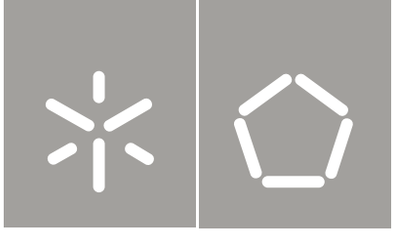
Universidade do Minho
Escola de Engenharia

Fábio Raul da Costa Gonçalves

Arquitectura de Segurança para a Prestação de Serviços de Saúde em Mobilidade

outubro de 2013

Arquitectura de Segurança para a Prestação de Serviços de Saúde em Mobilidade
Fábio Raul da Costa Gonçalves
UMinho | 2013



Universidade do Minho
Escola de Engenharia

Fábio Raul da Costa Gonçalves

Arquitectura de Segurança para a Prestação de Serviços de Saúde em Mobilidade

Tese de Mestrado
Ciclo de Estudos Integrados Conducentes ao
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do
Professor Alexandre Júlio Teixeira Santos
Professor Joaquim Melo Henriques Macedo

outubro de 2013

DECLARAÇÃO

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Guimarães, ___/___/_____

Assinatura: _____

A escrita da presente dissertação não segue o novo acordo ortográfico.

Resumo

O crescente custo associado ao tratamento de pacientes leva à sua relocação para o próprio domicílio. Esta relocação conduz à necessidade de uso de ferramentas automatizadas permitindo a diminuição dos erros que a mesma acarreta.

O uso de *Radio Frequency Identification* (RFID) permite não só a identificação dos medicamentos como também de pacientes, médicos, enfermeiros e qualquer outro tipo de prestador de cuidados de saúde. A combinação do uso de identificação de etiquetas RFID com soluções Internet of Things (IoT) bem estruturadas e seguras permite um acesso fácil e ubíquo a registos médicos, oferecendo em simultâneo controlo e segurança a todas as interacções.

Nesta dissertação é definida uma arquitectura de segurança, facilmente implementável em plataformas móveis, que permite o estabelecimento e gestão de um serviço de prescrições médicas, num contexto de mobilidade usando Registo Pessoal de Saúde (PHR) electrónico. Esta arquitectura de segurança tem como objectivo o uso com uma aplicação móvel de saúde (M-Health) através de um interface simples e intuitivo, suportado pela tecnologia RFID. Esta arquitectura, capaz de suportar interacções seguras e autenticadas, vai permitir uma fácil implementação de aplicações M-Health. É apresentado o caso especial de administração de medicamentos e o sistema controlo de medicação ubíqua, de acordo com o contexto da IoT.

A arquitectura de segurança e os seus protocolos, juntamente com o serviço seguro *Ambient Assisted Living* (AAL) para controlo de medicação, são analisados no contexto da IoT.

De forma a verificar a exequibilidade dos protocolos e da arquitectura de segurança, foram implementadas aplicações protótipos (fixas e móveis) que permitem verificar o funcionamento total do sistema. Foram ainda efectuados alguns testes de segurança e desempenho para verificar a usabilidade de todo o sistema.

Abstract

The increasing healthcare costs leads to their relocation to their own homes. This leads to the need of automated tools allowing the decrease errors that this entails.

The use of *Radio Frequency Identification* (RFID) technology allows not only drug identification, but also identification of patient, physicians, nurses or any other healthcare giver. The combination of RFID tag identification with structured and secured Internet of Things (IoT) solutions allows an ubiquitous and easy access to medical records, while providing control and security to all interactions.

In this thesis is defined a security architecture, easily deployable on mobile platforms, which would allow to establish and manage a medication prescription service in mobility context, making use of electronic Personal Health Record (PHR). This security architecture is aimed to be used with a mobile e-health application (M-Health) through a simple and intuitive interface, supported by RFID technology. This architecture, able to support secured and authenticated interactions, will enable an easy deployment of m-health applications. The special case of drug administration and ubiquitous medication control system, along with the corresponding IoT context, is presented.

The secuirity architecture and its protocols, along with a general *Ambient Assisted Living* (AAL) secure service for medication control, is then analyzed int the context of IoT.

To verify the architecture and protocols implementability, they were deployed prototype applications (Fixed and mobile) allowing the verification of the whole system operating. They were also made some security and performance tests allowing system usability verification.

Índice

Índice de Figuras

Índice de Tabelas

Abreviaturas

AAL *Ambient Assisted Living* i

AES *Advanced Encryption Standard* 10

BD Base de dados v

CA Autoridade de Certificação vi

CRC *Cyclic Redundancy Check* 12

DBMS *Data Base Management System* 30

DES *Data Encryption Standard* 10

DOS *Denial of Service* 71

EEPROM *Electrically Erasable Programmable Read-Only Memory* 14

EPC *Electronic Product Code* 17

Gen 2 Protocolo de interface aérea UHF *standard* classe 1 geração 2 17

HTTP *HyperText Transfer Protocol* 40

HTTPS *HyperText Transfer Protocol Secure* 40

IoT Internet of Things i

IPsec *Intenet Protocol Security* 34

IS-RFID Inpatient Safety RFID 25

ITF *Interrogator-talks-first* 17

IP *Internet Protocol* 31

MAC *Message Authentication Code* 10

MD5 *Message-Digest algorithm 5* 31

MHOUA *M-Health Offline User Acess* vi

MHSP *M-Health Security Protocol* vi

MHSUA *M-Health Secure User Authentication* vi

MHUA *M-Health User Authentication* vi

NFC *Near Field Communication* v

OSI *Open Systems Interconnection* v

PC Puzzles Criptográficos 21

PHR Registo Pessoal de Saúde i

PRNG *Pseudo-Random Number Generation* 19

QR *Quick Response* 16

RFID *Radio Frequency Identification* i

SUA *Secure User Authentication* 40

SHA1 *Secure Hash Algorithm* 31

SQL *Structured Query Language* 31

SSH *Secure Shell* 31

SSL *Secure Sockets Layer* 34

TCP *Transport Control Protocol* 31

TLS *Transport Layer Security* 36

UA *User Authentication* 40

UDP *User Datagram Protocol* 31

WBAN *Wireless Body Area Network* 37

WSBC *Weakly Secret Bit Commitment* 29

Capítulo 1

Introdução

1.1 Enquadramento

Um problema com que se debatem os serviços de saúde em Portugal e outros países, advém dos custos acrescidos da prestação de cuidados de saúde essencialmente centrados em unidades de saúde centrais e internamento, especialmente quando os tratamentos implicam tratamento especializado. Assiste-se assim a uma deslocalização de muitos serviços de saúde que, sempre que possível, passam dos hospitais e unidades de saúde centrais para cuidados em casa do paciente.

A partir do momento em que é prescrita a medicação até à toma/administração, decorre um período onde muitos erros podem acontecer. Estes podem ocorrer devido a uma má comunicação (quer a nível escrito quer a nível oral) [1] ou erros efectuados durante a toma de medicação pelo paciente. De realçar, que este deve tomar a dose certa do medicamento certo no momento certo. Este processo pode ser controlado através de acesso seguro ao Registo Pessoal de Saúde (PHR) do paciente.

Grande parte dos pacientes toma a medicação sem qualquer tipo de assistência, aumentando a probabilidade de acontecerem erros, principalmente em utentes com idade avançada [2]. Este processo pode ser facilitado utilizando as diversas tecnologias disponíveis para a prestação de cuidados remotos, tendo em conta que se possa minimizar os riscos de segurança associados. Utilizando a *Pervasive* Internet pode aceder-se à informação a partir de qualquer lugar. Assim, com a identificação automática de todos os participantes envolvidos os processos podem ser (semi-) automatizados permitindo reduzir os erros.

A utilização da tecnologia no contexto de saúde tem que cumprir fortes requisitos de segurança devido à natureza que este sensível ambiente apresenta [3]. Falhas a nível da segu-

rança podem trazer complicações a nível financeiro ou mesmo riscos para a vida dos pacientes [4][5]. Logo, uma aplicação para e-health deve implementar fortes mecanismos de segurança, por forma a evitar o tratamento errado, má identificação ou acesso não autorizado. Deve ser também mantido um registo de todos as operações relevantes para o sistema tornando possível detectar qualquer tipo de negligência [4][5].

Este trabalho analisa e apresenta novos protocolos de segurança e soluções para integrar uma arquitectura de segurança para serviços M-Health onde, um sistema para controlo de medicação remoto para *Ambient Assisted Living* (AAL) [6], direccionado principalmente para pacientes idosos, é proposto.

Utilizando a tecnologia disponível (auto-identificação, criptografia e Internet of Things (IoT) [7]), pretende-se definir e avaliar componentes com fortes restrições de segurança tanto em termos de identificação como de autenticação, a nível de transmissão e de armazenamento. Estes componentes são combinados numa arquitectura de segurança desenhada para gerir um serviço m-health de prescrição e gestão de prescrições adaptado para um acesso ubíquo em ambiente móvel.

Como referido anteriormente um dos principais objectivos da arquitectura de segurança para serviços de saúde no contexto de saúde passa por evitar erros na toma de medicação. Este é particularmente importante e deve ser feito de uma forma segura e fácil sem necessitar de intervenção especializada.

Foi efectuado um estudo em Portugal, *"Adesão ao regime medicamentoso em idosos na comunidade"* [8], onde é possível ver que grande parte da população idosa necessita de ajuda para gerir a medicação. O estudo [8] feito sobre a população idosa verificou que 60.5% das pessoas não toma a medicação por esquecimento e 24.4% não o faz porque não tem a medicação consigo no momento da toma. Esse estudo permitiu também concluir que, informar a população sobre a medicamentação ajuda a aumentar a adesão à toma medicamentosa. Como pode ser visto na Tabela 1.1, grande parte dos idosos necessita de ajuda para controlar a medicação, sendo que 82.8% destes precisa de ajuda por razões que poderiam ser ultrapassados com recurso a sistemas AAL semi-automatizados.

1.2 Cenários de Utilização

Como se pode ler anteriormente, um dos principais objectivos da arquitectura de segurança para a prestação de serviços de saúde e mobilidade é evitar erros nas tomas de medicação por utentes em ambulatório, sem necessidade qualquer assistência. Assim, o principal cenário

Tipo de Ajuda	Número	%
Gestão da Medicação	119	36,1
Obter Informação sobre a Medicação	63	19,2
Explicar o Regime de Medicação	44	13,3
Interpretar o Regime de Medicação	26	7,9
Monitorizar o Regime de Medicação	20	6,1
Lembrar as Horas da Medicação	19	5,8
Encher o Dispensador de Medicação	10	3,0
Ajuda Monetária	7	2,1
Ler o Rótulo	8	2,4
Retirar os Medicamentos da Caixa	4	1,2

Tabela 1.1: Adesão ao regime de medicação em idosos na comunidade (Adaptado de [8])

de utilização será, tal como referido, a utilização deste sistema por tais utentes e a possibilidade da actualização das prescrições por parte de médicos de forma fácil e automática.

Neste cenário (Figura 1.1) é dado a um utente um *smartphone* ou *tablet* que irá conter a aplicação a partir da qual poderá consultar as suas prescrições. O médico, a partir do seu computador portátil, *smartphone* ou *tablet*, poderá preencher a prescrição do utente, indicando a dosagem e a hora a que a medicação deverá ser tomada, informação que será imediatamente actualizada na BD. O utente poderá depois utilizar o *smartphone* que lhe foi dado para aceder ao sistema e actualizar as suas prescrições ou descarregar prescrições novas. Estas serão armazenadas de forma segura no dispositivo para posterior consulta em modo offline. Recorrendo à tecnologia *Radio Frequency Identification* (RFID) o dispositivo poderá verificar se o utente certo está a tomar a medicação certa à hora certa.

1.3 Objectivos

O foco deste trabalho é a definição, avaliação e teste de uma arquitectura de segurança para sistemas de prestação de cuidados médicos em mobilidade com garantias de confidencialidade, disponibilidade, identidade e autenticidade, estabelecido no contexto da IoT. Devem ser implementados os protótipos dos componentes da arquitectura definida para se poder fazer a respectiva prova de conceito.

Pretende-se definir uma arquitectura base de um serviço seguro e implementável em plataformas móveis, que permita estabelecer um serviço para suporte a prescrições médica, também em mobilidade; complementarmente, deverá especificar-se e testar-se uma aplicação móvel de interface simples, intuitivo, seguro e autenticado, que se socorra de RFID e permita auxiliar a

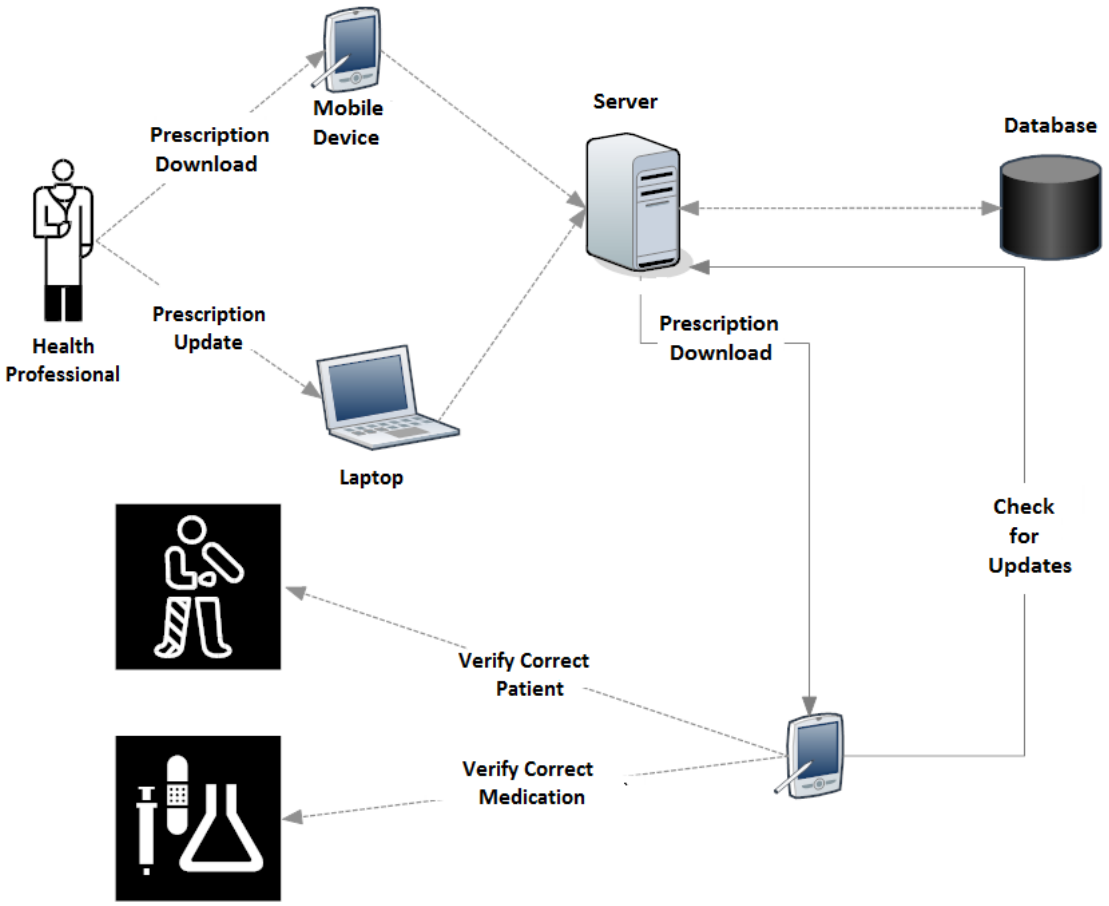


Figura 1.1: Cenário de Utilização [9]

confirmação local de toma bem como a monotorização remota, do cumprimento da prescrição medicamentosa estabelecida.

Admite-se que esta arquitectura seja testada com auxilio de uma Base de dados (BD) intermediária, mas deve também prever uma integração com um serviço geral de AAL no contexto da IoT, com acesso confidencial e seguro. É também objectivo deste trabalho a análise comparativa e selecção de *hardware* e drivers para *Near Field Communication* (NFC) [10] em dispositivos móveis.

1.4 Resultados Obtidos

Todo o trabalho efectuado e que irá ser descrito durante toda esta dissertação permitiu atingir com sucesso os objectivos propostos, entre eles obter uma arquitectura de segurança para sistemas de prestação de cuidados médicos em mobilidade.

Essa arquitectura teve como ponto de partida o cenário de aplicação referido anteriormente. Este foi cuidadosamente analisado e foram identificados quais os seus pontos de fragilidade. Após esse processo foram identificados quais os melhores métodos para eliminar as fragilidades identificadas.

As comunicações entre dispositivos RFID foram seguras recorrendo ao protocolo IS-RFID proposto por [11]. Este pode ser ainda complementado com o uso do protocolo PC [12].

Para os restantes pontos de fragilidade encontrados foram propostos 4 novos protocolos: MHSP, MHUA, acsmhsua, MHOUA.

O protocolo MHSP oferece ferramentas que permitem efectuar a autenticação entre aplicações e criar um canal seguro entre estas, possibilitando a comunicação segura entre entidades.

Foram criados dois tipos de autenticação:UA e SUA; que utilizam os protocolos MHUA e MHSUA, respectivamente, para efectuar a autenticação dos utilizadores.

Uma vez que nenhum destes protocolos permite efectuar uma autenticação offline, o protocolo MHOUA foi criado com esse fim. Este permite guardar dados temporários de uma forma segura no dispositivo por forma a poderem ser consultados offline.

Para o bom funcionamento dos protocolos referidos, foi definido também um processo de registo. Este é efectuado através de uma CA e fornece as credencias de acesso para os diversos utilizadores e aplicações.

Foi definido ainda uma forma de acesso seguro à BD, que passa pelo uso de uma aplicação especial que corre o protocolo MHSP e está instalada na máquina onde reside a BD. Esta permite uma troca de dados de forma segura com o servidor, reencaminhando a informação directamente para a BD. Não efectua qualquer tipo de processamento da informação.

De realçar que o trabalho efectuado para o desenvolvimento desta dissertação originou um artigo *"Security Architecture for Mobile E-Health Applications in Medication Control"* [9] apresentado na conferência *"SoftCom 2013 - 21th International Conference on Software, Telecommunications and Computer Networks"*.

1.5 Estrutura do Documento

A dissertação aqui apresentada encontra-se dividida em 7 capítulos, para melhor poder descrever todos os passos efectuados desde, a motivação que impulsionou o trabalho aos resultados obtidos. Os capítulos aqui apresentados são os seguintes: introdução, estado da arte, análise de requisitos e fragilidades, arquitectura e protocolos de segurança, implementação de sistema seguro para M-Health, análise de segurança e desempenho e, por fim, conclusões e

trabalho futuro.

A introdução pretende descrever qual a motivação do trabalho efectuado e quais os objectivos pretendidos. É também descrito qual o cenário para o qual a solução será desenhada e feito um pequeno resumo do trabalho que foi desenvolvido e dos objectivos atingidos.

No estado da arte, foram cuidadosamente estudadas as tecnologias envolvidas, de forma a poder encontrar-se as que melhor se adequam ao trabalho em questão. Efectua-se uma introdução à criptografia, enumerando-se os serviços comuns de segurança. São abordados os sistemas simétricos e assimétricos de criptografia, a Autoridade de Certificação (CA) e tecnologias de acesso como: *smart cards*, NFC e RFID. Relativamente às tecnologias RFID, são referidos os tipos de etiquetas existentes e os protocolos de segurança mais relevantes. Termina-se o capítulo com os protocolos de segurança relevantes para aplicação na BD e Internet.

No capítulo análise de requisitos e fragilidades faz-se uma análise às tecnologias descritas no estado da arte e ao cenário de aplicação descrito na introdução. A partir dessa análise é possível identificar as fragilidades encontradas no cenário de aplicação e quais as melhores ferramentas para as eliminar. São apresentadas soluções para efectuar diversos procedimentos em segurança, onde se podem identificar os seguintes: Comunicação sobre a Internet e autenticação das aplicações, autenticação dos utilizadores, ligação à BD e protecção de dados, armazenamento de dados para consulta offline e protocolos para comunicação sobre RFID

Seguidamente, no capítulo arquitectura e protocolos de segurança, é apresentada a arquitectura desenhada e os novos protocolos propostos. Esta arquitectura tem como objectivo minimizar ou eliminar as fragilidades encontradas, utilizando para isso os protocolos propostos. Esse protocolos são MHSP, MHUA, MHSUA e MHOUA. Define-se também como efectuar o processo de registo e como serão feitas as comunicações entre dispositivos RFID

De forma a verificar se os protocolos e arquitectura propostos são aplicáveis, foram desenhadas aplicações que permitam a sua implementação. O capítulo implementação de sistema seguro para M-Health tem como objectivo descrever a implementação efectuada. São apresentadas as diversas aplicações desenhadas e o modelo relacional da BD. Foram desenhadas aplicações para todos os dispositivos (fixos e móveis), tal como, para o servidor BD e para a CA

Após a implementação realizaram-se alguns testes de segurança e de desempenho de forma a verificar a relação desempenho/segurança oferecida pela arquitectura proposta. Estes são apresentados no capítulo análise de segurança e desempenho.

No capítulo conclusões e trabalho futuro é apresentado, na forma de um pequeno resumo, tudo o que foi alcançado com este trabalho. Desta forma, é possível verificar se os objectivos

propostos e a segurança desta arquitectura foram alcançados. São também propostas futuras implementações de forma a melhorar o sistema. Entre estes, podemos realçar alguns melhoramentos a nível de segurança e a incorporação do Cartão de Cidadão no sistema desenhado.

Capítulo 2

Estado da arte

Neste capítulo será descrito o estado corrente das tecnologias envolvidas. Ir-se-á descrever em pormenor quais os protocolos e métodos de segurança encontrados durante a pesquisa efectuada, que sejam relevantes tendo em conta requisitos de segurança (Capítulo 3.1). Serão também descritas algumas tecnologias e técnicas relevantes para o desenho da arquitectura de segurança para a prestação de serviços de saúde em mobilidade.

2.1 Criptografia

Criptografia é a ciência que estuda a escrita secreta [13][14]. Classicamente, a criptografia oferecia métodos para poder enviar informação secreta através de um canal inseguro. O emissor escolhia a cifra e a chave que dava directamente ao receptor ou enviava por um canal seguro.

Uma cifra é um método que permite transformar uma mensagem em texto limpo (*Cleartext*) em uma mensagem em texto cifrado (*cyphertext*). O processo de transformar o texto limpo em cifrado é chamado de cifragem. Os processos de cifrar e de decifrar são controlados por uma ou várias chaves criptográficas.

Criptoanálise é a ciência que estuda métodos para quebrar as cifras. Como pode ler-se em [13][14], uma cifra é quebrável quando for possível descobrir o texto ou a chave do texto cifrado, ou a chave dos pares texto cifrado, texto limpo .

Segundo [13][14], existem três métodos básicos de ataques: apenas texto cifrado, texto limpo conhecido e texto limpo escolhido.

Nos ataques em que o criptoanalista possui apenas blocos de apenas texto cifrado, este deve descobrir qual a chave, tendo apenas acesso a texto cifrado interceptado. Podem ser sabidas certas propriedades do texto cifrado como a linguagem deste, o modo como foi cifrado, o seu assunto e algumas palavras prováveis.

O ataque pode ser feito através de texto limpo conhecido. Neste ataque criptoanalista sabe alguns pares de texto limpo, texto cifrado.

O caso mais favorável ao atacante é aquele em que o atacante tem acesso a blocos de texto limpo escolhido. Neste, são conhecidos os pares de texto cifrado correspondentes aos pares texto limpo seleccionados. As BD são mais vulneráveis a este tipo de ataques se for possível a um utilizador inserir elementos na BD e verificar as mudanças ocorridas;

A informação transmitida sobre linhas electrónicas pode sofrer escutas passivas ou activas [13][14]. Nas escutas passivas as mensagens são interceptadas, normalmente sem detecção da escuta. Este ataque pode ser utilizado para determinar o conteúdo da mensagem ou a sua origem e destinatário. No caso das escutas activas as mensagens são alteradas deliberadamente. As mensagens podem sofrer alterações arbitrárias ou ser totalmente substituídas por mensagens anteriores.

A cifragem das mensagens protege contra a modificação ou injecção de novas mensagens, mas não contra ataques de repetição. Estes ataques utilizam mensagens previamente recolhidas.

Como se pode ler em [13][14], existem dois tipos de sistemas criptográficos: Sistemas simétricos ou de uma chave (Capítulo 2.1.2) e sistemas assimétricos ou de duas chaves (Capítulo 2.1.3). Nos sistema simétricos é utilizada a mesma chave para efectuar a cifragem e decifragem de uma mensagem, ao passo que nos assimétricos são utilizadas chaves diferentes para efectuar os dois processos referidos.

Na figura 2.1 é apresentado o modelo de comunicações básico utilizado para representar alguns exemplos de aplicações de criptografia, os seu desafios e implementações. A (Alice) e B (Bob) representam duas entidades que pretendem comunicar através de um meio inseguro e E (Eve) representa a entidade que pretende quebrar os serviços de segurança fornecidos a A e B [15].

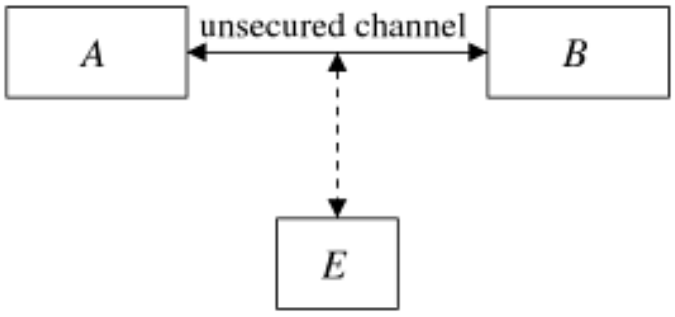


Figura 2.1: Arquitetura comunicações básica (Retirado de [15])

De forma a poder modelar ameaças realistas é assumido que E consegue ver todos os dados transmitidos no canal, consegue alterar dados transmitidos e ainda injectar dados criados por ele. Esta entidade tem, também, grande poder computacional e conhece todos os protocolos de comunicação, tal como os algoritmos criptográficos utilizados.

2.1.1 Objectivos de segurança

Segundo [15], existem alguns objectivos fundamentais das comunicações seguras que devem ser atingidos para se conseguir um sistema seguro:

- Confidencialidade: Manter a informação secreta a todas as entidades, excepto às quais esta é destinada. As mensagens entre A e B não devem ser compreensíveis por E.
- Integridade: Assegurar que os dados não são alterados por meios não autorizados. Deve ser possível a entidade B detectar se a informação enviada por A foi alterada por E.
- Autenticação da origem da informação: Verificar a autenticidade dos dados. B deve poder verificar que os dados supostamente enviados por A foram realmente enviados por A.
- Autenticação das entidades: Verificar a autenticidade de uma entidade. B deve poder confirmar a identidade de outras entidades.
- Não repúdio: Impedir uma entidade de negar acções ou compromissos prévios. Se B receber uma mensagem de A, além de ficar convencido que esta originou de A, consegue convencer terceiros que esta foi originada por A, fazendo com que A não consiga negar ter enviado a mensagem a B.

