MATEUS AUGUSTO SILVA SANTOS

**Segurança em redes de sensores sem fio
desassistidas com aplicações em redes
heterogêneas**

São Paulo

2014

MATEUS AUGUSTO SILVA SANTOS

**Segurança em redes de sensores sem fio desassistidas
com aplicações em redes heterogêneas**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Doutor em Ciências

São Paulo
2014

MATEUS AUGUSTO SILVA SANTOS

**Segurança em redes de sensores sem fio desassistidas
com aplicações em redes heterogêneas**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Doutor em Ciências

Área de Concentração:
Engenharia de Computação

Orientador: Profa. Dra. Cíntia Borges Margi

São Paulo

2014

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de novembro de 2014.

Assinatura do autor _____

Assinatura do orientador _____

Catálogo-na-publicação

Santos, Mateus Augusto Silva Segurança em redes de sensores sem fio desassistidas com aplicações em redes heterogêneas / M.AS. Santos. -- versão corr. -- São Paulo, 2014.89 p.Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.1.Sensor 2.Redes de sensores 3. Segurança de redes 4. Criptologia I.Universidade de São Paulo. Escola Politécnica. Depar-tamento de Engenharia de Computação e Sistemas Digitais II.t.

Agradecimentos

Este espaço é dedicado àqueles que deram a sua contribuição para que esta tese fosse realizada. A todos eles deixo aqui o meu agradecimento sincero.

À minha esposa e filha, por me apoiar e compreender que há extrema dificuldade em conciliar o tempo com a família e realizar um trabalho de pós-graduação com dedicação.

Aos meus pais, pelo grande incentivo e suporte financeiro.

À minha irmã, pelos conselhos e por torcer pelo meu sucesso.

À minha orientadora, pela orientação, colaboração com artigos e apoio com as atividades de doutorado, incluindo seminários, disciplinas, discussões de pesquisa, entre muitas outras.

Aos professores da Universidade Federal de Itajubá, pelo excelente trabalho com o curso de graduação que serviu como base para meus estudos no doutorado.

Aos professores do LARC-EPUSP, pelo compromisso com ensino e pesquisa que foram importantes para a minha formação durante o período de doutorado. Em especial, agradeço aos professores com quem escrevi artigos.

Aos colegas e amigos da pós-graduação, em especial aos amigos do SEMBEI-LARC-EPUSP, pelas produtivas conversas e principalmente pela companhia e amizade.

À minha supervisora de doutorado sanduíche no exterior, pela colaboração e interação com artigos e atividades de pesquisa.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio com uma bolsa de doutorado sanduíche no exterior através do processo número 245588/2012-4.

À Fundação de Apoio à Universidade de São Paulo (FUSP) pelo auxílio financeiro através da Scopus Inovação e Tecnologia. Agradeço também aos coordenadores e colaboradores dos projetos em que participei, os quais também foram importante para minha formação durante o período de doutorado.

Resumo

Em redes de sensores sem fio (RSSFs), o nó sorvedouro geralmente é a única entidade confiável. Uma RSSF desassistida é aquela na qual o nó sorvedouro está indisponível por um período de tempo, sendo necessário armazenar os dados coletados ao invés de transmiti-los para a entidade segura. Portanto, até o nó sorvedouro estar novamente disponível para a recepção dos dados, um adversário pode comprometer nós sensores distribuídos em uma região geográfica com o objetivo de encontrar e apagar determinadas unidades de dado. Com o objetivo de evitar este tipo de ataque, fornecendo “sobrevivência de dados”, estratégias geralmente utilizam confidencialidade e redundância de dados de forma que a agregação de dados por nós sensores seja inviável. Além disso, o decorrer do tempo permite que um adversário tenha maior poder de ataque através do comprometimento de diferentes conjuntos de nós sensores. Apresenta-se um protocolo de segurança para RSSFs desassistidas que, através da renovação do estado da rede, reduz a vantagem de um adversário em montar ataques a partir de dados obtidos com o decorrer do tempo. O mecanismo proposto fornece segurança para múltiplas unidades de dados e permite o uso de agregação de dados por nós sensores. Como forma de avaliação da proposta, foram realizados experimentos em nós sensores reais, além da elaboração de um modelo analítico e de simulações. Resultados indicam como usar o protocolo proposto em diferentes cenários para maximizar a sobrevivência de dados. Adicionalmente, apresenta-se a aplicabilidade do método proposto em cenários de redes heterogêneas, caso das redes de segurança pública, utilizando-se do paradigma de redes definidas por *software* (SDN) para implantação da rede.

Palavras-chave: Redes de sensores sem fio. Sobrevivência de dados. Criptografia.

Abstract

In Wireless Sensor Networks, the base station is usually the only unconditionally trusted entity. If it is not connected for a period of time, the network is left unattended and sensor nodes cannot offload data in real time. Thus, until the base station becomes available, adversaries can compromise some sensor nodes and selectively destroy data. In order to prevent such attacks, providing the so-called “data survival”, strategies generally employ the use of cryptography in a scenario where decryption keys are exclusively in possession of the base station, which restricts the use of in-network aggregation. In addition, as time increases, mobile adversaries become stronger due to the ability of compromising different sets of nodes. We present a secure protocol that refreshes the state of the network and prevents adversaries from taking advantage of data obtained in prior rounds. The scheme provides security to multiple units of data and allows data aggregation to be performed by nodes. In order to evaluate the performance of the proposed scheme, we carried out measurements on real devices. We analyzed the security of the scheme analytically and through simulations. Results indicate increased time epochs before an adversary succeeds the attack, leading to more probability of data survival. We also apply our protocol to scenarios of heterogeneous networks, namely a public safety network scenario, which is deployed under the paradigm of software-defined networking (SDN).

Keywords: Wireless sensor networks. Data survival. Cryptography.

Lista de Figuras

Figura 1 – Método adotado.	16
Figura 2 – Representação genérica de um sistema de cifração.	19
Figura 3 – Supercifração: cifração sucessiva com chaves distintas. Adversário em posse do nó de índice 3 deve obter as chaves K_1 e K_2 para recuperar M	24
Figura 4 – Processo de evolução da chave. Uma vez obtida a chave, é possível utilizar uma função unidirecional para calcular os próximos valores a serem utilizados, porém valores anteriores não são viáveis (computacionalmente) de se calcular.	24
Figura 5 – Modelos de distribuição de chaves utilizados em RSSFs desassistidas. Para realizar agregação de dados, os nós sensores (a) ou o nó sorvedouro (b) precisam revelar pelo menos uma chave privada.	26
Figura 6 – Tempo de execução da supercifração (100 execuções).	27
Figura 7 – Exemplo de um protocolo de compartilhamento de segredos. A inicialização (a) consiste em calcular as 5 quotas a partir do segredo e distribuí-las. A recuperação do segredo (b) é feita através da agregação das quotas, considerando que se tenha pelo menos 3 quotas (limiar).	31
Figura 8 – Uso eficiente de um mecanismos de compartilhamento de segredos, representado pela caixa SS	38
Figura 9 – Tempo de reconstrução do segredo (<i>Desktop</i>).	43
Figura 10 – Exemplo de sequência de passos de um adversário em um esquema CSS-(3,5). Cada nó possui um par (k_i, E_i) . O adversário pode distinguir quotas e iniciar a apagar dados a partir da rodada 2. Na rodada 3, o adversário vence o jogo ao apagar a terceira unidade de dado.	46
Figura 11 – Probabilidade de não se apagar quaisquer dados até a rodada r . Evento L_r condicionado ao evento CL_{r-1} para os parâmetros $n = 100$, $k = 2$, $p = 10$, $t = 5$ e $\varepsilon = r/2$	48
Figura 12 – Marcas definidas para o modelo da rede de Petri colorida: nós, quotas e pares (nó,quota).	52
Figura 13 – Exemplo de conflito. Não há como prever exatamente quais marcas devem se posicionar no lugar <code>NodesAndShares</code> após o disparo da transição.	53

