para este depósito e para o primeiro depósito (i.e., para o depósito onde a moeda foi utilizada de forma válida).

- Actualização de *RT*

  Quando é realizada uma actualização de *RT*, quer devido à geração de nós utilizados para agregar nós do nível hierárquico inferior, quer devido a uma operação de levantamento ou invalidação de moedas (e.g., devido a uma operação de chantagem realizada sobre o sistema bancário), o conjunto dos nós gerados é enviado aos utilizadores que necessitem de actualizar cadeias de nós representando moedas válidas.

Como descrito, as moedas electrónicas são representadas pelos nós folha de uma Árvore de Merkle. Cada moeda é validada, de modo similiar ao protocolo apresentado em 3.2.2, através da utilização dos nós que compõem o caminho desta mesma até a um nó de topo. Assume-se que a integridade de cada um dos nós de topo se encontra garantida. O sistema contempla também a possibilidade de invalidação (e.g., devido a um ataque) de moedas inseridas no sistema. Esta última propriedade é definida pelo protocolo como "não-rigidez".

O sistema garante que os intervenientes mantêm o estatuto de anónimos ao longo do protocolo. Embora um utilizador tenha de se autenticar no momento prévio à execução de um levantamento, é garantida a impossibilidade de relacionamento entre intervenientes em levantamentos e em depositos de moedas electrónicas.

A utilização de uma Árvore de Merkle consegue, para além da auditabilidade do sistema, solucionar problemas comuns em sistemas de gestão de dinheiro electrónico. No caso de chantagem, um atacante não pode extorquir dinheiro válido ao sistema sem que tal seja detectado, uma vez que as novas moedas teriam de ser adicionadas à Árvore de Merkle, e por conseguinte os nós de topo desta actualizados. Como estes são de conhecimento público, seria fácil para qualquer entidade detectar que moedas foram emitidas para pagar a acção de chantagem. Para além disso, o sistema poderia, posteriormente e em qualquer altura, invalidar essas mesmas moedas. Um outro tipo de ataque bastante comum, o roubo através da emissão de novas moedas por um atacante, está à partida colocado de parte, uma vez que o sistema não assenta na utilização de items que necessitem de protecção (e.g., chaves privadas) para manutenção da integridade dos dados.

3.3 Garantias de Segurança em Sistemas de Troca de Mensagens

Os algoritmos apresentados nas secções anteriores demonstram que já existem actualmente bastantes estudos realizados tanto em torno das questões relacionadas

com o fornecimento de garantias de segurança básicas como das questões relacionadas com o fornecimento de garantias de segurança avançadas em sistemas de troca de mensagens. Por outro lado, não se consegue encontrar num só sistema a satisfação de todas as garantias de segurança que se pretendem obter nesta dissertação (como apresentado no capítulo 1).

A melhor base para a construção de um sistema como o que este trabalho se propõe a desenvolver encontra-se actualmente na utilização de algoritmos de não-repúdio como os apresentados na secção 2.4. Em primeiro lugar, porque estes são os únicos a fornecer garantias de não-repúdio que tornam o protocolo justo. A utilização de assinaturas digitais em todas as mensagens confere integridade ao conteúdo das mesmas. Para se obter autenticação sobre estas, basta que as chaves públicas correspondentes às chaves de assinatura utilizadas estejam associadas a certificados digitais reconhecidos por todos os participantes numa transacção.

Mesmo o tipo de garantias que não se encontram em todos os sistemas de não-repúdio são integráveis nestes. A garantia de confidencialidade, muitas vezes ausente devido a questões relacionadas com desempenho, torna-se extremamente fácil de adicionar a qualquer um dos tipos de algoritmos de não-repúdio existentes (e.g., através da cifra das mensagens utilizando certificados digitais). O mesmo se passa com a utilização de selos temporais sobre as mensagens.

É, no entanto, notória a falta de informação relacionada com a auditabilidade de transacções em protocolos de não-repúdio. Nenhum deles apresenta qualquer sistema que permita uma auditoria por uma entidade credenciada para o efeito e independente de todos os intervenientes no protocolo, no conjunto de transacções realizadas num sistema. A maior parte deste tipo de protocolos refere que cada um dos participantes deve manter as mensagens que lhe permitam salvaguardar a sua posição no caso de disputas perante um juiz, sem no entanto referir o modo como estas devem ser mantidas. Se, por exemplo, os intervenientes decidirem colaborar de modo a eliminar por completo uma transacção, basta eliminarem as provas de não-repúdio geradas pelo protocolo.

Os mais recentes desenvolvimentos relacionados com manutenção da auditabilidade de um repositório encontram-se nos sistemas de comércio e dinheiro electrónico (como apresentado na secção 3.2). Os esquemas apresentados nestes trabalhos assentam na utilização de árvores (como apresentado por Merkle). Nestas últimas, cada nó folha representa informação relativa ao protocolo subjecente e cada nó intermédio representa informação relacionada com a manutenção da auditabilidade do repositório de dados. No entanto, todos estes sistemas foram realizados tendo

em mente um fim e um protocolo de utilização específico, não podendo ser directa e genericamente aplicados em sistemas de troca de mensagens.

Os factos enumerados nesta secção revelam a motivação para a realização do trabalho apresentado no próximo capítulo. Torna-se premente criar um esquema de auditabilidade que possa ser genericamente utilizado em sistemas de trocas de mensagens, de modo a construir um sistema que forneça garantias fortes e avançadas de segurança. Os sistemas de não-repúdio actuais são, pelas garantias de segurança fornecidas por defeito, complementos naturais ao esquema de auditabilidade apresentado no capítulo 4.

Capítulo 4

Um Novo Esquema de Auditabilidade

Neste capítulo é descrito um novo esquema de auditabilidade. Embora o esquema possa ser aplicado a qualquer cenário de transacções electrónicas que necessite de manter registos a longo termo de forma auditável, este é aqui apresentado com o pressuposto da sua utilização num sistema de troca de mensagens. O esquema de auditabilidate foi apresentado em [61].

4.1 Requisitos de Segurança

A auditabilidade de um sistema de troca de mensagens está baseada na possibilidade de recriar uma transacção ou um conjunto de transacções. Este requisito implica a manutenção, num repositório confiável (de aqui por diante referido apenas como "repositório"), do conjunto de mensagens pertencentes a uma determinada transacção. O repositório é composto por unidades básicas a que chamamos registos. Cada mensagem, bem como quaisquer atributos adicionais, é mapeada para um registo específico.[1] Deste modo podemos definir o conceito de manutenção confiável de mensagens como uma série de pressupostos aplicados a um registo:

- **Integridade de conteúdo**: Não é possível corromper (e.g., alterar) o conteúdo de um registo sem detecção.
- **Ordenação temporal**: Os registos devem estar ordenados cronologicamente, e esta ordem não pode ser corrompida sem detecção.
- **Eliminação de registos**: Não é possível eliminar um registo sem detecção.

[1]Cada registo é responsável pela manutenção de uma e só uma mensagem.

- **Inserção de registos**: Não é possível inserir um registo sem autorização. Por autorização entende-se possuir ou ter acesso a um segredo necessário para a inserção de registos.

Daqui em diante será utilizada a palavra "validade"sempre que é feita referência a uma situação em que todos estes pressupostos são verdadeiros.

4.2 Esquema de Auditabilidade

Nesta secção é apresentado um novo esquema que fornece garantias fortes de corresponder a todos os pressupostos identificados na secção 4.1 (i.e., que fornece garantias fortes de ser válido). É assumido que todos os registos são mantidos num repositório. É também assumido que o transporte de mensagens desde o sistema de troca de mensagens até ao repositório é feito sem corrupção das mesmas.

4.2.1 Notação

Será utilizada a seguinte notação para representar elementos e funções do esquema apresentado:

M : mensagem específica a uma determinada transacção

E : elemento de um registo

E_1,E_2 : concatenação dos elementos E_1 e E_2

f_m, f_e, f_h: etiqueta que indica o tipo de registo

L : etiqueta que liga uma mensagem com uma determinada transacção (identificador de transacção)

$R = \{E_1,...,E_n\}$: registo composto pela concatenação dos elementos E_1 a E_n

$H_k(E)$: algoritmo de sumário com cifra aplicado ao elemento E

$H(E)$: algoritmo de sumário aplicado ao elemento E

$s_k(E)$: assinatura digital do elemento E através da aplicação da chave privada k

V_A e S_A : chave de verificação (pública) e de assinatura (privada) da entidade A

$E_{\langle n \rangle}$: elemento pertencente à posição n de uma lista ordenada

$T(E)$: selo temporal aplicado a um elemento E

Seq : identificador de posição de inserção de registo no repositório

4.2.2 Constituição do Repositório

Os elementos básicos de um repositório são registos R_m. Cada registo R_m representa uma mensagem de um sistema de troca de mensagens. Tomando como exemplo um protocolo de não-repúdio (ver secção 2.4.4), cada registo R_m poderá representar uma prova de submissão, recepção ou origem de uma transacção específica. Os registos são inseridos ordenadamente, de acordo com a ordem de entrega das mensagens no repositório. De seguida será apresentado um esquema que fornece garantias fortes de que tanto o conteúdo dos registos como a ordenação dos mesmos não possam ser corrompidos.

Sempre que uma mensagem de uma determinada transacção é enviada para o repositório, um novo registo R_m é gerado da seguinte forma:

$$R_m = \{f_m, Seq_m, L, M, Mac\}$$

$$Mac = H_k(f_m, Seq_m, L, M)$$

O elemento Mac é gerado com base em todos os outros elementos constituintes do registo. Se algum destes elementos for alterado o elemento Mac também o terá de ser, de modo a ser garantida a integridade de todo o registo. Deste modo, apenas quem possuir o segredo k pode adicionar ou alterar registos R_m no repositório, ficando assegurado através deste mecanismo o pressuposto da integridade de conteúdo.

O propósito do elemento Seq_m é garantir que não existem alterações na ordenação dos registos R_m (i.e., garantir que a ordem pela qual estes são inseridos é a ordem pela qual existem no repositório). Este objectivo é alcançado tornando o valor do elemento Seq_m único e sequencial entre os registos R_m. Assim, o valor do elemento Seq_m de um registo R_m é obtido de forma incremental sobre o valor do elemento Seq_m do último registo R_m existente até ao momento.

É agora introduzido o conceito de *época*. Uma época é definida como um conjunto de registos R_m, inseridos em posições adjacentes no repositório, seguidos de um registo R_e. Este último registo é inserido no repositório na posição imediatamente seguinte à posição do último registo R_m da época, e é definido da seguinte forma:

$$R_e = \{f_e, k, V_A, Sig_e, T(Sig_e)\}$$

$$Sig_e = s_{S_A}(f_e, k, H(Mac_{<1>}, ..., Mac_{<n>}))$$

O elemento Sig_e representa uma assinatura digital realizada sobre um subconjunto de elementos pertencentes ao registo R_e, juntamente com um sumário. Este sumário é gerado com base em elementos pertencentes a todos os registos R_m que formam uma época. Sempre que um registo R_e é introduzido no repositório, diz-se que a época subjacente se encontra *fechada*.

O elemento k representa o segredo utilizado para gerar os registos R_m agregados pelo registo R_e. Isto implica que seja utilizado o mesmo segredo na geração dos elementos Mac de todos os registos R_m agregados, e que o valor deste seja alterado de época para época.[2]

Como explicado anteriormente para registos R_m, registos R_e apenas podem ser alterados ou adicionados ao repositório por entidades que possuam um segredo. Neste caso em particular, esse segredo é representado pela chave de assinatura S_A. Através da utilização de um sumário gerado com base em elementos obtidos, de forma ordenada, a partir de todos os registos R_m incluídos na época, o elemento Sig_e fornece garantias de integridade de conteúdo, ordenação temporal, não-eliminação de registos sem detecção e impossibilidade de inserção de registos sem autorização aos registos R_m que constituem a própria época.

[2]Para mais detalhes consultar a secção 4.5.

O sumário acima referido é gerado com base em elementos Mac. Desta forma, não só se garante a integridade destes elementos em cada registo R_m, mas também a integridade de todo o conjunto de registos R_m pertencentes a esta época.

Utilizando somente os elementos e registos definidos até aqui apenas se garantem os pressupostos definidos na secção 4.1 para cada época em particular. No entanto, é também necessário garantir a ordenação temporal entre épocas, bem como a impossibilidade de eliminar por completo uma ou mais épocas sem detecção. Com a concretização destes objectivos garantimos que os pressupostos acima referidos se verifiquem em todo o repositório. Para isso, é necessário modificar a definição do elemento Sig_e da seguinte forma:

$$Sig_{e<n>} = s_{S_A}(f_e, k, H(Mac_{<1>}, ..., Mac_{<n>}), H(Sig_{e<n-1>}))$$

Com esta definição consegue-se que todos os registos R_e se encontrem temporalmente ordenados. Cada registo R_e possui agora um elemento ($Sig_{e<n-1>}$) que representa uma referência inequívoca para o registo R_e que se encontra na posição imediatamente anterior em termos temporais (como ilustrado na Figura 4.1). Para uma entidade conseguir corromper uma ou mais épocas sem detecção de um sistema de validação, todas as épocas geradas posteriormente às épocas alteradas também necessitariam de ser modificadas.

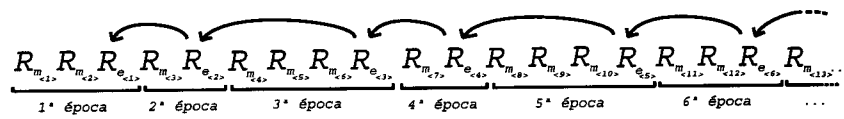

Figura 4.1: Repositório com número não fixo de registos R_m por época

Cada registo R_e actua como um ponto de controle de validação de registos ao longo do repositório. O conceito de gerar pontos de controle e torná-los interdependentes provem da definição de Árvore de Merkle.[3]

Quando pelo menos um dos elementos de um registo do repositório possui uma referência inequívoca (como é o caso do elemento Sig_e) para um outro registo, dizemos que estes registos se encontram "directamente ligados". O primeiro registo

[3]Na secção 4.7 é feita uma comparação entre o esquema aqui apresentado e uma Árvore de Merkle.

R_e a ser gerado no repositório representa um caso particular, visto que não possui nenhuma referência para outro registo R_e. Neste caso o valor de $H(Sig_{e<n-1>})$ deve ser fixo e pré-definido.

Se se assumir que a chave de assinatura utilizada no último registo R_e a ser gerado se encontra sempre sobre o controlo do repositório[4] (i.e., nunca se encontra comprometida) pode-se afirmar que, utilizando o esquema apresentado, os pressupostos definidos na secção 4.1 estão garantidos para todo o repositório.

4.2.3 Verificação da Integridade de Registos

Para validar o conteúdo de um registo R_m é necessário executar os seguintes passos:

1. Identificar a época à qual o registo R_m pertence, identificando assim também o registo R_e que fecha essa mesma época.
2. Validar o conteúdo do registo R_m, verificando para isso se o valor do elemento Mac é igual ao valor de $H_k(f_m, Seq_m, L, M)$. Para realizar esta operação é necessário utilizar o segredo k presente como um dos elementos do registo R_e que fecha a presente época.
3. Obter o elemento $H(Mac_{<1>}, ..., Mac_{<n>})$. Os elementos Mac são obtidos a partir de cada registo R_m pertencente à época corrente.
4. Obter o registo R_e que fecha a época anterior, utilizando-o para obter o elemento $H(Sig_{e<n-1>})$.
5. Verificar se o conteúdo do registo R_e não está corrompido, utilizando a chave de verificação V_A e os elementos obtidos nos passos anteriores para validar a assinatura digital presente no elemento Sig_e.
6. Repetir os passos 3 até 5 para todas as épocas geradas depois da presente. Com este procedimento valida-se a cadeia de registos R_e deste o que fecha a época actual até ao último a ser gerado no repositório.

[4]Para mais detalhes consultar a secção 4.4.

4.3 Partição Hierárquica dos Registos

De modo a validar um registo R_m, o procedimento definido na secção 4.2.3 implica a verificação da integridade de um número de registos directamente proporcional ao número total de registos existentes no repositório. Este procedimento torna-se impraticável à medida que o número de épocas aumenta.