2.1.2 Sistemas simétricos

Nos sistemas simétricos (Figura 2.2), como referido anteriormente, existe apenas uma chave. A chave para cifrar e decifrar é a mesma [13][14]. As entidades que pretendem comunicar combinam previamente uma chave que seja secreta e autêntica [15].Estes sistemas garantem confidencialidade.

Nos sistemas simétricos pode ser necessário o uso de um *Message Authentication Code* (MAC) [16] para obter integridade dos dados e autenticação da origem destes. Como exemplo de algoritmos de chaves simétricos temos o *Data Encryption Standard* (DES) [17] , o 3 DES e o *Advanced Encryption Standard* (AES) [18].

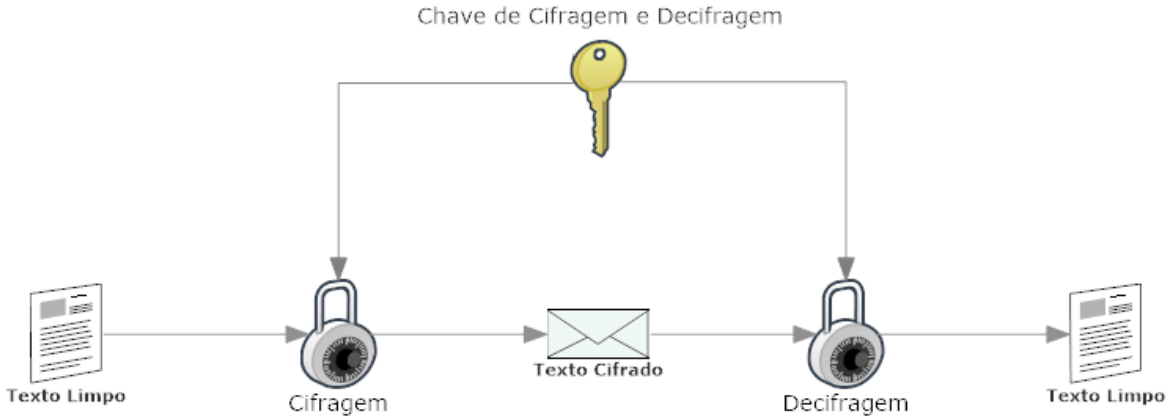


Figura 2.2: Sistema simétrico

O algoritmo DES utiliza uma chave binária de 64 bits. 56 desses bits são gerados e utilizados directamente pelo algoritmo e os restantes 8 são utilizados para detecção de erros. Este algoritmo não oferece muita segurança devido à sua pequena chave [19].

Como um melhoramento da cifra DES existe o algoritmo 3 DES, que utiliza chaves de 192 ou 256 bits. Apesar do aumento da chave, continua a não oferecer muita segurança. Segundo [19], a cifra DES foi desenhada para cifragem por *hardware* e como hoje em dia é implementada na maioria das vezes por *software*, torna-se bastante ineficiente.

O algoritmo AES pode utilizar chaves de 128, 192 ou 256 bits. Oferece maior segurança que as cifras referidas anteriormente. Um ataque por dicionário precisa de 2^{128} blocos de texto livre para poder cifrar ou decifrar uma mensagem arbitrária com uma mensagem desconhecida. Este ataque aplica-se a qualquer cifra de blocos determinística com blocos de 128 bits independentemente do seu desenho [19]. Uma cifra é denominada por determinística se gerar sempre o mesmo texto cifrado para um determinado texto limpo.

2.1.3 Sistemas assimétricos

Os sistemas assimétricos (Figura 2.3) são também conhecidos como sistemas de chaves públicas. Estes sistemas utilizam um par de chaves assimétricas para cifragem e decifragem. Cada par consiste numa chave pública e uma privada. A chave pública é distribuída indiscriminadamente enquanto que a privada é mantida sempre secreta. Os dados cifrados com a chave

pública podem apenas ser decifrados com a chave privada, tal como os dados cifrados com a chave privada apenas podem ser decifrados com a chave pública [20].

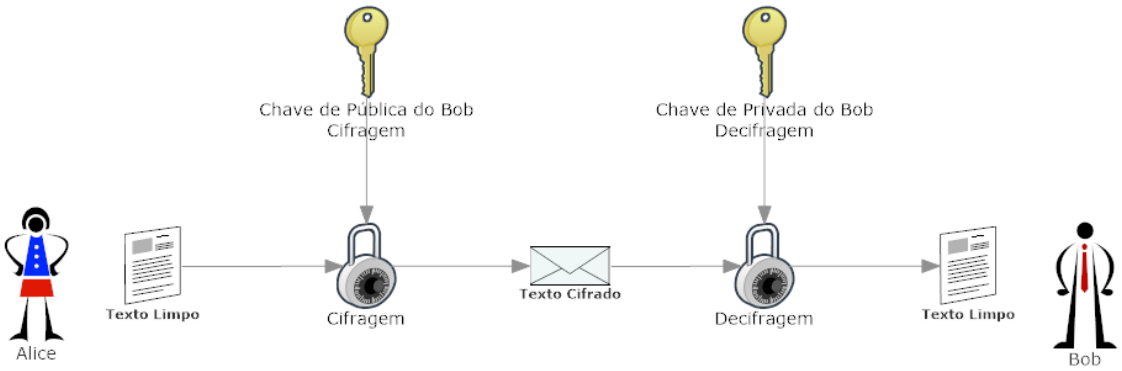


Figura 2.3: Sistema assimétrico

2.1.4 Funções de *Hash*

Uma função de *hash* [21] é uma função criptográfica que permite criar um resumo matemático especial da informação de uma mensagem. Se a informação da mensagem for alterada e o seu *hash* recalculado, terá um resultado diferente [20].

Para prevenir que uma mensagem seja alterada, a Alice calcula o *hash* da mensagem a enviar juntamente com um valor secreto que apenas ela e o Bob sabem. Ela junta o *hash* calculado à mensagem e envia para o Bob. O Bob pode posteriormente confirmar a validade da mensagem recebida. As funções de *hash* são algo parecidas com os códigos *checksum* ou o *Cyclic Redundancy Check* (CRC) [22], normalmente utilizados como mecanismos de detecção de erros. Estas diferem principalmente a nível de funcionalidades. Enquanto que os códigos CRC são utilizados para detectar erros acidentais, as funções de *hash* são desenhadas para detectar erros deliberados, tendo em conta que o atacante conhece o algoritmo para a sua criação e pode explorar as suas fraquezas.

2.1.5 Assinaturas Digitais

Uma assinatura digital [23] é função matemática utilizada para assinar mensagens. Se B receber uma mensagem assinada por A, segundo [13][14], a assinatura de A deve satisfazer os seguintes requisitos:

- B deve ser capaz de validar a assinatura de A em M.

- Deve ser impossível a qualquer entidade conseguir falsificar a assinatura de A.
- Se A quiser negar ter assinado a mensagem M, deve ser possível que uma terceira parte resolva o conflito entre A e B.

Uma assinatura digital garante a autenticidade de uma tal como dos dados da mensagem. De seguida ir-se-à descrever o processo para criar e validar uma assinatura. Para tal, é utilizado $dec(M, A_{privKey})$ para representar a decifragem da mensagem M com a chave privada de A e $enc(M, A_{pubKey})$ a cifragem da mensagem M com a chave pública de A [13][14]:

1. A assina a mensagem M calculando $C = enc(M, A_{privKey})$.
2. B valida a mensagem de A verificando se calculando $dec(M, A_{pubKey})$ obtém novamente a mensagem M.
3. Qualquer terceira parte pode resolver um conflito entre A e B efectuando o mesmo processo de B para verificar a autenticidade de M.

Uma vez que apenas A conhece a sua chave privada, é impossível a uma outra entidade calcular $enc(M, A_{privKey})$ e falsificar a sua assinatura.

2.1.6 Certificados de chave pública

Segundo [20] certificados de chave pública são documentos digitais que comprovam a ligação de uma chave pública a um individuo ou entidade. Permitem impedir que uma entidade utilize uma chave pública falsa, tentando se fazer passar por outra entidade.

Na sua forma mais simples, um certificado contém um nome e uma chave pública, podendo também conter data de validade, nome da entidade certificadora que emitiu o certificado, número de série , etc. Um dos elementos principais que o compõe é assinatura digital do emissor do certificado.

Os certificados são emitidos por uma Autoridade de Certificação (CA) que pode ser qualquer central de administração de confiança que esteja disposta a atestar a identidade dos certificados e associações feitas com as chaves dos certificados que emitiu. Uma CA deve publicitar a sua chave pública ou fornecer uma certificado de uma CA de mais alto nível, que possa comprovar a validade da sua chave pública.

2.2 Tecnologias de Acesso

2.2.1 Smart Cards

Um *Smart Card* [24] é um dispositivo que tem sensivelmente as dimensões de um cartão de crédito e que contém pelo menos um circuito integrado. Estes podem implementar diversas tecnologias, entre elas: código de barras (de uma ou duas dimensões), transmissores de radio frequência que não necessitam contacto, informação biométrica, cifragem, autenticação, identificação por foto.

A área de armazenamento principal é *Electrically Erasable Programmable Read-Only Memory* (EEPROM). Esta permite que o seu conteúdo seja alterado e permaneça inalterado depois da fonte de energia externa ser retirada.

O circuito integrado embebido no *Smart Card* pode funcionar como um microprocessador ou um computador; os dados são armazenados na memoria do cartão. A memória pode conter também algoritmos de cifragem que tornam quaisquer dados ou aplicações, que o *Smart Card* contenha, indecifráveis. Estes, quando utilizados juntamente com as aplicações e implementações apropriadas, podem aumentar a segurança total de um sistema e permitir a interoperabilidade entre agências. Os *Smart Cards* oferecem ferramentas que podem ser utilizadas para implementar um sistema utilizando identificação automática.

Segundo [24], existem três tipos de *Smart Cards* tendo em conta o tipo de funcionalidades que o circuito integrado oferece: *Memory-Only Integrated Circuit Chip Cards* (inclui os *Serial Protected Memory Chip*), *Wired Logic Integrated Circuit Chip Cards* e *Secure Microcontroler Integrated Circuit Chip Cards*.

Os *Memory-Only Integrated Circuit Chip Cards* são muito semelhantes aos cartões de banda magnética normais, oferecendo pouco mais segurança do que estes. De qualquer das formas, têm duas grandes vantagens sobre os cartões da banda magnética: muito maior capacidade de armazenamento de dados (16 Kbits em comparação com os 80 bytes por faixa) e os dispositivos que permitem fazer leitura/escrita são muito mais baratos.

Os *Wired Logic Integrated Circuit Chip Cards* contêm uma máquina de estados lógica que oferece cifragem e autenticação no acesso à memória e ao seu conteúdo.

Os *Secure Microcontroler Integrated Circuit Chip Cards* contêm um microcontrolador, um sistema operativo e uma memória de leitura/escrita que pode ser actualizada diversas vezes. Este é, no fundo, um minicomputador que se pode transportar no bolso e que necessita apenas de uma fonte de alimentação e um terminal.

Os *Smart Cards* permitem dois tipos primários de interfaces [24]: de contacto (Figura 2.4) e sem contacto (Figura 2.5). Os *Smart Cards* de contacto necessitam de uma ligação directa com o leitor para a troca de informação. Nos que não necessitam de contacto a informação é trocada com recurso a rádio frequência e costuma ter um alcance de cerca de 10 centímetros; necessitam de uma pequena antena para poder efectuar a transmissão. Além dos tipos primários referidos, existem outros dois tipos, que são uma junção dos dois tipos anteriormente referidos. Estes são *Hybrid Smart Cards* e *Dual-Interface Chip Smart Cards*. Os primeiros contêm dois circuitos no cartão: um suporta a interface com contacto e o outro sem contacto. Os *Dual Interface Chip Smart Cards* têm apenas um circuito o que permite os dois tipos de interface.

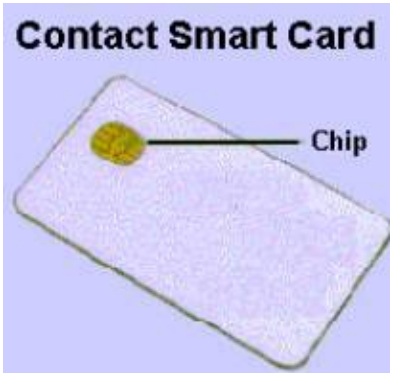


Figura 2.4: *Smart card* de contacto (Retirado de [25])

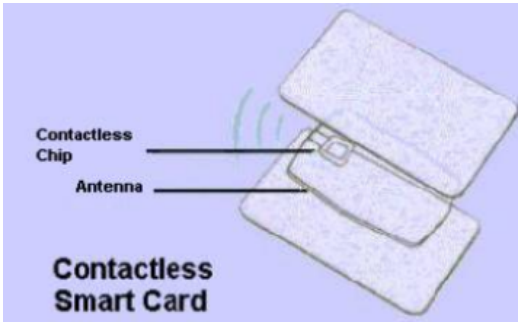


Figura 2.5: *Smart card* sem contacto (Retirado de [25])

2.2.2 *Near Field Communication* (NFC)

A norma *Near Field Communication* (NFC) [10] veio como resposta a um desafio com vários anos: ligar o mundo da Internet ao mundo físico. Este conceito é normalmente denominado por Internet of Things (IoT) [7]. Esta tecnologia permite que qualquer objecto, pessoa ou lugar esteja automaticamente associado com a sua informação online. Por exemplo, torna possível

utilizar um dispositivo como um *smartphone* ou um *tablet* PC para comprar um bilhete para o cinema, passando-o apenas perto do poster onde o filme está a ser anunciado.

Foram exploradas duas tecnologias para disponibilizar estas capacidades[10]: etiquetas ópticas como os códigos *Quick Response* (QR) e *Radio Frequency Identification* (RFID). A última é utilizada sob a forma de etiquetas electrónicas passivas, ou seja, uma etiqueta *transponder* que é activada pelo leitor por indução magnética.

2.2.3 *Radio Frequency Identification* (RFID)

Radio Frequency Identification (RFID) é uma tecnologia para identificação utilizando ondas de rádio [11] e que oferece meios para identificar itens aos quais uma etiqueta (*tag*) RFID está ligada. Com esse propósito é emitido um sinal de rádio para um determinado *transponder* que vai responder com outro sinal de rádio [26]. Como referido em 2.2.2 é uma das duas tecnologias de NFC.

Um sistema RFID é composto por 3 partes(Figura 2.6): Leitor (*Reader*), Etiqueta (*Tags*), Base de dados (BD).

Uma etiqueta é composta por uma antena para a emissão e recepção de dados e um chip para computar e armazenar informação. A informação contida na etiqueta pode ser lida ou escrita através de um leitor RFID [11].

No processo de comunicação entre os dispositivos RFID, os leitores (*transceivers*) interrogam as etiquetas (*transponders*) para acederem à informação que estas têm armazenada na memória. Posteriormente esta informação é enviada para uma BD que a usa como índice de procura. Assume-se que a ligação à BD é segura [12].

Esta tecnologia está em amplo crescimento, está inclusivamente a ser estudada como substituto dos códigos de barras actualmente utilizados. Esse crescimentos que está a ser um pouco atrasado devido aos riscos de segurança associados às comunicações sobre RFID [12].

2.2.4 Tipos de etiquetas RFID

Existem diversos tipos de etiquetas que suportam a comunicação RFID que podem ter vários tipos de características. Entre outras características podem variar as primitivas criptográficas implementadas, a capacidade de processamento, a memória, se são passivas ou activas, etc. Dependendo dessas características e funcionalidades o preço das etiquetas varia. É necessária uma análise exaustiva tanto aos tipos das etiquetas como aos requisitos de segurança para poder escolher uma etiquetas com uma boa relação preço/segurança. Como pode ser visto

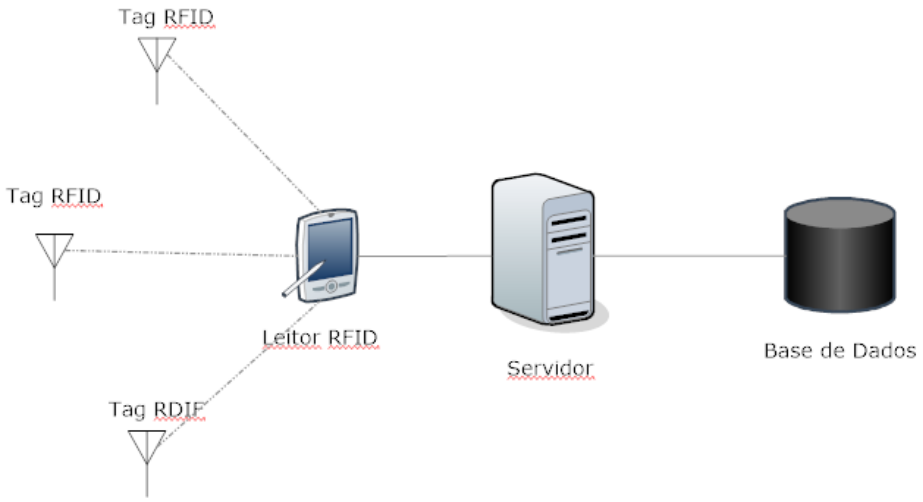


Figura 2.6: Arquitetura RFID

em [27] existem quatro classes nas quais as etiquetas podem ser classificadas. Na tabela 2.1 podem ser vistas as diferentes classes de etiquetas e que funcionalidades implementam. As funcionalidades apresentadas na tabela são cumulativas, ou seja, a classe 2 acrescenta às suas funcionalidades as apresentadas para a classe 1. O mesmo acontece para as restantes classes.

Classe	Tipo de Etiqueta	Características
Classe 1	Etiquetas de Identificação	*Electronic Product Code* (EPC), ID da etiqueta, memória opcional para o utilizador
Classe 2	Etiquetas de Identificação com mais funcionalidades	ID maior, Maior memória para o utilizador
Classe 3	Etiquetas passivas Assistidas por bateria	Fonte de energia, Registo de dados Opcional
Classe 4	Etiquetas activas	Comunicações sobre um transmissor autónomo

Tabela 2.1: Classes de Etiquetas RFID(Adaptado de [27])

2.2.4.1 Protocolo de interface aérea UHF *standard* classe 1 geração 2 (Gen 2)

Comummente conhecido como Gen 2 define os requisitos físicos e lógicos para um passive-backscatter, *Interrogator-talks-first* (ITF), sistema RFID a operar na banda de frequências entre os 860 e os 960 MHz.

Na versão actual do protocolo (V1.2.0) são acrescentadas três novas funcionalidades.

- Um indicador que permite mostrar se existem dados formatados na memória do utilizador.
- Protecção dos dados já escritos na memória do utilizador.
- Recolocação de uma etiqueta depois de uma operação de POS. Essa acção é indicada através da adição de bits de controlo no protocolo alargado.

2.2.5 Protocolos seguros para comunicação RFID

Tal como referido anteriormente (Capítulo 2.2.3), embora haja um grande crescimento da tecnologia RFID este está a ser um pouco reduzido devido aos riscos de segurança associados a estas comunicações.

A necessidade de garantir a autenticidade das partes envolvidas num processo de identificação RFID e a natureza critica da informação estão a encorajar o uso de criptografia apesar das limitações das etiquetas RFID. Recentemente foram desenvolvidos métodos que utilizam protocolos delimitadores de distância e utilizam normalmente funções pseudo-aleatórias e uma chave secreta partilhada. Mas o nível de segurança de um protocolo não depende apenas das primitivas criptográficas utilizadas, mas se um atacante consegue ou não entrar no sistema com sucesso [12].

Dos vários protocolos investigados vão ser descritos os mais interessantes tendo em conta: o nível de segurança apresentado, o desempenho e qual a classe de etiquetas que estes necessitam para ser implementados.

2.2.5.1 Protocolo RFID *Grouping Proof*

Um protocolo de RFID *grouping proof* permite a um leitor recolher a prova da presença de duas ou mais etiquetas em simultâneo e permitir depois que um verificador *offline* verifique essa informação. Neste sistema, tanto as embalagens dos medicamentos como os próprios pacientes têm que estar marcados com uma etiqueta RFID, assim o leitor pode verificar se o medicamento está a ser administrado ao doente certo [28].

Chien et al [28] propõe duas soluções, uma é baseada no protocolo de autenticação RFID e aplica-se a verificadores *online* e outra baseia-se no RFID *grouping protocol* e aplica-se a verificadores *offline*. Para um entendimento mais fácil das soluções apresentadas, vai denominar-se a pulseira que o paciente possui como *palete*. PIN_i e PIN_{palete} representam os códigos de identi-

ficação da Tag_i e da $Palete$, respectivamente. Os PIN, indicados, são segredos pré-partilhados entre o servidor (ou o leitor) e as etiquetas.

Verificador *online* Neste cenário a enfermeira tem na sua estação um computador portátil e um leitor RFID, ou os leitores RFID possuem as chaves secretas utilizadas para autenticar as etiquetas.

Como o verificador *online* interage com as etiquetas, pode executar qualquer protocolo de autenticação RFID para as autenticar dentro de uma janela de tempo predefinida e assim, obter dados de prova de *grouping*.

O protocolo proposto por [28] pretende utilizar apenas operações como CRC, *Pseudo-Random Number Generation* (PRNG) e simples operações ao bit.

Em seguida é apresentado o algoritmo proposto por [28] para verificadores *online*:

O leitor (L) começa o relógio.

1. $L \rightarrow Tag_i : pedido, N_R$. L escolhe um número aleatório N_R e envia como desafio a todas as etiquetas Tag_i vizinhas.
   para $i = 1$ até $n + 1$ (Tag_{n+1} é a palete das etiquetas)
   {

2. $Tag_i \rightarrow L : \text{EPC}_i, N_i, MACl_i$. A Tag_i escolhe um número aleatório N_i e responde com o valor, $MACl_i = \text{PRNG}(\text{EPC} \oplus \text{PRNG}(PIN_i) \oplus \text{PRNG}(N_R) \oplus \text{PRNG}(N_i))$
   O L verifica se i valor $MACl_i$ está correcto. Se sim aceita a etiqueta caso contrário termina o processo.

3. $L \rightarrow Tag_i : MAC2_i$. Se for necessária a autenticação do leitor, então L responde com, $MAC2_i = \text{PRNG}(R \oplus \text{PRNG}s(N_R) \oplus \text{PRNG}s(N_i))$. A Tag_i verifica se o valor $MAC2_i$ está correcto e aceita o leitor se tal acontecer.
   }

O leitor pára o relógio e verifica a associação $(Tag_1, Tag_2, ..., Tag_n, Palete)$ e se o tempo decorrido está dentro da janela de tempo predefinida. O principio base deste protocolo é que tanto as etiquetas como os leitores gerem os próprios desafios N_{ir} e esperam que o seu parceiro responda correctamente. O protocolo pode ser visto na figura 2.7.

Verificador *offline* Aqui como o verificador está *offline*, precisa de executar o protocolo de *grouping proof* para obter as provas. [28] propõe o protocolo de *grouping proof* que se segue.

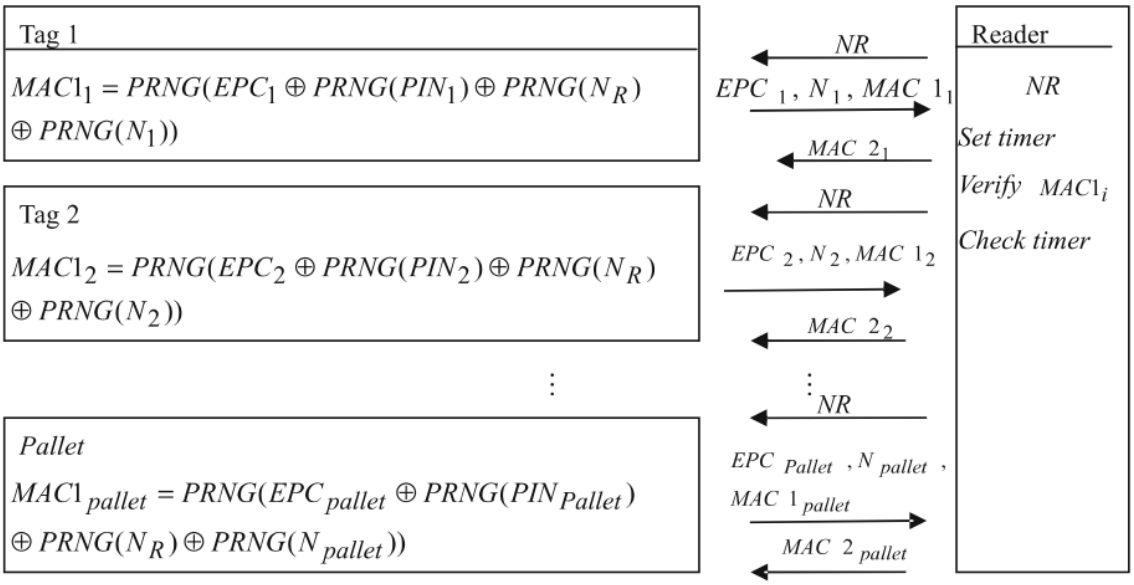


Figura 2.7: Verificador online do protocolo *Grouping Proof* (Retirado de[28])

1. Verificador → Leitor: $t = E_{K_V}(timestamp)$. O leitor obtém um *timestamp* cifrado $t = E_{K_V}$ $(timestamp)$ do verificador, onde $E_{K_V}(timestamp)$ é uma cifragem do *timestamp* com a chave secreta do verificador K_v.

2. $Leitor \rightarrow Tag_1, Palete : t$. O leitor envia o *timestamp* cifrado para a $Tag1$ e para a palete.

3. Para i=1, ..., n-1

   (a) $Tag_i \rightarrow Leitor : \text{EPC}_i, m_i$ Se $i = 1$, então $m_0 = t$. A Tag_i calcula $m_i = \text{PRNG}(\text{EPC} \oplus \text{PRNG}s(m_{i-1}) \oplus \text{PRNG}(PIN_i))$. A Tag_ienvia EPC_i e m_i para o leitor.

   (b) $Leitor \rightarrow Tag_{i+1} : m_i$. O leitor reenvia m_i para a próxima *tag* Tag_{i+1}.

4. (a) $Tag_n \rightarrow Leitor : \text{EPC}_n, m_n$. A Tag_n calcula $m_n = \text{PRNG}(\text{EPC} \oplus \text{PRNG}(m_{n-1}) \oplus \text{PRNG}(PIN_n))$. A Tag_n envia EPC_i e m_i para o leitor.

   (b) $Leitor \rightarrow Palete : m_n$. O leitor reenvia m_n para a palete.

   (c) $Palete \rightarrow Leitor : \text{EPC}_{Palete}, P$. Depois de receber m_n, a Palete calcula $m_{Palete} = \text{PRNG}(\text{EPC}_{Palete} \oplus \text{PRNG}(m_n) \oplus \text{PRNG}(PIN_{Palete}))$. A palete envia EPC_{Palete} e P para o leitor.

5. $Leitor \rightarrow Verificador : (t, \text{EPC}_1, m_1, ..., \text{EPC}_n, m_n, \text{EPC}_{Palete}, P)$. O leitor obtém a prova $(t, \text{EPC}_1, m_1, ..., \text{EPC}_n, m_n, \text{EPC}_{Palete}, P)$ e reencaminha para o verificador.