Figura 14	–Atribuição de uma marca para a variável y não é determinística. Neste exemplo, pode-se ter $y=1$ ou $y=2$	54
Figura 15	–Variável i não está no arco de entrada, mas é atribuída à variável j na função de guarda. Adicionalmente, variável b pode ser <i>true</i> ou <i>false</i> , pois também não tem associação em arco de entrada.	54
Figura 16	–Cenário em que as estratégias KEEP-MOVING (defesa) e FRANTIC (adversário) são combinadas com o protocolo proposto.	55
Figura 17	–Modelo da rede de Petri que inclui o jogo de defesa e ataque. O protocolo é parte do sistema de defesa.	56
Figura 18	–Subrede Receive Share: escolha de nós destinatários e recepção de quotas.	57
Figura 19	–Subrede Send Share: transmissão de quotas.	58
Figura 20	–Subrede ADV-Delete Share: modelo do adversário.	58
Figura 21	–Probabilidade de sobrevivência dos dados para comprometimento de dois nós por rodada, dada uma rede de 100 nós com 10 quotas e limiar de valor 5.	59
Figura 22	–A arquitetura CHORIST.	64
Figura 23	–Resultados de simulação em relação à renovação de quotas com o uso do simulador Qualnet.	66
Figura 24	–Exemplo de implantação de RSPs.	67
Figura 25	–Exemplo de uma rede de compartilhamento de capacidade. Dispositivos de usuário (DUs) podem ser conectados via pontos de acesso ou via outros DUs.	68
Figura 26	–Arquiteturas de GWs (a) e DUs (b).	70
Figura 27	–Etapas para o compartilhamento de capacidade: requisição do dispositivo de usuário, <i>handshaking</i> e <i>check-in</i>	71
Figura 28	–Exemplo de implementação do procedimento de <i>handshaking</i> entre DU e GW.	73
Figura 29	–Exemplo de <i>handover</i> de um DU entre dois GWs.	74

Lista de Tabelas

Tabela 1	– Consumo de energia da supercifração para diferentes tamanhos de mensagem (100 execuções)	28
Tabela 2	– Eficiência da evolução da chave criptográfica (100 execuções)	28
Tabela 3	– Custo das principais operações para o cálculo da recifração.	28
Tabela 4	– Consumo de tempo e energia para a inicialização de 80 quotas e reconstrução do segredo no dispositivo TelosB.	43
Tabela 5	– Cenário de simulação	65
Tabela 6	– Notação.	72

Lista de Abreviaturas e Siglas

CHORIST	<i>integrating Communications for enHanced enviroNmental RISk management and citizens safeTy</i>
CBI	Criptografia Baseada em Identidade
CSS	<i>Computational Secret Sharing</i>
DU	Dispositivo de Usuário
GW	<i>Gateway</i>
IDA	<i>Information Dispersal Algorithm</i>
RDS	Redes Definidas por <i>Software</i>
RSSFs	Redes de Sensores sem Fio
RSPs	Redes de Segurança Pública
TTP	<i>Trusted Third Party</i>

Sumário

1	Introdução	13
1.1	Motivação e Descrição do Problema	14
1.2	Objetivos	14
1.3	Justificativa	15
1.4	Métodos	16
1.5	Organização do documento	16
2	Segurança em RSSFs Desassistidas	18
2.1	Conceitos Fundamentais	18
2.1.1	Cifração	18
2.1.2	Criptografia Simétrica e Assimétrica	18
2.1.3	Criptografia Baseada em Emparelhamentos	19
2.1.4	Métodos de Criptografia em RSSFs Desassistidas	20
2.2	Modelo do Sistema e do Adversário	20
2.2.1	Modelo do Sistema	20
2.2.2	Modelo do Adversário	21
2.3	Estratégias Existentes para Sobrevivência de Dados	22
2.3.1	Estratégias de Defesa e Ataque	22
2.3.2	Estratégias com o Uso da Criptografia	23
2.3.3	Problema da Agregação de Dados	25
2.4	Implementação das Principais Estratégias	25
2.5	Considerações do Capítulo	29
3	Protocolo de Segurança para RSSFs Desassistidas	30
3.1	Conceitos Preliminares	30
3.2	Método da Proposta	31
3.3	Extensão do Modelo do Sistema e do Adversário	32
3.4	Descrição do Protocolo	34
3.5	Segurança do Protocolo Proposto	38
3.5.1	Reconstrução do Segredo	38
3.5.2	Distribuição de Quotas	39
3.5.3	Canal Seguro e Renovação de Chave Secreta	39
3.5.4	Segurança da Etapa de Atualização	39
3.5.5	Agregação de Dados	40
3.5.6	Outros Modelos de Adversário	40
3.6	Considerações do Capítulo	41

4	Avaliação da Proposta	42
4.1	Implementação e Experimentação	42
4.2	Análise de Segurança	44
4.2.1	Notação	44
4.2.2	Quotas de Tamanho Reduzido e Indistinguibilidade	45
4.2.3	Probabilidade de Reconstrução do Segredo	46
4.2.4	Probabilidade de Apagar Dados	47
4.3	Modelagem e Simulação com Redes de Petri Coloridas	49
4.3.1	Escopo do Modelo	51
4.3.2	Modelo	51
4.3.3	Simulação e Resultados	59
4.4	Considerações do Capítulo	60
5	Aplicação em Redes Heterogêneas	62
5.1	Redes de Segurança Pública	63
5.2	Implantação Segura de RSPs	66
5.2.1	Redes de Compartilhamento de Capacidade	67
5.2.2	Aplicação das Redes de Compartilhamento de Capacidade em RSPs	68
5.2.3	Arquitetura para Redes de Compartilhamento de Capacidade	68
5.3	Considerações do Capítulo	74
6	Considerações Finais	75
6.1	Trabalhos Futuros	76
6.2	Publicações	77
	Referências	80
	Apêndices	85
	APÊNDICE A Declarações dos Modelos em Redes de Petri	86

1 Introdução

As redes de sensores sem fio (RSSFs) são formadas por um grande número de dispositivos computacionais de baixo custo providos de elementos de sensoriamento, e que geralmente utilizam baterias como fonte de energia. Esses dispositivos utilizam comunicação sem fio de baixa potência para interagir, o que os diferencia de uma mera coleção de sensores, possibilitando a cooperação e a coordenação entre os nós sensores ativos (HONG et al., 2007). Por se tratarem de dispositivos de baixo custo e por ser interessante que sejam de tamanho reduzido, os nós sensores frequentemente apresentam recursos limitados de processamento, memória, comunicação e suprimento de energia (CULLER; ESTRIN; SRIVASTAVA, 2004).