4.3.1 Definições

De modo a resolver-se este problema, é introduzido um novo esquema, que recorre a partições hierárquicas de registos. Assim, em vez de todos os registos R_e se encontrarem directamente ligados, como definido anteriormente, apenas sub-conjuntos de registos R_e estão agora directamente ligados. É introduzida nova notação para apresentar este esquema:

$R_{[y]}$: registo pertencente ao nível hierárquico y

$R_{<n:w>}$: elemento pertencente à posição n do sub-conjunto de registos w

$R_{<l:w>}$: último registo do sub-conjunto de registos w

O último registo de um sub-conjunto de épocas (um registo R_e) já não se encontra directamente ligado ao primeiro registo da próxima época. Em vez disso, encontra-se directamente ligado a um registo hierarquicamente superior. Este novo tipo de registo será definido como R_h. Registos R_h estão também agrupados em sub-conjuntos e directamente ligados entre si, tal como definido para registos R_e. Genericamente, cada vez que um sub-conjunto de registos R_h é terminado com um registo $R_{h_{[y]<l:w>}}$ um novo registo $R_{h_{[y+1]<n:z>}}$ pertencente ao nível hierárquico superior é gerado (como ilustrado na Figura 4.2). Os registos R_h são definidos da seguinte forma:

$$R_{h_{[y]<n:w>}} = \{f_h, V_A, Sig_{h_{[y]<n:w>}}, T(Sig_{h_{[y]<n:w>}})\}, \text{ onde}$$

$$Sig_{h_{[1]<1:w>}} = s_{S_A}(f_h, H(Sig_{e_{<l:z>}}))$$

$$Sig_{h_{[1]<n:w>}} = s_{S_A}(f_h, H(Sig_{e_{<l:z>}}), H(Sig_{h_{[1]<n-1:w>}})) \text{ se } n > 1$$

$$Sig_{h_{[y]<1:w>}} = s_{S_A}(f_h, H(Sig_{h_{[y-1]<l:z>}})) \text{ se } y > 1$$

$$Sig_{h_{[y]<n:w>}} = s_{S_A}(f_h, H(Sig_{h_{[y-1]<l:z>}}), H(Sig_{h_{[y]<n-1:w>}})) \text{ se } n > 1 \text{ e } y > 1$$

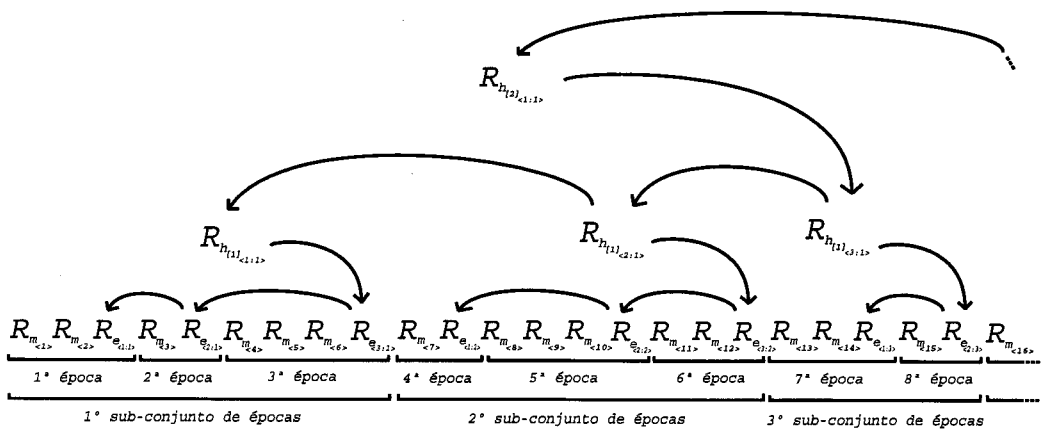

Figura 4.2: Repositório baseado num esquema hierárquico

À semelhança do definido na secção 4.2.2 para o último registo R_e presente no repositório (em determinado momento), assume-se que existe um conjunto de chaves de assinatura que se encontra sempre sobre o controlo do repositório (i.e., cuja privacidade se encontra sempre garantida). Cada uma destas chaves é aplicada na geração de um elemento Sig_e ou Sig_h de um registo identificado como "registo de topo". Os registos de topo têm um papel fundamental no processo de validação da integridade de registos (como apresentado na secção 4.3.2) e encontram-se definidos na secção 4.4.

4.3.2 Verificação da Integridade de Registos

A validação de um registo R_m segue os passos definidos na secção 4.2.3, apenas com algumas alterações. O último passo é redefinido e o procedimento é extendido da seguinte forma:

6. Repetir os passos 3 até 5 para todas as épocas geradas posteriormente à actual e que pertencem ao mesmo sub-conjunto de épocas. Desta forma, a cadeia de registos R_e directamente ligados é validada.

7. Identificar o registo $R_{h_{[1]<n:w>}}$ que se encontra directamente ligado ao último registo do sub-conjunto de épocas actual, $R_{e_{<l:z>}}$, validando a integridade do

seu conteúdo através da validação da assinatura digital presente no elemento $Sig_{h_{[1]<n:w>}}$. Verificar de seguida a integridade de todos os registos $R_{h_{[1]<m:w>}}$ pertencentes ao mesmo sub-conjunto para os quais $m > n$.

8. Identificar o registo $R_{h_{[y+1]<n2:w2>}}$ que se encontra directamente ligado ao último registo do sub-conjunto de épocas actual, $R_{h_{[y]<l:w>}}$, validando a integridade do seu conteúdo através da validação da assinatura digital presente no elemento $Sig_{h_{[y+1]<n2:w2>}}$. Verificar de seguida a integridade de todos os registos $R_{h_{[y+1]<m2:w2>}}$ pertencentes ao mesmo sub-conjunto para os quais $m2 > n2$.

9. Repetir o passo anterior até que seja validada a integridade de um registo de topo.

Utilizando o procedimento descrito acima, a verificação da validade de um registo R_m já não é directamente proporcional ao número total de registos no repositório (como explicado na secção 4.6).

4.4 Registos de Topo

Um registo de topo é definido como um registo R_e ou R_h para o qual não existem referências em qualquer outro registo presente no repositório (como ilustrado na Figura 4.3). O conjunto de registos de topo varia à medida que épocas vão sendo inseridas no repositório, tendo no mínimo um elemento e no máximo tantos elementos quantos os níveis hierárquicos existentes.[5]

Como apresentado na secção 4.3.2, a validação da integridade de um registo implica a validação da integridade de uma cadeia de registos. O último elemento desta cadeia é sempre um registo de topo. Assim, a integridade do repositório depende em última instância da integridade dos seus registos de topo.

A manutenção da integridade dos registos de topo é conseguida (à semelhança do definido para árvores de Merkle) através do anúncio e disponibilização de forma pública destes mesmos registos ou de uma representação inequívoca dos mesmos (e.g., um sumário). Isto pode ser conseguido por vários meios:

- Publicação num repositório electrónico de acesso público (e.g., LDAP, FTP[59], HTTP[34])

[5]Considera-se que o conjunto de registos R_e representa um nível hierárquico (o nível hierárquico mais baixo).

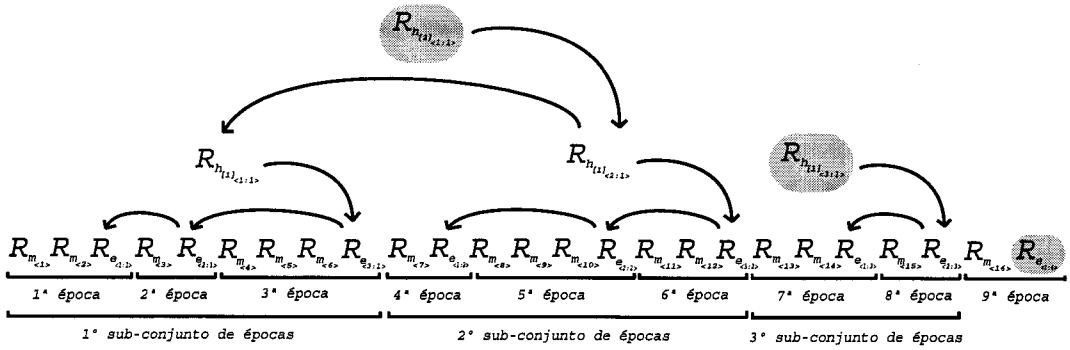

Figura 4.3: Conjunto de registos de topo existentes no repositório num determinado momento

- Publicação em papel (e.g., jornais[6], revistas) de uma representação "legível" dos registos (e.g., a representação codificada em Base 64[43]).

É essencial garantir que os registos de topo apenas possam ser publicados por quem de direito (neste caso pelo esquema de auditabilidade). Uma solução para a manutenção das garantias de segurança necessárias à publicação dos registos de topo passa por assinar um sumário do conjunto dos registos utilizando um segredo gerado especificamente para esse fim e reconhecido publicamente.[7] Essa assinatura seria publicada conjuntamente com a representação dos registos de topo.

4.5 Análise da Segurança

A integridade do conteúdo de um registo R_m está baseada na verificação da validade do elemento *Mac*. A segurança deste está, por sua vez, baseada na garantia de segurança do algoritmo de sumário utilizado para o calcular, bem como na manutenção da privacidade do segredo k.

Assim, deverá ser utilizado um algoritmo de sumário cuja inviolabilidade esteja perfeitamente demonstrada. O segredo k deverá ser gerado de acordo com as capacidades computacionais existentes durante a época à qual o registo R_m pertence, minimizando o risco de ataques bem sucedidos sobre o elemento *Mac*. Para além disso, a privacidade do segredo k deve ser mantida até que a época corrente seja fechada com a introdução de um registo R_e no repositório.

[6] A empresa *Surety* publica todas as semanas no jornal *New York Times* um sumário dos dados que lhe são entregues (no âmbito de um serviço de notário digital) nessa mesma semana.

[7] A utilização de certificados digitais fornece as garantias de segurança necessárias.

O segredo k utilizado na geração de elementos Mac de uma determinada época encontra-se presente como um dos elementos do registo R_e que finaliza a época. Este facto não diminui de modo algum as garantias de segurança do esquema apresentado, uma vez que nesse momento a época encontra-se fechada e a garantia da sua integridade reside agora na garantia de integridade do elemento Sig_e.

A integridade do conteúdo de um registo R_e ou R_h está baseada na integridade, respectivamente, dos elementos Sig_e ou Sig_h. A integridade destes, por sua vez, está baseada na segurança do algoritmo de assinatura utilizado, bem como na garantia de privacidade da chave de assinatura. Se esta chave ficar comprometida também toda a época ficará comprometida, devido à possibilidade de re-assinatura do registo subjacente sem detecção por parte de um sistema de validação.

Em todo o caso, para que a violação da integridade de um registo R_m se torne bem sucedida (i.e., totalmente indetectável) é necessário corromper a cadeia de registos directamente ligados (que termina sempre com um registo de topo). Este procedimento implica a corrupção do registo de época R_e bem como de todos os registos R_e e R_h que, directa ou indirectamente, mantêm uma referência para o primeiro (como ilustrado na Figura 4.4).

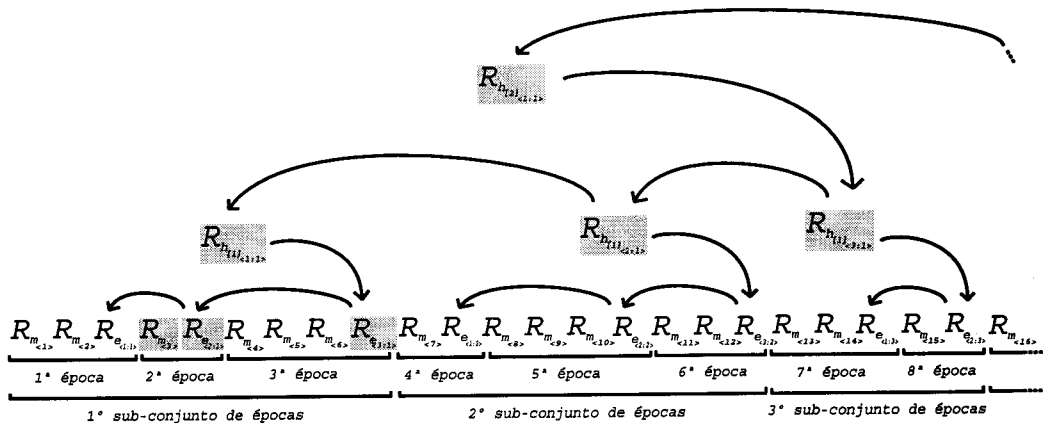

Figura 4.4: A violação do registo $R_{m<3>}$ implica a violação de todos os registos ligados

O facto descrito acima, juntamente com a possibilidade de renovação de chave de assinatura de uma época para a seguinte, tornam a corrupção sem detecção (por parte de um sistema de validação) dos registos do repositório um processo muito

difícil de concretizar com sucesso. Uma vez que as chaves de assinatura utilizadas nos registos R_e e R_h podem (e devem) ser adaptadas às capacidades computacionais existentes à data, o risco potencialmente maior de corrupção do repositório decorre da capacidade de manutenção da privacidade das próprias chaves de assinatura. De modo a minimizar este risco, deve-se eliminar cada chave de assinatura após a cessação de utilização da mesma. Uma vez que é utilizada criptografia de chave pública, a chave de assinatura não é necessária para verificar a validade dos registos.

Mesmo utilizando os esquemas de segurança acima descritos, é provável que chaves de assinatura utilizadas há bastante tempo sejam eventualmente comprometidas (e.g., através de ataques de força bruta[68] ou de avanços na área da criptografia). Mesmo nestes casos a integridade de um registo continua a ser possível de verificar. Apenas é necessário verificar se o elemento T permanece válido. Este elemento contém um selo temporal referente a um determinado elemento Sig_e ou Sig_h. Desta forma, é colocado um limite temporal sobre a data na qual a assinatura foi realizada. Se o selo temporal for anterior à data em que a chave de assinatura foi comprometida (ou se pensa ter sido), o registo permanece válido. O elemento T é gerado através da aplicação da assinatura digital de uma autoridade competente, como definido na secção 2.2. Assume-se que a chave de assinatura desta mesma autoridade não se encontra nunca comprometida.

4.6 Eficiência

A adaptação do tamanho tanto do segredo k como das chaves de assinatura aplicadas nos elementos Sig_e e Sig_h é uma operação trivial, uma vez que ambos podem ser substituidos no fim de cada época.

A geração de registos R_e e R_h é uma operação muito mais morosa do que a geração de registos R_m. Este facto deve-se sobretudo às características da criptografia de chave pública (utilizada nos primeiros) comparativamente às características da criptografia simétrica (utilizada no segundo). Para minimizar as diferenças de desempenho que advêm deste facto é sempre possível aumentar o número de registos R_m por época, mantendo sempre em consideração o facto de que nunca se deve permitir que as capacidades computacionais subjacentes à época permitam corromper tanto o segredo k como os elementos Mac.

A privacidade do segredo k termina no fim de cada época, tornando a verificação da validade de um registo R_m uma operação trivial e bastante rápida, mesmo após longos períodos de tempo. A operação de verificação da validade de um registo

R_m implica a validação de $O(\log_x n)$ registos, onde

n = número total de registos R_e e R_h existentes no repositório

x = número médio de registos em cada sub-conjunto

4.7 Comparação com Árvores de Merkle

Uma vez que o conceito de Árvore de Merkle é central ao esquema apresentado, torna-se importante estabelecer um paralelismo deste com o esquema definido e apresentado neste capítulo. Ambas as abordagens utilizam o conceito de níveis hierárquicos de registos (ou nós) interligados. Os registos do nível mais baixo mantêm os dados enquanto que os registos de topo (i.e., aqueles para os quais não existe nenhuma referência a partir de outro registo) são expostos publicamente, sendo também responsáveis últimos pela garantia da manutenção da auditabilidade dos dados. A validação de um registo de dados implica a validação de todos os registos que mantêm referências directas ou indirectas para este até ser validado um registo de topo.

Embora as abordagens utilizem vários conceitos base em comum, existem bastantes diferenças entre ambas. No esquema aqui apresentado os vários elementos de cada nível hierárquico estão directamente ligados (formando sub-conjuntos), o que limita o período temporal durante o qual são nós de topo (e logo diminui a possibilidade de ataques bem sucedidos sobre estes). Embora esta estrutura leve o procedimento de validação de um registo R_m a processar (em certos casos) mais nós que uma Árvore de Merkle, ambos os protocolos são da mesma ordem de grandeza.

O elemento L fornece um meio simples de relacionar mensagens no esquema apresentado, adequando-o quer para suporte de auditabilidade de outros tipos de algoritmos (e.g., protocolos de não-repúdio) quer para garantias de auditabilidade de sistemas de comércio electrónico. A utilização do elemento T permite limitar o momento no qual cada mensagem foi inserida no repositório, podendo adequar o esquema para aplicações para as quais a data de geração e a ordenação temporal de mensagens seja um factor de relevo (e.g., serviço de apostas, compra de bilhetes). A existência de diferentes tipos de registos para diferentes posições da árvore construída permite uma maior protecção sobre os registos mais expostos a ataques em cada momento (e.g., utilização do elemento Mac, conceito de "registos directamente ligados"e de sub-conjuntos de registos).

A possibilidade de utilização de ferramentas fornecidas por entidades externas ao sistema (com reconhecida independência deste) para garantir pressupostos de segurança subjacentes ao mesmo, adiciona-lhe um grau extra de credibilidade perante a comunidade de utilizadores. Nesta implementação poderão ser utilizadas como entidades externas uma AC (utilizada para a gestão do ciclo de vida dos certificados digitais aplicados na geração de registos R_e e R_h) e uma TSA (utilizada para geração dos selos temporais utilizados na geração de registos R_e e R_h).

4.8 Um Sistema Completo

O esquema apresentado neste capítulo fornece garantias fortes de auditabilidade a um sistema que o utilize, bem como informação temporal precisa relativamente ao momento de entrega de informação ao repositório.

Como demonstrado na secção 3.1, existem actualmente protocolos amplamente difundidos e cuja segurança foi exaustivamente analisada, que fornecem garantias como integridade, confidencialidade, autenticação e não-repudio de origem, submissão e recepção. Seguidamente é demonstrado que, através da aplicação do esquema acima definido a este tipo de protocolos, se consegue obter um sistema coerente com os objectivos propostos neste trabalho.