6. O verificador verifica se a associação ($EPC_1, ...EPC_n, EPC_{Palete}$) é verdade para a prescrição, se a prova $(m_1, ..., m_n, P)$ se verifica e se o *timestamp* decifrado $D_{K_v}(t)$ está dentro de uma janela de tempo razoável. Se as três condições se verificarem o protocolo *grouping proof* é terminado com sucesso. Na figura 2.8 pode ser visto o protocolo de *grouping proof* proposto para o um verificador offline.

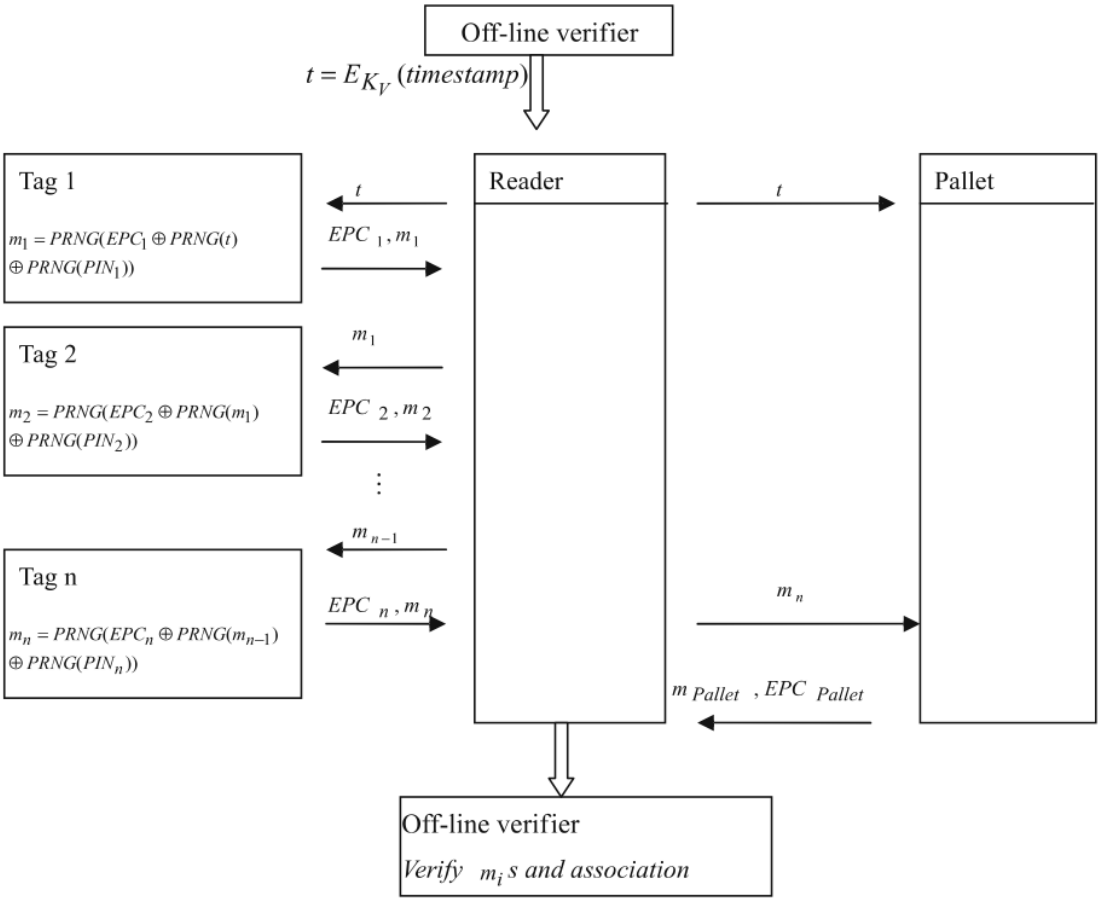


Figura 2.8: Verficador offline do protocolo *Grouping Proof* (Retirado de[28])

2.2.5.2 Puzzles Criptográficos (PC)

O método proposto por [12] permite de maneira simples proteger a privacidade e o rastreamento das etiquetas. Este método baseia-se no WSBC, e obriga à solução de um puzzle criptográfico.

Este método permite aumentar a segurança dos sistemas RFID fazendo com que os leitores tenham que efectuar um teste de esforço computacional. Os leitores têm que resolver um puzzle criptográfico para obter a identidade da etiqueta interrogada.

O objectivo é fazer com que os leitores que não queiram atribuir o tempo e o esforço computacional requerido para resolver o puzzle não irão aceder a nenhum dado que permita a identificação da etiqueta. Para isso, as etiquetas RFID geram puzzles que os leitores devem resolver ficando com a informação previamente cifrada e anonimizada em sua posse.

O problema do método referido é que tanto os leitores honestos como os outros vão ter que efectuar o mesmo esforço para resolver o puzzle criptográfico. [12] propõe duas soluções para este problema. Na primeira, a BD à qual o leitor está ligado pode delegar para leitores legítimos parte da chave secreta da etiqueta. A segunda solução baseia-se no aumento da dificuldade do puzzle com a distância entre o leitor e a etiqueta, tendo em conta que normalmente os leitores legítimos estão mais perto que os possíveis atacantes.

As duas soluções vão ser apresentadas mais à frente, para ambas vamos assumir que R e T são um leitor e uma etiqueta respectivamente, a ligação entre os leitores e a BD é segura e a informação a trocar entre as entidades é dada por ID (Identificador único da etiqueta). Da mesma forma, $enc_k(x)$ é um algoritmo de chave simétrica que cifra a mensagem x com a chave k e a concatenação entre duas variáveis é dada por $||$. O puzzle criptográfico enviado por T à j-ésima instância do protocolo, onde n é um número aleatório, é designado por $\varsigma = enc_k(n||ID||n||j)$. $w_j^k(k)$ representa uma função de WSBC, o autor sugere que para o id se utilize apenas $(\varsigma_j, w_j^{\Pi}(k))$. Assume-se também que $h(a||b)$ é uma função de *hash* que tem como entrada a e b, para este caso mais especificamente $v_j = h(j||n_1||ID||n_1)$ e representa o pseudónimo enviado a j-ésima entidade;

Esquema de autenticação WSBC Em seguida é apresentado o protocolo para autenticação WSBC proposto por [12] (Figura 2.9):

1. $R \rightarrow T : m_1 = request, n_1$

   R inicia o protocolo enviando um pedido a T que inclui um número aleatório n_1. n_1 serve também para manter a frescura da sessão tal como para seu identificador.

2. $T \rightarrow R : m_2 = n_2, \langle \varsigma_j, w_j^{\Pi}(k) \rangle, v_j, v_j^*$

   T gera um número aleatório n_2, um *commitment* $\langle \varsigma, w_j^{\Pi}(k) \rangle$ e um pseudónimo para o ID v_j formando uma mensagem m_2 que é por fim passada para o R.

   Aquando a recepção da mensagem, R obtém o PC ς_j que faz parte do *commitment* enviado por T, sendo o resto desse *commitment* o output da função WSBC $w_j^{\Pi}(k)$ que facilita a resolução do PC. Quando $w_j^{\Pi}(k)$ é recebido, l bit da chave secreta são passados para R [1].

[1] T pode enviar diferentes $w_j^{\Pi}(k)$ para R uma vez que para os formar são seleccionados l bits diferentes da

Finalmente é iniciado um processo de força bruta e R deve tentar em média $\binom{n}{l}2^{n-l-1}$ chaves (sendo n o número de bits da chave k a ser decifrada) para decifrar ς_j. Cada vez que é verificada uma nova chave, o sucesso da verificação é confirmado utilizando o pseudónimo do ID da etiqueta v_j incluído no fim da mensagem.

$$v_j \stackrel{?}{=} h(j||enc_k^{-1}(n_1||ID||n_1||j) >> p||n_2)$$

3. $R \rightarrow T_j : m_3 = n_4^*, \tau_j^*(Opcional)$

   R pode provar a Tque descobriu a solução para o PC. Para isso o leitor obtém o desafio n_3^* decifrando v_j^* e explorando o seu conhecimento da chave secreta k. Posteriormente R gera um número aleatório n_4^* computa τ_j^* e envia para T. T calcula então a sua versão de $\tau_j^* = enc_k(j||n_4||ID+1||n_3||n_1)$ e compara com o valor recebido, se for verificado com sucesso o processo de autenticação mútua está concluído.

Como referido anteriormente, com este esquema pode ser necessário a utilização da BD no processo de autenticação, para tal a BD deve guardar os pares $\{ID, k\}$. O leitor depois do passo 2 pode enviar certa informação privada contida pela etiqueta para a base de dados que confirma a sua veracidade autenticando ou não a etiqueta.

Esquema de autenticação WSBC com delimitador temporal Este protocolo assume que os leitores honestos deverão estar mais próximos que os desonestos, logo envia puzzles mais difíceis conforme o aumento da distancia entre o leitor e a etiqueta. O protocolo proposto por [12] (Figura 2.10) é descrito em seguida:

1. $R \rightarrow Tm_1 request, n_1, \gamma_j$

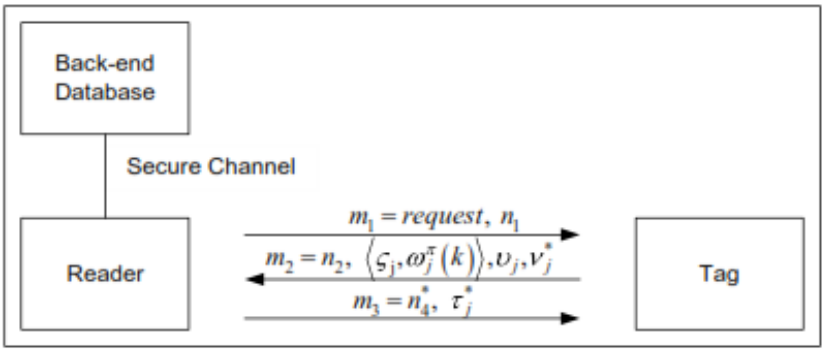
   R gera um valor aleatório s_j com t bits e compromete o seu valor enviando um um número n_1 e uma mensagem γ_j.

2. T e R começam uma troca de nível baixo de limitação de distância. São repetidos os seguintes passos t vezes.

   for i = 1, ..., t

   - R envia o bit $c(i)$ para T de modo a energizar a etiqueta.
   - T envia o bit $\alpha_j(i)$ para R.
   - R envia o bit $\beta_j(i) = \alpha_j(i) \oplus s_j(i)$ para T imediatamente após a recepção de $\alpha_j(i)$.

chave total aleatoriamente



1. $\mathcal{R} \rightarrow \mathcal{T}$: $m_1 = request, n_1$
2. $\mathcal{T} \rightarrow \mathcal{R}$: $m_2 = n_2, \langle \varsigma_j, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j^*$
3. $\mathcal{R} \rightarrow \mathcal{T}$: $m_3 = n_4^*, \tau_j^*$ (*Optional)

where $\{n_i\}_{i=0}^{4}$ are different *nonces*
$\varsigma_j = enc_k(n_1||ID||n_1||j)$
$\omega_j^{\pi}(k) = \{k_{\pi(0)}, k_{\pi(1)}, \ldots, k_{\pi(l-1)}\}$ is a *l-bit WSBC function* and $\pi()$ is a given permutation
$\upsilon_j = h(j||n_1||ID||n_2)$
$\nu_j^* = enc_k(j||n_3||ID||n_1)$ (Optional)
and $\tau_j^* = enc_k(j||n_4||ID+1||n_3||n_1)$ (Optional)

Figura 2.9: Esquema de autenticação WSBC (Retirado de[12])

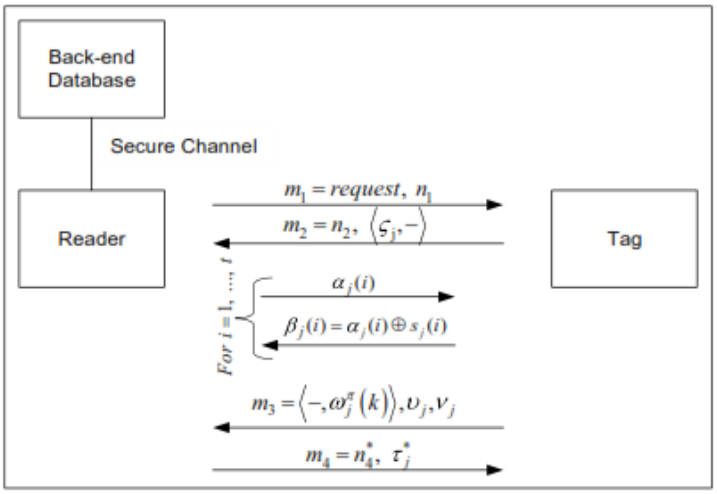
- Depois de completada a troca rápida de bits, R abre o *commitment* do valor oculto s_j enviado $\{n_2, s_j\}$.
- T consegue determinar o limite superior da distância d_{rt} utilizando o máximo dos atrasos entre enviar o bit $\{\alpha_j(i)\}$ e receber o bit $\{\beta_j(i)\}$.

3. $T \rightarrow R : m_2 = n_3, \langle \varsigma, w_j^{\Pi}(k) \rangle, u_j, v_j$

   A etiqueta gera um novo *nonce* n_3 e calcula WSBC$\langle \varsigma, w_j^{\Pi}(k) \rangle$ que depende da distancia d_{rt}. Mais precisamente a variavel l é condicionada pela distancia $\{l = f(d_{rt})\}$. Por fim a mensagem m_2 é terminada com uma mensagem de autenticação v_j

4. $R \rightarrow T : m_3 = n_5, \tau_j$

   R envia para T o *nonce* n_5 e a mensagem cifrada τ_j que um duplo propósito: a etiqueta poder autenticar o leitor e a etiqueta poder verificar que as mensagens durante a troca rápida de bits não foi alterada por um elemento externo.



1. $\mathcal{R} \rightarrow \mathcal{T}$: $m_1 = request, n_1$
2. $\mathcal{T} \rightarrow \mathcal{R}$: $m_2 = n_2, \langle \varsigma_j, - \rangle$
3. Distance-bounding protocol
   For $i = 1, \ldots t$
   $\mathcal{R} \rightarrow \mathcal{T}$: $\alpha_j(i)$
   $\mathcal{T} \rightarrow \mathcal{R}$: $\beta_j(i) = \alpha_j(i) \oplus s_j(i)$
4. $\mathcal{T} \rightarrow \mathcal{R}$: $m_3 = \langle -, \omega_j^{\pi}(k) \rangle, \upsilon_j, \nu_j$
5. $\mathcal{R} \rightarrow \mathcal{T}$: $m_4 = n_4^*, \tau_j^*$ (*Optional)

where $\{n_i\}_{i=0}^{4}$ are different *nonces*
$\varsigma_j = enc_k(n_1||ID||s_j||n_1||j)$
$\omega_j^{\pi}(k) = \{k_{\pi(0)}, k_{\pi(1)}, \ldots, k_{\pi(l-1)}\}$ is a *l-bit WSBC function* and $\pi()$ is a given permutation
$\upsilon_j = h(j||n_1||ID||s_j||n_2)$
$\nu_j = enc_k(j||n_3||ID||\alpha_j||\beta_j||n_1)$
and $\tau_j^* = enc_k(j||n_4||ID+1||n_3||n_1)$ (Optional)

Figura 2.10: Esquema de autenticação WSBC com protocolo delimitador de tempo (Retirado de [12])

2.2.5.3 *Inpatient Safety RFID (IS-RFID)*

Inpatient Safety RFID (IS-RFID) [11] é um sistema com base nos protocolos de *grouping proof*. No protocolo proposto por [11] as etiquetas estão ligadas aos pacientes (por exemplo, pulseiras) e aos recipientes da medicação (por exemplo, frascos de plástico marcados). Os leitores RFID lêem o identificador estático de cada etiqueta, os quais podem ser posteriormente utilizados como índice de procura numa BD. Para tal, assume-se que o leitor tem uma ligação segura à BD.

É assumido que são usadas etiquetas segundo o *standard* Gen 2, ou seja, são passivas, têm uma palavra passe de 32 bits e suportam uma função de *Pseudo-Random Number Generation (PRNG)* de 16 bits.

Segundo Periz-Lopez et al. [11] existem quatro procedimentos que devem ser seguidos de forma a garantir a segurança na administração da medicação. Estes serão descritos de seguida juntamente com uma pequena análise ao custo, performance e segurança do sistema.

Empacotamento

Um médico visita um paciente e diagnostica-o. Para isso lê a etiqueta ligada ao paciente através de um PDA com um leitor RFID, obtendo o identificador estático a ele associado, efectua o diagnóstico e emite uma nova prescrição. Depois de visitar os seus pacientes ele vai ao seu escritório e liga o PDA ao seu computador para registar as prescrições no sistema informático do hospital. O sistema informático do hospital informa a farmácia que começa a empacotar os medicamentos em doses unitárias de acordo com as ordens recebidas. Aqui pode ser utilizado um protocolo de *grouping proof* de forma a poder ser gerada uma prova de que os medicamentos foram introduzidas em simultâneo nas embalagens. É então gerado um identificador único(UD_i) para cada dose unitária que é escrito numa etiqueta passiva anexada à embalagem.O sistema informático do hospital fica com a entrada representada na Figura 2.11 para o paciente.

$Paciente_i$	UD_i	$Informacao - adicional_i$

Figura 2.11: Ligação paciente - dose unitária (Retirado de[11])

Estação de Enfermagem

Na estação de enfermagem uma enfermeira liga-se ao sistema e pede ao sistema informático do hospital os medicamentos que devem ser administrados num determinado piso, num determinado período de tempo. O sistema informático envia os tuplos {$Paciente_i$,UD_i,t_i, $Informacao - adicional_i$}. O elemento t_i representa um *timestamp* que será válido durante um determinado período de tempo, este é previamente especificado e registado no sistema informático. Ou seja, o medicamento UD_i tem que ser administrado ao $Paciente_i$ dentro dessa janela. Finalmente as enfermeiras transferem estes dados (Figura 2.12) para os seus PDAs e podem começar as rondas. Com este último passo é possível efectuar o próximo passo online.

$Paciente_1$	UD_1	$Informacao - adicional_1$
$Paciente_i$	UD_i	$Informacao - adicional_i$
$Paciente_n$	UD_n	$Informacao - adicional_n$

Figura 2.12: Informação guardada nos PDAs das enfermeiras (Retirado de [11])

Administração segura de medicamentos

Este processo foi desenhado para diminuir os erros humanos na administração de medicamentos fazendo com que a enfermeira possa confiar na ligação entre o paciente e os medicamentos correspondentes.

Cada enfermeira possui um PDA que funciona como um leitor RFID e dispositivo local. Este processo é dividido em duas fazes, cada uma delas divididas em vários passos como descrito em seguida:

- Processo de Verificação

  1. O leitor RFID gera um número aleatório r_p e envia um pedido $\{request,r_p\}$ para a etiqueta ligada ao paciente e para as etiquetas ligadas às embalagens dos medicamentos.
  2. Cada uma destas verifica o pedido e envia um identificador anónimo para o leitor.
     - A etiqueta do paciente calcula e envia $\{r_w,\text{PRNG}(Paciente_i,r_p,\ r_w)\}$
     - A etiqueta da embalagem calcula e envia $\{r_m,\text{PRNG}(UD_i,r_p,r_m)\}$

     Sendo r_w e r_m números aleatórios gerados pela etiqueta do paciente e da embalagem do medicamento respectivamente.
  3. O leitor recebe os dois valores e começa um processo de procura sobre os registos guardados no PDA. O par $\{Paciente_1,UD_1\}$ é obtido e o PDA gera uma versão local dos identificadores anónimos: $\text{PRNG}(Paciente_i,r_p,r_w)$ e $\text{PRNG}(UD_i,\ r_p,\ r_m)$. Se os valores computados forem iguais aos valores recebidos é mostrada uma mensagem de confirmação no ecrã e o medicamento pode ser administrado com segurança. Caso contrário é repetido o processo com $\{Paciente_2,UD_2\}$ até obter uma igualdade. Caso tal não aconteça a enfermeira para o processo de administração e investiga o problema.

- Geração de uma prova da administração do medicamento - As etiquetas do paciente e das embalagens dos medicamentos guardam um identificador e uma chave na sua memória: $\{Paciente_i,K_{paciente_i}\}$ e $\{K_{UD_i}\}$ respectivamente. Em seguida são descritas as mensagens trocadas entre as três entidades.

  1. O leitor RFID da enfermeira interroga a etiqueta do paciente utilizando o *timestamp* $\{t_i\}$ guardado no registo correspondente do PDA.

2. A etiqueta do paciente gera um número aleatório r'_w e calcula $m_t = \text{PRNG}\{Paciente_i, \oplus, \text{PRNG}(t_i), \text{PRNG}(Kpaciente_i)\}$ e envia $\{r'_w, m_t\}$ para o leitor.
3. O leitor guarda r'_t e envia m_t para a etiqueta da embalagem.
4. A etiqueta da embalagem gera um número aleatório r'_m e calcula $m_{UD} = \text{PRNG}(UD_i \oplus r'_m \oplus \text{PRNG}(m_t) \oplus K_{UD_i})$ e envia $\{r'_m, mUD_i\}$ para o leitor.
5. O leitor guarda r'_m e envia m_{UD} para a etiqueta do paciente.
6. A etiqueta do paciente calcula $m_{TUD} = \text{PRNG}(Paciente_i \oplus m_t \oplus \text{PRNG}(m_{UD}) \oplus K_{Paciente_i})$ e envia o resultado para o leitor.
7. O leitor gera a prova, $e_i = \{Paciente_i, UD_i, t_i, r'_w, r'_m, m_{TUD}\}$.
8. A intervenção da enfermeira (por exemplo: autenticação por *password*) é necessária para ser gerada uma assinatura digital da prova ($sign(e_i)$). Finalmente é gravado no registo correspondente a prova e a assinatura desta $\{e_i, sign(e_i)\}$.

Como resultado dos processos descritos anteriormente é guardada a seguinte informação no PDA da enfermeira 2.13

$Paciente_1$	UD_1	t_1	$\{e_i, sign(e_1)\}$	$Informacao - adicional_1$
$Paciente_i$	UD_i	t_i	$\{e_i, sign(e_i)\}$	$Informacao - adicional_i$
$Paciente_n$	UD_n	t_n	$\{e_i, sign(e_n)\}$	$Informacao - adicional_n$

Figura 2.13: Informação guardada nos PDAs das enfermeiras depois de todos os procedimentos (Retirado de [11])

Monitorização

Na estação da enfermagem, a enfermeira liga-se ao sistema, informando-o de seguida da administração do medicamento. São transferidos os dados guardados no PDA para a base de dados do sistema. O sistema informático verifica a validade das provas e verifica se os medicamentos foram administrados dentro de uma janela de tempo especificada. Se existiu algum erro é gerado um alarme.

Análise de segurança do algoritmo criptográfico

O protocolo IS-RFID como descrito anteriormente utiliza uma função PRNG e efectua operações XOR ao bit. A probabilidade de efectuar um ataque de força buta em tempo real durante ambas as fazes é de $1/2^{16}$, uma vez que a função de PRNG suportada pelas etiquetas utilizadas é de 16 bits. A fase do processo de verificação e a fase da geração da prova são independentes,

então a probabilidade de quebrar todo o sistema é de $1/2^{32}$. Periz-Lopez et al. [11] sugere que, para aumentar a segurança total do sistema, que se utilizem primitivas criptográficas mais fortes do que a funçao PRNG de 16 bits, ou então se liguem as duas fases, fazendo com que a entrada da fase B seja o *output* da fase A - {PRNG($t_i \oplus v_T \oplus v_{UD}$)}.

2.2.5.4 *Weakly Secret Bit Commitment* (WSBC)

Um bit *commitment* é um meio de exigir que uma entidade se comprometa a um valor, mantendo-o escondido até o poder revelar mais tarde [12]. Por exemplo, a Alice gera duas *strings* de bits aleatórias {R_1, R_2} e compromete-se a uma mensagem M fazendo h($R_1||R_2||$M) e enviando $R1$,h($R_1||R2||$M) para o Bob.

Quando a Alice quiser revelar a mensagem ao Bob ela envia {R_2,M}[29]. Pelas propriedades das funções de *hash* [12]:

- O Bob não consegue determinar a mensagem M apenas pela primeira parte da mensagem.
- A Alice não consegue encontrar um para {R_2',M'} diferente tal que, $h\{R_1||R_2||\text{M}\} = h\{R_1||R_2||\text{M'}\}$.

As funções WSBC funcionam segundo o mesmo principio, com a diferença que para estas é possível descobrir o segredo do bit *commitment* depois de um limite predefinido aceitável em termos de tempo e/ou computação [29].

Uma função WSBC tem as seguintes propriedades [29]:

1. 2^{nd}-*preimage resistance*: Dado um x, deve ser computacionalmente inviável encontrar um $x' \neq x$ tal que $w(x) = w(x')$;
2. *Weak-preimage resistance*: Para qualquer valor pré-especificado y de w deve ser moderadamente difícil calcular um x tal que $y = w(x)$;
3. *Collision resistance*: Deve ser computacionalmente inviável encontrar um x, x' tal que $w(x) = w(x')$;
4. *Near-premiage resistance*: Dado $y = w(x)$ deve ser difícil encontrar um x' tal que x e x' difiram em poucos bits;

As duas ultimas propriedades são apenas necessárias se os requisitos da aplicação assim o ditarem.

2.3 Base de dados (BD)

Uma Base de dados (BD) é qualquer colecção de dados ou informação especialmente organizada para ser possível efectuar uma rápida procura por um computador. Estas são feitas para facilitar o armazenamento, busca, alteração e remoção de dados, ou seja gerir e manipular informação estruturada [30].

Os dados são geralmente guardados como registos compostos de campos de um determinado tipo (inteiro, texto, data). Um grupo de registos com uma estrutura de campos idêntica é chamado de tabela [31].

Uma BD é então, uma colecção de dados nas tabelas que a compõe e o programa que organiza e gere a BD é um *Data Base Management System* (DBMS) [30].

As BD com tabelas ligadas e relacionadas com outras tabelas são chamadas de BD relacionais [30].

As BDs são uma das partes mais importantes de um sistema, uma vez que se estas ficarem comprometidas todos os dados guardados podem ir parar às mão dos atacantes [32].

Uma das protecções mais utilizadas são as *firewalls*, estas situam-se entre a rede interna de uma organização e a internet e monitorizam todo o trafego que passa para a rede interna bloqueando o que não for autorizado. Embora estas sejam uma boa protecção para bloquear ataques vindos da Internet devem ser apenas vistas como uma primeira linha de defesa uma vez que, se forem penetradas não fornecem qualquer tipo de protecção aos recursos internos [33].

Para proteger os dados de uma BD é necessário satisfazer os seguintes requisitos [33][32]:

- Identificação e autorização: O sistema deve conseguir identificar todos os seus utilizadores e deve também poder confirmar a sua identificação.
- Controlos de acesso: Mantém uma separação entre os utilizadores e os recursos de computação, protegendo os recursos internos de acessos ou modificações não autorizadas.
- Cifragem: Assegura que todos os dados enviados para a rede só possam ser vistos pelo receptor a que foram destinados.
- Integridade: Assegurar que os dados numa BD são de facto válidos e correctos. Seja qual for a importância dos dados guardados numa base de dados se estes não estiverem correctos a BD não tem qualquer utilidade.

- Controlo de modificações: devem ser registadas todas as alterações feitas à base de dados estruturais ou sejam alterações de dados.