RSSFs são uma valiosa tecnologia para o desenvolvimento de aplicações em diversas áreas. Monitoramento ambiental, vigilância militar em campos de batalhas e sistemas de segurança comerciais são exemplos de aplicações para RSSFs (ZHAO; GUIBAS, 2004).

Um tipo de RSSF é aquele em que pode não haver comunicação em tempo real entre o nó sorvedouro e os dispositivos sensores. Neste caso, considera-se que o nó sorvedouro é itinerante e que os nós sensores devem armazenar dados até que se tenha uma nova *visita* do sorvedouro. Este tipo de rede é chamado de desassistida¹ (PIETRO et al., 2008) e geralmente é definido em conjunto com um ambiente hostil. Em especial, considera-se um adversário móvel, o qual realiza ataques contra dispositivos sensores em diferentes regiões geográficas.

Um exemplo de cenário que poderia utilizar as RSSFs desassistidas é o monitoramento de fronteiras entre países (MA; SORIENTE; TSUDIK, 2009). Haveria um alto custo e complexidade em implantar uma quantidade razoável de nós sorvedouros em uma região geográfica de grande escala. Além disso, cada nó sorvedouro deveria ser implantado de forma que um adversário (i.e., uma pessoa que cruza ilegalmente a fronteira) não possa identificá-lo. Por outro lado, uma visita periódica de um nó sorvedouro móvel (e.g., implantado em um veículo de patrulha) seria uma solução adequada.

¹ Do inglês *unattended*.

1.1 Motivação e Descrição do Problema

Como a ausência do nó sorvedouro requer armazenamento local nos dispositivos sensores, um adversário poderia obter maior vantagem em RSSFs desassistidas ao realizar ataques físicos. Isso porque, nas RSSFs tradicionais, um nó sorvedouro receberia os dados dos dispositivos sensores e um adversário teria menor tempo para realizar ataques.

Vários modelos de adversário podem ser considerados (VITALI et al., 2011; PIETRO et al., 2009b; PIETRO et al., 2009a). Um adversário poderia, por exemplo, obter um conjunto de nós sob sua custódia e, com o objetivo de apagar determinadas unidades de dado, ler conteúdos da memória e chaves criptográficas, além das mensagens que estão sendo enviadas e recebidas pelos nós. Esse é o modelo de adversário considerado neste trabalho, o qual é definido na Seção 2.2.2.

Se um adversário obteve sucesso em apagar determinado conjunto de dados diz-se que o dado *não sobreviveu*, ou seja, o nó sorvedouro, em sua próxima visita, não receberia os dados importantes relacionados com o ambiente, o que comprometeria toda a motivação de implantação da rede. Portanto determinado conjunto de dados *sobrevive* se e só se o nó sorvedouro pode recebê-lo quando estiver disponível na rede (em sua próxima visita). Isso define informalmente o problema da **sobrevivência de dados**.

Um desafio das RSSFs desassistidas é projetar protocolos de segurança para que determinado conjunto de dados sobreviva. Para que isso possa acontecer, um mecanismo projetado para defender o sistema deve ser capaz de evitar que um adversário possa identificar e apagar um conjunto de dados considerado importante, o que geralmente é chamado de *dado alvo* para um adversário.

Portanto, é natural considerar alguns requisitos de um protocolo projetado para o problema da sobrevivência de dados:

- redundância: conjuntos de dados apagados até determinada quantidade poderiam ser reconstruídos pelo sistema;
- confidencialidade: um adversário não poderia encontrar e apagar determinada informação sem detectá-la;
- eficiência: mecanismos utilizados seriam eficientes em consumo de tempo e energia.

1.2 Objetivos

O objetivo geral da pesquisa de doutorado é maximizar a sobrevivência de dados em RSSFs desassistidas e permitir o uso de mecanismos de agregação de dados pelos dispositivos sensores. Vislumbra-se atingir este objetivo através da elaboração de um protocolo

de segurança. Como objetivo específico, considera-se a implementação em dispositivos reais e um estudo de caso através de modelagem, simulação e estudo teórico. Espera-se demonstrar que os mecanismos propostos são eficientes em consumo de tempo e energia, além de fornecer segurança e aplicabilidade no cenário adotado.

Além do problema da sobrevivência de dados, procura-se identificar outros tipos de cenários em redes de computadores que possam ter aplicações com os mecanismos elaborados. Considera-se redes de computadores que compartilham com as RSSFs desassistidas a característica de armazenamento local em dispositivos da rede, sejam eles sistemas finais ou roteadores.

Redes que possuem heterogeneidade de dispositivos e diferentes protocolos de comunicação, geralmente chamadas de *redes heterogêneas*, podem estar sujeitas a falhas temporárias de comunicação devido a problemas de interoperabilidade ou disponibilidade. Em especial, considera-se as redes de segurança pública (SANTOS; MARGI, 2011b), definidas no Capítulo 5.

1.3 Justificativa

A importância de elaborar um novo protocolo de sobrevivência de dados para RSSFs desassistidas vem do fato de que propostas anteriores consideram mecanismos de segurança com operações fundamentais de alto custo ou utilizam métodos que tornam inviável a agregação de dados entre nós sensores, como descrito nas seções 2.4 e 2.3.3, respectivamente. Essas características foram identificadas a partir de um estudo realizado na primeira etapa do presente projeto de pesquisa de doutorado, como descrito na Seção 1.4.

Além da importância do tema, é relevante justificar as hipóteses escolhidas para atingir os objetivos. No contexto da avaliação do protocolo proposto, as redes de Petri coloridas foram adotadas. Este tipo de rede facilita a modelagem porque permite a criação de modelos de alto nível, ao contrário das redes de Petri clássicas, como descrito na Seção 4.3. Além disso, é possível simular sequências de eventos e conduzir experimentos com a finalidade de entender o comportamento de um sistema ou avaliar suas funcionalidades. Observa-se ainda que as redes de Petri coloridas podem possuir uma representação gráfica, além de uma linguagem formal de especificação que pode trazer vantagens como especificações complexas livres de ambiguidade (JENSEN, 1987).

Considera-se também uma análise de segurança do mecanismo proposto através de um estudo teórico, também chamado de modelo analítico. Apesar deste tipo de estudo geralmente possuir premissas que simplificam o cenário avaliado, é possível utilizá-lo como forma adicional de análise dos resultados, além de compará-lo com o método supracitado, ou seja, com o uso das redes de Petri coloridas.