Cada transacção de um protocolo de não-repúdio é genericamente constituída por uma sequência de mensagens. Cada mensagem possui um identificador de transacção que poderá ser utilizado para definir o identificador L associado a um registo R_m. Desta forma torna-se simples a reconstituição de uma transacção a partir do sistema de auditabilidade. Uma vantagem clara para cada interveniente numa transacção (para além da óbvia possibilidade de reconstituição auditável da mesma) prende-se com o facto de deixar de ser necessário manter na sua posse as mensagens utilizadas na defesa do seu ponto de vista perante um juiz.

Os protocolos de não-repúdio onde existe uma EC pela qual passem todas as trocas de mensagens são facilmente integráveis com o esquema de auditabilidade proposto. A EC pode, desde que os intervenientes assim o permitam, realizar o papel de intermediário entre o protocolo e o sistema de auditabilidade, enviando para este último todas as mensagens trocadas. Sempre que um dos outros intervenientes na transacção pretender, pode verificar se uma determinada mensagem já foi inserida no sistema de auditabilidade e decidir, com base nessa informação, se deve abortar ou não o protocolo. Deste modo obtém-se uma garantia adicional sobre a honestidade da EC, bem como um maior controlo sobre a auditabilidade da transacção.

A participação da EC de modo parcial num protocolo de não-repúdio garante, de modo similar aos protocolos acima referidos, que uma transacção não possa ser eliminada por completo por uma conjugação de interesses dos restantes intervenientes na mesma. Uma vez que a EC tem sempre participação activa no protocolo, as mensagens recebidas e enviadas por esta última são inseridas no sistema de auditabilidade. Deve ficar à responsabilidade dos outros intervenientes na transacção a decisão sobre a entrega ou não ao sistema de auditabilidade das mensagens a que só estes têm acesso. É no entanto de realçar que na maioria dos casos o maior prejudicado pela não entrega de uma determinada mensagem ao sistema de auditabilidade é a própria entidade detentora da mensagem.

Pode suceder o caso em que, por razões de confidencialidade, uma determinada mensagem não possa ser inserida no sistema de auditabilidade. Embora o sistema garanta a auditabilidade dos dados inseridos, não garante que estes não possam eventualmente ser lidos por entidades com fins maliciosos. Estas últimas poderão conseguir (devido a avanços tecnológicos) quebrar dados cifrados. Mesmo a utilização de um sumário dos dados confidenciais em representação destes poderá não resolver o problema. Esta situação acontece no caso em que a mensagem provem de um conjunto limitado e conhecido de mensagens. Através de tentativas a todo o espaço de mensagens possíveis um atacante colocará eventualmente a mensagem a descoberto. Neste caso a mensagem poderá ser mantida num sistema de arquivamento de dados confiável e confidencial. Deste modo no sistema de auditabilidade pode simplemente ser mantida uma referência que permita identificar inequivocamente a mensagem no sistema de arquivamento.

Outro tipo de problema prende-se com a necessidade de manutenção da mensagem na posse de quem de direito durante todo o período temporal em que poderá ser iniciada uma disputa sobre a transacção subjacente. Neste caso as mensagens de uma transacção poderão não ser inseridas pela ordem pela qual foram geradas. Se a ordem pela qual estas foram geradas é crucial para o protocolo então estas poderão ser inseridas em conjunto com um selo temporal. Se tal não for possível então poderá ser inserida uma referência inequívoca à mensagem (como apresentado no parágrafo anterior).

A utilização do tipo de protocolo de não-repúdio apresentado na secção 3.1.3 implica a total ausência de participação de uma EC numa transacção, excepto quando se verifique uma situação anormal. Este tipo de protocolos não resiste à possibilidade de eliminação total de uma transação por acordo entre os intervenientes. Nestes casos a auditabilidade de uma transacção dependerá sempre da honestidade dos próprios intervenientes.

Capítulo 5

Implementação

Neste capítulo são primeiramente definidos procedimentos e regras que devem ser pontos de referência na implementação do esquema de auditabilidade proposto no capítulo 4 (de aqui por diante designado simplesmente por "Esquema"). Posteriormente é descrita uma implementação de referência do Esquema. Por fim são apresentados resultados de testes práticos realizados tendo por base essa mesma implementação. Esta implementação do Esquema foi apresentada em [62].

5.1 Protolocos, Estruturas e Algoritmos

Nesta secção são listados e justificados os protocolos, estruturas e algoritmos utilizados na implementação do Esquema. É colocada especial atenção na apresentação de soluções que apliquem tecnologia utilizada actualmente e com provas dadas nas áreas da segurança e eficiência.

O algoritmo de sumário utilizado para gerar elementos Mac deve fornecer garantias de segurança e eficiência, uma vez que é utilizado na geração de cada registo R_m. Um algoritmo de sumário com cifra como $HMAC$[23] satisfaz todos os requisitos acima definidos.

De modo similar, é importante utilizar um algoritmo de sumário que seja seguro e utilizado em larga escala, como SHA-1[56]. Este algoritmo é aplicado na geração dos dados necessários à construção de elementos Sig_e e Sig_h. Caso os últimos desenvolvimentos (como os apresentados na secção 2.1.2) revelem falhas no algoritmo SHA-1 existem já alternativas, também definidas em [56], como SHA-256, SHA-384 ou SHA-512.

Os elementos Sig_e e Sig_h são gerados recorrendo a criptografia assimétrica. A chave pública utilizada na verificação da validade desses elementos deverá estar presente num certificado digital $X.509$[37], devendo também este último ser emitido por uma AC reconhecida por todos os intervenientes na transacção em questão. Embora o repositório necessite de estar de acordo com múltiplos requisitos de segurança, operações delicadas como a gestão do ciclo de vida de certificados devem ser sempre feitas recorrendo aos serviços de uma AC.

De modo a maximizar a segurança necessária na geração de registos R_e e R_h, cada certificado digital deverá ser utilizado apenas durante um período temporal limitado. Deste modo minimiza-se o período temporal durante o qual a chave privada correspondente pode sofrer ataques que a tentem colocar a descoberto.[1]

É essencial para a segurança do Esquema que seja estabelecido um protocolo seguro para os pedidos de certificados digitais[17] serem realizados à AC. Existem múltiplas aproximações que fornecem garantias fortes de segurança, como por exemplo a utilização de uma ligação TLS mutuamente autenticada. Outra hipótese passa pela utilização de um segredo (partilhado pelo Esquema e pela AC), nas extensões do pedido de certificado. Este segredo deverá evoluir de um pedido para outro de modo a evitar ataques sobre os pedidos.

Outro ponto onde a segurança pode ser aumentada passa pela definição de um acordo entre a AC e o Esquema no qual se define um conjunto de extensões $X.509$ a ser incluídas em todos os certificados emitidos para este serviço. Estas extensões, para além de indicarem que um certificado apenas deve ser utilizado no Esquema, podem ligar inequivocamente o próprio certificado com um determinado registo R_e ou R_h. Através da utilização deste mecanismo pode-se, por exemplo, diminuir o risco de fraudes de substituição ou de re-utilização de certificados.

A validação de um registo R_e ou R_h deverá implicar a validação do certificado digital associado às assinaturas presentes. Esta operação passa pela verificação da data de expiração do certificado, bem como pela verificação da sua validade recorrendo quer a uma CRL, quer a um serviço de $OCSP$. Sempre que um certificado se encontrar inválido (e.g., revogado, suspenso, expirado) é necessário verificar (como definido na secção 2.2) o selo temporal presente no elemento T, de modo a ser tomada uma decisão sobre a validade do registo em questão.

[1] Após a utilização de uma chave privada para gerar um elemento Sig_e ou Sig_h esta deverá ser eliminada, uma vez que já não é necessária para o procedimento de validação do registo.

Um factor essencial de segurança prende-se com a protecção das chaves de assinatura (enquanto não forem eliminadas), bem como do segredo k (enquanto não for colocado a descoberto no fim de uma época). Assim, deve ser utilizado *hardware* criptográfico que permita gerar os elementos referidos e realizar as operações criptográficas associadas no seu interior (i.e., recorrendo a processador e memória próprios). O *hardware* deve obedecer aos requisitos de segurança definidos na especificação $FIPS\ 140-2$[53].

A implementação do repositório deve ser realizada sobre uma estrutura cujo desempenho (quer de operações de escrita quer de leitura) não sofra uma degradação assinalável com a inserção de um elevado número de dados. A robustez da estrutura é também uma característica essencial, bem como a disponibilização de mecanismos de realização de cópias de segurança (e respectiva operação de recuperação) sobre os dados. É desejável que existam diversas implementações da estrutura escolhida, com provas sólidas de fiabilidade e desempenho em sistemas com necessidades semelhantes. A utilização de uma base de dados relacional enquadra-se na perfeição nas características pretendidas para a estrutura de implementação do repositório.

5.2 Tecnologia Utilizada

A implementação do sistema é realizada recorrendo à linguagem de programação Java. A Máquina Virtual e o compilador Java utilizados pertencem à versão *Java 2 Platform, Enterprise Edition 1.4 SDK*. Esta linguagem preenche totalmente os requisitos necessários a uma implementação deste tipo, nomeadamente:

- Múltiplas possibilidades de utilização de pacotes criptográficos;
- *Interfaces* para acesso a serviços fornecidos por uma AC, como OCSP;
- *Interfaces* e *drivers*, com fiabilidade e desempenho amplamente testados, para as bases de dados mais comuns, como *MySQL*, *PostgreSQL* ou *Oracle*;
- *Interfaces* para acesso a directorias, nomeadamente para acesso a serviços de LDAP;
- *Interfaces* para acesso a serviços de *timestamping*;
- Independência de sistema operativo; e
- Facilidade de integração com diferentes servidores aplicacionais.

A restante tecnologia utilizada é a seguinte:

- **Sistema Operativo:** *Linux*
- **Capacidades Criptográficas:**
  - ***Software :*** IAIK[8]
  - ***Hardware :*** NCIPHER[11]
- **Servidor *LDAP*:** OpenLDAP[12]
- **Base de Dados:** MySQL[10]
- **Servidor Aplicacional:** JBoss[9]

5.3 Arquitectura

Nas próximas secções é apresentada a arquitectura da implementação do Esquema. São descritos os módulos que compõem o sistema, bem como a estrutura definida para o repositório e para a disponibilização dos registos de topo. É apresentado o sistema critográfico utilizado, o protocolo de interacção com a AC e o processo de gestão de chaves e certificados digitais.

5.3.1 Módulos

O Esquema encontra-se implementado em módulos que interagem entre si e com serviços externos (como ilustrado na Figura 5.1):

- **Serviço de Certificação Digital**

  Serviço externo ao Esquema composto por uma AC independente do mesmo. Esta é responsável pela emissão e revogação de certificados digitais utilizados na geração de registos R_e e R_h. O procedimento de interacção com o Esquema e os processos associados à emissão de certificados estão descritos na secção 5.3.5.

- **Serviço de Validação Cronológica**

  Serviço externo ao Esquema responsável pela emissão de selos temporais incluídos nos registos R_e e R_h. Este serviço encontra-se sincronizado com o Observatório Astronómico de Lisboa.[2] A comunicação com a Autoridade de Validação Cronológica associada é realizada de acordo com o modelo especificado em [19].

[2]Para mais detalhes consultar a secção 2.2.

- **Módulo de Gestão do Repositório**

  Módulo que assegura as operações realizadas sobre a base de dados (e.g., adição de registos, pedidos de leitura de mensagens) que implementa o repositório associado ao Esquema. Para além disso é responsável pela manutenção dos registos de topo em repositório público (como sugerido na secção 4.4).

- **Módulo de Gestão de Certificados**

  Módulo que assegura a interacção com o Serviço de Certificação Digital. É responsável pela entrega de pedidos de certificados à AC associada, recebendo posteriormente os certificados emitidos. Estes são mantidos num directório *LDAP* [69], do qual faz de *interface* para os restantes módulos.

- **Módulo de Gestão Criptográfica**

  Módulo responsável pela realização de todas as operações criptográficas (e.g., sumários, assinaturas) necessárias ao Esquema. Assegura também a geração de chaves secretas e privadas e a geração de pedidos de certificado e entrega dos mesmos ao Módulo de Gestão de Certificados.

- **Módulo de Processamento**

  Módulo responsável pelo processamento de cada mensagem M (em conjunto com o identificador de transacção L) recorrendo aos restantes módulos do Esquema (como apresentado na secção 5.4.1). Tenta garantir que existem sempre certificados prontos a ser utilizados quando uma época termina (requisitando a geração de pedidos de certificado ao Módulo de Gestão Critográfica antes de estes necessitarem de ser utilizados).

- **Módulo de Recepção de Pedidos**

  Módulo responsável por receber os pedidos de inserção de dados no repositório. Após a autenticação dos emissores entrega os pedidos ao Módulo de Processamento.

- **Módulo de Validação de Registos**

  Módulo que executa o procedimento de validação de registos do repositório (como descrito na secção 5.4.2). Este procedimento pode ser executado pelo próprio Esquema ou por pedidos de validação realizados por clientes.

Figura 5.1: Arquitectura do Esquema

5.3.2 Repositório

A utilização de uma base de dados como suporte à implementação do repositório levou à divisão deste por um conjunto de tabelas. Embora cada tabela represente um tipo de registo, os campos que compõem cada uma destas não correspondem na exactidão aos elementos definidos no Esquema. Estas alterações resultam, como justificado de seguida, quer de necessidades de desempenho, quer de ajustamentos resultantes da adaptação do Esquema a uma implementação real.[3]

- **Elementos da tabela de registos R_m:**

  M : mensagem específica a uma determinada transacção

  L : identificador de transacção. Este valor pode ser nulo no caso da mensagem M não se encontrar associada a nenhuma transacção

  Mac : sumário assinado de todos os elementos do registo

  Pr_m : posição de inserção do registo no repositório

- **Elementos da tabela de registos R_e:**

[3]Uma vez que diferentes tipos de registos estão identificados por diferentes tabelas, os elementos indicativos do tipo de registo (f_m, f_e e f_h) deixam de ser necessários. Para além disso os elementos V_A são implementados através da utilização de certificados digitais, identificados pelos campos $Cert_e$ e $Cert_h$.

k : chave simétrica utilizada na construção de elementos Mac

Sig_e : assinatura digital realizada sobre os elementos do registo e certificado digital associado

T : selo temporal aplicado à assinatura Sig_e

Pr_e : posição de inserção do registo no repositório

Sce_e : identificador do sub-conjunto de épocas ao qual pertence o registo

$Ultm_e$: valor indicativo sobre se o registo é o último de um sub-conjunto de épocas

Pr_i : identificador do primeiro registo R_m pertencente à época

Pr_f : identificador do último registo R_m pertencente à época

- **Elementos da tabela de registos R_h:**

Sig_h : assinatura digital realizada sobre os elementos do registo e certificado digital associado

T : selo temporal aplicado à assinatura Sig_h

Hr_h : nível hierárquico

Pr_h : posição de inserção do registo no repositório

Sce_h : identificador do sub-conjunto de épocas ao qual pertence o registo

$Ultm_h$: valor indicativo sobre se o registo é o último de um sub-conjunto de épocas

O tipo de implementação apresentado requer que sejam incluídos campos adicionais nos registos R_e e R_h. Estes campos são necessários para a validação dos registos do repositório (como justificado na secção 5.4.2). Assim, os campos adicionados foram os seguintes:

- Pr_e e Pr_h - À semelhança do que se passa para registos R_m, estes campos fornecem informações sobre a posição na qual o próprio registo foi inserido no repositório.
- Sce_e e Sce_h - Foram introduzidos com o propósito de aumentar substancialmente o desempenho do procedimento de validação de registos. A sua utilização permite identificar facilmente registos directamente ligados (como apresentado na secção 5.4.2).
- $Ultm_e$ e $Ultm_h$ - O seu propósito é similar ao da utilização dos campos Sce_e e Sce_h.

Os registos R_e incluem ainda dois campos que indicam qual o conjunto de registos R_m que formam a própria época. Este conjunto compreende todos os registos R_m que foram inseridos entre as posições Pr_i e Pr_f (i.e., cujo valor do campo Pr_m se encontra entre os valores dos campos Pr_i e Pr_f).
Aos registos R_h foi adicionado o campo Hr_h que indica qual o nível hierárquico ao qual o registo pertence.

De referir ainda que o elemento V_A foi eliminado quer dos registos R_e quer dos registos R_h. Este facto deve-se à utilização da estrutura *SignedData* (presente na especificação *PKCS7*[14]) nos campos Sig_e e Sig_h. Esta estrutura permite transportar junto com a assinatura dos dados o certificado utilizado para a validar.

5.3.3 Manutenção dos registos de topo

A disponibilização pública dos registos de topo é assegurada pelo módulo de Gestão do Repositório. Estes registos são publicados num serviço *LDAP* de acesso universal (i.e., podem ser livremente consultados). Para além disso é também publicado um valor utilizado para a verificação da integridade deste repositório. Este valor representa uma assinatura digital realizada sobre o conjunto de sumários representativos dos nós de topo.

A assinatura é realizada pelo Módulo de Gestão Criptográfica a pedido do Módulo de Gestão do Repositório. Esta é re-calculada sempre que é inserido e/ou retirado um novo registo R_e ou R_h do conjunto de registos de topo. A assinatura

é realizada através da utilização de uma chave privada correspondente a um certificado emitido pela AC associada ao serviço. Este certificado contém informações (através da utilização das extensões $X.509$) que indicam que o mesmo foi emitido especificamente para o propósito de assinar o conjunto de nós de topo.