Um dos DBMS *freeware* mais utilizados hoje em dia é o MySQL. Este utiliza como linguagem o *Structured Query Language* (SQL) e fornece mecanismos que permitem satisfazer os requisitos acima referidos [32]. Para tal é possível cifrar os dados com as cifras AES ou DES, utilizar as funções de *hash*, *Message-Digest algorithm 5* (MD5) ou *Secure Hash Algorithm* (SHA1). É possível também utilizar SSL ou *Secure Shell* (SSH) para garantir a segurança entre a DBMS e a aplicação MySQL cliente [34].

2.4 Internet

A Internet revolucionou o mundo da informática e comunicações como nada antes, sendo um mecanismo para a disseminação de informação, um meio de colaboração e interacção entre as pessoas e os seus computadores sem preocupações com a posição geográfica [35].

Nasceu durante a década de 1960 num ambiente militar para criar uma rede resistente a uma guerra globa. Essa rede teria a designação de ARPANET. O objectivo era distribuir o controlo pela rede de forma a que se um elemento fosse destruído ainda seria possível a comunicação entre os restantes elementos [36]

O protocolo que serve de base ao encaminhamento na internet é o *Internet Protocol* (IP) [37] e que possibilita o transporte de um pacote pela rede sem haver qualquer preparação para isso. As redes IP apresentam as seguintes características:

- Não são orientadas à ligação
- Não há garantias de fiabilidade para as camadas superiores.
- Não oferecem correcção de erros ou controlo de fluxo.
- O encaminhamento dos pacotes pode ser, *unicast* [2], *broadcast* [3] ou *multicast* [4].

Nas redes IP estão previstos dois protocolos, o TCP e o UDP. O protocolo *Transport Control Protocol* (TCP) [38] é orientado à conexão e dá garantias de entrega dos pacotes enquanto que, o *User Datagram Protocol* (UDP) [39] [36] embora consiga atingir menores atrasos, não é orientado à conexão, nem dá garantia de entrega dos pacotes.

[2] Um único destino
[3] Todos
[4] Vários destinos escolhidos

Como referido anteriormente, rede de computadores não é nada mais do que vários computadores interligados utilizando diferentes modelos de rede. Existem dois modelos de rede mais utilizados [40]: OSI, TCP/IP [5].

2.4.1 Modelo *Open Systems Interconnection* (OSI)

Segundo [36] o modelo OSI foi criado para poder dividir o problema complexo que é a comunicação entre aplicações informáticas residentes em diferentes máquinas, em vários problemas de menor complexidade (daí as várias camadas). As camadas não têm os protocolos definidos mas as funções que cada uma deve desempenhar. Cada camada utiliza os serviços da camada inferior e presta serviços á camada superior. Os protocolos que existem em cada camada definem as regras de comunicação entre as camadas correspondentes de equipamentos diferentes.

O modelo OSI é constituido por 7 camadas [36]:

- **Física** - define as características físicas e eléctricas da rede.
- **Ligação lógica** - garante a ligação fiável em cada um dos troços em que se divide uma ligação completa. Tem mecanismos de detecção de erros, numeração de tramas, etc.
- **Rede** - fornece processos para o estabelecimento, manutenção e terminação de ligações de rede. Assegura o endereçamento e encaminhamento correcto da informação até ao destino.
- **Transporte** - existe apenas nos equipamentos terminais e garante às camadas de aplicação uma ligação segura com o equipamento de destino. Pode incorporar mecanismos de detecção de erros e repetição de informação.
- **Sessão** - controla a sessão de dialogo entre aplicações. Pode ser visível ao utilizador solicitando introdução de identificação e senha de acesso.
- **Apresentação** - processos de cifra ou transcodificação de dados.
- **Aplicação** - interfaces com as aplicações.

O modelo OSI fornece uma arquitectura de rede completa e robusta, mas é difícil de adaptar à mudança e à realidade. No entanto se devidamente ajustado pode gradualmente vir a ser usado em grande escala.

[5] Embora sejam na verdade duas famílias de protocolos diferentes, o protocolo TCP e o protocolo *Internet Protocol* (IP), como são normalmente utilizados em conjunto são referidos como o modelo TCP/IP.

2.4.2 Modelo TCP/IP

Embora seja muito diferente do modelo OSI, o modelo TCP/IP apresenta apenas 4 camadas, são ambos semelhantes ao nível das camadas de transporte e de rede. A correspondência entre as camadas dos dois modelos pode ser vista na Figura 2.14.

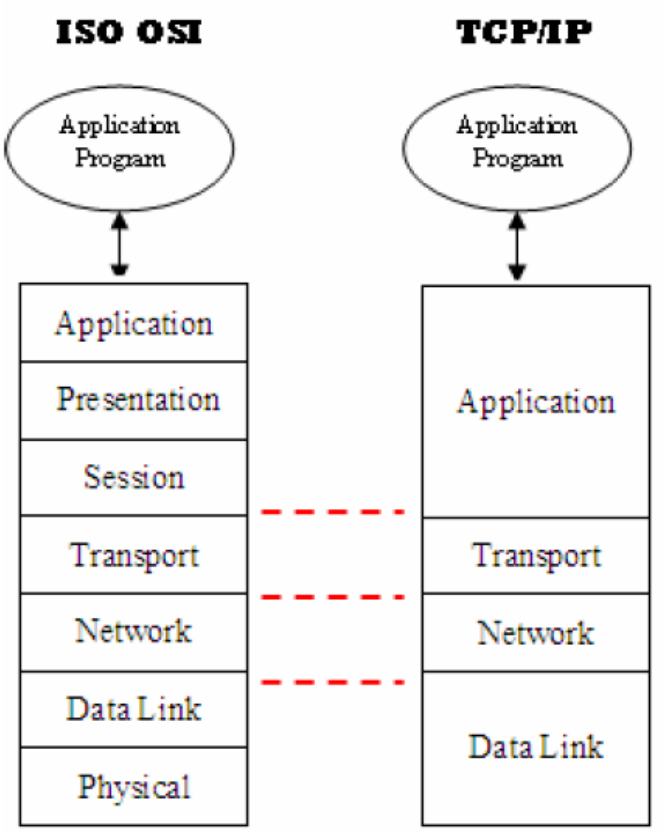


Figura 2.14: Comparação entre o modelo OSI e o modelo TCP/IP (Retirado de [40])

2.4.3 Protocolos seguros para comunicação sobre a Internet

Como visto anteriormente (Capítulo 2.4) existem dois tipos de modelos que permitem a comunicação entre as diversas entidades de uma rede de computadores, o modelo OSI e o modelo TCP/IP. O modelo OSI é pouco utilizado devido à sua complexidade, já o modelo TCP/IP é utilizado em grande escala, sendo um dos modelos mais utilizados na internet. Este será escolhido para um estudo mais aprofundado, averiguando como como se poderá segurar cada uma das suas camadas, uma vez que, qualquer uma poderá ser vitima de ataques [40].

Pretende-se que a ligação seja segura em todo o percurso. Desta forma, serão mais focadas as camadas que permitem assegurar segurança fim a fim. As camadas referidas são a camada

de rede, a camada de transporte e a de aplicação [36]. De realçar que durante a fase de pesquisa não foram encontrados quaisquer protocolos seguros para a camada de aplicação normalizados.

Protocolos de segurança para a comunicação sobre a Internent para a camada de rede

A protecção ao nível da rede fornece serviços que permitem aos utilizadores pensar nesta como uma rede segura. Uma técnica usual para tornar a rede segura, no caso das redes orientadas a pacotes, é a encapsulamento de datagramas [41], associado a técnicas de cifragem.

Nesse sentido um dos protocolos mais utilizados é o *Intenet Protocol Security* (IPsec) [42]. Este pode garantir a segurança a qualquer protocolo que funcione sobre a pilha IP. Introduzindo um certo *overhead* que diminui o *throughput* total da rede.

O IPsec verifica a origem dos pacotes IP prevenindo ataques de repetição e oferece ainda serviços de autenticação, integridade e cifragem de dados. Este pode funcionar em dois modos (Figura 2.15): modo túnel e transporte.

O modo de transporte assegura as comunicações seguras fim-a-fim cifrando toda a mensagem passada para o nível de rede. O modo de túnel é utilizador para proteger as comunicações entre duas redes de confiança.

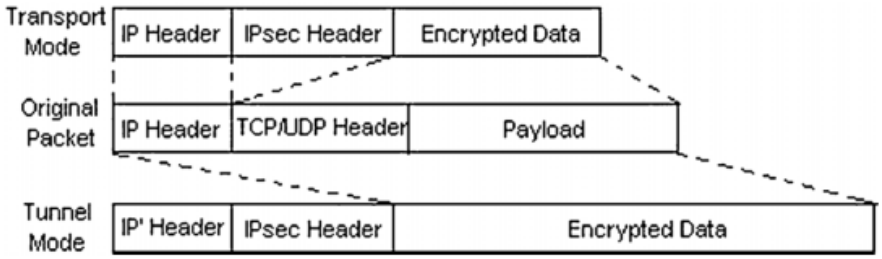


Figura 2.15: Tipos de Pacotes IPsec (Retirado de [42])

Protocolos de segurança para a comunicação sobre a Internet para camada de transporte

O protocolo utilizado como padrão para comunicações na internet a nível da camada de transporte é o protocolo *Secure Sockets Layer* (SSL) [20]. Embora este não seja o único que permite segurar a camada de transporte, este é o mais utilizado na Internet. Assim, vai ser este protocolo que vai ser aqui estudado.

Para a fácil implementação deste protocolo os *sockets* TCP/IP comuns são substituídos

por um *socket* SSL, diminuindo o tempo de desenvolvimento em contraste com o tempo necessário para construir e incorporar os componentes criptográficos necessários para garantir a mesma segurança que este [20].

O protocolo SSL inclui dois sub-protocolos: O SSL *record Protocol*, que define o formato das mensagens a ser trocadas, e o SSL *handshake Protocol*, que utiliza o sub-protocolo anterior para trocar uma série de mensagens entre um cliente e um servidor.

Segundo [20] a troca de mensagens efectuada durante o SSL *handshake Protocol* permite efectuar as seguintes tarefas:

- Autenticar o servidor perante o cliente.
- Negociar algoritmos criptográficos ou cifras que ambos suportem.
- Opcionalmente autenticar o cliente perante o servidor.
- Utilizar técnicas de chave pública para gerar segredos partilhados.
- Estabelecer uma ligação SSL.

O protocolo SSL suporta muitos tipos de cifras ou algoritmos criptográficos que podem ser utilizados em diversas operações como: autenticação do cliente e servidor, transmitir certificados ou estabelecer chaves de sessão. Durante a fase *handshake* do protocolo os mecanismos criptográficos são negociados de forma a serem suportados por cliente e servidor. Em seguida vão ser descritos os passos referentes à parte *handshake* do protocolo SSL [20].

1. O cliente envia para o servidor a sua versão SSL, definições de cifras, dados gerados aleatoriamente e informação adicional que o servidor necessita para comunicar utilizando SSL.
2. O servidor envia para o cliente a sua versão SSL, definições das cifras, dados gerados aleatoriamente e informação adicional que o cliente necessita para comunicar utilizando SSL.
3. O cliente utiliza parte da informação recebida para autenticar o servidor. Se este não poder ser autenticado o cliente é avisado de que não será possível iniciar uma sessão cifrada e autenticada. Se o servidor for autenticado o cliente segue para o próximo passo.
4. Utilizando os dados gerados no *handshake* o cliente cria uma chave pré-mestra para a sessão, cifra-a com a chave pública do servidor e envia-a.

5. (Passo opcional) Se o servidor pediu a autenticação do cliente, o cliente assina uma mensagem que é única para este *handshake* e conhecida pelo cliente e servidor. Neste caso o cliente envia a mensagem assinada, o seu certificado e a chave pré-mestra cifrada para o servidor.

6. Se foi pedida a autenticação do cliente pelo servidor, o servidor tenta autenticar o cliente. Se este não poder ser autenticado a sessão termina. Caso contrário, o servidor utiliza a sua chave para decifrar a sua chave pré-mestra começando uma série de passos (passos efectuados também pelo cliente) de forma a gerar a chave secreta mestra.

7. O cliente e o servidor utilizam a chave mestra para gerar chaves. Estas são chaves simétricas utilizadas tanto para cifrar e decifrar mensagens como para verificar a sua integridade durante a sessão SSL.

Um certificado SSL deve conter: o domínio para o qual foi emitido, o dono do certificado, a localização física do dono, as datas de validade do certificado.

De forma a evitar aparências de parcialidade com qualquer empresa particular as versões mais recentes do protocolo SSL foram renomeadas para *Transport Layer Security* (TLS).

Capítulo 3

Análise de Requisitos e Fragilidades

O principal objectivo desta dissertação é a elaboração de uma arquitectura de segurança para sistemas de prestação de cuidados médicos em mobilidade. Antes de se passar ao desenho da arquitectura é necessário efectuar uma análise dos requisitos de segurança, tendo em conta o contexto onde esta será aplicada. Esta arquitectura tem por base o desenho apresentado no cenário de aplicação (Figura 1.1), que deverá ser também analisado de forma a encontrar os seus pontos de fragilidade. Assim, será possível seleccionar das tecnologias descritas no estado da arte as que melhor se adequam à arquitectura a desenhar, e conseguir desse modo alcançar a segurança desejada.

3.1 Análise dos Requisitos de Segurança

O sistema a ser desenvolvido tem requisitos de segurança muito específicos uma vez que o ambiente onde está inserido, a saúde, assim o obriga. Devido à natureza sensível desse ambiente, é necessário negar o acesso não autorizado a informação que não seja pública e guardar um histórico de acessos que detecte qualquer falha no sistema [26].

Quando informação médica é disponibilizada através de uma aplicação informática torna-se mais tentador quebrar a segurança, devido à facilidade de acesso a quantidades massivas de informação dos pacientes. Os erros na segurança ou integridade dos dados tornam-se ainda mais devastadores, uma vez que os humanos possam não estar sempre disponíveis para filtrar ou rever os dados [4][5][43].

Em [3] pode-se ler os requisitos de segurança para uma rede *Wireless Body Area Network* (WBAN). Embora estes não se possam aplicar directamente à arquitectura a ser desenhada, alguns destes podem ser adaptados. Os requisitos de segurança do sistema podem ser complementados com os que podem ser inferidos a partir de uma análise do que foi descrito nesta

dissertação, principalmente o capítulo sobre criptografia (2.1.1) e o capítulo sobre segurança na BD (2.3). Devem ser implementados no mínimo os objectivos fundamentais de segurança que necessitam de ser complementados, de forma a atingir o nível de segurança pretendido. Através da análise efectuada foi possível retirar os seguintes requisitos de segurança:

- Confidencialidade: Toda a informação privada ou que permita deduzir informação privada deve ser devidamente cifrada. Esta não deve ser entendida pelas entidades a quem não é destinada;
- Integridade: Não deve ser possível alterar dados armazenados ou trocados na rede por meios não autorizados, ou seja, se a informação for alterada o destinatário deve se aperceber de tal alteração;
- Autenticação: Deve ser possível a qualquer uma das entidades verificar e confirmar a autenticidade de qualquer outra entidade, tal como a de qualquer mensagem recebida;
- Não repúdio: Não deve ser possível a qualquer uma das entidades recusar uma operação ou o envio de uma mensagem;
- Controlo de Acesso: Diferenciar o tipo de acesso de cada de utilizador de forma a proteger os dados de alterações não autorizadas.
- Controlo de mudanças: deve ser mantido um registo de todos os eventos e alterações efectuadas ao sistema;
- Disponibilidade: Assegurar o acesso ininterrupto a todas as entidades.

3.2 Análise dos pontos de Fragilidade do cenário de utilização

Segundo o desenho elaborado para o cenário de utilização podem ser identificados diversos pontos de fragilidade. A arquitectura proposta pretende reduzir ou mesmo eliminar essas fragilidades.

No desenho podem ser identificadas as seguintes vulnerabilidades:

- Comunicações:
  - Sobre a Internet;

  - RFID;

- Ligação à base de dados;
- Armazenamento de dados:
  - BD;
  - Do dispositivo para consulta offline;
- Autenticação:
  - Dos utilizadores perante a aplicação;
  - Entre as diversas aplicações;
  - Perante a BD.

Durante o estado da arte foram descritos diversos protocolos para a comunicação entre as diversas entidades e sobre os diversos meios. Nas próximas secções, será feita uma breve análise a esses protocolos de forma a escolher quais os mais adequados para atacar os pontos de fragilidades apontados.

Devido às fortes componentes de segurança impostas pelo sistema, é necessário proteger todas comunicações entre as aplicações e também os dados quando armazenados. Dependendo da utilização da aplicação, estes poderão ser armazenados tanto na BD como nos dispositivos.

Como foi visto no estado da arte, se for utilizado como DBMS o MySQL, este oferece mecanismos de segurança para os dados guardados na BD. Assim, é apenas necessário uma análise aos tipos de cifras e conexões que oferecem melhor segurança.

Quanto aos dados guardados nos dispositivos, durante a pesquisa não foram encontrados quaisquer métodos normalizados. Foi decidido então propor um método para assegurar que os dados são mantidos seguros. Esse método será descrito mais à frente e será denominado de *M-Health Offline User Acess* (MHOUA).

3.3 Protocolos seguros para a comunicação sobre a internet e autenticação das aplicações

Para obter uma comunicação segura sobre a internet, a segurança pode ser aplicada em diferentes camadas da pilha TCP/IP. No estado da arte foram abordados alguns protocolos para as diferentes camadas tendo em conta que, para este caso, apenas interessam protocolos

que proporcionam segurança fim a fim. Com isso em mente, foram apenas vistos protocolos para as camadas de rede, transporte e aplicação.

Para a camada de rede foi visto o protocolo IPsec. O protocolo IPsec pode não funcionar em todos os dispositivos o que, tendo em conta que os dispositivos móveis poderão também fazer parte da arquitectura, poderia acarretar problemas de portabilidade.

Para efectuar comunicações a nível da camada de transporte, foi analisado o protocolo SSL(ou TLS para versões mais recentes do protocolo SSL). Este protocolo é muitas vezes utilizado pelo protocolo *HyperText Transfer Protocol Secure* (HTTPS), versão segura do protocolo *HyperText Transfer Protocol* (HTTP), utilizado em grande escala para aplicações *web*. Este já deu provas de ser seguro e fiável e está sobre constantes actualizações.

Embora o protocolo SSL seja muito utilizado e ofereça um bom nível de segurança, este autentica apenas a máquina onde corre a aplicação e não a aplicação ou o utilizador em si. Sendo assim, uma aplicação pirata pode fazer-se passar por uma autêntica, bastando para isso apenas ter acesso ao servidor. As aplicações cliente que se conectem verificam apenas a autenticidade da máquina a que se conectaram e não da aplicação. Logo, não se percebem que estão a comunicar com a aplicação errada.

Uma vez que durante a pesquisa não foi encontrado qualquer protocolo seguro normalizado para a camada de aplicação, foi proposto um novo protocolo. Este permite efectuar autenticação e segurar os dados trocados a nível da aplicação. Este, tem por base o conhecido protocolo SSL e terá por nome MHSP.

3.4 Autenticação dos Utilizadores

No ambiente da saúde é extremamente importante ter em conta que, embora seja necessário implementar fortes componentes de segurança, a informação relativa aos pacientes tem que estar disponível a qualquer momento, principalmente em caso de emergência.

Para resolver esse problema e permitir que os acessos ao sistema sejam seguros mas que, simultaneamente, não comprometam a disponibilidade da informação, foram propostos dois possíveis cenários. Um que permita acessos apenas de leitura e outro que permita acessos de leitura-escrita. Esses cenários levam a dois tipos de autenticação: *User Authentication* (UA)[9] e *Secure User Authentication* (SUA)[9].

O tipo de autenticação UA foi desenhado para o cenário que permite apenas operações de leitura, ou seja, consultas ao sistema. Este tem mecanismos de segurança um pouco mais relaxados, permitindo uma acesso mais fácil à informação. Desta forma, garante-se que a infor-

mação esteja disponível em caso de necessidade mas apenas para consulta. Assim, aumenta-se a disponibilidade do sistema reduzindo-se as consequências, no caso da informação ser acedida por entidades não autorizadas. O protocolo aqui proposto para adereçar este problema vai ser descrito mais à frente e será denominado de *M-Health User Authentication* (MHUA).

Já o tipo de autenticação SUA foi desenhado para o cenário que permite leitura-escrita. Uma vez que este permite alterar informações contidas no sistema, necessita de implementar mecanismos de segurança mais fortes. Uma alteração não autorizada pode ser catastrófica para o sistema ou mesmo para a vida dos pacientes. Este protocolo será apresentado mais à frente e será chamado de *M-Health Secure User Authentication* (MHSUA).

3.5 Ligação à base de dados e protecção de dados

Para proteger uma base de dados é necessário cumprir alguns objectivos de segurança: identificação e autorização, controlo de acesso, cifragem, integridade e disponibilidade. Um DBMS como o MySQL oferece de raiz mecanismos que permitem satisfazer esses requisitos. Para tal, o MySQL oferece possibilidade de conexão à BD, utilizando um protocolo como o SSL ou SSH. Estes são protocolos de nível de transporte e, como foi dito anteriormente, os protocolos seguros de nível de transporte autenticam apenas as máquinas e não as aplicações. É aqui proposto que, para a ligação e autenticação das aplicações à BD, se utilize um protocolo que proteja os dados a nível da aplicação. Tal pode ser efectuado utilizando o protocolo MHSP aqui proposto. Deve ser então implementada uma aplicação na máquina onde está situada a BD que faça a comunicação com o exterior, ficando o DBMS confinado a comunicações com esta. Assim, o servidor pode ligar-se à BD utilizando o canal seguro fornecido pelo MHSP e a forte componente de segurança que este oferece.

Quanto ao processo de cifrar dos dados na BD, o MySQL oferece mecanismos que a permitam fazer, sendo apenas necessário ponderar as cifras existentes e as vantagens e desvantagens de a fazer. Se se assumir que a BD se encontra num local seguro, tal não é necessário. Não há acesso físico à máquina onde se encontra a BD e todos os dados que entram ou saem dessa máquina são cifrados. Se por outro lado a máquina onde se encontra a BD não estiver num local seguro, pode optar-se por cifrar os dados tendo em conta a perda de performance que vai existir. Além da cifragem e decifragem dos dados ser exigente computacionalmente, estas acarretam ainda outros problemas.

Se os dados da BD forem cifrados passa a ter-se um ponto único de falha, a chave de cifragem, ou seja, ir-se à criar outro problema que passa por como e onde armazenar a chave.

Se esta for guardada em apenas um local corre-se o risco de a perder e, consequentemente, perder todos os dados que estivessem cifrados com esta. Se for replicada, de forma a que existam cópias em caso de uma ser perdida, pode ser mais facilmente obtida por alguém que não deveria ter acesso.

Para o caso aqui descrito, assume-se que a BD está num local fisicamente seguro, não sendo necessária a cifragem dos dados. Esta tem como primeira linha de defesa uma *firewall* que apenas permite acesso à aplicação que corre o protocolo MHSP. Toda a comunicação com a máquina que contém a BD é feita utilizando o protocolo MHSP.

3.6 Armazenamento de dados para consulta offline

Todos os protocolos aqui referidos necessitam de uma conexão ao servidor para poder obter a informação necessária. Esta conexão pode não ser sempre possível. Então, prevê-se a utilização de um método que permita o acesso de um utilizador aos seus próprios dados, de uma forma *offline*. Tal objectivo pode atingir-se permitindo que os utilizadores guardem informação em cache nos seus próprios dispositivos.

Durante a pesquisa efectuada não foram encontrados quaisquer métodos normalizados com o fim anteriormente referido. Assim, é proposto aqui um método que permita armazenar dados no dispositivo de forma segura. Esse método dá pelo nome de MHOUA e será descrito mais à frente.

3.7 Protocolos seguros para comunicação RFID

Todos os protocolos para a comunicação segura sobre RFID estudados trabalham com etiquetas que cumprem o *standard* Gen 2. Estas são etiquetas *low cost* e são consideradas como tendo um nível baixo de segurança, como se pode ler em [11]. O uso destas proporciona a construção de um sistema de médio custo com a contra partida do nível de segurança que oferecem, principalmente no que toca a protecção da privacidade

Os protocolos mais relevantes encontrados durante a pesquisa efectuada foram o PC e o IS-RFID. O protocolo de *Grouping Proof* apresentado no estado da arte é utilizado pelo protocolo IS-RFID e por isso não vai ser aqui referido.

Dos protocolos referidos, o mais apropriado para o uso no cenário de *M-Health* é o protocolo IS-RFID. Este protocolo foi desenhado por [11], precisamente para ser usado neste ambiente e oferece ferramentas para fazer a ligação paciente-medicação-prescrição, assim como,

um bom nível de segurança[1] que depende da entropia do PIN da etiqueta.

O protocolo PC permite trocar dados entre dispositivos RFID, oferecendo um bom nível de segurança. Este oferece uma ferramenta que permite aumentar o nível de segurança conforme a distância do leitor. O protocolo PC tem como contrapartida não oferecer ferramentas para fazer a ligação paciente, medicação, prescrição e a complexidade de implementação. Como tal, será utilizado o protocolo IS-RFID, podendo utilizar o protocolo de PC para complementar a segurança do protocolo IS-RFID, aumentando a segurança total do sistema.

[1] Tendo em conta a segurança que uma etiqueta RFID Gen 2 pode oferecer

Capítulo 4

Arquitectura e Protocolos de Segurança

Neste capítulo será apresentado o desenho da arquitectura de segurança para a prestação de serviços de saúde em mobilidade. Este terá como ponto de partida o modelo apresentado para o cenário de utilização, tendo em conta a análise feita no capítulo anterior aos diversos pontos de fragilidade encontrados. Esta arquitectura tenta minimizar, ou mesmo eliminar, os pontos de fragilidade identificados. A arquitectura proposta (Figura 4.1) é composta por 7 entidades: profissional de Saúde, utilizador comum, dispositivos fixos e móveis, servidor, BD, medicamento e CA. A CA passa agora a ser uma importante parte da arquitectura, uma vez que é essencial para o correcto funcionamentos dos protocolos propostos. Esta é responsável pelo registo dos utilizadores, emissão e verificação dos certificados de utilizadores e aplicações.

Como referido anteriormente as *firewalls* servem de primeira linha de defesa, impedindo conexões não desejadas. O servidor e a BD devem estar protegidos por uma *firewall* que deve impedir conexões com o exterior, exceptuando as que tenham como destino ou origem as aplicações a correr o protocolo MHSP. Um outro ponto importante passa por actualizar devidamente o sistema operativo dos dispositivos, eliminando possíveis falhas de segurança. Estas actualizações são imperativas principalmente na BD, uma vez que a ligação entre a aplicação da BD e o DBMS não está segura. Uma outra razão passa pela não cifragem dos dados da BD que permite a qualquer vírus ou ligação não desejada à BD o acesso aos dados que esta contém, o que pode ter consequências devastadoras.

Nesta arquitectura, as ligações entre aplicação cliente e aplicação servidor e aplicação servidor e aplicação BD, serão seguras utilizando o protocolo MHSP, o qual também responsável pela sua autenticação.

Os dados temporários guardados pelas aplicações clientes serão seguros utilizando o método MHOUA.