Em relação à eficiência da proposta, considera-se a implementação em dispositivos reais. Isso permite conduzir experimentos com base em métricas como consumo de tempo e energia.

1.4 Métodos

A Figura 1 ilustra a metodologia adotada neste trabalho, o qual é dividido em quatro etapas.

Figura 1: Método adotado.

Na Etapa 1 foi realizado um estudo e avaliação de propostas anteriores. A principal estratégia foi identificar as deficiências dos trabalhos relacionados, incluindo questões de implementação e resultados de experimentação.

Na Etapa 2 foram utilizados conceitos do trabalho realizado na etapa anterior para identificar quais mecanismos seriam promissores para uma nova proposta, o que foi elaborado na Etapa 3. Esta etapa também inclui a avaliação da proposta através de modelagem e simulação com redes de Petri, modelo analítico e implementação em dispositivos reais, incluindo experimentação.

A Etapa 4 concluiu o trabalho relacionado com a pesquisa de doutorado. Foram identificados cenários de aplicação para os mecanismos de segurança elaborados. Considerou-se aplicações para redes heterogêneas, em especial as redes de segurança pública e sua implantação através de redes de compartilhamento de acesso e do paradigma das redes definidas por *software*.

1.5 Organização do documento

O Capítulo 2 apresenta premissas e definições fundamentais sobre o problema da sobrevivência de dados, além de uma revisão da literatura, incluindo trabalhos relacionados com as RSSFs desassistidas. Além disso, ainda apresenta resultados obtidos com a implementação e experimentação de propostas anteriores. A proposta de um protocolo

de segurança para RSSFs desassistidas é apresentada no Capítulo 3. O Capítulo 4 apresenta a avaliação do protocolo proposto através de implementação e experimentação em dispositivos reais, análise de segurança com abordagem teórica e modelagem com o uso das redes de Petri, incluindo simulação do modelo que foi elaborado. Em relação a outras aplicações para o protocolo proposto, foram consideradas as redes heterogêneas, descritas no Capítulo 5 através das redes de segurança pública. Finalmente, no Capítulo 6, são apresentadas as considerações finais e os artigos publicados e em submissão.

2 Segurança em RSSFs Desassistidas

Neste capítulo apresentamos conceitos preliminares, premissas e definições fundamentais que tem sido utilizadas para o estudo do problema da sobrevivência de dados em RSSFs desassistidas. Em seguida, apresentamos uma revisão das estratégias existentes na literatura e resultados obtidos a partir de implementação e experimentação, o que resultou na seguinte publicação:

- **Santos, M. A. S.**; Margi, Cíntia B.;Simplicio, M.A.;Geovandro, C.C.F.P.;de Oliveira, B.T.. *Implementation of data survival in unattended Wireless Sensor Networks using cryptography*. In: 5th IEEE International Workshop on Practical Issues in Building Sensor Network Applications. Co-located with 35th IEEE Local Computer Networks, 2010, Denver, CO, USA. In Proceedings: 35th IEEE Local Computer Networks (**IEEE LCN**), 2010. p. 961-967.

2.1 Conceitos Fundamentais

2.1.1 Cifração

Um método utilizado para fornecer confidencialidade de dados é a cifração. De uma forma genérica, a cifração é uma função E que recebe uma chave criptográfica de cifração K_e e um texto claro M , gerando como resultado um texto cifrado C , definido através da seguinte expressão: $C = E_{K_e}(M)$. Para que um texto cifrado C retorne a sua forma original M , o texto claro deve ser recuperado pela expressão: $M = D_{K_d}(C)$, onde D é a função de decifração e K_d é a chave criptográfica para decifração (STALLINGS, 2008). A Figura 2 ilustra esta generalização.

2.1.2 Criptografia Simétrica e Assimétrica

Um sistema criptográfico no qual o emissor e o receptor utilizam o mesmo valor de chave ($K_e = K_d$) é classificado como *criptografia simétrica*. Já um sistema no qual o emissor e o receptor utilizam chaves diferentes ($K_e \neq K_d$) é classificado como *criptografia assimétrica* (STALLINGS, 2008).

Figura 2: Representação genérica de um sistema de cifração.

Fonte: (OLIVEIRA, 2012).

As chaves criptográficas utilizadas em um sistema de criptografia assimétrica são conhecidas como chave pública e chave privada. A chave pública pode ser distribuída arbitrariamente, podendo qualquer entidade conhecê-la sem que a segurança seja comprometida, enquanto a chave privada deve ser mantida em segredo por parte de seu proprietário.

2.1.3 Criptografia Baseada em Emparelhamentos

Informalmente, um emparelhamento associa pares de pontos em curvas elípticas a um elemento de um corpo finito (OLIVEIRA et al., 2010). A criptografia baseada em emparelhamentos permite a realização de protocolos com propriedades interessantes. Por exemplo, com o uso do protocolo proposto por Sakai, Ohgishi e Kasahara (SOK) (SAKAI; OHGISHI; KASAHARA, 2000) dois nós sensores podem realizar um *acordo de chaves* a partir de um canal inseguro. Ao final do processo tem-se um mesmo valor de chave compartilhado entre o par de nós de forma segura contra um terceiro.

Uma definição mais precisa de emparelhamento pode ser feita com base no conceito de grupos. Seja q um número primo. Seja $\mathbb{G}$ um grupo cíclico aditivo e $\mathbb{G}_T$ um grupo cíclico multiplicativo, ambos de ordem q . Um mapeamento bilinear ou emparelhamento é uma função $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ que satisfaz as seguintes propriedades. (1) bilinearidade: $e(aP, bQ) = e(P, Q)^{ab} \forall P, Q \in \mathbb{G} \text{ e } \forall a, b \in \mathbb{Z}_q^*$; (2) não-degeneração: $e(P, P) \neq 1$; (3) computabilidade: existe um algoritmo eficiente ou em tempo polinomial para o cálculo de $e(P, Q) \forall P, Q \in \mathbb{G}$.

Uma aplicação do emparelhamento em RSSFs desassistidas é através de protocolos de recifração, e.g., (ATENIESE et al., 2006). Como descrito na Seção 2.3.2, a recifração requer operações em corpos finitos como exponenciação e inversão. Nesse caso, as operações devem ser realizadas a partir do elemento do corpo finito que foi obtido com o cálculo do emparelhamento.

2.1.4 Métodos de Criptografia em RSSFs Desassistidas

Em RSSFs, a cifração é realizada principalmente através de algoritmos simétricos, já que apresentam melhor desempenho que mecanismos de chave pública (SANTOS et al., 2010). No caso das RSSFs desassistidas, métodos convencionais de cifração não são seguros devido ao modelo do adversário. Geralmente considera-se que um adversário pode ler o conteúdo de memória de um nó sensor e portanto obter o material criptográfico armazenado (e.g., chaves secretas).