5.3.4 Sistema Criptográfico

A manutenção da privacidade das chaves secretas e privadas utlizadas na geração de registos é essencial na implementação proposta. Para a geração das chaves é utilizado *hardware* criptográfico especializado. Este sistema possui processador e memória próprios, encontrando-se de acordo com a especificação $FIPS\ 104-2$ *Level* 3. Todas as operações criptográficas que impliquem a utilização ou geração de alguma das chaves referenciadas são executadas recorrendo ao processador e à memória do próprio *hardware*. A interacção do Módulo de Gestão Criptográfica com o *hardware* é realizada através de um *interface* PKCS11[18].

A utilização de PKCS11 torna possível definir quais as chaves que podem ser reconstituidas num meio exterior ao próprio *hardware*. Assim, as chaves privadas utilizadas na geração dos elementos Sig_e e Sig_h são definidas como não-exportáveis (i.e., impossíveis de utilizar fora do *hardware*), enquanto que o segredo k utilizado na geração de registos R_m é definido como exportável (uma vez que é utilizado como elemento do registo R_e). Este procedimento torna extremamente improvável a captura de qualquer chave por parte de um atacante.

O facto de a chave se encontrar protegida no interior do *hardware* criptográfico não impossibilita por si só o abuso desta para execução de operações não autorizadas. É necessário proteger o acesso a todas as operações relacionadas com a manipulação da chave (e.g., geração de uma assinatura digital, geração da própria chave). O *hardware* criptográfico utilizado funciona com base em cartões inteligentes. Cada chave pode ser dividida num conjunto de cartões, apenas podendo ser utilizada quando se reúne um sub-conjunto pré-definido desses mesmos cartões.

Uma forma de limitar as quebras de segurança passa pela divisão do conjunto de cartões por diferentes entidades. É importante que os cartões sejam entregues a entidades envolvidas no processo de interacção com o Esquema, de modo a que todas possam ter algum tipo de responsabilidade pela manutenção da segurança do mesmo. Para que as chaves associadas ao conjunto de cartões possam ser utilizadas é necessário que um sub-conjunto de entidades detentoras de cartões executem um procedimento de autorização junto do *hardware* criptográfico. O procedimento passa pela utilização de cada cartão no *hardware* criptográfico em conjunto com a

introdução de uma frase secreta no Módulo de Gestão Criptográfica. Este procedimento permite ao *hardware* a obtenção da informação suficiente para reconstituir as chaves privadas associadas ao conjunto de cartões (como ilustrado na Figura 5.2).

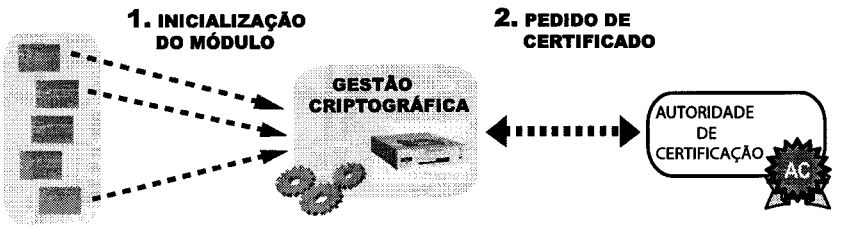

Figura 5.2: Para inicializar o Módulo de Gestão Criptográfica são necessários 3 cartões de um conjunto de 5

O procedimento de autorização descrito apenas necessita de ser executado uma vez por cada inicialização do Módulo de Gestão Criptográfica. A partir desse momento esse módulo pode utilizar as chaves existentes no *hardware* criptográfico. Cada chave é gerada juntamente com uma etiqueta identificativa da mesma. Cada etiqueta é única no conjunto de cartões ao qual pertence a chave, sendo definida pela entidade responsável pela geração desta última (neste caso pelo Módulo de Gestão Criptográfica).

5.3.5 Interacção com a Autoridade de Certificação

De modo a maximizar a segurança do protocolo de pedido de certificados é necessária a definição de um protocolo de interacção com uma AC. Este última é responsável pela emissão e revogação de todos os certificados utilizados nos registos R_e e R_h. A geração de pedidos de certificados é (como definido na secção 5.3.4) sempre realizada no interior de *hardware* criptográfico.

O protocolo de interacção definido passa pela entrega dos pedidos de certificado à AC sobre uma ligação segura TLS autenticada mutuamente. A chave privada utilizada para autenticação do sistema de auditabilidade deve corresponder à chave pública presente no último certificado emitido para o nível hierárquico para o qual se está a pedir um novo certificado. Para além disso deve ser enviada informação referindo se o certificado pedido será o último de um sub-conjunto de épocas. Esta

informação é transmitida como uma extensão $X.509$ pré-definida do pedido de certificado. Este pedido é por sua vez enviado em formato $PKCS10$ [17].

Existe um caso especial no qual o pedido não é transmitido utilizando para autenticação a chave privada correspondente ao último certificado existente para o nível hierárquico pretendido. Isto acontece quando é gerado o primeiro certificado de um novo nível. Neste caso deve ser utilizada a chave privada correspondente ao último certificado emitido para o nível hierárquico mais alto existente na altura. Este último, por sua vez, deve conter uma extensão indicando ser o último de um sub-conjunto.

A AC possui conhecimento sobre qual o último certificado emitido para cada nível hierárquico. Esta, com base nessa informação em conjunto com a informação obtida na transmissão do pedido, emite um novo certificado digital contendo as seguintes extensões (como ilustrado na Figura 5.3):

NH Nível hierárquico a que pertence. O valor desta extensão é indicado no pedido de certificado e, no caso de se tratar de um pedido para um certificado a utilizar na construção de um registo R_h,[4] deve corresponder ao valor do campo Hr_h.

PR Posição relativa dentro do nível hierárquico. O valor desta extensão é incremental sobre o correspondente valor existente no último certificado emitido para o nível em questão. Este valor deve ser igual ao valor presente no campo Pr_e ou Pr_h sempre que o certificado pedido seja utilizado respectivamente num registo R_e ou R_h.

SCE Sub-conjunto de épocas a que pertence. O valor desta extensão é obtido a partir do correspondente valor existente no último certificado emitido para o nível em questão. Este valor é incrementado sempre que o último certificado emitido finalizar um sub-conjunto de épocas e deve corresponder ao valor presente no campo Sce_e ou Sce_h sempre que o certificado pedido seja utilizado respectivamente num registo R_e ou R_h.

ULTM Indicação sobre se o certificado é o último de um sub-conjunto. O valor desta extensão deve corresponder ao valor presente no campo $Ultm_e$ ou $Ultm_h$ sempre que o certificado pedido seja utilizado respectivamente num registo R_e ou R_h.

[4]No caso de se tratar de um pedido de certificado para um registo R_e este valor tanto pode ser nulo como 0.

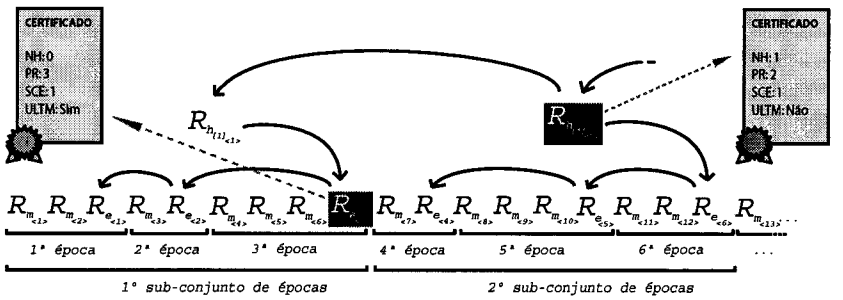

Figura 5.3: Cada certificado contém 4 extensões: NH (Nível Hierárquico), PR (Posição Relativa), SCE (Sub-conjunto de épocas) e ULTM (valor indicativo sobre se o registo é o último de um sub-conjunto de épocas)

As extensões apresentadas estão identificadas como críticas e são privadas ao Esquema (i.e., não são entendidas por qualquer outro sistema que tente interagir com algum dos respectivos certificados). As extensões NH e PR juntas formam um identificador que é único entre todos os certificados emitidos para este serviço, evitando deste modo fraudes de substituição. As extensões SCE e ULTM são utilizadas para optimizar o desempenho do processo de validação de registos do Esquema (que se encontra descrito na secção 5.4.2).

Uma vez que todos os valores presentes nas extensões dos certificados digitais emitidos também se encontram representados em campos das tabelas que compõem esta implementação do Esquema, poder-se-ía pensar em eliminar uns ou outros (de modo a evitar duplicação de informação). No entanto a duplicação de valores encontra-se justificada nos parágrafos seguintes.

A existência dos valores como extensões $X.509$ justifica-se como forma de controle da sequência de certificados emitidos. Ao transferir-se esta responsabilidade para um serviço externo ao Esquema (i.e., para a AC) consegue-se evitar ataques de substituição de certificados (sendo cada certificado emitido apenas uma vez para cada registo específico) bem como ataques de eliminação de épocas ou inserção de épocas forjadas.

A existência dos valores como campos da base de dados justifica-se com o aumento substancial do desempenho do protocolo de validação de registos. Se estes campos não existissem o protocolo referido teria de interpretar as extensões presentes nos certificados para conseguir recriar uma cadeia de registos directamente ligados, o que não seria um procedimento viável do ponto de vista do desempenho das operações de pesquisa no repositório.

5.3.6 Gestão de Chaves e Certificados Digitais

É importante que o pedido do primeiro certificado a utilizar pelo Esquema não seja feito de forma automática. Uma vez que ainda não existem certificados emitidos, este pedido encontra-se mais exposto a ataques do que qualquer outro emitido posteriormente. Isto leva a que este não possa ser transmitido de acordo com as especificações apresentadas na secção 5.3.5. Uma das possibilidades é a entrega manual do pedido (e respectiva emissão do certificado) pelo Esquema à AC.

Em cada momento existe, no conjunto de cartões associado ao Esquema, apenas uma chave secreta utilizada para gerar elementos *Mac*. Tal é possível porque o Esquema não necessita de nenhum tipo de interacção com outras entidades (e.g., com a AC) para a geração desta mesma chave. Deste modo esta pode ser gerada imediamente após o fim de uma época sem causar alterações significativas no desempenho do próprio Esquema. Uma vez que o serviço prestado pelo Esquema pode ter de ser re-iniciado em qualquer altura (e.g., por falha de *hardware*) é necessário que esta chave seja identificada inequivocamente por uma etiqueta pré-definida.

A optimização do desempenho do Esquema pode levar o Módulo de Gestão Critográfica a gerar chaves privadas e pedir os respectivos certificados várias épocas antes de estes elementos serem utilizados. Este procedimento tenta minimizar eventuais impactos no desempenho do Esquema que o pedido e respectiva geração de certificado pela AC possam causar. Para além disso deve existir, no conjunto de cartões associado ao Esquema, pelo menos uma chave privada por cada nível hierárquico existente. Desse modo é necessário que a etiqueta associada a cada chave privada se consiga relacionar de forma inequivoca com o registo R_e ou R_h no qual vai ser utilizada.

É também essencial que exista a possibilidade de relacionar de modo eficaz a chave privada com o respectivo certificado. Esta premissa justifica a geração de etiquetas para as chaves com base em informação presente nos respectivos certificados. Como já referido, as extensões NH e PR são as necessárias e suficientes para gerar uma etiqueta identificativa de cada certificado utilizado em registos R_e e R_h. Embora seja necessário gerar as chaves privadas (e correspondentes etiquetas) antes dos respectivos certificados, é fácil para o Esquema calcular quais os valores que a AC utiliza para gerar as extensões referidas (como apresentado na secção 5.3.5).

A manutenção dos certificados já emitidos mas ainda não utilizados faz-se recorrendo a um repositório *LDAP*, gerido pelo Módulo de Gestão de Certificados. De

modo a aumentar o desempenho das operações de pesquisa, cada vez que um certificado é inserido também o são os seguintes elementos:

- *Distinguished Name*
- Data de início de validade
- Data de fim de validade
- Entidade Emissora
- Número de série
- Extensões NH e PR

5.4 Inserção e Validação de Dados

Nesta secção são descritos os processos de inserção de mensagens e de validação de registos no repositório.

5.4.1 Processamento de Mensagens

O processamento de uma mensagem M conjuntamente com uma etiqueta L pelo Módulo de Processamento segue os passos descritos de seguida. Para cada passo é indicada a correspondente operação no Módulo de Processamento e nos *interfaces* java representativos dos restantes Módulos que compõem o sistema (numeradas e ilustradas nas Figuras 5.4 e 5.5).[5]

Geração do registo R_m

1. A mensagem M e a etiqueta L são entregues ao Módulo de Processamento através de um ficheiro, como apresentado na secção 5.5.1 (operação 1).
2. É pedido ao Módulo de Gestão Criptográfica o sumário com cifra (campo Mac) dos campos Pr_m, L e M (operação 1.1).
3. É entregue o registo R_m ao Módulo de Gestão do Repositório (operação 1.2).

[5]Nas Figuras 5.4 e 5.5 *CryptographyInterface* representa o Módulo de Gestão Critpográfica, *RepositoryInterface* representa o Módulo de Gestão do Repositório e *TimestampInterface* representa o *interface* de acesso ao Serviço de Validação Cronológica (gerido pelo Módulo de Processamento).

4. É verificada a necessidade de finalizar a época. Esta decisão é baseada na data na qual o registo de topo actual R_e foi gerado e no número de registos R_m a ser incluídos na época. Em caso afirmativo é executado o procedimento de geração do registo de fim de época R_e (operação 2).

Geração do registo de fim de época R_e

1. É pedida ao Módulo de Gestão Criptográfica a chave secreta k utilizada na época a finalizar (operação 2.1).
2. São pedidos ao Módulo de Gestão do Repositório os registos R_m para a época que ainda não se encontra finalizada (operação 2.2).
3. É pedido ao Módulo de Gestão Criptográfica o sumário dos valores Mac presentes nos registos R_m obtidos no passo anterior (operação 2.3).
4. É pedido ao Módulo de Gestão do Repositório o registo R_e correspondente à última época finalizada (operação 2.4).
5. O cálculo do sub-conjunto de épocas no qual o registo R_e vai ser inserido pode gerar duas situações:
   (a) Se o registo obtido no passo 4 não for o último de um sub-conjunto de épocas é pedida a geração do campo Sig_e ao Módulo de Gestão Criptográfica com base na chave secreta k, no sumário obtido no passo 3 e no valor do sumário do campo Sig_e do registo obtido no passo 4 (operações 2.5.1 e 2.6).
   (b) Se o registo obtido no passo 4 for o último de um sub-conjunto de épocas é iniciado um novo sub-conjunto e é pedida a geração do campo Sig_e ao Módulo de Gestão Criptográfica com base na chave secreta k e no sumário obtido no passo 3 (operação 2.6).
6. É pedido ao Serviço de Validação Cronológica o selo temporal T aplicado ao campo Sig_e (operação 2.7).
7. É verificada a necessidade de finalizar o sub-conjunto de épocas actual (definindo-se assim o valor dos campos Sce_e e $Ultm_e$). Esta decisão é baseada no número de registos R_e que compõem o sub-conjunto (operação 2.8).
8. É entregue o registo R_e ao Módulo de Gestão do Repositório. Os valores dos campos Pr_i e Pr_f são calculados a partir dos registos obtidos no passo 2 (operação 2.9).
9. Caso o sub-conjunto seja finalizado é iniciado o processo de geração de registos R_h (operação 3).

Geração dos registos R_h

1. É definido um registo temporário R_{temp}, inicializado com os valores do último registo R_e gerado (operação 3.1).

2. Caso o registo R_{temp} seja o último de um subconjunto é iniciado o processo de geração de um novo registo para o nível hierárquico imediatamente superior ao do próprio. Caso contrário o processo de geração de registos R_h é finalizado.

3. É pedido ao Módulo de Gestão Criptográfica o sumário do campo Sig presente no registo R_{temp} (operação 3.2.1).

4. É pedido ao Módulo de Gestão do Repositório o último registo R_h do nível hierárquico imediatamente seguinte ao nível hierárquico Hr_{temp} (operação 3.2.2).

5. O cálculo do sub-conjunto de épocas no qual o registo R_h vai ser inserido pode gerar duas situações:

   (a) Se o registo obtido no passo 4 não for o último de um sub-conjunto é pedida a geração do campo Sig_h ao Módulo de Gestão Criptográfica com base no sumário obtido no passo 3 e no valor do sumário do campo Sig_h do registo obtido no passo 4 (operações 3.2.3.1 e 3.2.4).

   (b) Se o registo obtido no passo 4 for o último de um sub-conjunto (ou não existir ainda nenhum registo gerado para esse nível hierárquico) então é iniciado um novo sub-conjunto e é pedida a geração do campo Sig_h ao Módulo de Gestão Criptográfica com base no sumário obtido no passo 3 (operação 3.2.4).

6. É pedido ao Serviço de Validação Cronológica o selo temporal T aplicado ao campo Sig_h (operação 3.2.5).

7. É verificada a necessidade de finalizar o sub-conjunto actual (definindo-se assim o valor dos campos Sce_h e $Ultm_h$). Esta decisão é baseada no número de registos R_h que compõem o sub-conjunto (operação 3.2.6).

8. É entregue o registo R_h ao Módulo de Gestão do Repositório (operação 3.2.7).

9. Caso o sub-conjunto actual seja finalizado são repetidos os passos deste o passo 2 até ao passo 9. O registo R_{temp} passa a representar o registo R_h que foi inserido no passo 8.