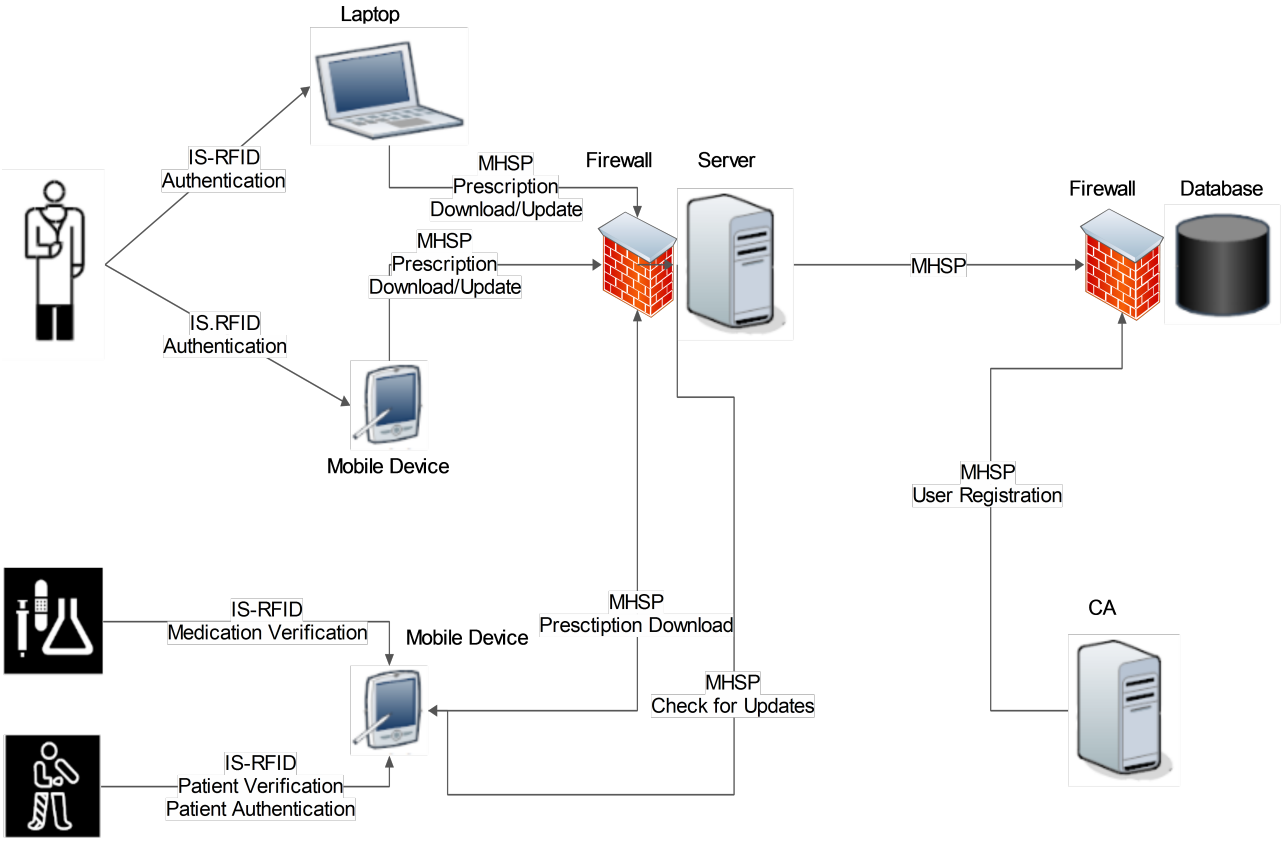


Figura 4.1: Arquitectura de segurança para a prestação de serviços de saúde em mobilidade [9]

Para efectuar a ligação entre paciente, medicamento e prescrição será utilizado parte do protocolo IS-RFID. A transmissão do id da etiqueta, durante o processo de autenticação dos utilizadores, será efectuada utilizando parte do mesmo protocolo. Embora este não tenha como objectivo a autenticação, oferece métodos para transmitir a informação da etiqueta com alguma segurança.

Os processos a ser efectuados para tornar esta arquitectura segura, são os seguintes:

- Comunicação e autenticação entre aplicações: *M-Health Security Protocol* (MHSP);
- Comunicação entre dispositivos RFID: IS-RFID;
- Autenticação:
  - Processo de Registo;
  - *User Authentication* (UA): *M-Health User Authentication* (MHUA);
  - *Secure User Authentication* (SUA):*M-Health Secure User Authentication* (MHSUA);
- Armazenamento de dados para consulta offline: *M-Health Offline User Acess* (MHOUA).

4.1 *M-Health Security Protocol* (MHSP)

O protocolo *M-Health Security Protocol* (MHSP) [9] é semelhante ao SSL, mas foi desenhado para a camada de aplicação ao invés da camada de transporte. O uso do protocolo SSL como base do protocolo aqui proposto, deve-se à segurança que este oferece, de tal forma que é utilizado em grande escala na internet.

O protocolo SSL inclui dois sub-protocolos (2.4.3): um para definir as mensagens a ser trocadas (*Record Protocol*) e um para efectuar efectuar a ligação (*handshake*). Como não se pretende que este seja um protocolo genérico, mas apenas utilizado pelas aplicações da arquitectura de segurança, a primeira parte não será necessária, já que o formato das mensagens trocadas entre as aplicações será previamente conhecido.

A segunda parte do protocolo SSL (o *handshake*) vai servir de base ao protocolo aqui proposto. O passo opcional nesse protocolo passa a ser agora obrigatório; a autenticação do cliente é agora tão importante como a do servidor. Este protocolo oferece um canal seguro sobre a Internet autenticando as aplicações intervenientes. A autenticação entre as aplicações é alcançada utilizando certificados de chave pública, cuja veracidade pode ser confirmada através de uma CA.

O protocolo MHSP (Figura 4.2) será descrito em seguida. Para uma melhor leitura deste, vai denominar-se a aplicação cliente de $CApp$, a aplicação servidor de S, CC o certificado da aplicação cliente e SC o certificado do servidor. Vai assumir-se que a mensagem de ordem n trocada entre as aplicações é representada por m_n e RN são um conjunto de dados aleatórios de ordem N gerados por uma aplicação. Da mesma forma, $enc(M, K)$ é a cifragem da mensagem M com a chave K e $dec(M, K)$ é o processo oposto. S_{pubKey} e $S_{privKey}$ são, respectivamente, as chaves públicas e privadas da aplicação servidora e $CApp_{pubKey}$ e $CApp_{privKey}$ as chaves da aplicação cliente. Finalmente, $gen(key)$ gera a chave desejada e V é a versão do protocolo.

1. A aplicação cliente envia o seu certificado para o servidor.

   $CApp \rightarrow S : CC$

2. A aplicação servidor verifica a veracidade do certificado recebido e termina a conexão caso esta não se verifique.

   ```
   if (CC is invalid)
      then Connection terminated
   ```

else $S \rightarrow CApp : SC$

3. A aplicação cliente verifica a veracidade do certificado recebido e termina a conexão caso esta não se verifique. Caso contrário, gera um desafio para que o servidor possa verificar a sua autenticidade.

   if SC is invalid
      then Connection terminated
   else $m_1 = enc(< V, R1 >, S_{pubKey})$
        $C_{App} \rightarrow S : m_1$

4. Nesta fase, o servidor tem que provar a sua autenticidade perante a aplicação cliente utilizando para isso o desafio recebido. Posteriormente gera as chaves de sessão e de comunicação.

   $< V, R1 >= dec(m_1, S_{privKey})$
   $gen(Sess_{key})$
   $m_2 = enc(< R1, Sess_{key} >, CApp_{pubKey})$
   $S \rightarrow CApp : m_2$
   $gen(Comm_{key})$
   $m_3 = enc(Comm_{key}, Sess_{key})$
   $S \rightarrow CApp : m_3$

5. Com a reposta a aplicação cliente pode verificar a autenticidade do servidor e obter as chaves de sessão e comunicações.

   $< R1, Sess_{key} >= dec(m_2, C_{priv_{key}})$
   if $R1_{sent}$ equal $R1_{received}$
      then $Comm_{key} = dec(m_3, Sess_{key})$
           $gen(R2)$
           $m_4 = enc(< "Hello", R2 >, Comm_{key})$
           $CApp \rightarrow S : m_4$
   else Connection terminated

6. Finalmente, o servidor decifra a mensagem m_4 enviada pelo servidor. Se esta for decifrada correctamente significa que a aplicação cliente foi capaz de decifrar correctamente a $Comm_{key}$ e está correctamente autenticada.

```
if dec(m4, Comm_key)
   then Authenticated
else Connection terminated
```

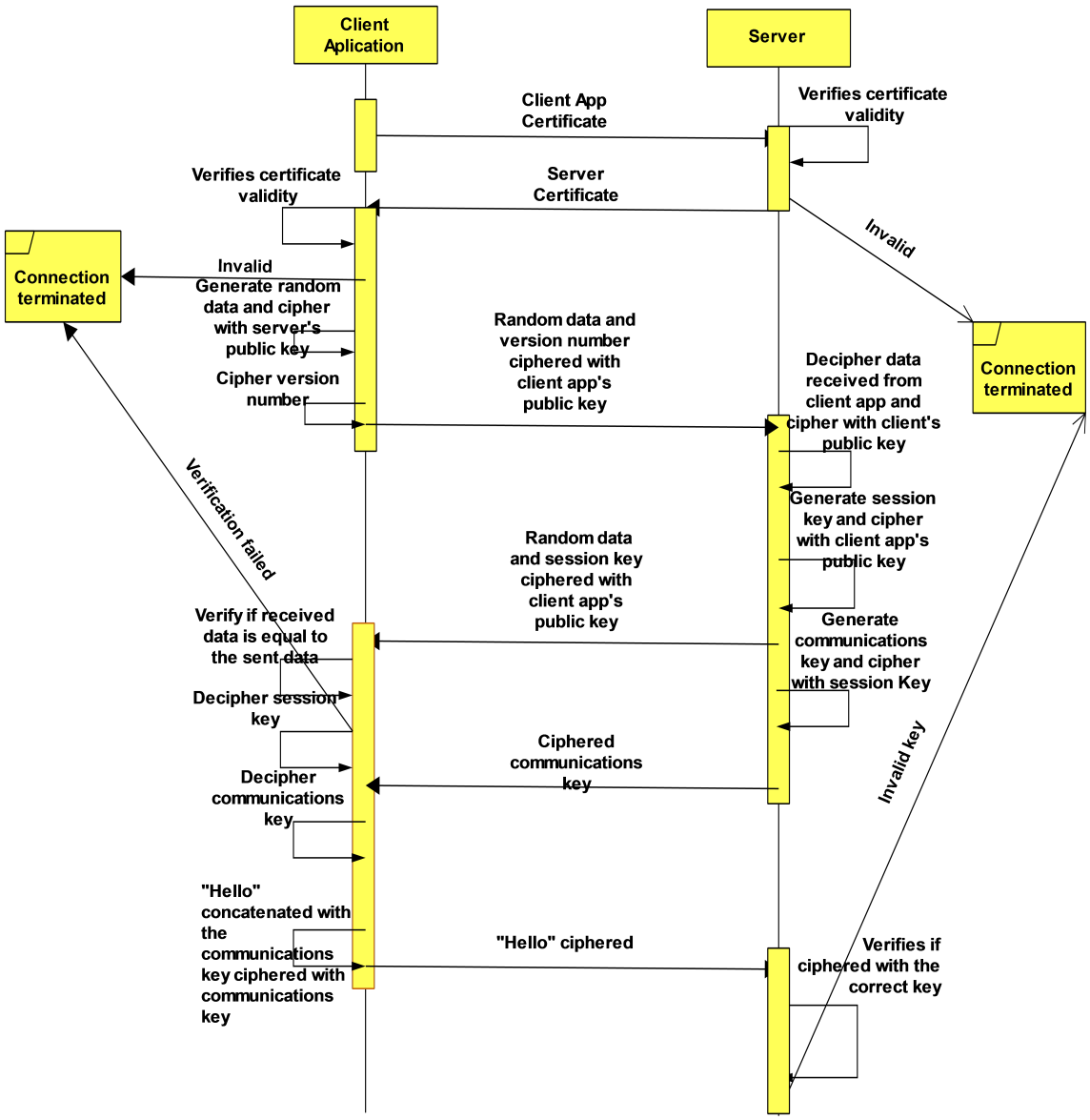


Figura 4.2: *M-Health Security Protocol* (MHSP) [9]

O protocolo MHSP permite resolver o problema da autenticação entre as aplicações com recurso aos certificados. Esses certificados terão um formato semelhante ao descrito no estado

da arte (2.4.3).

De forma a garantir-se integridade, autenticidade e não repudio das mensagens trocadas, pode recorrer-se a assinaturas digitais com certificados de chaves públicas ou ao uso de um método como o MAC. Aqui, sugere-se os uso das chaves públicas; desta forma é sempre possível verificar a autenticidade da mensagem sendo necessário apenas o conhecimento da chave pública do assinante.

Um dos pontos mais importantes do protocolo MHSP é a definição da duração das chaves e das validades dos certificados. Caso contrário, as chaves poderão ser comprometidas e a segurança do sistema ser posta em risco.

4.2 Comunicação entre dispositivos RFID

Como se pode ver no desenho da arquitectura (Capítulo 4.1), o protocolo a ser utilizado para a comunicação entre os diversos dispositivos RFID é parte do protocolo Inpatient Safety RFID (IS-RFID) proposto por [11]. Como foi referido na análise de requisitos e fragilidades, este oferece um bom nível de segurança e meios para fazer a correspondência entre paciente, prescrição e medicação. Existem alguns passos deste protocolo que podem ser ignorados, uma vez que os dispositivos que os profissionais de saúde ou os pacientes têm, possuem a capacidade para aceder ao servidor e actualizar os dados. Além do referido este protocolo foi desenhado para funcionar dentro de um hospital ou instituição de saúde e a arquitectura desenhada tem como principal objectivo o auxilio na toma de medicação em ambulatório. Este protocolo é composto por 4 procedimentos: empacotamento, estação de enfermagem, administração segura de medicamentos, monotorização.

Neste caso, o empacotamento não tem necessidade de existir, uma vez que a arquitectura desenhada tem como objectivo a administração de medicamentos. Assume-se que o empacotamento destes tenha sido efectuado correctamente.

O procedimento efectuado na estação de enfermagem passa agora a ser efectuado automaticamente pelo dispositivo. Este pode aceder ao servidor e descarregar a informação que necessita.

O procedimento de administração segura de medicamentos é composto por dois passos: processo de verificação e geração da prova. Como o dispositivo agora contém toda a informação que necessita, pode gerar a prova com a informação recolhida no primeiro passo. O processo final, monotorização, é efectuado automaticamente pelo dispositivo aquando a leitura das etiquetas RFID.

4.3 Processo de registo

O registo deverá ser feito por uma entidade de confiança, uma CA. Deverá ser feito em pessoa e assim, evitar identificações incorrectas. Uma vez que este protocolo é desenhado principalmente para o uso com dispositivos móveis (para facilitar o seu uso), o tradicional nome de utilizador será substituído por uma etiqueta RFID e o seu ID correspondente. Esta etiqueta apenas permite a autenticação quando complementada por outros meios; a mera posse não torna possível o acesso ao sistema.

Para o registo dos utilizadores do cenário UA, a CA irá apenas emitir uma palavra passe e uma etiqueta RFID. A palavra passe deverá ser dada aos utilizadores através de um canal paralelo, como por exemplo um envelope fechado, processo semelhante ao efectuado pelos bancos para enviar o pin para os seus clientes. A CA será ainda responsável por guardar um *hash* da palavra passe na BD. Optou-se por guardar apenas o *hash* para proteger a palavra passe real; este *hash* dá acesso apenas como UA. A posse deste não permite acesso a operações SUA ou a ficheiros temporários armazenados nos dispositivos para consulta offline.

O acesso UA tem por base um método de autenticação de nome de utilizador e palavra-passe e, como tal, não possui fortes componentes de segurança. Propõe-se que para o tipo de acesso SUA seja utilizado um método com base em certificados de chave pública e, assim, aumentar a segurança total do sistema. Para este tipo de acesso a CA deverá emitir, para cada utilizador, um certificado de chave pública, uma etiqueta RFID e ainda uma palavra passe (a palavra passe deverá ser entregue num envelope fechado).

Para impedir o roubo da chave privada associada à chave pública do certificado, a CA irá cifrá-la com uma chave simétrica. Essa chave será obtida através de uma *hash* obtido do tuplo `<palavra passe, ID do utilizador>`. A chave privada e o certificado poderão ser entregues ao utilizador através de um *smart card*. Este poderá ser simplesmente um *Memory Card*, uma vez que não necessitará de efectuar qualquer computação.

Será possível alterar a palavra passe do utilizador para qualquer um dos cenários. Para o cenário UA o utilizador poderá alterar a sua palavra passe depois de se autenticar correctamente. A aplicação cliente envia para o servidor uma mensagem com o *hash* da palavra passe antiga e da nova e a alteração será efectuada.

No caso dos utilizadores SUA não será necessária a intervenção do servidor. Após o utilizador inserir a sua palavra passe, a aplicação cliente pode decifrar a chave privada. Se o utilizador inserir então uma nova palavra passe, a aplicação poderá cifrar a chave secreta com a nova palavra passe.

4.4 *M-Health User Authentication* (MHUA)

De forma a segurar o tipo de acesso UA é proposto um método de autenticação um pouco mais relaxado. Este método baseia-se num simples esquema de palavra-passe e nome de utilizador e será identificado como *M-Health User Authentication* (MHUA) [9] (Figura 4.3).

Neste processo de autenticação mais simples, a aplicação lê o ID do utilizador da etiqueta RFID e pede a palavra passe do utilizador. A aplicação cria então um *hash* da palavra passe do utilizador e envia para o servidor, juntamente com o ID do utilizador. O servidor após receber esses dados limita-se a verificar a sua veracidade com recurso à BD. Este método é bastante simples de utilizar e, embora não seja tão seguro como o método proposto para SUA, oferece um bom nível de segurança tendo em conta que todas as mensagens são trocadas sobre o canal já seguro com o protocolo MHSP.

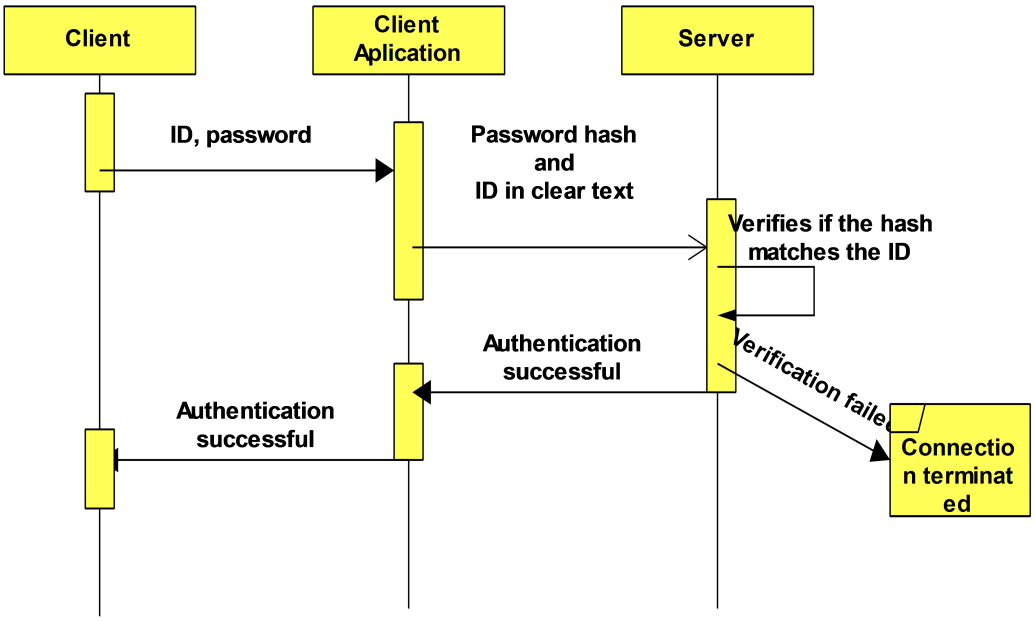


Figura 4.3: *M-Health User Authentication* (MHUA)

4.5 *M-Health Secure User Authentication* (MHSUA)

Para o tipo de autenticação SUA é necessário ter em mente que um ataque com sucesso pode permitir a alteração de informação privada de um paciente, podendo trazer danos para o sistema ou para a vida deste, sendo necessário um protocolo de autenticação seguro. Para tal, é proposto o protocolo *M-Health Secure User Authentication* (MHSUA) [9] (Figura 4.4). Este tem por base os certificados de chave pública e oferece uma forte componente de segurança. Como mencionado na fase de registo, cada utilizador deve possuir uma palavra-passe, uma

etiqueta RFID e um suporte físico que contenha o seu certificado de chave pública e a chave secreta correspondente, cifrada com o *hash* do tuplo `<palavra passe, ID do utilizador>`. De forma a tornar o protocolo seguro, não existe qualquer informação secreta (tal como a palavra-passe) a ser trocada na rede.

De seguida será descrito o protocolo em detalhe; os identificadores assumidos em 4.1 vão ser aqui utilizados. Será também assumido que *pass* é a palavra passe do utilizador e $TagID$ é o ID do utilizador (lido da etiqueta). UC é o certificado do utilizador e U_{pubKey} e $U_{privKey}$ são as chaves pública e privada do utilizador. Finalmente $enc_{private}$ é a chave privada cifrada e guardada no *smart card*, $a||b$ é a concatenação de a com b e $hash(m)$ é o *hash* feito através da entrada m.

1. Primeiramente a aplicação cliente lê a $TagID$;
2. O utilizador selecciona o tipo de autenticação SUA e a aplicação irá pedir o *smart card* do utilizador;
3. Depois de lido o UC e a chave privada correspondente do *smart card* a aplicação irá pedir a palavra passe do utilizador.
4. A aplicação irá então obter a chave privada utilizando a palavra passe.

   $U_{privKey} = dec(enc_{private}, hash(pass||Tag_ID))$
   $CA \rightarrow S : UC$

5. O servidor vai verificar a autenticidade do UC e gerar o desafio para testar a autenticidade do utilizador.

   `if` UC `is valid`
   `then Connection terminated`
   `else` $m_1 = enc(R1, U_{pubKey})$
   $S \rightarrow CApp : m_1$

6. Para provar a sua autenticidade a aplicação cliente deve ser capaz de decifrar a mensagem m_1 utilizando a chave $U_{privKey}$.

$m_2 = dec(m_1, U_{privKey})$

$CApp \rightarrow S : m_2$

7. Se a mensagem recebida for igual aos dados aleatórios enviados a autenticação do utilizador será verificada.

```
if m2 = r1
   then User authenticated
else Connection terminated
```

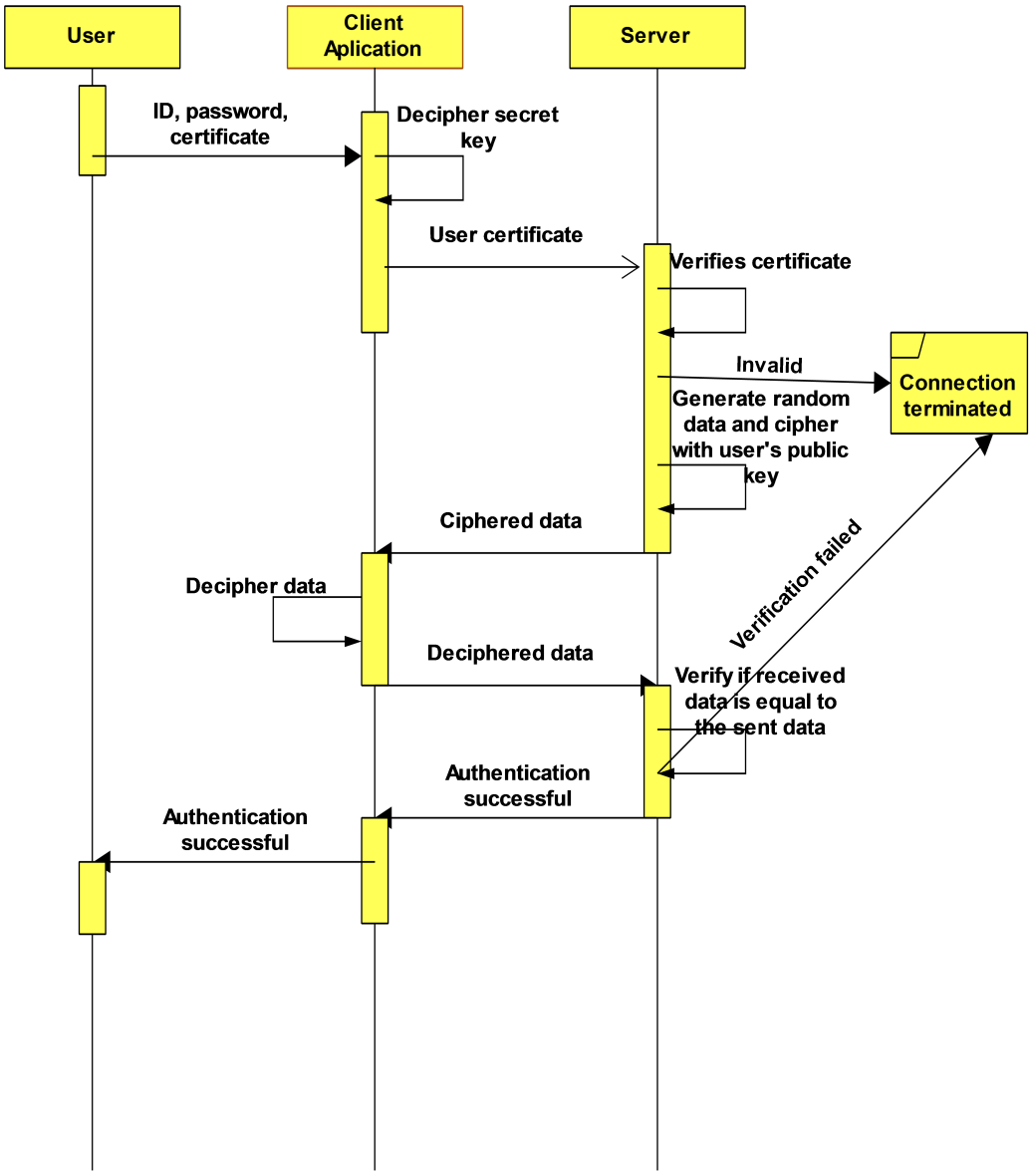


Figura 4.4: *M-Health Secure User Authentication* (MHSUA)

Todos os passos do método anteriormente referido são efectuados sobre a ligação já segura com o MHSP (4.1), não sendo necessária a cifragem dos dados no último passo. Este protocolo garante a autenticidade dos utilizadores e é mais seguro que um protocolo com base em nome de utilizador e palavra passe (Como é o caso do MHUA). Assim, garante-se que os utilizadores com acesso SUA têm que o fazer com uma componente de autenticação forte, de forma a dificultar ou mesmo eliminar possíveis ataques.