Como será apresentado nas seções 2.2 e 2.3, esquemas de criptografia simétricos e assimétricos são utilizados de forma que se explore a cooperação entre nós sensores. Por exemplo, uma unidade de dado pode ser cifrada por dois ou mais nós sensores, cada um utilizando a sua chave secreta. Portanto um adversário deve obter dados armazenados em pelo menos dois nós sensores para restaurar o dado original. Da mesma forma, protocolos de recifração são executados por diversos nós para aumentar a segurança contra adversários.

2.2 Modelo do Sistema e do Adversário

Chama-se de modelo o conjunto de premissas adotadas para o estudo do problema da sobrevivência de dados. Nesta seção, apresenta-se os modelos que possuem intersecção com trabalhos relacionados (PIETRO et al., 2008; PIETRO et al., 2009a; HERZBERG et al., 1995; PIETRO et al., 2009; REN; ZHAO; REN, Dec. 2009). Na Seção 3.3, apresenta-se uma extensão do modelo em questão.

2.2.1 Modelo do Sistema

O modelo do sistema possui as seguintes premissas:

- sempre existe um caminho entre dois nós sensores, ou seja, quaisquer dois nós da rede podem se comunicar direta ou indiretamente, via múltiplos saltos. Observa-se que o esquema proposto foi elaborado para operar na camada de aplicação, sendo independente de protocolos de roteamento;
- não há perda de pacotes durante a transmissão de dados, ou seja, dados transmitidos são recebidos com sucesso no destinatário. Perdas não são consideradas para permitir que se isole a dinâmica do sistema de problemas de rede, o que também permite uma comparação entre modelo analítico e simulação. Esta premissa é removida na aplicação do protocolo em redes heterogêneas através do uso de um simulador de redes (SANTOS; MARGI, 2011b);

- o sistema de defesa não possui conhecimento sobre as ações de um adversário. O sistema não sabe quais nós em particular estão atualmente comprometidos ou sob custódia do adversário. Além disso, não se conhece a estratégia de um adversário (a Seção 2.3.1 apresenta as estratégias adotadas).

2.2.2 Modelo do Adversário

Define-se *dado alvo* como uma unidade de informação crítica ou valiosa, a qual foi coletada por um nó sensor. O objetivo de um adversário, ou $\mathcal{ADV}$, é localizar um dado alvo (ou dados alvo) e apagá-lo(s), de forma que os nós sensores não possam transmiti-lo(s) ao nó sorvedouro móvel em sua próxima visita. Se um dado alvo qualquer foi apagado e não houve recepção pelo nó sorvedouro, o adversário venceu o “jogo”. Observa-se que o $\mathcal{ADV}$ conhece o dado em particular a ser procurado.

O modelo do adversário possui as seguintes premissas:

- o $\mathcal{ADV}$ pode comprometer um conjunto de nós sensores a cada rodada. Apesar de ser permitido que os elementos deste conjunto sejam alterados, considera-se que a quantidade de nós comprometidos entre rodadas distintas não se altere;
- o $\mathcal{ADV}$ pode ler a memória de nós comprometidos, obter dados armazenados e apagá-los. Além disso, esses dados podem ser transferidos para um *catálogo* para futuro acesso pelo $\mathcal{ADV}$;
- apesar do $\mathcal{ADV}$ poder apagar unidades de dado de nós comprometidos, não é permitido que dados sejam apagados sem critério. O dado deve ser identificado pelo $\mathcal{ADV}$ antes de ser apagado. Não é permitido apagar toda a memória de um nó sensor. Se não for possível distinguir um valor pseudo-aleatório de um dado alvo que foi criptografado, não há qualquer ação a ser tomada pelo $\mathcal{ADV}$;
- o $\mathcal{ADV}$ conhece a topologia da rede, estratégias de defesa e parâmetros do esquema de segurança;
- no momento que o $\mathcal{ADV}$ conhece o dado a ser procurado, a inicialização do sistema de defesa já foi concluída. Logo, o protocolo de segurança utilizado também foi inicializado, ou seja, considera-se que material criptográfico secreto ou público já tenha sido distribuído e o $\mathcal{ADV}$ não possui qualquer informação sobre a alocação desses dados;
- o $\mathcal{ADV}$ não pode interferir na comunicação entre os nós sensores ou corromper dados em transmissão. Dados apagados da memória de um nó sensor não podem ser recuperados;

- o $\mathcal{ADV}$ não pode realizar ataques relacionados com integridade e autenticação, o que está fora do escopo deste trabalho.

Apesar deste trabalho não estar relacionado com problemas de integridade e autenticação em RSSFs, o protocolo proposto pode ser estendido para fornecer estes serviços, como discutido na Seção 3.5.6. A respeito de exemplo, o trabalho de Pietro et al. (2009b) estuda o problema da autenticação em RSSFs desassistidas.

2.3 Estratégias Existentes para Sobrevivência de Dados

2.3.1 Estratégias de Defesa e Ataque

O primeiro método proposto como mecanismo de defesa para o problema da sobrevivência de dados foi mover unidades de dados de um dispositivo para outro, escolhido de forma pseudo-aleatória. As principais estratégias propostas neste contexto são (PIETRO et al., 2008):

- **MOVE ONCE:** o dado alvo é movido apenas uma vez de um dispositivo sensor para outro;
- **KEEP MOVING:** o dado alvo é sempre movido entre dispositivos sensores, sendo que um dispositivo tem posse de apenas uma unidade de dado em uma determinada rodada.

Um adversário pode ser *móvel* e adotar suas próprias estratégias para maximizar sua chance de ataque. Alguns exemplos são (PIETRO et al., 2008):

- **LAZY:** um adversário compromete sempre os mesmos nós a cada rodada, ou seja, uma vez escolhidos os nós não há alteração no conjunto selecionado;
- **FRANTIC:** um adversário compromete conjuntos disjuntos de nós a cada rodada, repetindo apenas se houve escolhas exaustivas em rodadas sucessivas.

É importante observar que a existência de um adversário móvel não implica que os nós sensores sejam também móveis. Isso significa que um adversário pode se mover ao alcance de conjuntos de nós fixos dispostos em uma região geográfica. Observa-se ainda que as estratégias de defesa e ataque têm natureza probabilística. Dois conjuntos são escolhidos de forma pseudo-aleatória: os nós comprometidos e os destinatários do dado alvo. O comprometimento de nós é uma escolha do adversário, já os nós de destino de mensagens estão no escopo do sistema de defesa.

2.3.2 Estratégias com o Uso da Criptografia

As principais estratégias utilizadas em conjunto com os métodos supracitados são baseadas em criptografia. Geralmente emprega-se a confidencialidade para evitar que o $\mathcal{ADV}$ encontre um dado alvo (PIETRO et al., 2008; PIETRO et al., 2009a). Informalmente, uma unidade de dado indistinguível de um valor pseudo-aleatório pode ser considerada como dado criptografado que não será detectado pelo $\mathcal{ADV}$.