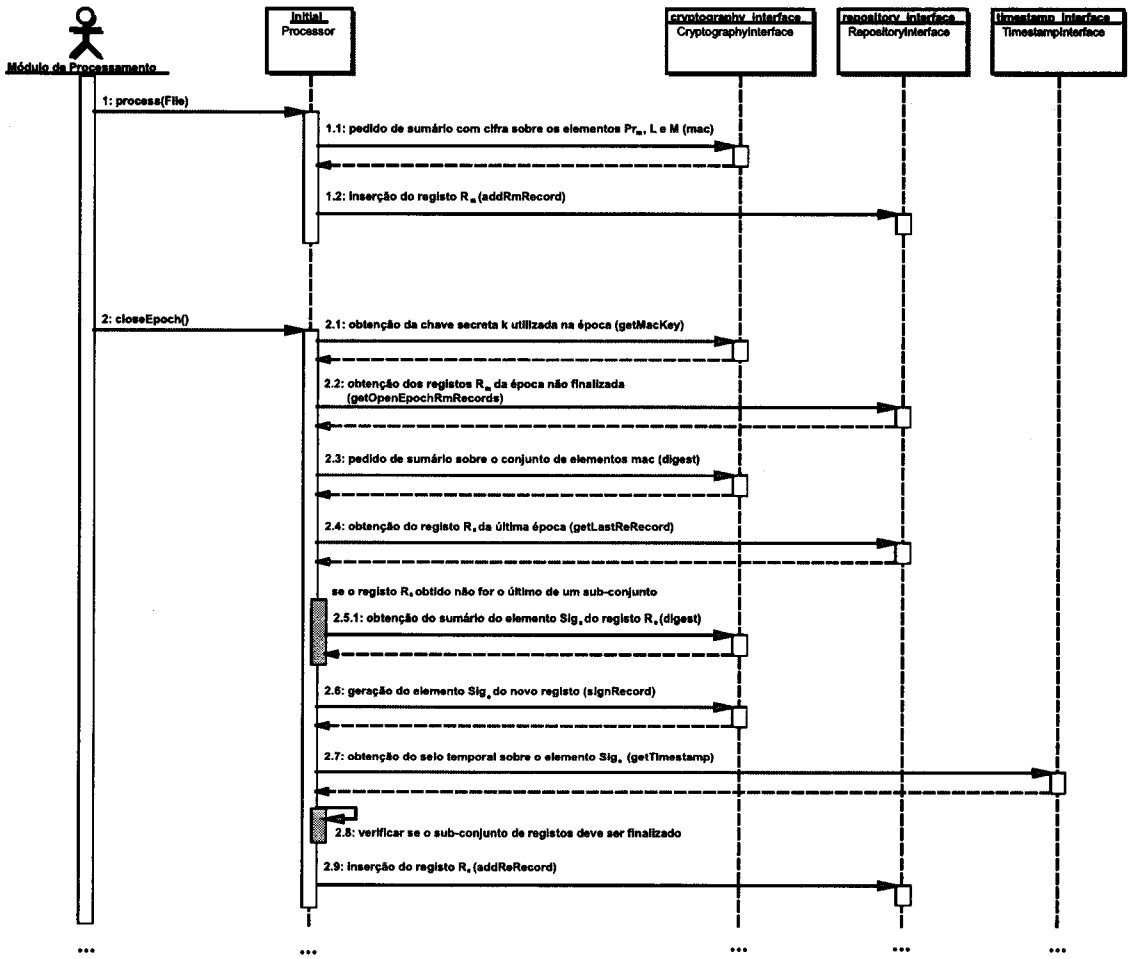

Figura 5.4: Diagrama de sequência do procedimento de inserção de um registo R_m (continua)

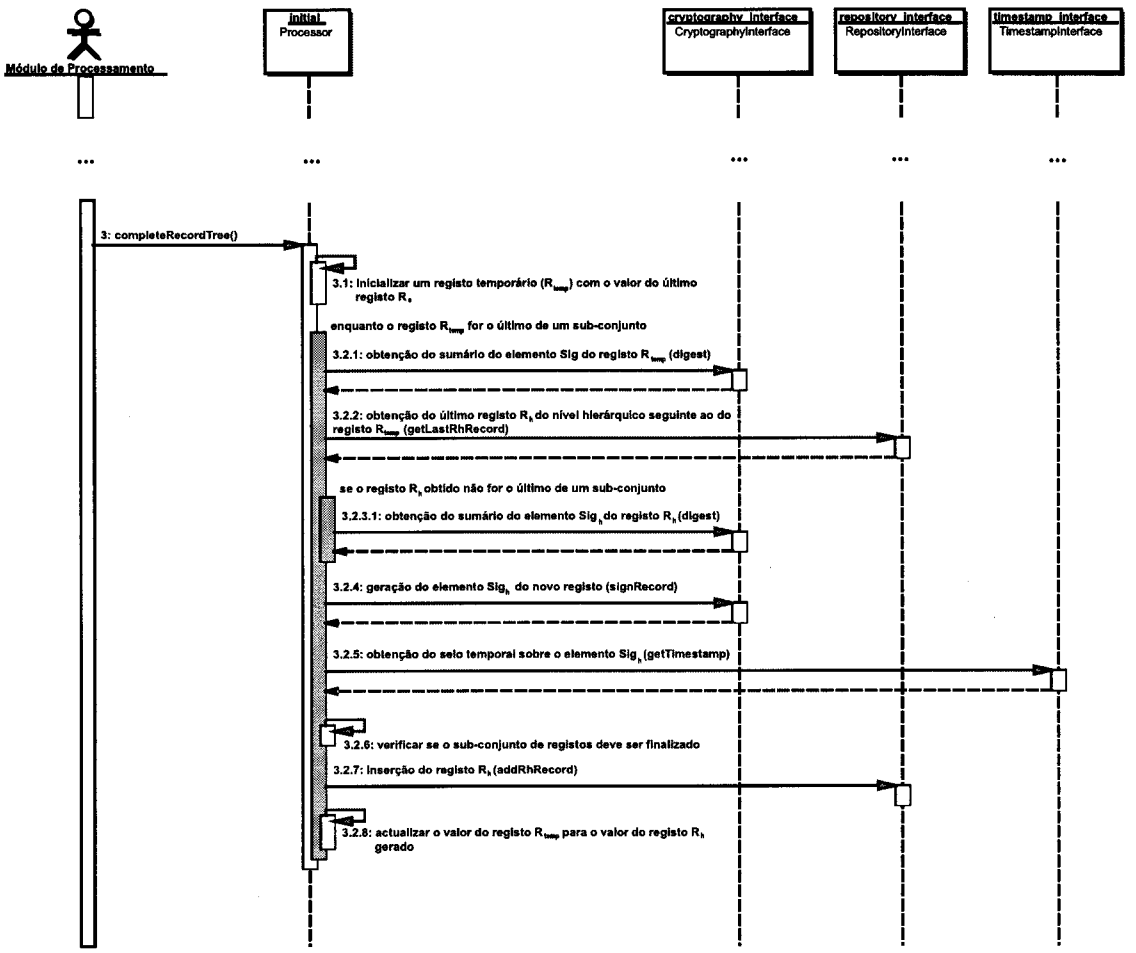

Figura 5.5: Diagrama de sequência do procedimento de inserção de um registo R_m (continuação)

5.4.2 Processo de Validação do Sistema de Auditabilidade

O processo de validação de registos segue os passos descritos, a alto nível, na secção 4.3.2. De seguida são apresentados os detalhes desse mesmo processo tendo por base a implementação proposta na secção 5.3.2. Para cada passo é indicada a correspondente operação no Módulo de Processamento e nos *interfaces* java representativos dos restantes Módulos que compõem o sistema (numeradas e ilustradas nas Figuras 5.6 e 5.7).

Processo de validação da integridade de um registo R_m:

1. É obtido, com base no valor do campo Pr_m, o registo R_m a validar (operação 1.1).
2. É obtido o conjunto de registos de topo através do Módulo de Gestão do Repositório (operação 1.2).
3. É obtido, com base no valor do campo Pr_m, o registo de fim de época R_e (operação 1.3).
4. O registo R_m é validado recorrendo à chave secreta k presente no registo R_e (operação 1.4).
5. É executado o processo de validação de um sub-conjunto de registos R_e (iniciado no registo de fim da época actual). Este termina com um erro ou com o último registo R_e do sub-conjunto (operação 1.5.1). Este último registo é de aqui por diante designado como R_{next}.
6. Através do campo *Ultm* é verificado se o registo R_{next} é o último de um sub-conjunto:
   (a) Em caso negativo deve ser terminado o processo de validação do registo R_m e verificada a existência do registo R_{next} no conjunto de registos de topo (operação 1.5.3).
   (b) No caso deste registo ser o último de um sub-conjunto deve ser verificada a sua existência no conjunto de registos de topo. Em caso positivo a validação do registo R_m termina correctamente. Em caso negativo devem ser executados os seguintes passos:
      i. É verificada a existência de um registo R_h pertencente ao nível hierárquico imediatamente superior ao do registo R_{next} que o referencie directamente. Esse registo é obtido com base no valor do campo Pr_h (que deverá corresponder ao valor do campo *Sce* do

registo R_{next}) e pelo nível hierárquico (operação 1.5.2.1). Caso este registo R_h não exista, o processo termina com um erro.

ii. É calculado o sumário do elemento Sig do registo R_{next}. Este é utilizado para a validação do registo R_h obtido (operação 1.5.2.2).

iii. É executado o processo de validação de um sub-conjunto de registos R_h (iniciado no registo R_h obtido no passo 6(b)i. Este termina com um erro ou com o último registo R_h do sub-conjunto (operação 1.5.2.3).

iv. Seguidamente são executados os passos deste o passo 6 até ao actual com o registo R_{next} a representar o registo R_h resultante do passo 6(b)iii.

Processo de validação de um sub-conjunto de registos R_e:

1. A partir dos campos Pr_i e Pr_f do registo de fim de época R_e são obtidos os registos R_m pertencentes à época (operação 2.1.1).

2. É obtido o sumário dos elementos Mac dos registos R_m pertencentes à época (operação 2.1.2).

3. É verificado, com base no valor dos campos Pr_e e Sce_e, se existe um registo R_e anterior ao actual e pertencente ao mesmo sub-conjunto de épocas. Em caso positivo este é obtido (operação 2.1.3.1) e é calculado o sumário do seu elemento Sig_e (operação 2.1.3.2).

4. Após a validação da assinatura Sig_e (com base nos valores obtidos nos passos anteriores) do registo R_e (operação 2.1.4) é necessário verificar se existe correspondência entre os valores das extensões do certificado $Cert_e$ e os valores dos campos do registo.

5. Através do campo $Ultm_e$ é verificado se o registo R_e é o último de um sub-conjunto:

   (a) Em caso positivo este processo termina.

   (b) Em caso negativo é necessário verificar se existe um registo R_e que referencie directamente o actual (operação 2.1.5). Este é identificado inequivocamente pelo do valor do campo Pr_e (e.g., se o valor deste elemento no registo R_e actual for n então o registo que o referencia directamente é identificado pelo valor $n + 1$ nesse mesmo campo). Se este registo não existir o processo termina. Caso contrário são repetidos os passos desde o passo 1 para o registo R_e obtido neste passo.

Processo de validação de um sub-conjunto de registos R_h:

1. É verificado, com base no valor dos campos Hr_h, Pr_h e Sce_h, se existe um registo R_h anterior ao actual e pertencente ao mesmo sub-conjunto de épocas. Em caso positivo este é obtido (operação 3.1.1.1) e é calculado o sumário do seu elemento Sig_h (operação 3.1.1.2).[6]

2. Após a validação da assinatura Sig_h (com base nos valores obtidos nos passos anteriores) do registo R_h (operação 3.1.2) é necessário verificar se existe correspondência entre os valores das extensões do certificado $Cert_h$ e os valores dos campos do registo.

3. Através do campo $Ultm_h$ é verificado se o registo R_h é o último de um sub-conjunto:

   (a) Em caso positivo este processo termina.

   (b) Em caso negativo é necessário verificar se existe um registo R_h que referencie directamente o actual (operação 3.1.3). Este é identificado inequivocamente pelo do valor do campo Pr_h (e.g., se o valor deste elemento no registo R_h actual for n então o registo que o referencia directamente é identificado pelo valor $n+1$ nesse mesmo campo). Se este registo não existir o processo termina. Caso contrário são executados os seguintes passos:

      i. É obtido o registo do nível hierárquico imediatamente inferior para o qual o registo R_h obtido no passo anterior possuí uma referência (operação 3.1.4.1 ou 3.1.5.1) e calculado o sumário do seu campo Sig (operação 3.1.4.2 ou 3.1.5.2). Este valor é necessário para a validação do campo Sig_h do registo obtido no passo 3b.

      ii. São repetidos os passos desde o passo 2 para o registo R_h obtido no passo 3b.

5.5 Interacção com Clientes

Nesta secção são apresentados os protocolos de comunicação e definidos os formatos das mensagens trocadas com os clientes do serviço. É também definido o formato das etiquetas correspondentes ao campo L do repositório.

[6]Este passo apenas é necessário para o primeiro registo R_h a validar no processo. Nos passos seguintes estes valores estão disponíveis uma vez que o registo obtido neste passo é precisamente o que acabou de ser validado.

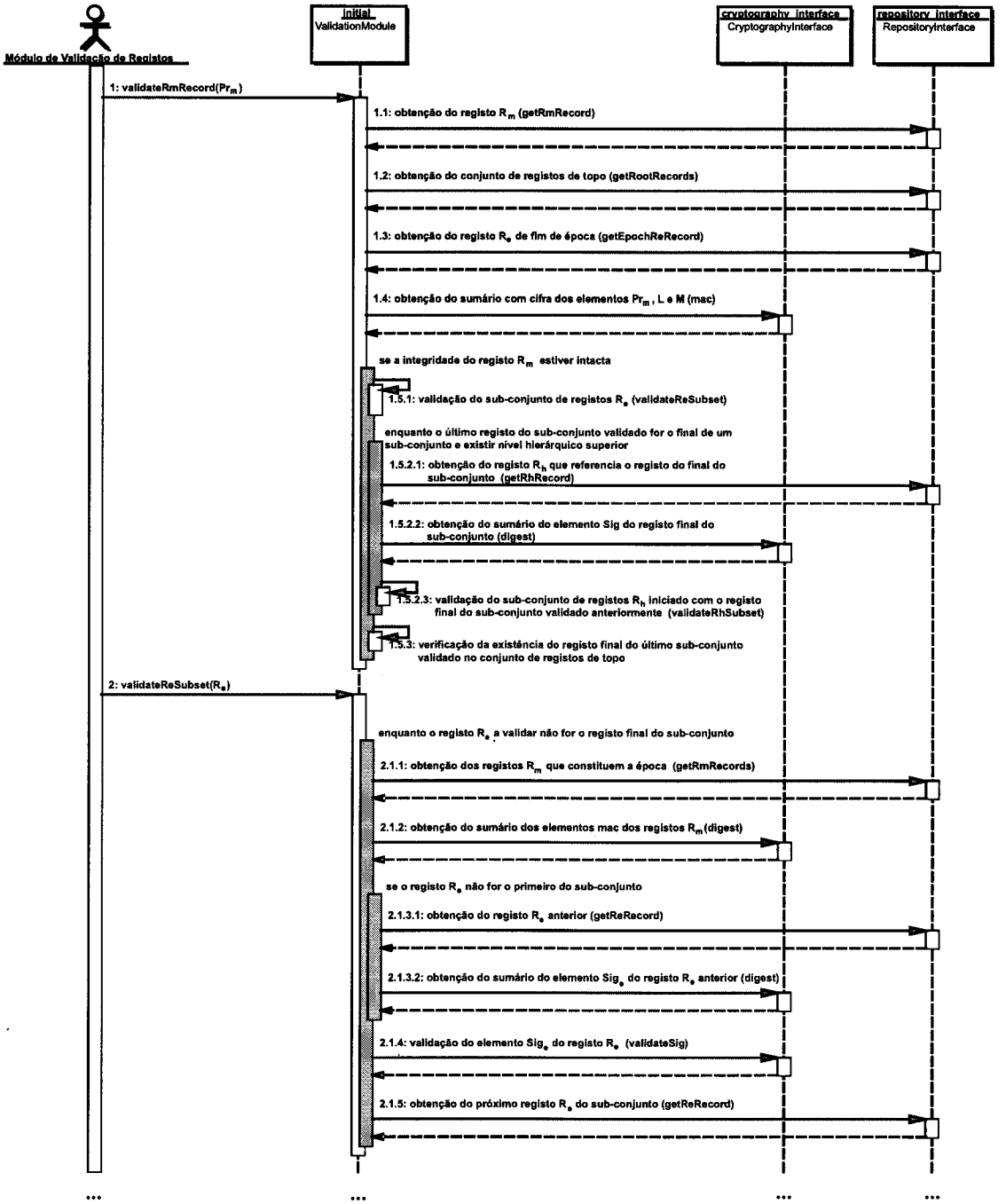

Figura 5.6: Diagrama de sequência do procedimento de validação de um registo R_m (continua)