4.6 *M-Health Offline User Acess* (MHOUA)

Para não ser necessária uma ligação constante ao servidor, o protocolo MHOUA [9] permite ao utilizador guardar dados temporários no seu próprio dispositivo para consulta offline. O utilizador terá que se autenticar previamente, utilizando um dos métodos referidos anteriormente (MHUA ou MHSUA), de modo a aceder ao sistema e fazer download dos respectivos dados. A aplicação guarda os dados utilizando um parâmetro *time-to-live*, para que possam apenas ser consultados durante um certo período pré-definido.

O método aqui proposto consiste em utilizar uma cifra simétrica para guardar os dados de forma segura, não sendo necessária uma ligação ao servidor para obter parte da chave. Propõe-se que a chave seja um *hash* criado a partir do tuplo `<palavra passe do utilizador,id da etiqueta, nome do ficheiro, data de criação do ficheiro>`, assegurando-se que cada ficheiro tem uma chave diferente, mesmo que exista uma utilização da aplicação no mesmo dispositivo por diferentes utilizadores.

O tamanho máximo de cada ficheiro deve ser cuidadosamente definido dependendo da força da cifra utilizada, para impedir ataques de força bruta. Ficheiros que tenham tamanho grande devem ser partidos em diversos ficheiros mais pequenos.

Capítulo 5

Implementação de Sistema Seguro para M-Health

Depois da devida análise de todos os protocolos e métodos envolvidos e do desenho da arquitectura, passou-se à implementação da mesma, de modo a poder testar-se o bom funcionamento e a segurança oferecida. Para tal, foram desenhadas as aplicações: cliente (para dispositivos fixos e móveis), servidor, para a BD e para a CA.

As aplicações desenhadas implementam apenas funções de gestão básicas, uma vez que o principal objectivo destas é o teste da segurança, desempenho e falhas do sistema implementado. Todas as aplicações foram implementadas utilizando a linguagem de programação JAVA. Esta escolha foi tomada devido à vasta quantidade de bibliotecas existentes e à grande portabilidade que esta oferece. O JAVA permite que uma aplicação funcione em qualquer sistema operativo para dispositivos fixos e, devido à sua semelhança e compatibilidade com o sistema operativo Android, torna mais fácil a implementação em dispositivos móveis. As aplicações feitas para os dispositivos fixos foram facilmente transportadas para os dispositivos móveis, necessitando de apenas pequenas alterações. Toda a implementação do protocolo MHSP, os métodos de autenticação MHUA e MHSUA e o método para guardar informação temporária offline MHOUA se manteve.

Foi desenhada ainda uma aplicação para simular as etiquetas RFID, uma vez que não houve disponibilidade de tais dispositivos.

5.1 Autoridade de Certificação (CA)

A aplicação para a CA tem como principais funções: efectuar o registo de utilizadores e aplicações, passar e atestar a veracidade dos certificados.

Esta aplicação dá a possibilidade de fazer registo de utilizadores normais ou profissionais de saúde, aplicações para dispositivos móveis ou fixos, aplicações para a BD e ainda aplicações servidoras. Quando se fala de registo, no caso das aplicações para a BD, significa apenas a passagem de um certificado. Para as restantes aplicações, a CA faz um registo da aplicação na BD e assim, poder-se-à verificar se, por exemplo, a chave de alguma das aplicações ou utilizadores foi revogada.

Para fazer o registo, a aplicação permite seleccionar (Figura 5.1) qual o tipo de certificado a ser utilizado e, dependendo dessa escolha, têm que ser preenchido um registo diferente.

Figura 5.1: Autoridade de Certificação (CA) - Ecrã inicial

Para os dispositivos, móveis (Figura 5.2) ou fixos (Figura 5.3), os campos a ser preenchidos são os mesmos: endereço MAC da placa de rede, nome do dispositivo e tempo de validade. A única excepção é que, enquanto para o dispositivo móvel a aplicação apenas pode ser cliente, para o dispositivo fixo esta pode ser cliente, servidor ou ainda BD. Os campos endereço MAC e nome do dispositivo permitem à aplicação verificar se o certificado é o correcto para a aplicação, impedindo que haja de trocas de certificados entre as aplicações ou dispositivos.

A aplicação gera para cada registo um ficheiro com o certificado da aplicação e outro com a chave secreta correspondente.

Para os utilizadores (Figura 5.4) o registo necessita de mais alguns dados. Esse irá ser o registo pessoal do utilizador no sistema e deverá ser efectuado pessoalmente, evitando erros de identificação. A aplicação permite o registo de dois utilizadores diferentes: utilizador normal ou

Figura 5.2: Autoridade de Certificação (CA) - Dispositivo Móvel

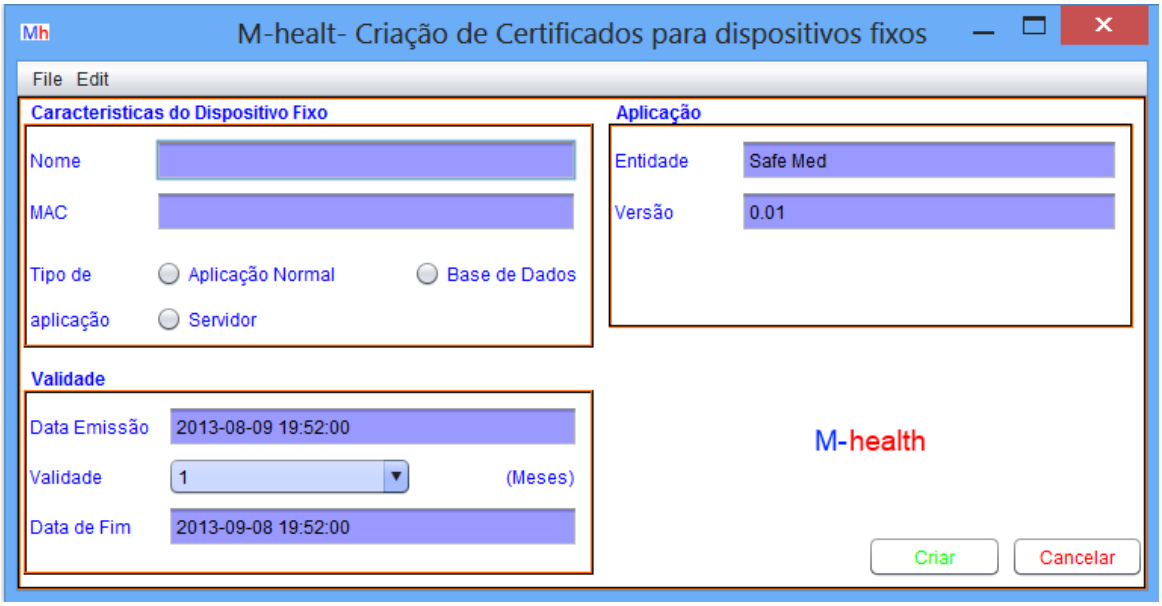


Figura 5.3: Autoridade de Certificação (CA) - Dispositivo Fixo

utilizador profissional de saúde. Para o primeiro a aplicação vai gerar apenas um ficheiro[1] com a palavra passe do utilizador. No caso do utilizador profissional, a aplicação gera um certificado, um ficheiro com a chave secreta cifrada com a chave *hash* do tuplo `<palavra passe, ID do utilizador>` e um ficheiro com a palavra passe.

[1]Este ficheiro é criado para simular o envelope fechado referido na análise

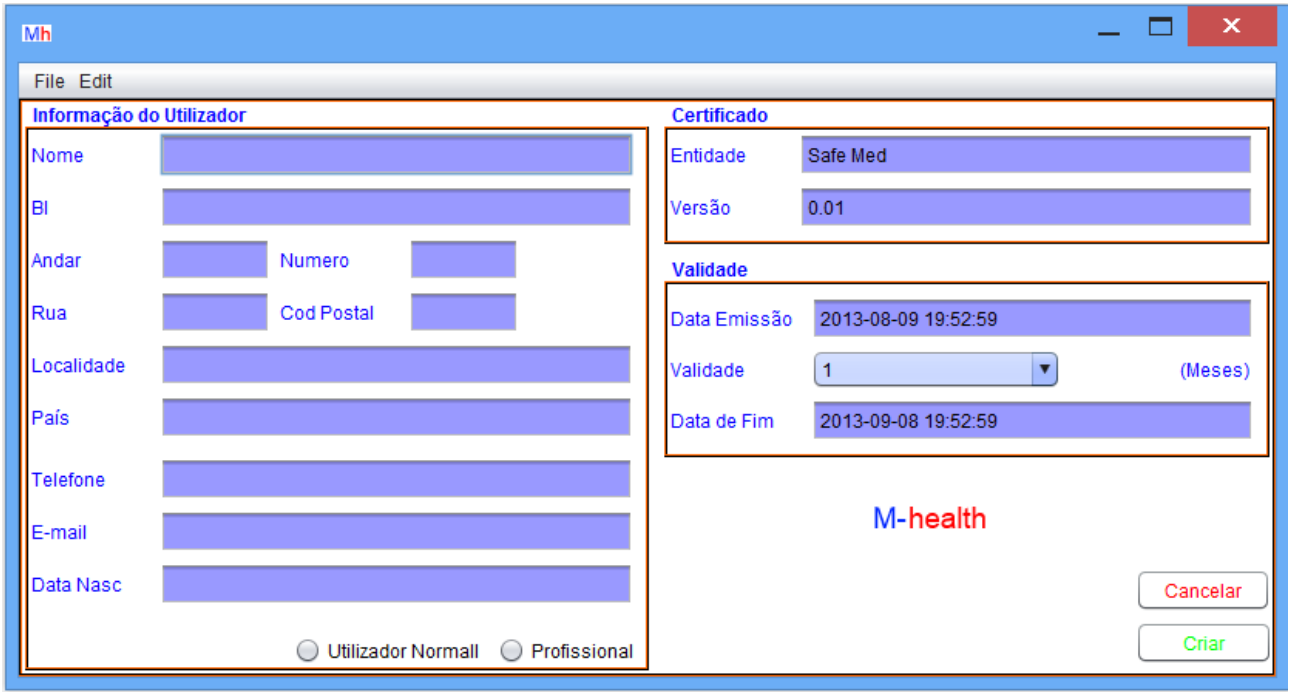


Figura 5.4: Autoridade de Certificação (CA) - Utilizador

Esta aplicação vai ligar-se à BD através do protocolo seguro MHSP, transmitindo todos os dados de uma forma segura. A aplicação para a CA foi desenhada apenas para efeitos de teste, pelo que não existe autenticação de utilizador, não sendo possível verificar quem efectuou o registo de cada utilizador. Sugere-se que, para implementações futuras seja necessário um tipo de utilizador especial que tenha permissão apenas para registo de utilizadores, não podendo consultar utentes ou prescrever medicamentos. Assim tornando-se possível descobrir quem efectuou registos indevidos. Apesar disso, a aplicação da CA apenas pode ser utilizada na máquina (ou máquinas) para as quais os certificados de CA forem passados. Este certificado pode ser passado pela própria CA ou por uma de mais alto nível.

Não existe qualquer certificado para os utilizadores normais. Estes podem apenas aceder ao sistema utilizando MHUA. Apenas a aplicação cliente pode existir nos dispositivos móveis e fixos. Os restantes tipos de aplicações só podem existir nos dispositivos fixos.

O certificado de aplicação (Figura 5.6) permite que uma aplicação se autentique perante outra e, simultaneamente, verificar se o dispositivo onde a aplicação corre é o correcto, dificultando possíveis ataques. Desta forma, o certificado aqui utilizado contém os seguintes campos: autoridade certificadora, tipo de Certificado, tipo de dispositivo, tipo de aplicação, nome do dispositivo, endereço MAC da placa de rede do dispositivo, data de emissão, data de validade, ID da aplicação, versão do certificado, chave pública da aplicação, chave pública da autoridade

certificadora e assinatura da autoridade certificadora. Dos campos referidos, o tipo de dispositivos e o tipo de aplicação varia consoante o dispositivo e o tipo de aplicação ao qual se destina. Os restantes campos são iguais para todas as aplicações.

```
Certificado por: Safe Med
Tipo de Certificado: Utilizador

Data de Emissão: 2013-08-01 21:51:08
Data de Validade: 2014-01-28 20:51:08


ID do Utilizador: 2

Versão: 0.01

Chaves

Chave Publica:
30819f300d06092a864886f70d010101050003818d0030818902818100cb621639
d3f2d80902433fe087520d4611f941f17befefb0c72fc3a72b91aa54a1996609be
c7503994a1cd9f149ac0f9fae65c9d0754e4ae19ae026dc395dded5c03933b0888
1fe78213c7a329da8d3dc52a24139734919799424db29c6ddda508f3bc61363e35
733002c7d277f42c51c88a2d217fec7e83b044394419a9f2830203010001

Chave Pública da CA:
30819f300d06092a864886f70d010101050003818d0030818902818100 84f6e6e81
c3b7ebfad8349c31c88fccaa5de89b52e4131ef160b6b257b93d01cf360546aad5c
1388be9e61110af0ef8640321c8b5e97b7c01b56caf973fec0e5e0caff46e31a2ff
60af5a64bd6a027948a6aa1de66fd05f4b3c2f9e2e7fcff5f16e4689fe254ba0b3f
cd4b6d5ebb4544117c8ba9059169240634fdefbac4f4c90203010001

Assinatura:
084d75b822b29fe2f74719b94342c7f7fed28b673f81acd99988cd922e8b9b87305b
4d1a5ba3a32d4524a98f66258ecfc03fcc9309c208b6c0bcf2c3e81208bf66c3d7d1
0f4fb24c2f3f05157664d4c2b7ebfcecbe14c46d5151216b27060a9f4c673e48b929
db26191939233c54dfaf5e623dd7c0c7dbe83903c37f6fdc9d2d
```

Figura 5.5: Certificado de utilizador

Os certificados de utilizador (Figura 5.5) permitem que os utilizadores profissionais de saúde se autentiquem utilizando MHSUA. Estes certificados são muito semelhantes aos certificados de aplicação mudando apenas o tipo de certificado (passa agora a ser "Utilizador") e são eliminados os campos MAC e nome.

5.2 Base de dados (BD)

Como foi referido anteriormente, a única comunicação com o exterior, que será possível efectuar da máquina onde reside a BD, será através de uma aplicação que corre o protocolo

```
Certificado por: Sate Mea
Tipo de Certificado: Aplicação

Tipo de Dispositivo: Fixo
Tipo de Aplicação: Aplicacao Normal

Identificação do Dispositivo

Nome: App-PC
MAC: MAC-APP

Data de Emissão: 2013-07-24 21:11:31
Data de Validade: 2013-08-23 21:11:31


ID da Aplicação: 17

Versão: 0.01

Chaves

Chave Publica:
30819f300d06092a864886f70d010101050003818d0030818902818100a39a12d4
b3328db8fd0c36966bfae792a7c0cee2fbeed40917256c54731a52854f9aec5ac2
06d750f05659f2301e6ac7be5921fe9b16d5bbb29c5434f93dfc9157dfdea27107
cd808d6677638c0f130db838016c1839ce85cda8de88bf21173c3cb2a620bc6c28
5c6d00afe669f2a60de2d3d3bd80fd4367bf31025d27aae0810203010001

Chave Pública da CA:
30819f300d06092a864886f70d010101050003818d0030818902818100 4f6e6e8
1c3b7ebfad8349c31c88fccaa5de89b52e4131ef160b6b257b93d01cf360546aad
5c1388be9e61110af0ef8640321c8b5e97b7c01b56caf973fec0e5e0caff46e31a
2ff60af5a64bd6a027948a6aa1de66fd05f4b3c2f9e2e7fcff5f16e4689fe254ba
0b3fcd4b6d5ebb4544117c8ba9059169240634fdefbac4f4c90203010001

Assinatura:
14505e33fb2f33eb68a356974ac4ab396ab2a04ed2f0468c340df0c9b9a30c5d27
18d90b97310523c992c0f2f78d16d9c6704d0a1405f7f2f462d0b5398ab4e700e6
82d9e52c312b930d56ae827311aca5fc30591672af9d7df1cae647b88b278f2d16
8369569ba5b3d30dd7744a6ba95dac2197a8c404fda16f30c2ed39f0e6
```

Figura 5.6: Certificado de aplicação

MHSP (Figura 5.7). Essa aplicação será responsável apenas por encaminhar o tráfego vindo do servidor para a BD, tornando todo tráfego devidamente seguro e autenticado. Tal como foi dito anteriormente, um DBMS como o MySQL oferece meios para cifrar e autenticar os dados, mas apenas a nível da camada de transporte. Todo o tráfego entre a aplicação da BD e a BD será em texto limpo. Assume-se que a BD está fisicamente segura. Logo a ligação sem protecção entre a aplicação da BD e a BD não traz falhas de segurança. A aplicação da BD apenas aceita ligações de aplicações servidor e aplicações CA; todas as outras serão imediatamente rejeitadas.

Foi tomado como opção ligar a aplicação CA directamente à BD apenas por motivos de teste e, assim, facilitar a implementação. Futuramente, as ligações entre a CA e a BD deverão ser efectuadas através do servidor, assim como para todas as outras aplicações.

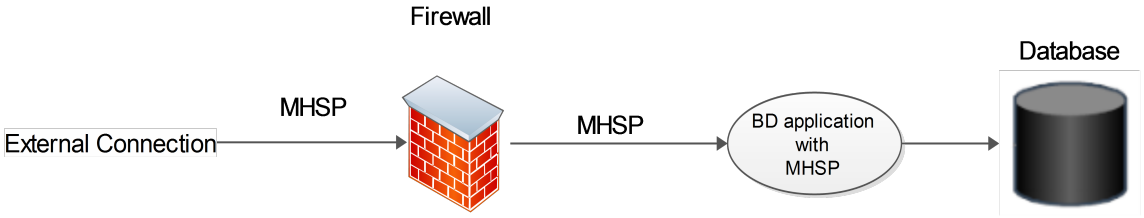


Figura 5.7: Ligação externa para a BD

Para testar as funcionalidades das várias aplicações foi desenhada uma BD em MySQL (Figura 5.8). Esta permite efectuar algumas operações básicas de gestão, tal como inserir utentes e profissionais de saúde, medicamentos, prescrições, etc.

Algumas das tabelas têm como objectivo ajudar a melhorar a segurança total do sistema. A tabela acessos permite guardar todos os acesso efectuados por um utilizador e qual a aplicação utilizada. Se existirem dois ou mais acessos em simultâneo da mesma aplicação de dois locais ou máquinas diferentes, o seu certificado deve ser imediatamente revogado. Com a revogação de certificados pode dificultar-se ainda mais a vida a um possível atacante.

A tabela chaves permite guardar todas as chaves das aplicações e utilizadores, tornando possível impedir o acesso a utilizadores com chaves inválidas.

Existe ainda uma outra tabela relevante, a tabela ChavesCA, a qual possibilita guardar todas as chaves da CA, permitindo a uma aplicação verificar a veracidade de um certificado e chave correspondente.

A BD desenhada permite também saber que profissional de saúde passou uma determinada prescrição e guardar os IDs e PINs das etiquetas necessários para propósitos de autenticação. São guardadas na BD também as provas das tomas medicamentosas.

Esta aplicação gera um registo de eventos relevantes que, neste caso, são apenas conexões, uma vez que esta apenas reencaminha informação. Guardar registos da informação recebida cria ficheiros demasiadamente grandes e é um pouco redundante uma vez que a aplicação servidor regista todas as operações.

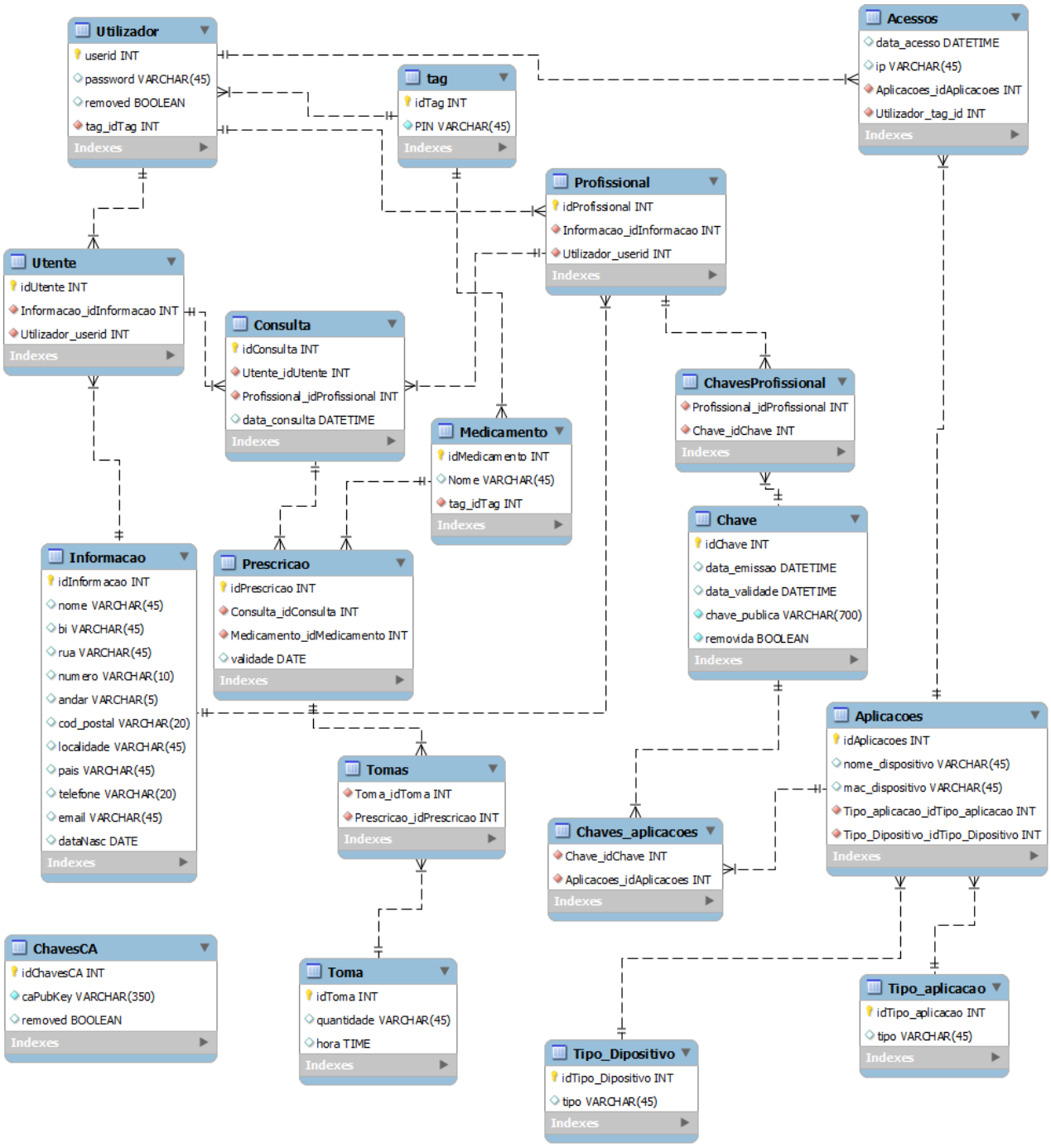


Figura 5.8: Modelo relacional para BD

5.3 Aplicação Servidor

A aplicação servidor é uma das partes centrais de toda a arquitectura,já que é responsável pela autenticação dos utilizadores e das aplicações clientes. É também responsável por efectuar todas as operações de gestão, enviando as *querys* prontas a executar para a aplicação BD. A última tem apenas que as reencaminhar correctamente para a BD.

O servidor verifica, aquando da autenticação das aplicações, se existe alguma conexão activa desta, mas proveniente de um dispositivo diferente. Se for esse o caso, a conexão é imediatamente terminada e o certificado da aplicação revogado; o mesmo acontece para a autenticação dos clientes.

Esta aplicação aceita apenas conexões de aplicações cliente e liga-se única e exclusivamente a aplicações BD.

O servidor deve estar num local fisicamente seguro. Embora a informação que aqui circula esteja devidamente cifrada, o acesso físico ao sistema pode permitir o acesso ao certificado e à chave secreta deste, podendo comprometer o sistema.

5.4 Aplicação Cliente

A aplicação cliente permite fazer o interface com o utilizador, seja ele do tipo UA ou SUA. Todas as informações trocadas entre esta e o servidor são seguras, utilizando o protocolo MHSP. Esta aplicação apenas se pode conectar a aplicações servidor.

Foram desenhadas duas aplicações diferentes, uma para dispositivos fixos e outra para dispositivos móveis. A aplicação para dispositivos móveis desenhada apenas permite acessos de utilizadores UA, embora em implementações futuras se possa facilmente implementar os dois tipos de acesso. A aplicação fixa aceita os dois tipos de autenticação UA e SUA.

5.4.1 Aplicação cliente para dispositivos fixos

A aplicação cliente para dispositivos fixos permite que um utilizador interaja com o sistema M-health utilizando para isso um dispositivo fixo, como um portátil ou um computador fixo.

Esta aplicação é composta por 3 partes principais: ecrã inicial, interface MHSUA e interface MHUA. Permite ainda um acesso MHOUA, o qual efectuado se a aplicação não obter ligação com o servidor.

No ecrã inicial (Figura 5.9) o utilizador pode seleccionar qual o tipo de acesso que pretende fazer e, consoante a sua selecção, ser-lhe-ão pedidos diferentes requisitos. No caso de acesso MHUA, o utilizador terá introduzir a sua palavra passe e o id da sua etiqueta será lido automaticamente. Se o utilizador pretender um acesso MHSUA, será necessária a introdução de um certificado (caminho para o certificado) e, só após a sua validação[2], será pedida a intro-

[2]validação apenas do formato, das datas de validade da entidade certificadora e da assinatura. Os restantes campos serão apenas validados após a introdução da palavra passe e id da etiqueta

dução da palavra passe ao utilizador, sendo mais o id da etiqueta será lido automaticamente. Em ambos os casos o utilizador será autenticado com a ajuda da ligação ao servidor.

Para o caso da autenticação offline (MHOUA), de forma a poder ser lido o id da etiqueta, é necessário ter acesso ao id e PIN desta. Sendo esse o caso, cada vez que um utilizador faz um acesso MHUA ou MHSUA, o PIN e ID da etiqueta são guardados num ficheiro que será cifrado, utilizando a chave privada da aplicação. Assim, é possível verificar o id da etiqueta sem necessidade de recurso ao servidor.

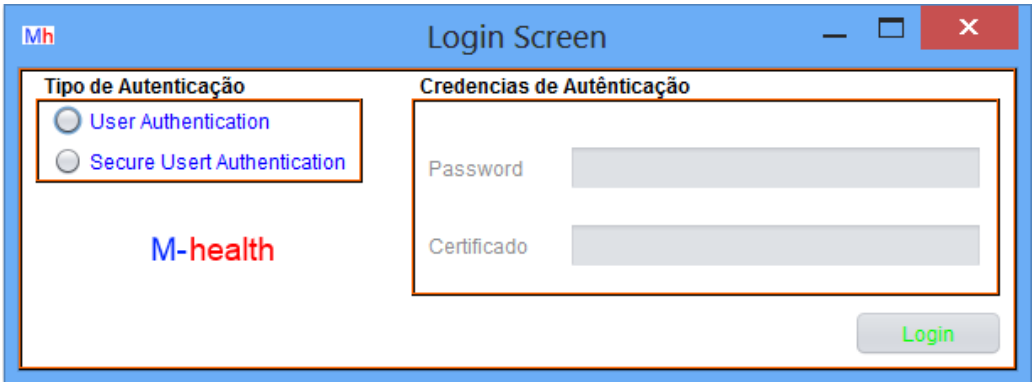


Figura 5.9: Ecrã Inicial

Depois de devidamente autenticado, existem duas possibilidades diferentes dependendo do tipo de autenticação. Para o caso do utilizador UA existe o interface MHUA (Figura 5.10) referido anteriormente. Neste, o utilizador poderá consultar todas as suas prescrições, quem as prescreveu, quando foram prescritas e quais as horas das tomas. O utilizador pode também fazer a confirmação entre etiqueta do medicamento, etiqueta do utilizador e hora da toma. A aplicação irá mostrar uma mensagem de confirmação se as etiquetas lidas corresponderem correctamente entre elas e com a hora (dentro de um limite aceitável). Caso contrário, será mostrada uma mensagem de erro, sendo necessário efectuar novamente a sua leitura.