É importante observar que o modelo utilizado em RSSFs desassistidas define que o $\mathcal{ADV}$ conhece o dado alvo. Logo, é possível que seja realizada uma operação criptográfica (e.g., cifração) no dado alvo com o objetivo de comparar o valor obtido com dados criptografados armazenados em nós sensores. Esse tipo de ataque é possível se não houver uma fonte segura para geração de números aleatórios. Por exemplo, geradores de números pseudo-aleatórios geralmente utilizam uma *semente* para obter de forma determinística os demais números. A geração de números aleatórios reais evita esse tipo de vulnerabilidade. Refere-se ao trabalho de Re, Milazzo e Ortolani (2014) para mais informações sobre geração de números aleatórios de forma segura para a criptografia em RSSFs.

É coerente afirmar que sem critério não existe um ataque, pois não seria permitido apagar dados repetidamente até esvaziar a memória de todos os dispositivos sensores. Esse procedimento, geralmente chamado de *força bruta*, poderia ser observado pelo sistema de defesa (PIETRO et al., 2008).

As estratégias supercifração, evolução de chave e recifração são revistas nos próximos parágrafos da presente seção.

Supercifração e Evolução de Chave

As principais propostas de sobrevivência de dados baseadas em criptografia simétrica são:

- supercifração: cifração sucessiva de uma mensagem com o uso de diferentes chaves criptográficas;
- evolução de chave¹: alteração frequente da chave criptográfica com o uso de uma função unidirecional. É importante observar que a chave anterior ao processo é apagada do dispositivo sensor.

No caso da supercifração, cada uso da estratégia aumenta a segurança do sistema, pois um adversário em posse de uma mensagem supercifrada teria que obter diferentes chaves criptográficas para recuperar a informação por decifração sucessiva. A Figura 3 ilustra o processo.

¹ Do inglês, *key evolution*.

Figura 3: Supercifração: cifração sucessiva com chaves distintas. Adversário em posse do nó de índice 3 deve obter as chaves K_1 e K_2 para recuperar M .

A evolução de chave, ilustrada na Figura 4, permite que mensagens cifradas ou supercifradas não possam ser recuperadas por um adversário que não tem posse do dispositivo até o momento do processo. Isso porque a chave é apagada após sua alteração. O nó sorvedouro poderia realizar as decifrações, pois considera-se que este possui toda a cadeia de chaves em seu repositório.

Figura 4: Processo de evolução da chave. Uma vez obtida a chave, é possível utilizar uma função unidirecional para calcular os próximos valores a serem utilizados, porém valores anteriores não são viáveis (computacionalmente) de se calcular.

Recifração

Em relação à criptografia assimétrica, a principal estratégia proposta foi a recifração, como descrito no trabalho de Pietro et al. (2009a). Isso porque o número de mensagens que um adversário não consegue decifrar é maior se comparado com outras estratégias. Com o intuito de exemplificar o processo, considera-se uma mensagem cifrada para Alice (i.e., com o uso da chave pública de Alice). Neste exemplo, a recifração poderia ser aplicada na mensagem para que esta possa ser transformada em um texto exclusivamente cifrado para um outro usuário, Bob. A recifração permite que um terceiro seja autorizado a fazer a conversão do texto cifrado (i.e., de Alice para Bob) sem possuir a chave privada de Alice e portanto sem revelar o conteúdo da mensagem.

A recifração dificulta o rastreamento do par mensagem cifrada e chave criptográfica. Logo, o adversário teria menor chance de identificar e apagar dados. Isso porque

dados cifrados podem estar em um nó, digamos de índice 1, e a chave criptográfica para decifração armazenada no nó de índice 5, o qual poderia estar localizado em uma região geográfica diferente da área de ataque do adversário.

2.3.3 Problema da Agregação de Dados

O processo de agregação em RSSFs é a ideia de combinar dados de diferentes nós fontes com o objetivo de eliminar redundâncias, minimizar o número de transmissões e portanto economizar energia (KRISHNAMACHARI; ESTRIN; WICKER, 2002). Protocolos de roteamento centrado em dados, como é o caso do *Directed Diffusion* (INTANAGONWIWAT; GOVINDAN; ESTRIN, 2000), apresentam melhor desempenho com o uso de mecanismos de agregação de dados. Ao invés de encaminhar dados através do menor caminho, o paradigma de roteamento centrado em dados define que nós realizem o roteamento com base no conteúdo de um pacote, sendo que o próximo salto pode ser escolhido para se promover a agregação de dados (FASOLO et al., 2007).

No contexto da segurança, o uso de mecanismos de criptografia simétrica não permite que se tenha confidencialidade fim-a-fim e agregação de dados de forma simultânea (OZDEMIR; XIAO, 2009). No caso das RSSFs desassistidas, destaca-se que a dificuldade de realizar agregação de dados vem do fato de que as propostas estão fundamentadas em duas características principais:

- **criptografia simétrica com chaves par-a-par:** como ilustrado na Figura 5a, cada dispositivo possui uma chave compartilhada com o nó sorvedouro e, portanto, dados cifrados por diferentes dispositivos não poderiam ser consolidados para agregação, já que cada participante teria que revelar sua chave secreta para permitir tal operação;
- **criptografia de chave pública com decifração exclusiva do nó sorvedouro:** qualquer cifração de dados é realizada com o uso da chave pública do nó sorvedouro, o que delega a decifração dos dados com exclusividade para a entidade confiável. A Figura 5b mostra esta característica, em que os nós sensores não têm posse da chave para decifração.

2.4 Implementação das Principais Estratégias

Consideramos a implementação e avaliação das principais propostas existentes na literatura (SANTOS et al., 2010); em especial, o trabalho de Pietro et al. (2009a). O objetivo principal foi estudar as estratégias em cenários reais, identificando vantagens e deficiências do ponto de vista do desempenho dos algoritmos.

Figura 5: Modelos de distribuição de chaves utilizados em RSSFs desassistidas. Para realizar agregação de dados, os nós sensores (a) ou o nó sorvedouro (b) precisam revelar pelo menos uma chave privada.

A contribuição vem do fato de que as propostas existentes não foram implementadas e avaliadas em dispositivos reais. O que havia sido feito está relacionado apenas com a elaboração de um modelo analítico e simulação em alto nível, de forma que o comportamento das estratégias fosse combinado com o de um adversário.

Como mencionado na Seção 2.3.1, as estratégias avaliadas geralmente são combinadas com a transmissão periódica de dados entre nós da rede, o que aumenta a segurança principalmente devido ao uso de materiais criptográficos de diferentes dispositivos. Apesar da transmissão de dados gerar sobrecusto de comunicação, considerou-se na experimentação apenas o desempenho dos algoritmos no contexto local do dispositivo.

Todos os experimentos foram realizados em nós sensores sobre o sistema TinyOS 2.1.0 e uma aplicação desenvolvida em nesC (GAY et al., 2003). O compilador adotado foi o msp-gcc².