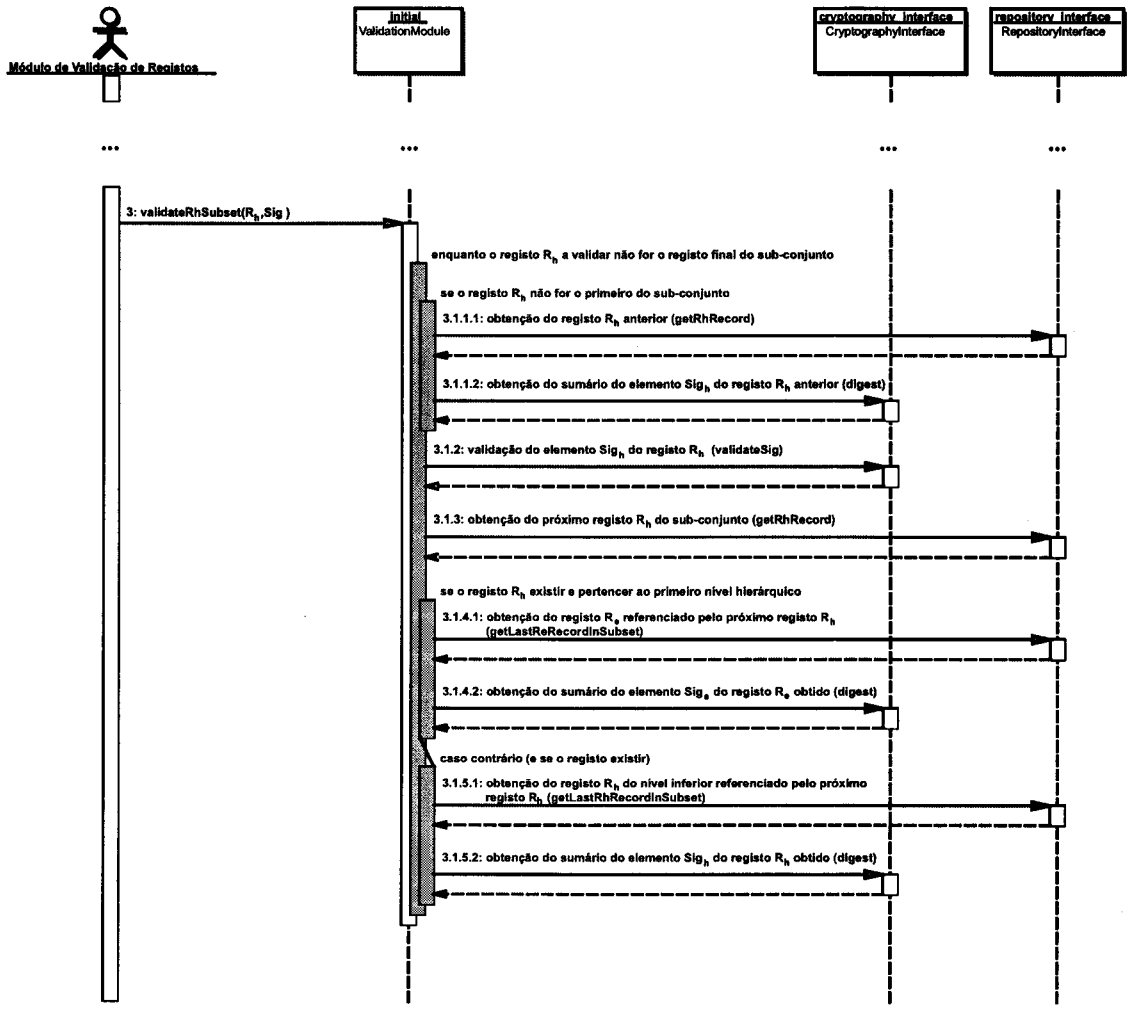

Figura 5.7: Diagrama de sequência do procedimento de validação de um registo R_m (continuação)

5.5.1 Protocolos de Comunicação

Cada cliente do serviço apenas comunica directamente com o Módulo de Recepção de Pedidos. A comunicação entre ambos pode ser realizada de dois modos distintos:

- Modo Síncrono - Recorrendo a um serviço de HTTP sobre TLS mutuamente autenticado; e
- Modo Assíncrono - Recorrendo a um serviço de SMTP que transporta uma mensagem S/MIME[60] assinada.

Ambos os modos garantem a autenticação de um cliente com base em certificados digitais. No modo síncrono a autenticação é realizada ao nível do protocolo TLS enquanto que no modo assíncrono a autenticação é realizada através da validação da assinatura digital presente na estrutura S/MIME. A autenticação de clientes é um requisito para a inserção de mensagens no Esquema. Quando a confidencialidade dos dados for essencial para o cliente este pode utilizar a capacidade de cifrar dados presente na estrutura S/MIME.[7]

Após a correcta autenticação de um cliente, o Módulo de Recepção de Pedidos necessita de verificar o tipo do pedido e processar o mesmo:

- Pedido de inserção de dados

  É gerado um ficheiro contendo os dados a inserir no Esquema num directório pré-definido. O directório representa o ponto de recepção de dados pelo Módulo de Processamento. O nome do ficheiro gerado referencia inequivocamente a etiqueta representativa dos dados (como apresentado na secção 5.5.2).

- Pedido de consulta a dados

  Os dados requisitados pelo cliente são obtidos a partir do Módulo de Gestão do Repositório. No caso da comunicação ser assíncrona estes têm de ser transmitidos ao cliente recorrendo a informação sobre o mesmo pré-definida no Esquema (e.g., modo de transmissão, credenciais necessárias para comunicação).

[7] O protocolo TLS já garante a confidencialidade dos dados transmitidos.

A comunicação entre módulos do sistema e serviços externos faz-se da seguinte forma:

- O Módulo de Gestão Criptográfica lê os pedidos da sua directoria de entrada e, após o processamento dos mesmos, move-os para uma directoria de ficheiros correctamente processados ou para uma directoria de erros (onde poderão mais tarde ser re-processados). O processamento dos pedidos recorre aos Módulos de Gestão Critpográfica, de Gestão de Certificados, de Gestão do Repositório e ao Serviço de Validação Cronológica. O acesso aos módulos referidos faz-se recorrendo a instâncias dos mesmos (propriedade da linguagem de programação Java). A obtenção de selos temporais faz-se recorrendo a um cliente que implementa os protocolos de HTTP e TCP[38] definidos em [19].

- Após a geração de um pedido de certificado por parte do Módulo de Gestão Criptográfica, este é encaminhado para o Módulo de Gestão de Certificados. Para isso é gerado um ficheiro contendo o próprio pedido na directoria de dados de entrada deste último módulo. O nome do ficheiro é aquele pelo qual o Módulo de Processamento vai referenciar o respectivo certificado quando o requisitar ao Módulo de Gestão de Certificados.

- A comunicação do Módulo de Gestão de Certificados com a AC é realizada por HTTPS mutuamente autenticado (como definido na secção 5.3.5). A recepção dos certificados por parte deste módulo é realizada recorrendo a um directório LDAP disponibilizado para o efeito pela AC. A pesquisa do certificado no repositório LDAP é realizada recorrendo ao nome do ficheiro do pedido associado ao mesmo.

5.5.2 Definição de Etiquetas

Cada etiqueta (ou identificador de transacção) é gerada pelo Módulo de Recepção de Pedidos com base em informação fornecida pelo cliente e pelo próprio Esquema. Assim, após a autenticação de um cliente, este módulo obtém (com base em informação previamente definida) o identificador único do mesmo perante o Esquema. O ficheiro a ser colocado na directoria de entrada do Módulo de Processamento será nomeado agregando o identificador único do cliente com o identificador de transacção.[8]

[8]Exemplificando, se o identificador único do cliente for "ABC"e o identificador de transacção for "123"o identificador final de transacção será "ABC.123".

A utilização de um prefixo distinto para cada cliente do Esquema permite fazer a distinção de registos R_m por cliente. Esta propriedade é essencial para o Esquema, pois só assim se garante que cada cliente possa consultar apenas os registos que o próprio inseriu ou sobre os quais tem permissões de leitura. Podem também ser definidos identificadores associados a grupos de utilizadores permitindo, por exemplo, que todos os intervenientes numa transacção possam consultar os registos associados à mesma.

5.5.3 Formato das Mensagens

Os pedidos de clientes e respostas do Esquema são realizados recorrendo a mensagens XML[71]. De seguida é apresentado o DTD (*Document Type Definition*) XML para cada uma das mensagens.

```
<!ELEMENT RequestEnvelope   (Header, RequestMsg+)>
<!ELEMENT ResponseEnvelope   (Header, ResponseMsg+)>

<!ELEMENT Header        (Date)>

<!ELEMENT Date          (#PCDATA)>

<!ELEMENT RequestMsg       (Id, (InsertData | GetData))>
<!ELEMENT ResponseMsg       (Id, (InsertResponse | GetResponse))>


<!ELEMENT Id            (#PCDATA)>

<!ELEMENT InsertData (Record)>

<!ELEMENT Record (Label, Data)>

<!ELEMENT Label         (#PCDATA)>
<!ELEMENT Data          (#PCDATA)>

<!ELEMENT GetData (Filter+)>

<!ELEMENT InsertResponse (Status)>

<!ELEMENT Status        (#PCDATA)>

<!ELEMENT GetResponse (Record*)>

<!ELEMENT Filter        (#PCDATA)>

<!ATTLIST Filter
   OpType         (intersect|subtract|union) #REQUIRED >
```

Uma funcionalidade que permite aos clientes maior controle sobre os resultados de uma pesquisa de mensagens ao Esquema é a possibilidade de serem incluídos filtros na mesma. O elemento *Filter* permite realizar uma sequência de operações

de união, intersecção e subtracção sobre um conjunto de registos R_m. Este conjunto é inicialmente representado por todos os registos R_m existentes no repositório, sendo actualizado após cada uma das operações requisitadas pelo cliente. Para que um registo faça parte do conjunto resultante de uma operação, o elemento L do mesmo tem de incluir o parâmetro associado à própria operação. De seguida é apresentado um exemplo ilustrativo da utilização do filtro de mensagens acima definido.

Suponha-se que uma determinada entidade é responsável pela inserção no Esquema de todos os dados relativos a transacções de um serviço do qual é responsável. Suponha-se também que esta entidade é reconhecida pelo Esquema através do identificador $ENTITY.ID.ABC$ e que os identificadores de mensagens inseridas são do tipo:

```
<nome do serviço>.SN+
<identificador interno da transacção>.TID+
<nome da entidade emissora da mensagem>.S+
<nome da entidade receptora da mensagem>.R+
<tipo da mensagem>.T+
[CIPHER]
```

Veja-se, como exemplo, o processamento do seguinte pedido:

```
<RequestEnvelope>
  <Header>
     <Date>01-09-2004 14:45:32</Date>
  </Header>
  <RequestMsg>
     <Id>ABC1234</Id>
     <GetData>
        <Filter OpType="intersect">USER.123.S</Filter>
        <Filter OpType="intersect">XPTO.SERVICE.SN</Filter>
        <Filter OpType="union">PAY.FINAL.MSG</Filter>
        <Filter OpType="subtract">CIPHER</Filter>
     </GetData>
  </RequestMsg>
</RequestEnvelope>
```

Este pedido pretende obter todas as mensagens não cifradas que o utilizador $USER.123$ enviou através do serviço $XPTO.SERVICE$ juntamente com todas as mensagens não cifradas relativas a pagamentos efectuados (i.e., mensagens do tipo $PAY.FINAL.MSG$). Para além destes filtros está também implícita a filtragem de todas as mensagens com base no identificador da entidade (nesta caso $ENTITY.ID.ABC$). O filtro apresentado pode ser utilizado, por exemplo, para calcular a percentagem do total de valores transaccionados originada no utilizador $USER.123$. Um elemento L pertencente ao conjunto de registos resultantes da

pesquisa acima apresentada poderá ser, por exemplo, o seguinte:

$$ENTITY.ID.ABC{+}XPTO.SERVICE.SN{+}TRANS.678.TID{+}USER.123.S{+}$$
$$USER.XYZ.R + MSG.TYPE.AB12.T+$$

A notação utilizada para obter o conjunto de registos de mensagens pretendido foi inspirada na notação definida em [26].

A possibilidade de filtrar os resultados de uma pesquisa em conjunto com a possibilidade de definição das etiquetas por parte do cliente tornam o sistema adequado para utilização em ambientes transaccionais (i.e., em ambientes em que um conjunto de troca de mensagens represente uma transacção). Estas propriedades levam a que se possam definir etiquetas representando esquemas hierárquicos de transacções que mais tarde serão facilmente reconstituídas numa operação de pesquisa.

5.6 Desempenho do Sistema

Nesta secção são apresentadas medidas de desempenho da implementação do Esquema. São apresentados dados que permitem retirar conclusões relativas ao desempenho das operações de validação da integridade de um registo e de inserção de uma nova mensagem M no repositório.

As tabelas 5.1 e 5.2 apresentam dados relativos ao estado do repositório quando este mantém um determinado número de registos. São indicados o número de níveis hierárquicos existentes e o número de elementos que compõem a cadeia de registos directamente ligados para um conjunto pré-definido de registos R_m. O número médio de registos R_m por época é 120 em ambas as tabelas. Estas distinguem-se pelo número médio de elementos por sub-conjunto de registos R_e e R_h.

Dados do repositório			# de registos validados por cadeia			
$\#R_m$	$\#R_e + \#R_h$	# níveis	$R_{m<1>}$	$R_{m<5.000>}$	$R_{m<50.000>}$	$R_{m<500.000>}$
$\approx$ 10.000	$\approx$ 90	3	13	7	-	-
$\approx$ 100.000	$\approx$ 970	5	21	15	15	-
$\approx$ 1.000.000	$\approx$ 10.400	6	27	21	21	18

Tabela 5.1: Número de elementos que compõem a cadeia de validação de $R_{m<n>}$ quando o número médio de registos por sub-conjunto é 5

Dados do repositório			# de registos validados por cadeia			
$\#R_m$	$\#R_e + \#R_h$	# níveis	$R_{m_{<1>}}$	$R_{m_{<5.000>}}$	$R_{m_{<50.000>}}$	$R_{m_{<500.000>}}$
$\approx$ 10.000	$\approx$ 80	2	17	9	-	-
$\approx$ 100.000	$\approx$ 1100	4	28	20	17	-
$\approx$ 1.000.000	$\approx$ 8.300	5	37	29	26	20

Tabela 5.2: Número de elementos que compõem a cadeia de validação de $R_{m_{<n>}}$ quando o número médio de registos por sub-conjunto é 9

Como se pode constatar pelos dados apresentados nas tabelas 5.1 e 5.2, à medida que o número de registos presentes no repositório aumenta, o número de registos validados por operação de verificação da integridade de um registo R_m também aumenta, embora de forma muito mais lenta. Mais, comprova-se que a relação entre o crescimento destes dois valores corresponde ao valor apresentado na secção 4.6.

À medida que são gerados sub-conjuntos com um maior número de registos, o número total de níveis hierárquicos existentes no repositório tende a diminuir, aumentando o desempenho da operação de inserção de mensagens. Esta característica leva, no entanto, à geração de cadeias de registos directamente ligados com um número cada vez maior de elementos, diminuindo o desempenho da operação de validação de cada registo R_m.

As propriedades acima referidas possibilitam que o número médio de registos por sub-conjunto possa ser facilmente adaptado à relação expectável entre o número de operações de escrita (i.e., operações de inserção de dados) e o número de operações de leitura (i.e., operações de validação de registos ou pedidos de leitura de dados por parte de clientes) para cada utilização específica do Esquema.

Importa referir que o número de registos validados apresentado não inclui os registos R_m pertencentes a cada uma das épocas validadas. Pode-se assumir que este número se mantém constante,[9] e logo não influencia a relação de crescimento do número de registos acima apresentada.

[9] É trivial definir na implementação do Esquema um limite para o número de registos R_m agregados por época.

Capítulo 6

Caso de Estudo

Neste capítulo é feita a apresentação de um sistema de facturação electrónica onde a implementação do Esquema apresentada no capítulo 5 é aplicada. Por um lado o Esquema serve de base de auditabilidade a um protocolo que fornece garantias fortes de não-repúdio entre os intervenientes numa transacção. Por outro garante a auditabilidade das operações internas ao sistema. Este sistema encontra-se em fase de exploração comercial pela Novis Telecom, SA.

6.1 Requisitos Legais

Tal como apresentado em legislação específica[4, 6, 2], um sistema de facturação electrónica necessita de cumprir uma série de requisitos técnicos:

- A conservação das facturas e documentos equivalentes pela ordem cronológica de emissão e recepção;
- A manutenção da integridade, disponibilidade e autenticidade do conteúdo original das facturas e documentos equivalentes;
- O não-repúdio das mensagens; e
- A não duplicação das facturas ou documentos equivalentes.

A utilização do sistema de auditabilidade proposto nesta dissertação (como apresentado nas próximas secções) garante que o sistema de facturação cumpre todos os requisitos enumerados.

6.2 Apresentação do Sistema

Os objectivos do projecto (daqui em diante designado por "Serviço") são os seguintes:

- Permitir a troca electrónica de facturas entre parceiros comerciais com formatos distintos de facturação;
- Proceder à salvaguarda de facturas; e
- Criar e manter um repositório de mapas recapitulativos.

O Serviço encontra-se implementado em módulos. Cada módulo é responsável por uma parte do processamento de uma factura electrónica. Este modo de implementação torna o sistema flexível para:

- Futuras alterações no conjunto de módulos utilizado que obriguem à re-implementação do fluxo de mensagens pelo Serviço;
- Processamento faseado de facturas; e
- Gestão das garantias de segurança necessárias por módulo.