Os utilizadores SUA terão acesso à interface MHSUA (Figura 5.11). Esta permite aos utilizadores ver a lista de pacientes existentes e toda a informação relacionada com estes: data de nascimento, nome, prescrições activas, profissional que as prescreveu, data da prescrição, etc. Estes utilizadores têm também acesso a uma outra interface (Figura 5.12) onde é possível passar prescrições para os pacientes, especificando os medicamentos, como deverão efectuar a toma destes e qual a validade da prescrição.

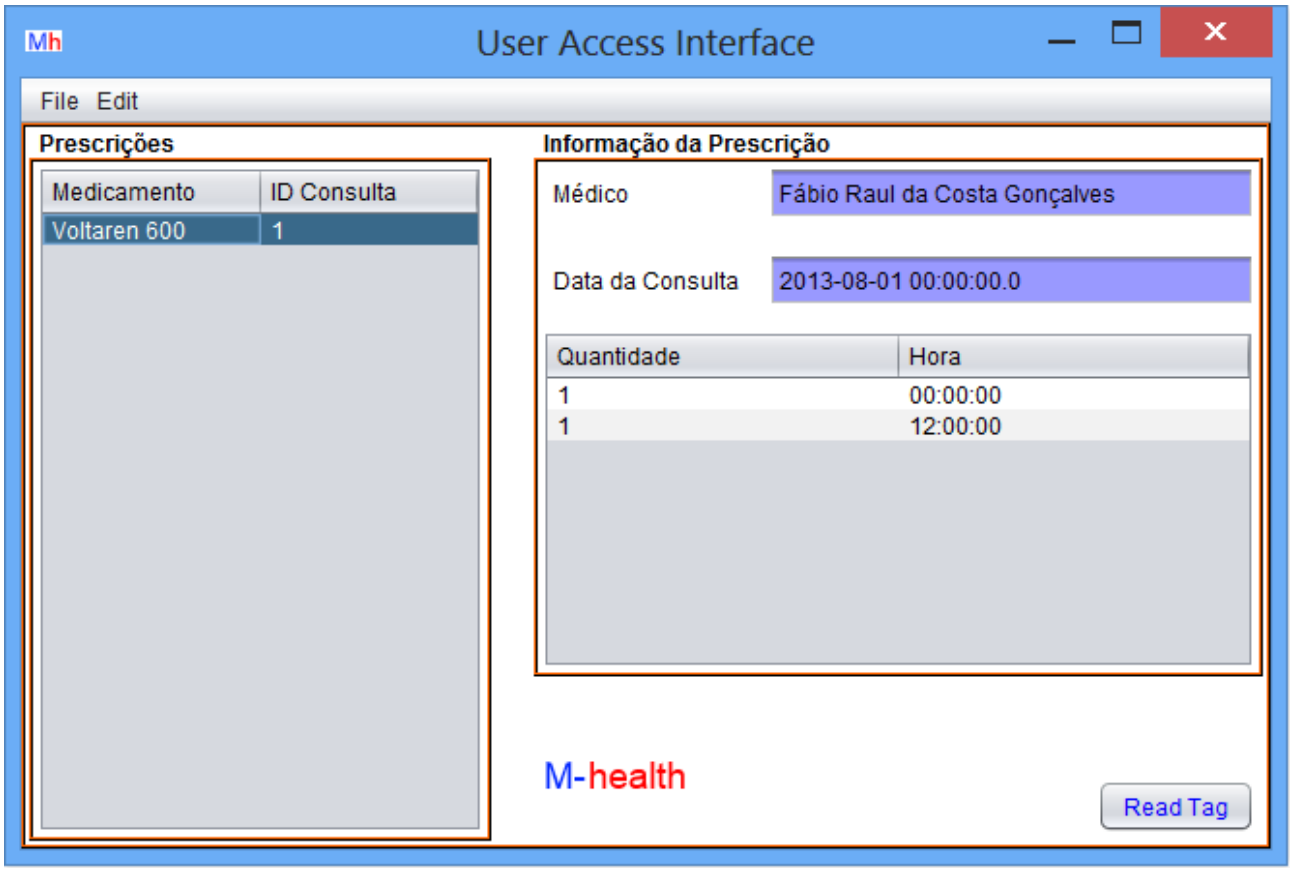


Figura 5.10: Interface MHUA

5.4.2 Aplicação cliente para dispositivos móvel

A aplicação desenhada para dispositivos móveis utiliza o mesmo código de base da aplicação cliente para dispositivos fixos, sendo apenas necessário o desenho de uma interface gráfica para *android* e algumas pequenas alterações a que o sistema *android* obriga, tal como, alterações no caminho para os ficheiros. O sistema de ficheiros de *android* é diferente do utilizado numa plataforma fixa e o caminho para os ficheiros que acompanham a aplicação (como o certificado da aplicação e chave privada desta) são diferentes.

As funcionalidades e interface desta aplicação são limitados, uma vez que esta foi desenhada apenas para testar a compatibilidade dos protocolos com o sistema Android. Esta permite demonstrar que é possível a ligação MHSP entre um dispositivo móvel e o servidor implementado, tal como a autenticação MHUA.

A aplicação foi desenhada e implementada num emulador, tendo sido efectuados alguns testes também num dispositivo móvel. Uma vez que esta não possui um leitor RFID este é simulado através de um identificador de utilizador, como se pode ver na Figura 5.13. Esta necessita ainda da introdução de uma palavra passe, de forma a poder autenticar o utilizador.

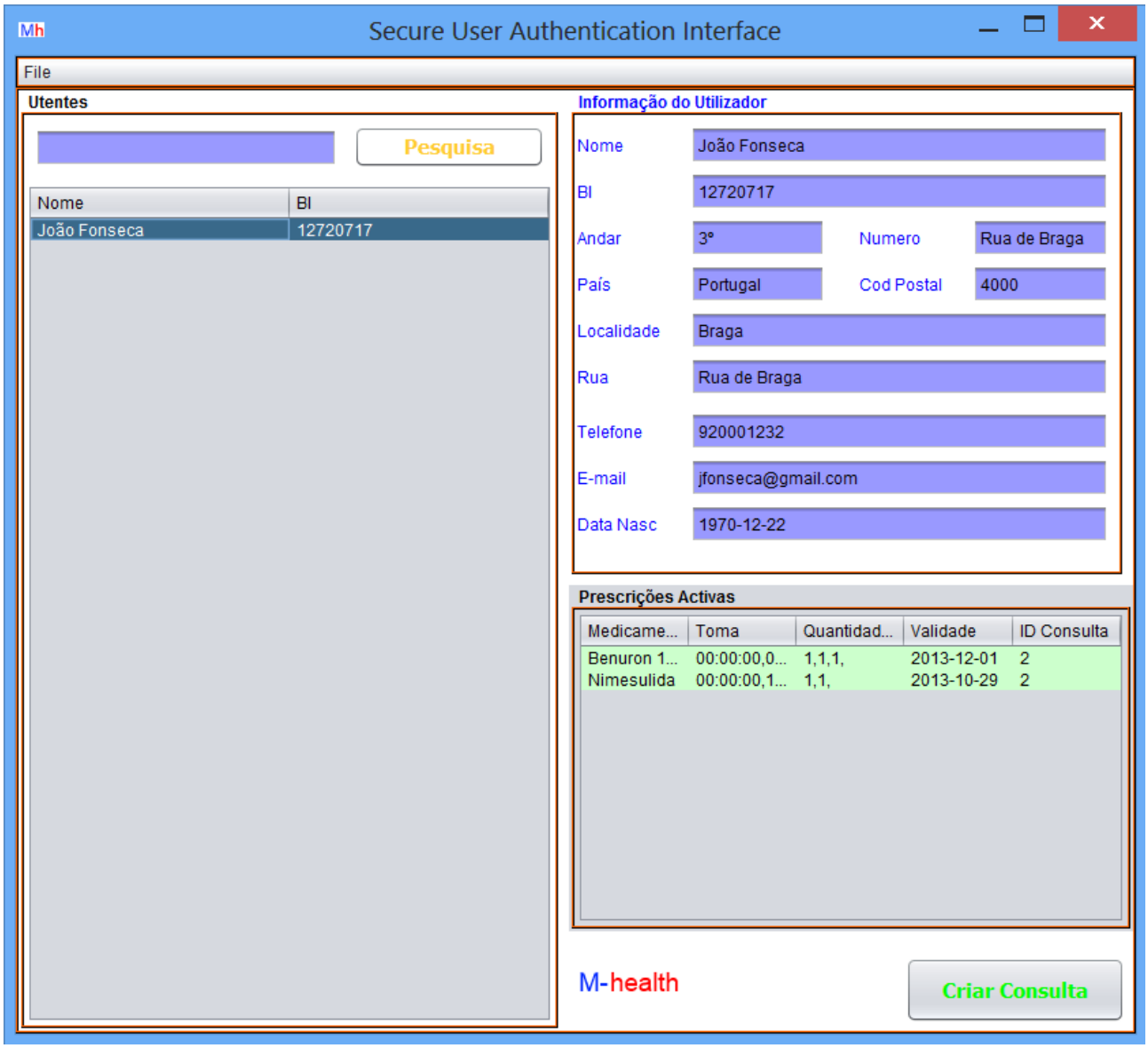


Figura 5.11: Interface MHSUA

Além de permitir uma autenticação MHSP e MHUA, permite também o download de prescrições, como se pode ver na Figura 5.14.

O protocolo MHSUA não foi implementado. O método de autenticação deste é muito semelhante ao do protocolo MHSP e, uma vez que a sua autenticação é efectuada sem qualquer problema, a do protocolo MHSUA será efectuada igualmente.

Foi possível provar que os protocolos propostos são possíveis de implementar em dispositivos móveis e fixos, sem necessidade de grandes alterações para isso.

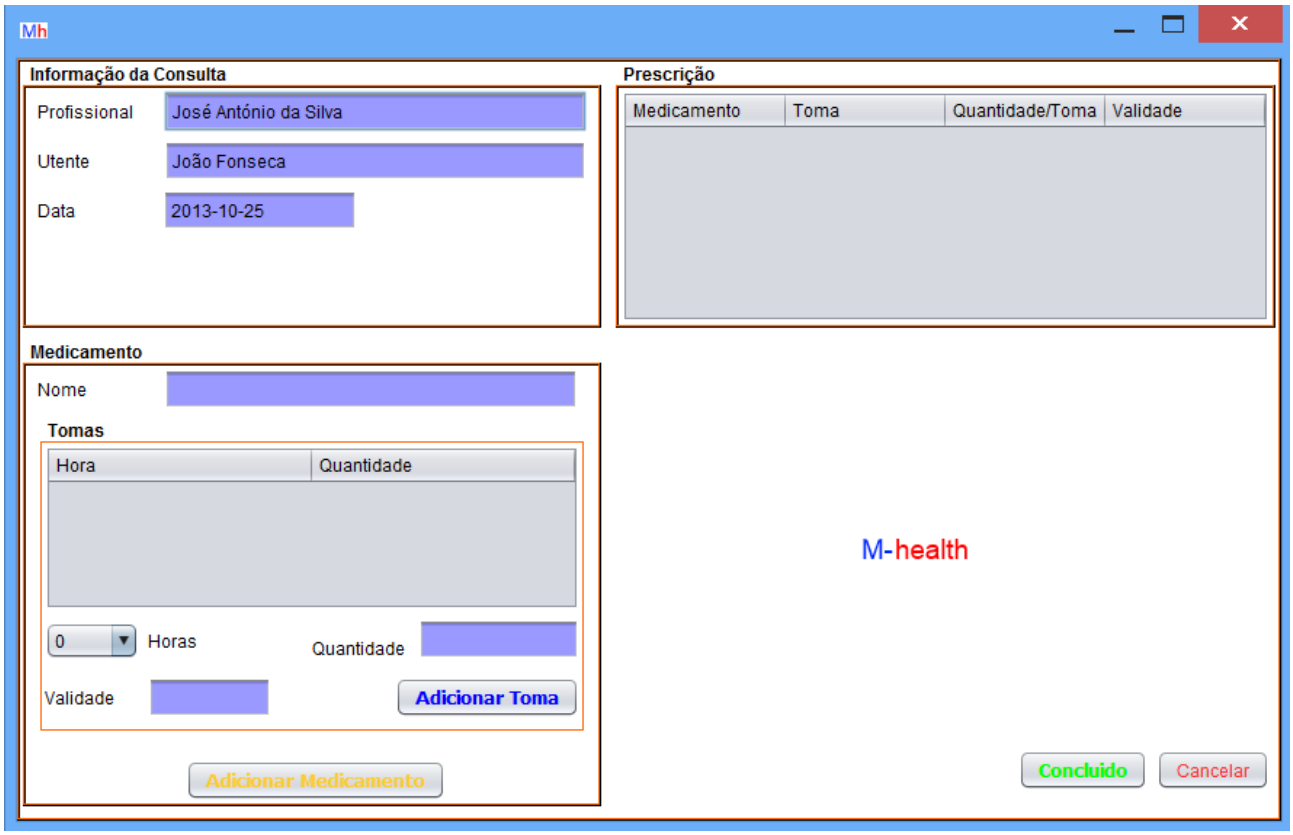


Figura 5.12: Interface MHSUA - Consulta

5.5 Aplicação simula Etiqueta

Uma vez que não foi possível o acesso a nenhum dispositivo RFID, foi desenhada uma aplicação para simular as etiquetas (Figura 5.15). Esta aplicação, embora utilize métodos muito diferentes dos oferecidos pelos dispositivos RFID, permite simular o funcionamento deste.

Esta permite a introdução de IDs de etiquetas e dos PINs correspondentes e cria uma *thread* para cada uma dessas etiquetas. Todas essas *threads* estão à escuta num endereço *multicast* e, quando recebem o pedido do leitor (neste caso a aplicação cliente), geram os dados e enviam para o mesmo endereço *multicast*. Este método permite simular o funcionamento das etiquetas RFID, onde os leitores enviam uma mensagem para energizar as etiquetas que respondem cada uma a seu tempo. É uma aplicação de implementação simples mas permite a simulação eficiente de uma etiqueta RFID.

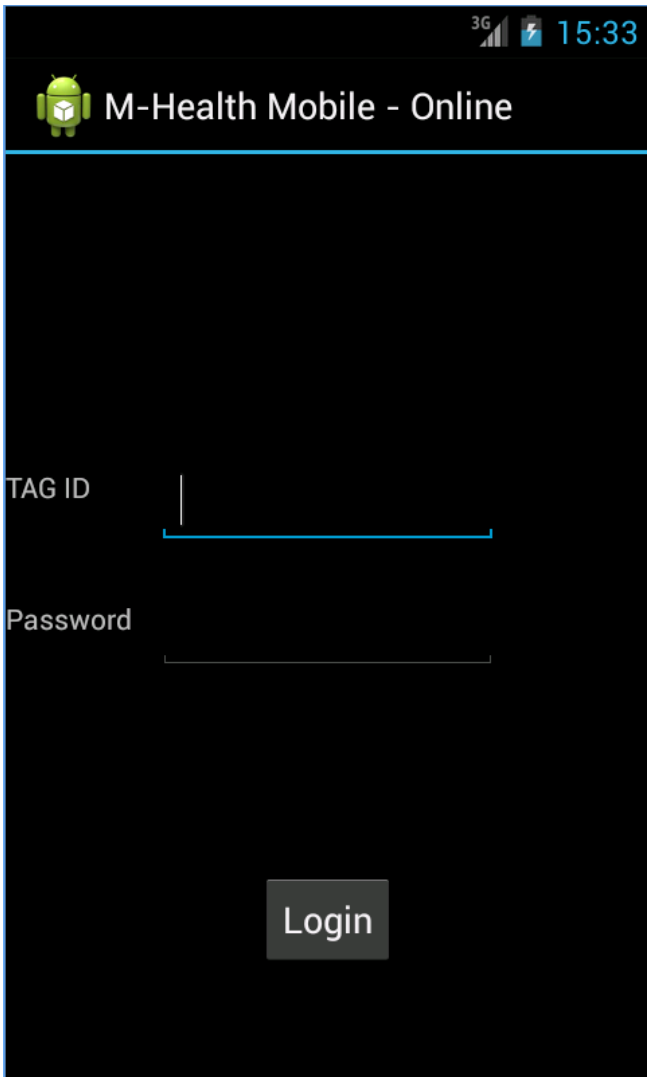


Figura 5.13: Interface MHUA - Dispositivo Móvel

Figura 5.14: Prescrições do Utente - Dispositivo Móvel

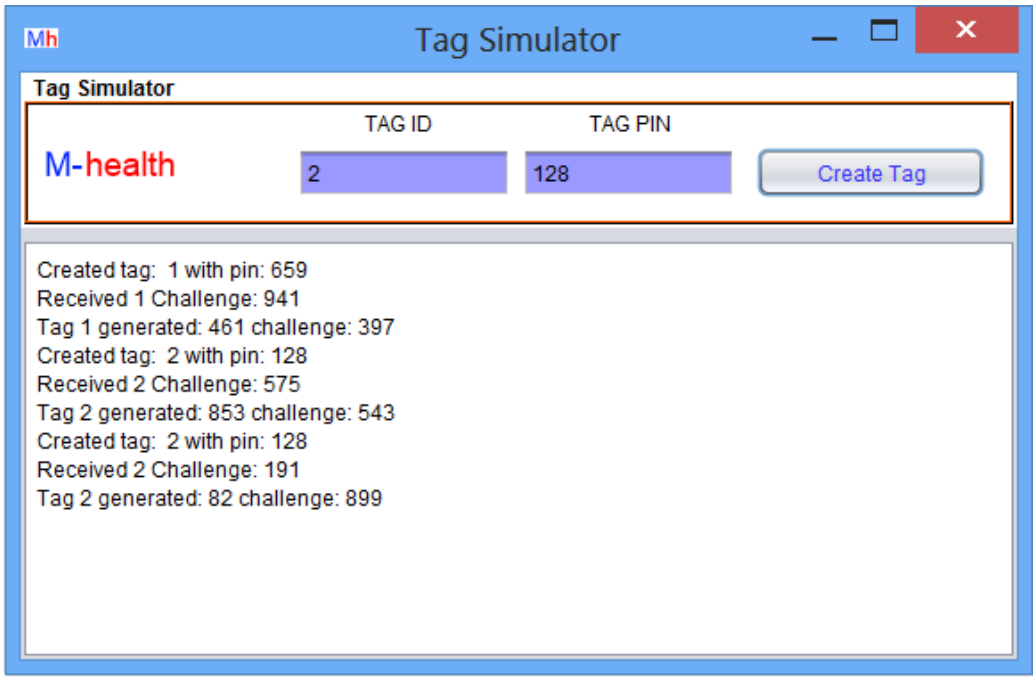


Figura 5.15: Simula Etiqueta

Capítulo 6

Análise de Segurança e Desempenho

Neste capítulo será apresentada uma pequena análise de segurança feita aos protocolos apresentados, de modo a compreender até que ponto estes são seguros e quais as falhas de segurança que podem apresentar.

Será apresentada também uma análise ao desempenho das aplicações desenhadas, para verificar se a segurança implementada compensa a perda de desempenho.

6.1 Análise de Segurança

O protocolo MHSP tem por base o protocolo SSL, que é conhecido pelas componentes de segurança fortes que oferece. Nesse sentido, o protocolo implementado é também seguro, sendo apenas necessário garantir que as cifras utilizadas sejam fortes o suficiente. Um dos pontos fulcrais de toda a segurança do sistema passa pela tempo de vida das chaves utilizadas. Este tem que ser correctamente definido de forma a que estas não sejam quebradas.

Os protocolos de autenticação funcionam sobre o canal seguro já estabelecido com o protocolo MHSP, logo toda a informação é trocada de forma segura.

Para um ataque com sucesso ao sistema, um atacante deve adquirir pelo menos uma aplicação correctamente autenticada com o protocolo MHSP e as credenciais para um dos tipo de autenticação. Na tabela 6.1 podem ser visto os itens necessários para um ataque com sucesso a este protocolo.

Se, tal como feito na implementação da CA (Capítulo 5.1), forem acrescentados ao certificado campos como o endereço MAC da placa de rede do dispositivo e nome do dispositivo, o atacante terá ainda que alterar os valores dessas propriedades no seu dispositivo. Sugere-se ainda que, se duas aplicações forem ligadas ao servidor em simultâneo de dispositivos diferentes, o certificado deve ser imediatamente marcado como inutilizável.

Se por acaso um atacante tiver acesso a um aplicação com o protocolo MHSP já a correr, este tem ainda que se autenticar para aceder ou alterar informações privadas. Para um ataque MHUA com sucesso o atacante deverá possuir, além da aplicação correctamente autenticada, os itens indicados na tabela 6.1.

Para obter a ID do utilizador terá que conseguir quebrar a segurança implementada no canal RFID. Mesmo depois de a obter, terá ainda que descobrir qual a palavra passe do utilizador.

Pode verificar-se nas tabela 6.1 os itens necessários para um ataque com sucesso ao protocolo MHSUA. Neste caso, além dos itens necessários para uma autenticação MHUA, este necessitará ainda um certificado do utilizador. O certificado terá que ser válido e correctamente assinado pela CA, caso contrário será rejeitado.

Pode ainda ver se na tabela 6.1, o que um atacante terá que possuir para poder decifrar com sucesso um dos ficheiros armazenados com o protocolo MHOUA.

Protocolo	Dados necessários para Ataque com Sucesso
MHSP	Certificado de aplicação válido, Chave secreta do certificado válido, Aplicação
MHUA	Palavra passe do utilizador, ID do utilizador
MHSUA	Palavra passe do utilizador, ID do utilizador, Certificado do utilizador
MHOUA	Palavra passe do utilizador ID do utilizador

Tabela 6.1: Itens necessários para ataque com sucesso aos protocolos propostos

Podemos concluir que os protocolos aqui apresentados são seguros, principalmente contra ataques em grande escala. Para um atacante tentar apenas ver a informação trocada entre aplicações terá que possuir pelo menos o certificado da aplicação e a chave privada correspondente. De realçar, que este deverá ter uma validade e que ficará invalido quando esta expirar.

Sendo assim, é possível atacar o sistema, mas torna-se impraticável, principalmente em larga escala. Um atacante necessita de obter demasiada informação do utilizador e da aplicação, sendo necessário acesso físico aos dispositivos.

O sistema pode ser sensível a um ataque de *Denial of Service* (DOS) mas, uma vez que neste caso o sistema foi desenhado para a toma de medicação em ambulatório, se este estiver offline durante alguns minutos não será problemático, principalmente uma vez que o sistema

permite guardar dados temporários nos seus dispositivos.

Além dos pontos aqui referidos, a aplicação da BD guarda registos detalhados de todas as conexões efectuadas. Assim, é possível verificar se houve tentativas de ataques e impedir que aconteçam novamente.

A aplicação servidor guarda também registos de todas as operações efectuadas. Assim, se apesar de todas as adversidades algum atacante conseguir ter sucesso, é possível verificar com se procedeu o ataque, impedir futuros ataques e ainda é reverter algumas ou mesmo todas as alterações não autorizadas.

Foram efectuados alguns testes recorrendo às ferramentas Wireshark (Disponível em `http://www.wireshark.org/`) e RawCap (Disponível em http://www.netresec.com/). A ferramenta RawCap permite capturar pacotes trocados na rede, mais precisamente na interface *loopback*. A ferramenta Wireshark permite analisar esses pacotes e verificar o seu conteúdo, tornando possível verificar qual a informação trocada entre as aplicações e se esta é facilmente visível na rede. As Figuras 6.1,6.2, 6.3 permitem ver as mensagens trocadas entre a aplicação cliente e aplicação servidor. A Figura 6.1 corresponde ao início do protocolo, fase onde a aplicação cliente envia o seu certificado. Na seguinte fase (Figura 6.2), o servidor envia o seu certificado. Finalmente, temos a representação de parte da troca de chaves entre as aplicações (Figura 6.3). Como se pode verificar, apenas a informação trocada nas duas primeiras fases é visível. Esta informação é pública, logo não existe necessidade de a cifrar. Na fase de troca de chaves é impossível conseguir perceber qual a informação está a ser trocada, uma vez que esta está devidamente cifrada. Assim, podemos concluir que o protocolo, tal como proposto, segura com sucesso todos os dados trocados entre as aplicações.

6.2 Análise de Desempenho

Para uma melhor verificação do desempenho das aplicações e dos protocolos desenhados, foi desenhada uma aplicação especial que tem como objectivo efectuar uma mesma operação diversas vezes e calcular o tempo em gasto em média. Os testes foram efectuados estando todas as aplicações na mesma máquina[1]. Será possível melhorar estes resultados se as aplicações estiverem distribuídas por diferentes máquinas e se a aplicação servidor estiver situada numa máquina que seja optimizada para desempenhar essa função. A vantagem de ter todas as aplicações numa só máquina é minimizar a interferência de possíveis atrasos na rede. Os

[1] Computador Portátil com um CPU de 2.00 GHZ com 4 núcleos, 8 GB de RAM e sistema operativo Windows 8

Figura 6.1: MHSP - Certificado do Cliente

Figura 6.2: MHSP - Certificado do Servidor

resultados obtidos podem ser verificados na tabela 6.2.

Uma vez que existem outras aplicações em execução na máquina onde foram efectuados os testes, existem picos nos tempos de execução. De forma a minimizar esses picos, cada um dos testes foi efectuado 100 vezes, sendo a média o valor colocado na tabela. Esse número (100) foi calculado por tentativa e erro e foram verificados qual o número de testes necessários de forma a minimizar o desvio padrão das amostras. Foi possível verificar que, à medida que se aumentava o número de testes, o valor do desvio padrão diminuía. O desvio padrão obtido para

Figura 6.3: MHSP - Troca de chaves cifradas

100 testes apresenta já um valor bastante baixo, o qual tem um decrescimento pouco acentuado a partir desse número.