A fim de obter dados precisos, foram realizadas medidas diretas sobre o sensor TelosB (POLASTRE; SZEWCZYK; CULLER, 2005). O método é similar ao de Margi et al. (Margi, C. B. et al., 2010). Foi utilizado o multímetro digital de alta precisão Agilent modelo 34401A (AGILENT, 2007) para medir o fluxo de corrente. Determinou-se quando o microcontrolador do sensor torna-se ativo e inativo durante a execução dos testes. Uma fonte de alimentação precisa Agilent modelo E3631A (AGILENT, 2009) foi configurada

² <http://mspgcc.sourceforge.net/>

para fornecer 3 V de tensão ao sensor TelosB. Em relação à medição, primeiro é obtida a corrente drenada quando o sistema está em modo ocioso, ou seja, nenhuma operação está sendo executada. Após isso, são obtidas medidas com operações sendo processadas. Então, a partir dos níveis de drenagem de correntes obtidos, é possível determinar quando o sistema está executando uma operação ou está em modo ocioso.

Para obter os resultados apresentados nesta seção foram realizados 10 conjuntos de experimentos, sendo que cada conjunto de dados foi obtido com a execução da mesma operação 100 vezes. Isso está identificado nos gráficos e tabelas como "100 execuções". Em suma, foram obtidas 10 amostras com cada uma representando a execução de 100 repetições de uma mesma tarefa. A média das amostras foi apresentada como resultado. Este procedimento foi realizado devido a limitações do equipamento adotado, já que não haveria granularidade suficiente para a medição de apenas uma operação.

É importante observar que o procedimento supracitado não foi utilizado para obter os resultados de recifração (Tabela 3), pois as operações relacionadas possuem maior custo. Portanto neste caso foi possível obter a média de 10 amostras sem qualquer repetição das operações.

As principais métricas adotadas foram consumo de energia e tempo de execução. Os resultados para as estratégias supercifração e evolução da chave são mostrados na Figura 6 e nas Tabelas 1 e 2, respectivamente. Como esperado, tais estratégias apresentam bom desempenho em consumo de tempo e energia (faixa de milijoules e milissegundos), já que são baseadas em algoritmos de criptografia simétrica. No entanto, essas estratégias tornam inviável o uso de agregação de dados e são dependentes de um grande número de transmissões periódicas. No caso da supercifração, por exemplo, a estratégia só pode ser segura se dados forem movidos de um dispositivo para outro várias vezes, com a premissa de que cada nó possui uma chave criptográfica diferente – caso contrário o $\mathcal{ADV}$ pode realizar a decifração ao obter as chaves de nós sensores.

Figura 6: Tempo de execução da supercifração (100 execuções).

Fonte: (SANTOS et al., 2010).

Tabela 1: Consumo de energia da supercifração para diferentes tamanhos de mensagem (100 execuções)

Consumo de Energia (mJ)	
2 rounds	$1,09 \pm 0,04$
4 rounds	$2,21 \pm 0,01$
8 rounds	$4,45 \pm 0,02$

Fonte: (SANTOS et al., 2010).

Tabela 2: Eficiência da evolução da chave criptográfica (100 execuções)

Tempo (ms)	Consumo de Energia (mJ)
$145,31 \pm 9,19$	$0,74 \pm 0,02$

Fonte: (SANTOS et al., 2010).

O desempenho da recifração, mostrado na Tabela 3, depende de operações fundamentais como inversão e exponenciação. Esta última, especialmente, é a de maior consumo de tempo e pode apresentar um ponto de risco para a estabilidade da rede, já que o processamento de muitas mensagens é inviável se a operação não for otimizada. Infelizmente, otimizar tal operação pode requerer grande consumo de memória (SANTOS et al., 2010).

Adicionalmente, outras operações custosas podem ser necessárias, como é o caso do emparelhamento. O cálculo de cada função de emparelhamento em nós sensores geralmente requer alto consumo de memória para que se tenha um tempo de execução próximo de 2 segundos (OLIVEIRA et al., 2010). Portanto, a recifração é um mecanismo que possui alto custo computacional e de energia se executada com frequência, o que motiva a elaboração de novos mecanismos para a sobrevivência de dados.

Tabela 3: Custo das principais operações para o cálculo da recifração.

	Tempo (s)	Consumo de Energia (mJ)
exponenciação	$9,28 \pm 0,01$	$53,22 \pm 0,03$
inversão	$0,76 \pm 0,01$	$4,25 \pm 0,01$

Fonte: (SANTOS et al., 2010).

É possível comparar os resultados de medição supracitados com tarefas básicas realizadas por dispositivos sensores. Por exemplo, o trabalho de Margi, C. B. et al. (2010) apresenta o custo para transmitir e receber dados. Em especial, a transmissão de 12 bytes equivale a 634 milissegundos, sendo seu consumo de energia equivalente a 33.5 milijoules. Isso mostra que a recifração apresenta alto custo pelo menos em relação ao tempo, se comparada com a transmissão de apenas 12 bytes. Observa-se que 12 bytes é um tamanho razoável, pois o tamanho máximo de um quadro é 127 bytes.

2.5 Considerações do Capítulo

Devido ao modelo de adversário adotado, métodos de criptografia convencionais não são suficientes para fornecer segurança em RSSFs desassistidas. Um dos principais problemas é a fácil exposição de chaves criptográficas armazenadas em nós sensores. Por isso, as principais propostas da literatura utilizam a cooperação entre nós em conjunto com mecanismos de criptografia simétrica e assimétrica.

A implementação dos mecanismos existentes na literatura mostrou que a criptografia assimétrica, em especial a recifração, requer operações fundamentais de alto custo. No caso da criptografia simétrica, os mecanismos utilizados são eficientes em consumo de tempo e energia. No entanto, para a criptografia simétrica, não há como utilizar confidencialidade fim-a-fim e agregação de dados de forma simultânea. Além disso, o uso de chaves secretas par-a-par compartilhadas com o nó sorvedouro implica em revelar a chave secreta para outro nó sensor em caso de decifração com múltiplos nós sensores. O modelo simétrico ainda requer a cooperação de diversos nós sensores para evitar que o $\mathcal{ADV}$ realize a decifração.

Conclui-se que a cooperação entre nós pode ser utilizada para aumentar a segurança em RSSFs desassistidas. Como mecanismos de criptografia tradicionais apresentaram desvantagens, é interessante investigar novos modelos de criptografia. Vislumbra-se manter o serviço de confidencialidade e a interação entre nós que fornece a propriedade de cooperação. Além disso, como o objetivo do $\mathcal{ADV}$ é apagar dados, espera-se que alguma forma de redundância seja utilizada.

3 Protocolo de Segurança para RSSFs Desassistidas

Neste capítulo apresenta-se inicialmente conceitos fundamentais utilizados para a elaboração de um protocolo de segurança para RSSFs desassistidas, o qual é descrito em conjunto com formas de uso e escolha de parâmetros.

O protocolo proposto é uma das contribuições da pesquisa de doutorado e resultou na seguinte publicação:

- **Santos, M. A. S.;** Margi, Cíntia B.. *Design and Implementation of Data Survival in Unattended Wireless Sensor Networks*. In: The 2011 International Workshop on Hot Topics on Wireless Network Security and Privacy. Co-located with 30th IEEE International Performance Computing and Communications Conference (**IEEE IPCCC**), 2011, Orlando, FL, USA. In Proceedings: 30th IEEE International Performance Computing and Communications Conference (IPCCC 2011), 2011. p. 1-6.