Os módulos que compõem o Serviço são os seguintes:

- **Módulo de Certificação Digital**

  Na base das garantias de segurança que a legislação requer para as facturas electrónicas (integridade, autenticidade e não-repúdio) estão as assinaturas digitais. Assim, um dos módulos do Serviço é a infra-estrutura de certificação digital, responsável pela emissão, renovação e revogação de certificados para os intervenientes (em particular, compradores e vendedores).

  Além das garantias referidas acima, existe ainda uma componente que garante a confidencialidade dos dados a trocar entre as diversas partes no âmbito do Serviço. Isto implica a cifragem desses dados, recorrendo aos certificados digitais das entidades a quem se destinam.

A emissão de certificados está sujeita a procedimentos de identificação das entidades a certificar, os quais serão definidos de acordo com as boas práticas estabelecidas no mercado[27] e pelas normas definidas pela legislação relevante[3, 5, 1, 7].

- **Módulo de Validação Cronológica**

O módulo de validação cronológica permite dar resposta a um dos requisitos da legislação para a factura electrónica, que se refere à validação cronológica das mensagens emitidas como facturas. Este serviço permite a obtenção de informação horária de alta precisão, obtida a partir da fonte da Hora Legal de Portugal (i.e., a partir do OAL) de forma segura e inalterável, recorrendo a assinaturas digitais.

O módulo de validação cronológica disponibilizado é compatível com a norma do IETF definida em [19] e obtém a sincronização com o OAL através de uma infra-estrutura segura que garante não só a precisão como a integridade da informação horária.

- **Módulo de Directório**

Um dos módulos fundamentais do Serviço é um directório, baseado na norma LDAP, que dá acesso a elementos como certificados digitais e listas de certificados revogados.

Este directório é usado essencialmente para obter informações sobre a validade de certificados digitais, por forma a possibilitar a cifra de mensagens ou a verificação de assinaturas digitais. No entanto, também pode ser utilizado para guardar e disponibilizar informações sobre as partes envolvidas, como dados de facturação de compradores e vendedores.

- **Módulo de Repositório**

Uma das funcionalidades a que a legislação para a factura electrónica obriga é a da manutenção da disponibilidade dos documentos, correspondentes a facturas electrónicas, enviados e recebidos, por ordem cronológica

de emissão e recepção. Neste sentido, o Serviço conta com um repositório que permite o armazenamento daqueles documentos pelo período obrigatório por lei. Este repositório oferece ainda as seguintes garantias adicionais:

- Salvaguarda da informação, com políticas de *backup* apropriadas que garantem a disponibilidade total e permanente das facturas, mesmo em caso de falha de um ou mais componentes do Serviço.

- Confidencialidade da informação, de modo a que os dados críticos das facturas sejam acessíveis apenas às partes interessadas (emissor, receptor e administração fiscal). Em particular, as pessoas afectas à operação do Serviço não podem ter acesso a estes dados.

A informação contida neste repositório serve como base do serviço de geração e disponibilização de mapas recapitulativos, conforme preconizado pela legislação. A legislação obriga também à existência de arquivo de facturas tanto para quem as emite como para quem as recebe. O repositório tem a possibilidade de armazenar as facturas, distinguindo-as e controlando o seu acesso por entidade e natureza (i.e., uma entidade pode ter dois repositórios associados, um para a qualidade de vendedor e outra para a qualidade de comprador).

- **Módulo de Troca de Mensagens**

Este módulo assegura a recepção das mensagens contendo as facturas do emissor e fá-las chegar ao receptor. Inclui um sistema que assegura a notificação de eventos relevantes para cada uma das partes. Os tipos de notificações a disponibilizar são os seguintes:

- Armazenamento, após recepção pelo Serviço, de uma factura (notificação ao emissor e ao receptor).
- Consulta da factura pelo receptor (notificação ao emissor).

Estas notificações, todas elas assinadas digitalmente pelo Serviço, oferecem a este garantias fortes de segurança, dado que às garantias de confidencialidade, integridade e autenticação, dadas pela utilização de criptografia e assinaturas digitais, são adicionadas garantias de não-repúdio:

- De envio, através da notificação de armazenamento ao emissor; e

– De recepção, através da notificação de consulta, também ao emissor.

Ambas as garantias protegem o emissor, uma vez que, no primeiro caso, este fica com uma prova do envio da factura, e no segundo caso, uma prova da recepção. Em relação à protecção do receptor pelo não-repúdio da autoria da factura pelo emissor, está já estava garantida pela assinatura digital aposta por este.

Faltam, no entanto, fornecer outras garantias essenciais ao serviço. O protocolo de geração das notificações acima definidas implica trocas de mensagens entre o emissor e o Serviço e entre o Serviço e o receptor. É essencial garantir que este protocolo é justo (como definido na secção 2.4.4) e que as trocas de mensagens realizadas durante o mesmo são auditáveis. De forma similar é importante que as operações realizadas sobre uma factura pelos módulos do Serviço sejam auditáveis.

De modo a fornecer garantias fortes de segurança entre parceiros comerciais e o sistema, este módulo implementa o protocolo de não-repúdio apresentado na secção 3.1.2. Este protocolo tem como base de auditabilidade o Esquema implementado no capítulo 5. A integração do protocolo de não-repúdio com o esquema de auditabilidade segue as indicações apresentadas na secção 4.8. A implementação do Esquema é também utilizada para registar as acções que cada módulo do Serviço toma sobre o processamento de uma factura.

6.3 Arquitectura

A Figura 6.1 ilustra o modelo de comunicação e os módulos do Serviço. Para simplificar a exposição, o emissor de mensagens correspondentes a facturas passa a ser identificado como sendo o vendedor, e o receptor dessas mensagens passa a ser identificado como comprador.

O Serviço posiciona-se como um intermediário entre o vendedor e o comprador. Qualquer que seja a operação que cada um destes pretenda fazer, terá sempre de recorrer ao Serviço e nunca directamente à outra parte. Internamente, os vários módulos interagem entre si para providenciar as funcionalidades relativas a cada operação solicitada pelos vendedores e compradores. De seguida descrevem-se

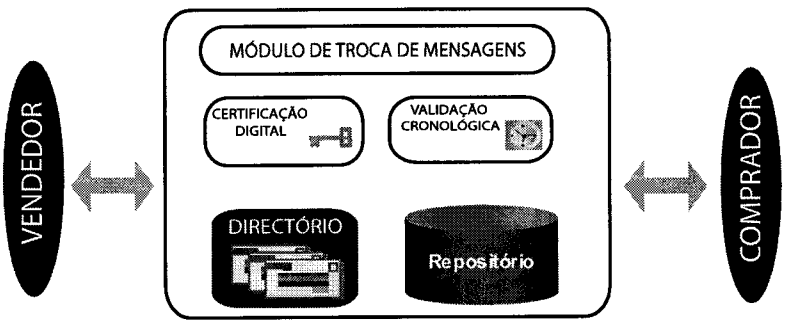

Figura 6.1: Módulos do Serviço

em pormenor as interacções externas (com os vendedores e compradores). As interacções internas (entre módulos) do Serviço não são apresentadas uma vez que não são relevantes para a apresentação do protocolo de não-repúdio e do esquema de auditabilidade implementados.

A entrega de uma factura do vendedor ao Serviço requer os seguintes passos:

1. Geração da factura;
2. Assinatura digital da factura utilizando certificados digitais;
3. Aplicação de um selo temporal sobre a assinatura da factura;
4. Opcionalmente pode proceder-se à cifra dos dados a enviar (utilizando o certificado digital do comprador); e
5. Envio da factura ao Serviço através da execução do protocolo de não-repúdio (iniciado pelo vendedor).

A leitura de uma factura pelo comprador requer os seguintes passos:

1. Envio de mensagem de pedido de factura do comprador para o Serviço;
2. Obtenção da factura através da execução do protocolo de não-repúdio (iniciado pelo Serviço);
3. Verificação da integridade e autencicidade do selo temporal; e
4. Verificação da assinatura digital presente no documento. Esta operação implica a utilização do selo temporal associado para a validação de assinaturas realizadas com a chave privada correspondente a um certificado que já não é válido no momento da verificação.

6.4 Integração com o Protocolo de Não-Repúdio

A escolha do tipo de protocolo de não-repúdio está regulada por várias condicionantes. Em primeiro lugar é importante que exista uma EC, quer do Serviço quer dos compradores e vendedores. Por outro lado pode ser importante para o emissor e para o receptor não revelarem a ninguém, nem mesmo à EC utilizada no protocolo, qualquer informação sobre o conteúdo das mensagens trocadas. Uma outra característica desejável no protocolo é a possibilidade de qualquer um dos intervenientes o abortar ao fim de um tempo limite pré-estabelecido. Por fim há que ter em conta que o tamanho das mensagens trocadas possa ser arbitrariamente grande.

Os condicionalismos apresentados levaram à implementação de um protocolo de não-repúdio com participação parcial da EC. Mais concretamente, foi decidida a implementação do protocolo apresentado na secção 3.1.2. Neste protocolo a primeira mensagem trocada (a única onde é enviada a factura) não necessita de ser transmitida através da EC, sendo enviada directamente para o Serviço. Existe também um tempo limite (conhecido e acordado por ambos os intervenientes) para a finalização do protocolo. Por outro lado é possível garantir a confidencialidade da mensagem em relação à EC (e por conseguinte a qualquer outra entidade não participante no protocolo), através de uma alteração no protocolo original:

$$A \rightarrow EC : f_{Sub}, B, L, T, c_{C_B}(k), sub\_k$$

A chave k passa a ser entregue à EC cifrada com a chave pública do receptor. Deste modo apenas este último a poderá ler. Este procedimento aplica-se tanto à recepção de facturas pelo Serviço como à leitura de facturas pelo comprador.

A utilização do protocolo acima apresentado permite evitar a utilização de um canal de comunicação seguro entre os intervenientes, optimizando por consequência o desempenho deste. Como já referido, tanto a mensagem como a chave utilizada para a cifrar são enviadas cifradas. Para além disso, a assinatura de todas as mensagens através da utilização de certificados digitais garante integridade e autenticação das mesmas. A própria identidade dos intervenientes, revelada pela utilização de certificados digitais, pode ser salvaguardada de entidades externas ao serviço. Para tal basta que cada comprador e vendedor estabeleça um acordo com o Serviço no qual ficam definidos quais os certificados a utilizar por cada um. Deste modo os certificados não necessitarão de conter qualquer referência à verdadeira identidade destas entidades.

6.5 Aplicação do Esquema de Auditabilidade

O Esquema é aplicado não só no protocolo de não-repúdio utilizado mas também nas interacções entre módulos do Serviço (como ilustrado na Figura 6.2). A aplicação do Esquema entre módulos não consegue por si só garantir, na presença de um Serviço não-idóneo, a total auditabilidade das transacções realizadas internamente por este último. O próprio Serviço pode decidir se, como e quando regista uma determinada transacção. No entanto para esta situação ser viável (i.e., indetectável para um auditor independente) é necessário que o Serviço, compradores ou vendedores e a EC utilizada no protocolo de não-repúdio sejam coniventes entre si. Isto porque qualquer transacção entre estas entidades também fica registada nos protocolos de não-repúdio de envio e de leitura de facturas.

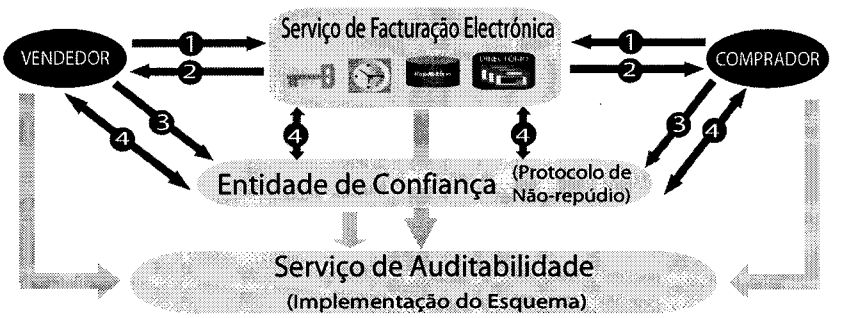

Figura 6.2: Interacções entre intervenientes no protocolo de não-repúdio (numeradas entre 1 e 4) e inserções de mensagens no Esquema

A auditabilidade do protocolo de não-repúdio depende da vontade dos intervenientes para registarem cada transacção. A execução deste protocolo com sucesso não depende, no entanto, do correcto registo de uma transacção. Cada interveniente no Serviço pode, por exemplo, decidir registar todas as transacções de um determinado dia apenas num momento futuro pré-definido. Claro que esta situação pode não ser exequível para certas transacções para as quais é importante que a data de registo esteja de acordo com a data de geração da factura.

É essencial para o Serviço a possibilidade de reconstrução das operações realizadas sobre uma factura desde o momento em que foi entregue ao Serviço por um vendedor até ser lida pelo respectivo comprador. Deste modo é necessário que se gere uma ligação inequívoca entre as transacções envolvidas. No Serviço isto foi conseguido recorrendo à possibilidade que o Esquema tem de etiquetar mensagens

(e logo, transacções). Assim tornou-se extremamente fácil produzir etiquetas que distinguissem, mas ao mesmo tempo relacionassem, os protocolos de não-repúdio executados, quer na entrega quer na leitura de uma factura, com as operações de processamento desta realizadas entre módulos do sistema.

Capítulo 7

Conclusões

Este capítulo faz um balanço sobre o trabalho apresentado ao longo da dissertação. Mais concretamente, na secção 7.1 são enumerados os objectivos alcançados, na secção 7.2 é realizada uma análise comparativa com trabalhos relacionados, na secção 7.3 são apresentadas limitações conhecidas ao sistema proposto e na secção 7.4 são sugeridos possíveis futuros melhoramentos ao trabalho apresentado.

7.1 Objectivos Alcançados

O objectivo deste trabalho era conceber e implementar um sistema de troca segura de mensagens que, para além de fornecer as garantias de segurança básicas presentes em sistemas de trocas de mensagens que envolvam gestão de informação crítica (integridade, auditabilidade, confidencialidade e não-repúdio de origem), conseguisse também fornecer um conjunto de garantias de segurança avançadas, necessárias em sistemas que operem em ambientes de alta segurança (e.g., sistema militar, comércio electrónico seguro).

Garantias como não-repúdio de submissão ou de recepção são fornecidas pela utilização de um protocolo de não-repúdio justo. Este tipo de protocolos encontra-se amplamente estudados e a sua segurança perfeitamente demonstrada. Deste modo procurou-se conceber um sistema que fornecesse garantias fortes de auditabilidade de dados. A definição do sistema de auditabilidade apresentado no capítulo 4 representa a principal contribuição desta dissertação para a concepção do sistema de troca segura de mensagens proposto no capítulo 1.

O sistema de auditabilidade proposto fornece garantias fortes de integridade e ordenação temporal sobre os dados. Para além disso não é possível para um atacante remover dados sem que tal acto não seja detectado pelo sistema de validação

do Esquema, nem inserir dados sem a devida autorização. Por outro lado o processo de validação de registos pode ser executado por qualquer entidade recorrendo apenas à informação disponível no próprio repositório. A disponibilização desta informação no próprio repositório não limita de forma alguma qualquer garantia de segurança fornecida por este. As propriedades enumeradas fornecem ao Esquema garantias fortes de auditabilidade dos dados inseridos.

O repositório confiável definido assenta na utilização de três tipos de registos. Os registos R_m são responsáveis pela manutenção dos dados de entrada. Cada registo R_e agrupa um conjunto ordenado de registos R_m, formando uma época. Os registos R_e encontram-se estruturados por sub-conjuntos. Dentro de cada sub-conjunto cada registo R_e contem uma referência para o registo R_e presente na posição temporal imediatamente anterior. Os registos R_h são definidos como nós de controle, sendo utilizados para optimizar o desempenho do procedimento de validação do repositório.

Um ponto essencial nos objectivos do trabalho proposto era conseguir uma implementação do sistema utilizando algoritmos, estruturas e protocolos cuja segurança e fiabilidade se encontre plenamente demonstrada. A implementação apresentada no capítulo 5 obedece a esse conjunto de requisitos. A estruturação do sistema num conjunto de módulos torna o mesmo bastante flexível a alterações. Cada módulo é responsável por uma tarefa específica e é facilmente substituível.

A manutenção da integridade do repositório baseia-se na manutenção da privacidade quer da chave simétrica utilizada para gerar elementos *Mac* dos registos R_m (enquanto a época subjacente não se encontrar terminada) quer das chaves privadas utilizadas para gerar elementos *Sig* dos registos R_e e R_h. Uma vez que a geração e utilização destas chaves é sempre realizada no interior de *hardware* criptográfico a sua privacidade encontra-se assegurada. Por outro lado, e sempre que avanços criptográficos o aconselharem, torna-se fácil adaptar o Módulo de Gestão Criptográfica de modo a minimizar a possibilidade de ataques bem sucedidos sobre o mesmo.

7.2 Discussão e Comparação com Trabalhos Relacionados

Como apresentado no capítulo 3, não são conhecidos trabalhos cujos objectivos sejam similares aos do sistema proposto. As implementações de protocolos de não-repúdio existentes não referem soluções para a manutenção a longo prazo da

auditabilidade das transacções (i.e., das trocas de mensagens) realizadas. Este tipo de protocolos fornecem, no entanto, a base ideal para a implementação do tipo de garantias de segurança acima referidas. Esta conclusão deriva do facto destes disponibilizarem um conjunto de garantias de segurança que mais nenhum tipo de protocolo pode oferecer.