Operação	Dados (Bytes)	Tempo (Segundos)	Desvio Padrão (Segundos)
Ligação MHSP	-	0.862	0.039
Autenticação MHUA	-	1.020	0.068
Autenticação MHSUA	-	1.170	0.084
Download de Dados MHSP	1000	0.055	0.004
Download de Dados MHSP	10000	0.059	0.011
Download de Dados MHSP	100000	0.078	0.008
Download de Dados MHSP	1000000	0.305	0.017
Download de Sem segurança	1000000	0.154	0.016

Tabela 6.2: Desempenho dos Protocolos

Como se pode verificar na tabela 6.2, apesar da segurança implementada, pode concluir-se que o protocolo consegue assegurar um bom desempenho. Para termo de comparação, foi testado qual o tempo que duas aplicações, também desenhadas em Java, levam para trocar um pacote de 1000000 bytes entre si, sem qualquer tipo de segurança. Depois de efectuar alguns testes, na máquina utilizada para efectuar também os testes anteriores, verificou-se que esse tempo era de aproximadamente 0.154 segundos. A transmissão do mesmo pacote de dados utilizando o protocolo MHSP leva cerca de 0.305, ou seja, cerca de 0.15 segundos a mais do que

sem qualquer segurança.

Utilizando a ferramenta para análise de tráfego Wireshark foi possível verificar qual o *overhead* introduzido pelo protocolo desenhado. Para efectuar o download de um pacote de 1000 Bytes, o tráfego realizado na verdade é de 8665 Bytes, sendo que parte deste tráfego é utilizado para a autenticação efectuada pelo protocolo MHSP. Esta autenticação é sempre necessária de modo a obter um canal seguro antes de trocar informação entre as aplicações e, necessita de um total de 6779 bytes para autenticar correctamente as aplicações. Logo, para trocar 1000 bytes entre as duas aplicações utilizando um canal seguro com protocolo MHSP, necessita de 1886 bytes, sendo o *overhead* real introduzido de apenas 886 bytes. Para termo de comparação, foi efectuado o mesmo teste, utilizando um canal não seguro, introduzindo um *overhead* de apenas 35 bytes.

Pode verificar-se que, como em qualquer outro sistema, à medida que se aumenta a segurança total do sistema existe uma perda no desempenho. Uma vez que este acontecimento é inevitável, é importante que se minimize essa perda e que se considere as perdas *versus* ganhos que a relação desempenho/segurança acarretam. Neste caso, tal como se pode verificar pelos resultados indicados na tabela, pode concluir-se que existe uma pequena perda de desempenho devido aos fortes mecanismos de segurança implementados. Estes são indispensáveis devido à natureza sensível deste. Logo, os ganhos que o sistema oferece (forte componente de segurança) compensa as perdas deste (atraso nas comunicações).

Capítulo 7

Conclusões e Trabalho Futuro

O custo cada vez mais acrescido dos tratamentos de saúde obriga a uma relocação dos pacientes, se possível para a sua residência, reduzindo o custo desses tratamentos. No entanto, este procedimento leva a um aumento de erros que ocorrem durante os tratamentos, principalmente quando se trata de pacientes com uma certa idade. Os erros nas tomas de medicamentos podem ocorrer devido a erros efectuados durante a comunicação das prescrições ou durante a toma dos mesmos. Foi visto que, utilizando a *pervasive* Internet era possível aceder à informação a partir de qualquer, local tornando possível o uso da autenticação automática dos participantes e, assim, automatizar os processos e reduzir o acontecimento de erros.

O ambiente que rodeia este trabalho, a saúde, tem uma natureza muito sensível. A informação que este contém é privada e a sua fuga ou alteração pode levar a consequências catastróficas. A automatização dos processos na administração da medicação traz consigo alguns riscos de segurança, uma vez que um ataque com sucesso pode dar acesso a grandes quantidades de informação, tornando-se assim mais apetecível a potências atacantes.

É de relevar a importância de implementar fortes mecanismos de segurança, de forma a evitar o tratamento errado, identificação errada, ou acesso não autorizado. Deve ser mantido um registo de todos os eventos para ser possível detectar qualquer ataque ao sistema ou até casos de negligência. De modo a resolver as possíveis falhas de segurança que o sistema poderia, ter é aqui proposta uma nova arquitectura de segurança.

Esta arquitectura foi desenhada de forma a cumprir os fortes requisitos de segurança necessários a um sistema deste tipo e minimizar ou mesmo eliminar potências ataques e riscos de segurança. Para este desenho, definiu-se qual o cenário (Figura 1.1) onde esta iria ser aplicada. Este cenário é composto por todas as entidades que poderão interagir com o sistema e as ligações entre si. Aqui todos os profissionais de saúde, utentes e medicamentos são identificados através

de etiquetas RFID.

No cenário desenhado assume-se que os profissionais de saúde, utilizando um portátil ou dispositivo móvel, preenchem as prescrições, indicando a dose e a hora a que a esta deve ser tomada. Esta Informação que é guardada na BD e ligada ao identificador RFID dos utentes. Posteriormente, os utentes utilizando um dispositivo móvel poderão aceder ao sistema e descarregar ou actualizar as suas prescrições. A conexão efectuada entre utente, medicamento e prescrição deve ser feita utilizando identificadores RFID.

Depois de se ter chegado ao cenário de aplicação e identificado devidamente as suas entidades e interacções, passou-se à sua análise, de forma a descobrir as suas fragilidades de segurança. Nesta análise foram identificados quatro grandes grupos de fragilidades: nas comunicações entre dispositivos, na ligação à BD, no armazenamento de dados e na autenticação. As fragilidades encontradas nas comunicações entre os dispositivos acontecem nas comunicações sobre a Internet e sobre RFID. Relativamente ao armazenamento de dados é necessário garantir a sua segurança na BD e nos dados guardados nos dispositivos dos utilizadores para consulta offline. A nível da autenticação é imperativo garantir a correcta autenticação: dos utilizadores perante a aplicação, entre aplicações e das aplicações perante a base de dados.

A arquitectura de segurança proposta e apresentada na Figura 4.1 visa eliminar as fragilidades identificadas. Esta foi construída a partir do cenário de aplicação, mas apresenta os protocolos através dos quais as entidades irão comunicar entre si. Foram investigados diversos protocolos de forma a averiguar que tipo de segurança se poderia aplicar nos diversos pontos de fragilidade e como se poderia aplicar tal segurança. A arquitectura desenhada acrescenta também uma CA necessária para o funcionamento de alguns dos protocolos utilizados.

De forma a resolver o problema das comunicações entre dispositivos RFID, foi encontrado durante a investigação o protocolo IS-RFID proposto por [11]. Este foi desenhado por [11] para ser utilizado num ambiente de M-Health e oferece mecanismos para efectuar a correspondência entre utente, medicação e prescrição. O protocolo IS-RFID oferece um bom nível de segurança que depende da entropia do PIN. Este pode ser ainda complementado com o protocolo Puzzles Criptográficos (PC) e assim aumentar a segurança total do sistema.

Foram investigados diversos protocolos que poderiam ser aplicados nas diversas camadas da pilha TCP/IP, para resolver o problema das comunicações sobre a internet. Depois de alguma investigação concluiu-se que o ideal seria aplicar segurança a nível da aplicação. Como não foi encontrado qualquer protocolo normalizado para isso, foi decidido que irá ser proposto um novo protocolo. Este tem nome de *M-Health Security Protocol* (MHSP) e tem por base o protocolo SSL. O protocolo MHSP permite resolver o problema das comunicações sobre a

Internet tal como o problema da autenticação das aplicações.

A autenticação dos utilizadores é um problema um pouco mais complexo porque, para uma arquitectura para um sistema tão sensível (como é o caso da saúde) é importante ter uma boa relação entre a segurança e a disponibilidade do sistema. O acesso a dados privados pode ser mau para o sistema mas, em caso de emergência, a não disponibilidade destes pode ser catastrófica para a vida dos pacientes. Assim, foram criados dois tipos de acesso para os utilizadores: *User Authentication* (UA) e *Secure User Authentication* (SUA). O primeiro permite apenas operações de leitura tendo requisitos de segurança mais relaxados, permitindo um acesso mais fácil à informação e acesso online e offline ao sistema. O tipo de acesso SUA permite operações de leitura e escrita, tendo uma segurança mais apertada. Este tipo de acesso tem componente de segurança mais forte uma vez que, a alteração de dados relativos a doentes pode ser extremamente danoso para o sistema ou mesmo para a vida dos pacientes.

Os utilizadores terão que estar registados de forma a poder aceder ao sistema. O processo de registo é efectuado pela CA e deve ser efectuado em pessoa para evitar erros de identificação. Durante este processo vão ser dados determinados itens ao utilizador que dependem do tipo de acesso que estes têm. Para os utilizadores com acesso UA, vai lhes ser fornecido uma palavra-passe (preferencialmente através de um canal paralelo como por exemplo, um envelope selado) e uma etiqueta RFID. Os utilizadores com acesso do tipo SUA vão receber uma palavra-passe, uma etiqueta RFID e um certificado de chave pública (juntamente com a chave privada correspondente).

Foram desenhados dois protocolos para permitir os tipos de acesso referidos anteriormente. Para o tipo de acesso UA foi proposto o protocolo*M-Health User Authentication* (MHUA) e para o acesso SUA o protocolo MHSUA. O protocolo MHUA é simplesmente um protocolo de nome de utilizador e palavra-passe; a aplicação lê o identificador da etiqueta do utilizador e pede a palavra passe. Esses dados são enviados para o servidor para serem verificados. O protocolo MHSUA é mais complexo que o anteriormente referido. Este, tal como o MHSP, tem por base o protocolo SSL e serve-se de certificados de chave pública para autenticar os utilizadores. O certificado deve ser acompanhado por uma chave privada que é cifrada utilizando uma chave simétrica. Esta obtida a partir do *hash* do tuplo `<palavra passe, ID do utilizador>`, fazendo que só seja possível obter pelo respectivo dono. Durante a fase de autenticação, todos os dados são trocados sobre o canal já seguro com o protocolo MHSP.

Os protocolos falados anteriormente não permitem que os utilizadores acedam aos seus dados sem estarem ligados à Internet. Com esse fim, foi criado um outro protocolo denominado de MHOUA. Este necessita de um acesso online correctamente autenticado para poder obter

os dados e armazena-os de forma segura no dispositivo do utilizador, para posterior consulta offline. Para isso, utiliza uma chave simétrica obtida do *hash* do tuplo `<palavra passe do utilizador, id da etiqueta, nome do ficheiro, data de criação do ficheiro>`. As prescrições devem ter uma data de validade que as tornas inutilizáveis após expirar.

A BD encontra-se num local fisicamente seguro, não sendo necessário segurar os dados aí guardados; caso contrário seria, necessário cifrar os dados. Este caso traria ainda outro problema que se prende pela chave que seria usada para cifrar a BD e como e onde seria armazenada.

Para a ligação à BD propõe-se a utilização de uma aplicação especial que corra o protocolo MHSP e permita encaminhar dados de um servidor para a BD. Esta seria colocada na mesma máquina da BD e, todas as ligações efectuadas dessa e para essa máquina serão apenas e só através dessa aplicação. Assim, resolve-se o problema da ligação e autenticação das aplicações perante a BD.

De forma a poder verificar se os protocolos e a arquitectura propostos são seguros, funcionais e fornecem um bom desempenho, foram implementadas aplicações em Java (Aplicações Fixas) e android (Aplicações móveis), para representar as diversas entidades. O sistema implementado permite apenas operações básicas. Entre estas podemos identificar as seguintes: o registo de utentes e profissionais de saúde, a possibilidade de registar consultas e emitir prescrições. Além das operações referidas, o sistema implementado permite testar todos os protocolos aqui propostos, já que também permite também a autenticação dos utilizadores e segurar os dados trocados entre as aplicações com os protocolos aqui propostos.

Uma vez que não houve acesso a dispositivos RFID, foi desenhada uma aplicação que permitisse simular este tipo de comunicações. Assim, foi possível verificar a funcionalidade do protocolo IS-RFID. A aplicação desenhada utiliza *sockets Multicast* para efectuar a simulação.

Após desenhada a arquitectura de segurança foram efectuadas análises de segurança e desempenho. Concluiu-se que o protocolo MHSP tem por base um protocolo conhecido pela segurança que fornece, sendo o MHSP também seguro desde que as cifras utilizadas sejam fortes o suficiente e o tempo de vida das chaves seja correctamente definido. Verificou-se que, para um ataque com sucesso, é necessário no mínimo uma aplicação que esteja a correr o protocolo MHSP correctamente autenticada (necessita da aplicação, do certificado da aplicação e da chave privada a este associada; implica acesso físico ao dispositivo), da palavra passe e o id da etiqueta RFID do utilizador, isto para ter apenas acesso MHUA. Para o acesso MHSUA, além dos itens referidos, o atacante necessitará ainda do certificado do utilizador e da sua chave privada. Foram ainda efectuados testes utilizando a ferramenta *wireshark* que permitiu verificar

que todos os dados privados trocados entre aplicações são correctamente cifrados.

Foi possível concluir que o sistema desenhado oferece mecanismos fortes de segurança e que tornam um ataque ao sistema inexequível (principalmente em larga escala), uma vez que o atacante necessita de obter demasiada informação da aplicação e do utilizador.

Na análise ao desempenho verificou-se que os protocolos desenhados introduzem algum *overhead* e atraso ao sistema desenhado mas, que mesmo assim, oferecem um boa relação segurança/desempenho.

Os protocolos aqui apresentados foram todos eles desenhados para ser utilizados no ambiente de M-Health. Ainda assim, estes apresentam um funcionamento bastante genérico e deverão poder ser aplicados em qualquer cenário.

O trabalho efectuado durante o percurso da dissertação originou também um artigo científico *"Security Architecture for Mobile E-Health Applications in Medication Control"* [9] apresentado na conferência *"SoftCom 2013 - 21th International Conference on on Software, Telecommunications and Computer Networks"* na Croácia. Neste artigo foram publicados todos os protocolos propostos nesta dissertação, tal como a arquitectura aqui apresentada.

De seguida serão propostos alguns objectivos que tem como fim melhorar o sistema desenhado, tanto a nível de segurança como desempenho e usabilidade. O principal objectivo proposto para trabalho futuro prende-se pela eliminação do problema encontrado durante processo de análise de segurança, relacionado com a chave privada da aplicação. Esta não é cifrada e é susceptível a ataques, sendo necessário apenas o acesso físico ao sistema. Pretende-se no futuro utilizar algum método para guardar esta chave de forma segura no dispositivo.

O sistema desenhado pode ser sensível a ataque DOS; este tipo de ataques foi um pouco negligenciado, uma vez que esta arquitectura foi desenhada para a administração de medicação em ambulatório. Nesse sentido, um ataque DOS que deixe o sistema inutilizável durante algum tempo não tem quaisquer consequências para o sistema. No entanto, o acesso ou alteração a informações privadas pode ter consequências catastróficas.

Futuramente, pretende-se efectuar testes exaustivos de segurança, desempenho e usabilidade de forma a melhor avaliar as limitações da arquitectura desenhada e a sua implementabilidade num sistema real.

A implementação de um sistema de rastreamento do dispositivo traria a possibilidade de encontrar o dispositivo se este fosse roubado ou perdido. Desta forma, poderia impedir-se certos gastos acrescidos para o sistema ou para o utente lesado.

As aplicações desenhadas permitem apenas operações básicas de gestão, tal como, a inserção, remoção e listagem de medicamentos e utilizadores, criação de consultas e emissão de

prescrições. As funções de gestão devem ser melhoradas, de modo a poder implementar-se o sistema num ambiente real.

Seria ideal a interligação do sistema com o sistema de gestão existente nas farmácias. Essa interligação tornaria possível operações como por exemplo, o envio de medicamentos para casa do utente quando algum destes estivesse a terminar. Assim, poderia-se-ia simplificar bastante a vida dos utentes, principalmente os com mobilidade condicionada.

Como foi referido anteriormente, a autenticação efectuada pelo protocolo MHSUA serve-se da utilização de um *smart card*. Em Portugal o documento de identificação pessoal, o Cartão de Cidadão, é um *smart card*. Este oferece mecanismos de autenticação como certificados de chave pública e a possibilidade de criar assinaturas.

Durante o trabalho efectuado nesta dissertação, foi efectuada alguma investigação sobre como incorporar o Cartão de Cidadão na arquitectura de segurança. Para tal, é necessário efectuar algumas mudanças ao sistema desenhado. Sugere-se a troca do tipo dos certificados utilizados para o tipo X509, uma vez que são estes os utilizados pelo Cartão do Cidadão, normalizando-se o tipo dos certificados utilizados.

A aplicação desenhada deve ser capaz de ler os certificados do Cartão de Cidadão e de verificar a sua autenticidade. O protocolo MHSUA teria que ser alterado, uma vez que agora seria o cartão a cifrar e assinar mensagens com a chave privada em vez da aplicação. O processo de registo poderia ser efectuado automaticamente recorrendo ao Cartão do Cidadão. Este possui toda a informação relativa a quem o possui.

A implementação do Cartão de Cidadão traria vantagens para o sistema, principalmente a nível de usabilidade. Todos os utentes possuem (ou possuirão num futuro próximo) cartão de cidadão, não necessitando de possuir um outro cartão para a autenticação no sistema de medicação. No entanto, sugere-se uma investigação aprofundada para averiguar os possíveis problemas que este processo poderia trazer, a nível de implementação e segurança.

Referências

[1] D. M. Benjamin, "Reducing medication errors and increasing patient safety: case studies in clinical pharmacology," *The Journal of Clinical Pharmacology*, vol. 43, no. 7, pp. 768–783, 2003. available at http://onlinelibrary.wiley.com/doi/10.1177/0091270003254794/pdf (Online, última visita em 31/10/2013).

[2] S. Meredith, P. H. Feldman, D. Frey, K. Hall, K. Arnold, N. J. Brown, and W. A. Ray, "Possible medication errors in home healthcare patients," *Journal of the American Geriatrics Society*, vol. 49, no. 6, pp. 719–724, 2001. available at http://onlinelibrary.wiley.com/doi/10.1046/j.1532-5415.2001.49147.x/pdf (Online, última visita em 31/10/2013).

[3] H. B. Elhadj, N. Bradai, L. Chaari, and L. Kamoun, "A survey of security proposals and issues in wireless body area networks for healthcare applications," 2012. available at http://onlinelibrary.wiley.com/doi/10.1046/j.1532-5415.2001.49147.x/pdf (Online, última visita em 31/10/2013).

[4] D. Hamilton, "Application layer security requirements of a medical information system," in *National Computer Security Conference Proceedings, 1992: Information Systems Security*, p. 9, DIANE Publishing, 1992.

[5] S. Hameed, H. Yuchoh, and W. Al-Khateeb, "A model for ensuring data confidentiality: In healthcare and medical emergency," in *Mechatronics (ICOM), 2011 4th International Conference On*, pp. 1–5, 2011. available at http://dx.doi.org/10.1109/ICOM.2011.5937139 (Online, última visita em 31/10/2013).

[6] A. Belbachir, M. Drobics, and W. Marschitz, "Ambient assisted living for ageing well–an overview," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 7-8, pp. 200–205, 2010. available at http://link.springer.com/article/10.1007/s00502-010-0747-9 (Online, última visita em 31/10/2013).

[7] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, no. 1, pp. 49–69, 2011. available at http://link.springer.com/article/10.1007/s11277-011-0288-5 (Online, última visita em 31/10/2013).

[8] M. A. P. Henriques, *Adesao ao regime medicamentoso em idosos na comunidade: eficacia das intervencoes de enfermagem (in Portuguese)*. PhD thesis, Universidade de Lisboa, 2011. available at http://hdl.handle.net/10451/3801 (Online, última visita em 31/10/2013).

[9] F. Gonçalves, J. Macedo, M. J. Nicolau, and A. Santos, "Security architecture for mobile e-health applications in medication control," SoftCom 2013 - 21th International Conference on on Software, Telecommunications and Computer Networks (ISBN 978-953-290-041-5), 2013, FESB, University of Split, 18-20 Sep, 2013.

[10] R. Want, "Near field communication," *Pervasive Computing, IEEE*, vol. 10, no. 3, pp. 4–7, 2011. available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5958681 (Online, última visita em 31/10/2013).

[11] P. Peris-Lopez, A. Orfila, A. Mitrokotsa, and J. C. van der Lubbe, "A comprehensive {RFID} solution to enhance inpatient medication safety," *International Journal of Medical Informatics*, vol. 80, no. 1, pp. 13 – 24, 2011. available at http://dx.doi.org/10.1016/j.ijmedinf.2010.10.008 (Online, última visita em 31/10/2013).

[12] P. Peris-Lopez, J. Hernandez-Castro, J. Tapiador, E. Palomar, and J. C. A. van der Lubbe, "Cryptographic puzzles and distance-bounding protocols: Practical tools for rfid security," pp. 45–52, 2010. available at http://dx.doi.org/10.1109/RFID.2010.5467258 (Online, última visita em 31/10/2013).

[13] D. E. Robling Denning, *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc., 1982. available at http://dl.acm.org/citation.cfm?id=539308 (Online, última visita em 31/10/2013).

[14] W. Stallings, *Cryptography and Network Security: Principles and Practice (6th Edition)*. Prentice Hall, 2013. 752 pages,ISBN: 978-0133354690.

[15] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to elliptic curve cryptography*. Springer, 2004.

[16] J. R. Black Jr, *Message authentication codes*. PhD thesis, UNIVERSITY OF CALIFORNIA, 2000. available at http://www.cs.colorado.edu/~jrblack/papers/thesis.pdf (Online, última visita em 31/10/2013).

[17] S. H. Standard, "Federal information processing standard publication# 180," vol. 56, pp. 57–71, 1993. available at http://csrc.nist.gov/publications/fips/fips185/fips185.txt (Online, última visita em 31/10/2013).

[18] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer, 2002.

[19] R. Anderson, E. Biham, and L. Knudsen, "Serpent: A proposal for the advanced encryption standard," 1998. available at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.5585 (Online,08/10/2013).

[20] M. S. Bhiogade, "Secure socket layer," *InSITE-Where Parallels Intersetc*, 2002. available at http://dx.doi.org/10.2139/ssrn.291499 (Online, última visita em 31/10/2013).

[21] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya, "Merkle-damgard revisited: How to construct a hash function," in *Advances in Cryptology - CRYPTO 2005* (V. Shoup, ed.), vol. 3621 of *Lecture Notes in Computer Science*, pp. 430–448, Springer Berlin Heidelberg, 2005. available at http://dx.doi.org/10.1007/11535218_26 (Online, última visita em 31/10/2013).

[22] A. Houghton, "Cyclic redundancy checking," in *The Engineer's Error Coding Handbook*, pp. 12–24, Springer US, 1996. available at http://dx.doi.org/10.1007/978-1-4613-0447-0_3 (Online, última visita em 31/10/2013).

[23] J. Pieprzyk, T. Hardjono, and J. Seberry, "Digital signatures," in *Fundamentals of Computer Security*, pp. 283–305, Springer Berlin Heidelberg, 2003. available at http://dx.doi.org/10.1007/978-3-662-07324-7_7 (Online, última visita em 31/10/2013).

[24] General Services Administration Office of Governmentwide Policy, Smart Card Interoperability Advisory Board, *Govermnment smart card handbook*. February 2004. available at http://www.smartcardalliance.org/resources/pdf/smartcardhandbook.pdf/ (Online, última visita em 27/09/2013).

[25] Kruthi, "Smart cards," June 2002. available at http://ewh.ieee.org/r10/bombay/news5/SmartCards.htm (Online, última visita em 10/08/2013).

[26] I. Laranjo, J. Macedo, and A. Santos, "Internet of things for medication control: Service implementation and testing," *Procedia Technology*, vol. 5, no. 0, pp. 777 – 786, 2012. available at http://dx.doi.org/10.1016/j.protcy.2012.09.086 (Online, última visita em 31/10/2013).

[27] GS1 EPCgloba, "Tag classification definitions," November 2007. available at http://www.gs1.org/docs/epcglobal/TagClassDefinitions_1_0-whitepaper-20071101.pdf/ (Online, última visita em 28/09/2013).

[28] H.-Y. Chien, C.-C. Yang, T.-C. Wu, and C.-F. Lee, "Two rfid-based solutions to enhance inpatient medication safety," *Journal of Medical Systems*, vol. 35, no. 3, pp. 369–375, 2011. available at http://dx.doi.org/10.1007/s10916-009-9373-7 (Online, última visita em 31/10/2013).

[29] P. Syverson, "Weakly secret bit commitment: Applications to lotteries and fair exchange," in *Computer Security Foundations Workshop, 1998. Proceedings. 11th IEEE*, pp. 2–13, IEEE, 1998.

[30] J. Fairfield, "Databases," *Anaesthesia & Intensive Care Medicine*, vol. 5, no. 12, pp. 407 – 409, 2004. available at http://dx.doi.org/10.1383/anes.5.12.407.55122 (Online, última visita em 31/10/2013).

[31] A. Hope and G. Forrest, "Databases," *Anaesthesia & Intensive Care Medicine*, vol. 11, no. 12, pp. 495 – 496, 2010. available at http://dx.doi.org/10.1016/j.mpaic.2010.09.006 (Online, 31/010/2013).

[32] K. Westphal, "Secure mysql database design," 2003. available at http://66.14.166.45/sf_whitepapers/database/Secure%20MySQL%20Database%20Design.pdf (Online, última visita em 31/10/2013).

[33] E. Bertino, S. Jajodia, and P. Samarati, "Database security: Research and practice," *Information Systems*, vol. 20, no. 7, pp. 537 – 556, 1995. available at http://dx.doi.org/10.1016/0306-4379(95)00029-4 (Online, última visita em 31/10/2013).

[34] MySQL, "A guide to securing mysql on windows," January 2010. available at http://www.mysql.com/why-mysql/white-papers/a-guide-to-securing-mysql-on-windows/ (Online, última visita em 31/10/2013).

[35] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet," *Contributions In Librarianship and Information Science*, vol. 96, pp. 3–24, 2001.

[36] R. Sa, *Introducao as Telecomunicacoes*. FCA, 2010.

[37] J. Postel, "Internet protocol," *RFC791*, 1981. available at http://xml2rfc.tools.ietf.org/html/rfc791 (Online, última visita em 31/10/2013).

[38] J. Postel, "Transmission control protocol," *RFC793*, 1981. available at http://tools.ietf.org/html/rfc793 (Online, última visita em 31/10/2013).

[39] J. Postel, "User datagram protocol," *RFC768*, 1980. available at http://tools.ietf.org/html/rfc793 (Online, última visita em 31/10/2013).

[40] S. Zaman and F. Karray, "Tcp/ip model and intrusion detection systems," in *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, pp. 90–96, 2009. available at http://dx.doi.org/10.1109/WAINA.2009.12 (Online, última visita em 31/10/2013).

[41] M. Blaze, "Trust management and network layer security protocols," in *Security Protocols* (B. Christianson, B. Crispo, J. Malcolm, and M. Roe, eds.), vol. 1796 of *Lecture Notes in Computer Science*, pp. 109–118, Springer Berlin Heidelberg, 2000. available at http://dx.doi.org/10.1007/10720107_17 (Online, última visita em 31/10/2013).

[42] J.-S. Li, C.-J. Hsieh, C.-Y. Chang, and N. Chilamkurti, "Improved ipsec performance utilizing transport-layer-aware compression architecture," *Security and Communication Networks*, vol. 4, no. 9, pp. 1063–1074, 2011. available at http://dx.doi.org/10.1002/sec.257 (Online, última visita em 31/10/2013).

[43] W. Liu, E. K. Park, and U. Krieger, "ehealth interconnection infrastructure challenges and solutions overview," in *e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on*, pp. 255–260, 2012. available at http://dx.doi.org/10.1109/HealthCom.2012.6379417 (Online, última visita em 31/10/2013).