Os protocolos de não-repúdio apresentados são justos. Isto implica que nenhum dos intervenientes no protocolo ganha qualquer tipo de vantagem (e.g., provas) que lhe permita defender a sua posição perante um juiz ao mesmo tempo que impede os restantes intervenientes de defenderem a sua própria posição. Os mesmos protocolos fornecem garantias de não-repúdio de origem e recepção, bem como a garantia de não-repúdio de submissão nos casos onde esta pode ser fornecida (i.e., quando está envolvida uma EC). Para além destas garantias também se encontram garantidas a integridade e autenticação das mensagens através da utilização de assinaturas e certificados digitais. A utilização de criptografia assimétrica torna por sua vez bastante fácil a introdução da garantia de confidencialidade em todas as mensagens trocadas durante a execução de um protocolo de não-repúdio.

A manutenção da auditabilidade das mensagens trocadas durante a execução de um protocolo de não-repúdio passa pela salvaguarda destas de modo a que seja possível reconstituir a transacção subjacente. Esta manutenção implica a existência de um repositório que deve ser gerido de modo seguro, confiável e transparente.

O trabalho apresentado em [58] apresenta argumentos válidos para a separação das operações de autenticação de utilizadores, gestão e validação do repositório. A implementação do Esquema apresentada baseou-se nestes argumentos para a separação das operações referidas em diferentes módulos do sistema. Embora nesta implementação os módulos sejam geridos por um mesmo sistema, é extremamente fácil torná-los independentes e geridos por diferentes entidades.

A estrutura do Esquema foi definida tendo por base conceitos utilizados nas soluções para auditabilidade de dados apresentadas em [35] e em [67]. Mais concretamente, foram aplicados os principios de estruturação dos dados em árvore. Este conceito deriva da definição de Árvore de Merkle e consiste em gerar uma hierarquia de registos de dados onde um registo pertencendo a um determinado nível hierárquico contém referências inequívocas para um ou mais registos pertencentes ao nível hierárquico imediatamente inferior. Deste modo a corrupção de um registo sem detecção de tal acto por parte de um sistema de validação do

repositório implica sempre a corrupção sem detecção de um conjunto de registos que se referenciam em cadeia.

7.3 Limitações

Embora o Esquema e a implementação do mesmo garantam o cumprimento integral dos objectivos propostos para este trabalho, existem limitações que no futuro deverão ser eliminadas de modo a tornar o sistema mais preciso e robusto.

A limitação mais notória prende-se com a falta de informação sobre a data exacta na qual cada registo R_m foi inserido no repositório. Este foi um compromisso assumido na definição do Esquema, justificado pela optimização do desempenho e simplificação do mesmo. Para além disso, e embora a data não seja conhecida, esta encontra-se delimitada pelas datas presentes nos registos R_e da época anterior e da actual. Esta limitação impede o Esquema de ser utilizado em ambientes onde a precisão horária seja essencial, como por exemplo em operações relacionadas com transacções de títulos cotados em bolsa ou em concursos que definam um tempo limite para participação nos mesmos.

O sistema proposto não faz referência a mecanismos de replicação e reposição de dados. Após a detecção de uma violação da integridade do repositório (por parte do Módulo de Validação de Registos) é necessário recuperar a consistência do mesmo. Esta operação deve ser executada através do recurso a cópias dos dados previamente armazenadas. As cópias devem por sua vez ser armazenadas periodicamente e de forma segura e confiável.

Devem ser definidos mecanismos de replicação dos dados em suportes físicos distintos. Estes mecanismos são responsáveis por manter a coerência entre as diferentes instâncias do repositório e por minimizar os períodos temporais durante os quais este se encontra inacessível. As instâncias replicadas serão utilizadas como alternativa ao suporte principal no caso de falha deste último (e.g., devido a operações de manutenção, ataques bem sucedidos).

7.4 Trabalho Futuro

O Esquema apresentado pode futuramente ser optimizado de modo quer a minimizar ou eliminar por completo as limitações apresentadas na secção 7.3, quer a desenvolver novas funcionalidades que possam contribuir positivamente para a optimização do sistema. Estas possíveis alterações são de seguida enumeradas.

- **Precisão Temporal:** Encontrar uma forma de colocar selos temporais nos registos R_m sem que tal operação altere de forma notória o desempenho do Esquema. Deste modo poder-se-á utilizar o Esquema para operações que necessitem de dispor da data exacta na qual os dados foram inseridos no repositório.

- **Estrutura do Repositório:** Pesquisar a possibilidade de redefinição dos elementos das tabelas de registos de modo a optimizar o processo de validação de registos. Deve também ser testada a fiabilidade e o desempenho de outras estruturas (e.g, LDAP ou bases de dados não-relacionais) para utilização como base do repositório dos dados.

- **Gestão Criptográfica:** Definir e validar um modo de utilizar remotamente o *hardware* criptográfico, de modo a agilizar o processo de inicialização do Módulo de Gestão Criptográfica.

- **Transacções:** Redefinir o elemento L de modo a aceitar de forma mais prática e rápida etiquetas que representem mensagens de sistemas transaccionais complexos.

- **Estrutura Distribuída:** Investigar a possibilidade de distribuir a estrutura do repositório por diferentes suportes físicos. Desta forma poder-se-ia entregar o controlo de diferentes partes de um mesmo repositório a diferentes entidades.

- **Replicação:** Definir procedimentos de manutenção de cópias de partes ou de todo o repositório de forma segura, confiável e facilmente acessível.

- **Formato dos dados:** Optimizar a inclusão dos dados (elemento M) nos registos R_m. Pode, por exemplo, ser utilizado um sistema de ficheiros (que forneça garantias fortes de disponibilidade) onde serão mantidos os dados de entrada no repositório. Deste modo o conteúdo de cada elemento M representará uma referência inequívoca para um ficheiro cujo conteúdo são os dados de entrada (e não os próprios dados).

- **Comunicação com os clientes do sistema:** Optimizar a estrutura dos pedidos de inserção e leitura de dados. A utilização de XML torna a estrutura actual bastante escalável, mas acaba por limitar também o desempenho do sistema. Devem ser procuradas alternativas que não necessitem de transmitir os dados num formato textual (i.e., legível).

Apêndice A

Interfaces

Módulo de Gestão do Repositório

```
package novis.ts.tstp.interfaces;


public interface RepositoryInterface {

  public Vector getRootRecords();
  public boolean addRmRecord(RmRecord rmRec);
  public boolean addReRecord(ReRecord reRec);
  public boolean addRhRecord(RhRecord rhRec);
  public RmRecord getRmRecord(int pr);
  public ReRecord getReRecord(int pr);
  public RhRecord getRhRecord(int hr,int pr);
  public Vector getOpenEpochRmRecords(int firstRecord);
  public Vector getRmRecords(int firstRecord, int lastRecord);
  public ReRecord getEpochReRecord(int rm_pr);
  public ReRecord getLastReRecordInSubset(int sce);
  public RhRecord getLastRhRecordInSubset(int hr,int sce);
  public ReRecord getLastReRecord();
  public RhRecord getLastRhRecord(int hr);
  public Vector getRootRecords();

}
```

Módulo de Gestão Criptográfica

```
package novis.ts.tstp.interfaces;


public interface CryptographyInterface {

  public byte[] mac(byte[] data, String alias);
  public boolean createMacKey(String key_alias);
  public boolean deleteMacKey(String key_alias);
  public boolean createCertificateRequest(String hr_value,String ultm_value,String alias);
```

```
  public byte[] signRecord(byte[] data, String alias);
  public X509Certificate validateSig(byte[] data,byte[] sig);
  public byte[] digest(byte[] data);
  public byte[] getMacKey(String alias);

}
```

Módulo de Gestão de Certificados

```
package novis.ts.tstp.interfaces;


public interface CertificateInterface {

  public byte[] getCertificate(String alias);
  public boolean putCertificateRequest(byte[] certReq, String alias);

}
```

Módulo de Validação de Registos

```
package novis.ts.tstp.interfaces;


public interface ValidationInterface {

  public boolean validateRmRecord(int pr);
  public boolean validateRmRecord(String l);
  public ReRecord validateReSubset(ReRecord re_rec);
  public RhRecord validateRhSubset(RhRecord rh_rec,byte[] sig);

}
```

Módulo de Processamento

```
package novis.ts.tstp.interfaces;


public interface ProcessorInterface {

  public boolean process(File msg);
  public boolean closeEpoch();
  public void completeRecordTree();

}
```

Cliente do Serviço de Validação Cronológica

```
package novis.ts.tstp.interfaces;


public interface TimestampInterface {

  public byte[] getTimestamp(byte[] data) throws Exception;
  public Date validateTimestamp(byte[] timestamp, byte[] content);
  public Date getTimestampDate(byte[] timestamp);

}
```

Referências

[1] *Decreto-Lei n.º 165/2004, de 6 de Julho.*

[2] *Decreto-Lei n.º 256/2003, de 21 de Outubro.*

[3] *Decreto-Lei n.º 290-D/99, de 2 de Agosto.*

[4] *Decreto-Lei n.º 375/99, de 18 de Setembro.*

[5] *Decreto-Lei n.º 62/2003, de 3 de Abril.*

[6] *Decreto Regulamentar n.º 16/2000, de 2 de Outubro.*

[7] *Decreto Regulamentar n.º 25/2004, de 15 de Julho.*

[8] IAIK - Institute for Applied Information Processing and Communications. Graz University of Technology. http://www.iaik.tugraz.at.

[9] JBoss. JBoss Inc. http://www.jboss.org.

[10] MySQL. MySQL AB. http://www.mysql.com.

[11] nCipher. nCipher Corporation Ltd. http://www.ncipher.com.

[12] OpenLDAP. The OpenLDAP Foundation. http://www.openldap.org.

[13] The OpenSSL Project. http://www.openssl.org.

[14] PKCS #7 v1.5: Cryptographic Message Syntax Standard. Technical report, RSA Laboratories, 1993.

[15] ANSI X9.52-1998. Technical report, ANSI, 1998.

[16] ITU-T Recommendation X.509. Technical report, ITU-T, 2000.

[17] PKCS #10 v1.7: Certification Request Syntax Standard. Technical report, RSA Laboratories, 2000.

[18] PKCS #11 v2.20: Cryptographic Token Interface Standard. Technical report, RSA Laboratories, 2004.

[19] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. Time-Stamp Protocol (TSP). RFC 3161, Internet Engineering Task Force, 2001.

[20] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *ACM Conference on Computer and Communications Security*, pages 7–17, 1997.

[21] N. Asokan, Victor Shoup, and Michael Waidner. Asynchronous Protocols for Optimistic Fair Exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99, 1998.

[22] A. Bahreman and J. Tygar. Certified electronic mail. In *Proceedings of the Internet Society Symposium on Network and Distributed System Security*, pages 3–19, 1994.

[23] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying Hash Functions for Message Authentication. In Neal Koblitz, editor, *Advances in Cryptology – Crypto 96 Proceedings*, volume 1109 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[24] M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest. A Fair Protocol for Signing Contracts. *IEEE Transactions on Information Theory*, 36(1):40–46, 1990.

[25] S. Boeyen, T. Howes, and P. Richard. Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2. RFC 2559, Internet Engineering Task Force, 1999.

[26] John Boyer, Merlin Hughes, and Joseph Reagle. XML-Signature XPath Filter 2.0. Technical report, World Wide Web Consortium (W3C), 2002.

[27] S. Chokhani and W. Ford. Internet X.509 Public Key Infrastructure : Certificate Policy and Certification Practices Framework. RFC 2527, Internet Engineering Task Force, 1999.

[28] Tom Coffey and Puneet Saidha. Non-repudiation with mandatory proof of receipt. Technical report, ACM SIGCOMM Computer Communication Review 26, 1996.

[29] Bert den Boer and Antoon Bosselaers. Collisions for the compression function of md5. In *Workshop on the theory and application of cryptographic techniques*

on Advances in cryptology, pages 293–304. Springer-Verlag New York, Inc., 1994.

[30] T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246, Internet Engineering Task Force, 1999.

[31] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[32] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, 1985.

[33] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18. Springer-Verlag New York, Inc., 1985.

[34] R. Fielding, James Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, Internet Engineering Task Force, 1999.

[35] Stuart Haber and W. Scott Stornetta. Secure Names for Bit-Strings. In *ACM Conference on Computer and Communications Security*, pages 28–35, 1997.

[36] J. Hodges and R. Morgan. Lightweight Directory Access Protocol (v3): Technical Specification. RFC 3377, Internet Engineering Task Force, 2002.

[37] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, Internet Engineering Task Force, 2002.

[38] Information Sciences Institute. Transmission Control Protocol. RFC 793, Internet Engineering Task Force, 1981.

[39] Kwangjo Kim, Sangjoon Park, and Joonsang Baek. Improving Fairness and Privacy of Zhou-Gollmann's Fair Non-Repudiation Protocol. In *ICPP Workshop*, pages 140–145, 1999.

[40] S. Kremer and O. Markowitch. Optimistic Non-repudiable Information Exchange. In *21st Symposium on Information Theory in the Benelux*, 2000.

[41] S. Kremer and O. Markowitch. Selective Receipt in Certified E-mail. *Lecture Notes in Computer Science*, 2247:136+, 2001.

[42] Steve Kremer, Olivier Markowitch, and Jianying Zhou. An Intensive Survey of Fair Non-Repudiation Protocols. *Computer Communications*, 25-17, 2002.

[43] J. Linn. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures. RFC 1421, Internet Engineering Task Force, 1993.

[44] O. Markowitch, D. Gollmann, and S. Kremer. On Fairness in Exchange Protocols. Technical Report 461, 2001.

[45] R. C. Merkle. Protocols for public key cryptosystems. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 122–134. IEEE Computer Society Press, 1980.

[46] Silvio Micali. Certified E-mail with invisible post offices. In *RSA conference. San Francisco, California*, 1997.

[47] David L. Mills. Network Time Protocol (NTP). RFC 958, Internet Engineering Task Force, 1985.

[48] David L. Mills. Network Time Protocol (Version 3). RFC 1305, Internet Engineering Task Force, 1992.

[49] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560, Internet Engineering Task Force, 1999.

[50] National Institute of Standards and Technology (NIST). *NIST Federal Information Processing Standards Publication 186: Digital Signature Standard*, May 1994.

[51] National Institute of Standards and Technology (NIST). *NIST Federal Information Processing Standards Publication 46-3: Data Encryption Standard*, October 1999.

[52] National Institute of Standards and Technology (NIST). *NIST Federal Information Processing Standards Publication 197: Advanced Encryption Standard*, November 2001.

[53] National Institute of Standards and Technology (NIST). *NIST Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules*, Dec 2002.

[54] Oded Goldreich. Zero-knowledge twenty years after its invention. Technical report, Faculty of Mathematics and Computer Science. Weizmann Institute of Science, Israel, 2004.

[55] National Institute of Standards and Technology (NIST). *NIST Federal Information Processing Standards Publication 180: Secure Hash Standard*, May 1993.

[56] National Institute of Standards and Technology (NIST). *NIST Federal Information Processing Standards Publication 180-2: Secure Hash Standard*, August 2002.

[57] National Institute of Standards and Technology (NIST). *Proposed Withdrawal of FIPS for the Data Encryption Standard (DES)*, July 2004.

[58] Jon M. Peha. Electronic commerce with verifiable audit trails. In *Proceedings of ISOC*, 1999.

[59] J. Postel and J. Reynolds. File Transfer Protocol. RFC 959, Internet Engineering Task Force, 1985.

[60] B. Ramsdell. S/MIME Version 3 Message Specification. RFC 2633, Internet Engineering Task Force, 1999.

[61] M. Reis, A. Romão, and A. E. Dias. Practical Auditability in Trusted Messaging Systems. In *Proceedings of ICETE - International Conference on E-Business and Telecommunication Networks*, 2004.

[62] M. Reis, A. Romão, and A. E. Dias. Registos Auditáveis: Aplicação a um Sistema de Facturação Electrónica. In *Actas da CRC 2004 - 7ª Conferência sobre Redes de Computadores*, 2004.

[63] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, Internet Engineering Task Force, 1992.

[64] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[65] R. L. Rivest, A. Shamir, and L. Adleman. On Digital Signatures and Public-Key Cryptosystems. Technical report, MIT Laboratory for Computer Science, 1979.

[66] Matt Robshaw. On Pseudo-Collisions in MD5. Technical report, RSA Laboratories, 1994.

[67] Tomas Sander and Amnon Ta-Shma. Auditable, anonymous electronic cash. In *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, pages 555–572. Springer-Verlag, 1999.

[68] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* John Wiley and Sons, Inc., second edition, 1995.

[69] M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3). RFC 2251, Internet Engineering Task Force, 1997.

[70] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu. Collisions for hash functions md4, md5, haval-128 and ripemd. Cryptology ePrint Archive, Report 2004/199, 2004.

[71] François Yergeau, John Cowan, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler. Extensible Markup Language (XML) 1.1. Technical report, World Wide Web Consortium (W3C), 2004.

[72] Jianying Zhou. *Non-Repudiation in Electronic Commerce.* Artech House, first edition, 2001.

[73] Jianying Zhou and Dieter Gollmann. A Fair Non-repudiation Protocol. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 55–61. IEEE Computer Society Press, 1996.