3.1 Conceitos Preliminares

Mecanismos de compartilhamento de segredos¹ foram introduzidos de forma independente por Shamir (1979) e Blakley (1979) com o propósito de fornecer serviços de segurança para chaves criptográficas. Em geral, esse tipo de mecanismo produz valores denominados quotas a partir de um ou mais segredos que se deseja compartilhar entre usuários. As quotas são distribuídas para usuários autorizados e se tornam, portanto, os novos valores secretos do sistema. A partir de um determinado número de quotas agregadas por um usuário, é possível utilizar uma estrutura de acesso para reconstrução de um ou mais segredos.

Uma propriedade importante desse tipo de mecanismo é que a posse de um número de quotas inferior ao definido pelo sistema, chamado de *limiar*, implica que o problema computacional de reconstruir o(s) segredo(s) é difícil. Caso contrário, o problema é fácil de se resolver, ou seja, eficiente em consumo de tempo. A Figura 7 ilustra as etapas de um esquema de compartilhamento de segredos.

¹ Do inglês, *secret sharing*

Figura 7: Exemplo de um protocolo de compartilhamento de segredos. A inicialização (a) consiste em calcular as 5 quotas a partir do segredo e distribuí-las. A recuperação do segredo (b) é feita através da agregação das quotas, considerando que se tenha pelo menos 3 quotas (limiar).

Em um mecanismo de compartilhamento de segredos pode-se utilizar qualquer tipo de dado como segredo. Por exemplo, uma chave criptográfica pode ser utilizada como um segredo ou mesmo um documento sigiloso. Neste capítulo mostramos como utilizar uma chave criptográfica como um segredo de forma segura e eficiente.

3.2 Método da Proposta

Em relação à sobrevivência de dados, a estratégia adotada foi usar esquemas de compartilhamento de segredos (IFTENE, 2007) para combinar os serviços de confidencialidade e redundância. A parte de seu desempenho, as principais características do protocolo proposto são a inicialização simultânea da segurança de diversas unidades de dados e o emprego da renovação de dados confidenciais (HERZBERG et al., 1995). Esta última propriedade é importante no contexto da sobrevivência dos dados, pois aqueles que foram comprometidos por adversários podem ser invalidados ou tornados obsoletos, de forma que apenas os nós autorizados a participar de um processo de renovação é que podem futuramente se agrupar para reconstruir os dados originais em sigilo.

A estratégia proposta se baseia nos seguintes mecanismos:

- compartilhamento de um segredo por interpolação de polinômios, introduzido no trabalho de Shamir (1979);
- compartilhamento de múltiplos segredos sobre o modelo de Shamir (YANG; CHANG; HWANG, 2004);
- renovação de quotas sobre o modelo de Shamir (HERZBERG et al., 1995).

A diferença entre a estratégia proposta neste trabalho e os esquemas de Herzberg et al. (1995), Yang, Chang e Hwang (2004), Shamir (1979) é a renovação de quotas sobre múltiplos segredos e a aplicabilidade em RSSFs desassistidas. No exemplo da Figura 7, temos a renovação de quotas como uma etapa adicional, além das etapas de inicialização (7a) e recuperação do segredo (7b). Observa-se que o trabalho de Ren, Zhao e Ren (2009) utiliza mecanismos de redundância para o problema da sobrevivência de dados. No entanto, não há um mecanismo que possa renovar o estado da rede e tornar dados previamente comprometidos em informação obsoleta.

3.3 Extensão do Modelo do Sistema e do Adversário

Na Seção 2.2 foram apresentados o modelo do sistema e do adversário relacionados com a presente pesquisa de doutorado e também com trabalhos anteriores. Nesta Seção, apresenta-se uma extensão de tais modelos, o que é necessário para apresentar e analisar o protocolo proposto na seção que se segue (Seção 3.4).

Modelo do Sistema

Em relação ao sistema, o modelo estendido possui as seguintes premissas:

- existem n dispositivos de rede que compartilham p segredos através de um esquema de compartilhamento (t, n) . Isto significa que $t - 1$ quotas não fornecem informação sobre os segredos, enquanto t quotas são suficientes para a reconstrução do segredo. O número de segredos é menor ou igual ao limiar, i.e., $p \leq t$;
- existem canais seguros para a transmissão de quotas, ou seja, um nó que gera e distribui quotas deve ser capaz de utilizar mecanismos de cifração. A necessidade de canais seguros é discutida neste trabalho (e.g., Seção 3.5.3), porém é importante observar que esta questão é ortogonal ao problema da sobrevivência de dados, já que mecanismos de compartilhamento de segredos consideram a existência de canais seguros para transmissão de quotas;
- o tempo é dividido em intervalos fixos e de mesma duração, os quais são chamados de rodadas. Cada nó sensor pode coletar múltiplas unidades de dado por rodada. No início de uma rodada, dispositivos podem iniciar um *protocolo de atualização* iterativo a partir do qual, de acordo com o esquema proposto, termina sua execução com novas quotas geradas para o mesmo segredo. Tal protocolo também é chamado de *renovação de quotas*. Adicionalmente, esta fase de atualização possui tempo ín-

fimo se comparada com o tempo de uma rodada. Isto significa que o protocolo de atualização é eficiente em consumo de tempo;

- o protocolo de atualização supracitado é iniciado após um procedimento de atualização de chave secreta, como no trabalho de Herzberg et al. (1995). Esta premissa pode ser removida se o material criptográfico dos nós sensores for protegido por *hardware*. Alguns trabalhos anteriores removem esta premissa ao considerar que um adversário não pode monitorar a comunicação de nós que não estão atualmente comprometidos (REN; OLESHCHUK; LI, May, 2013; REN; OLESHCHUK; LI, 2010; MA; TSUDIK, 2008), o que é discutido na Seção 3.5.4;
- cada nó possui no máximo uma quota relacionada com o mesmo conjunto de segredos (i.e., segredos usados na mesma etapa de inicialização). Não há restrição para os casos de agregação de dados e renovação de quotas, os quais podem requerer que dados sigilosos sejam agregados para reconstrução do segredo ou para renovação de quotas.

A última premissa supracitada diminui o poder do $\mathcal{ADV}$ ao comprometer nós. Se um nó armazena uma certa quantidade de quotas relacionadas com o mesmo conjunto de segredos, o $\mathcal{ADV}$ precisa de menos esforço ao alternar entre diferentes nós para obter dados, já que um único nó sensor poderia possuir várias unidades de dado. Por outro lado, se cada quota armazenada em um nó sensor está relacionada com apenas um conjunto de segredos, um adversário não possuiria grande vantagem ao comprometer nós.

A premissa de que $t - 1$ quotas não fornecem informação sobre os segredos é uma propriedade do protocolo proposto, já que se usa interpolação de polinômios.

Modelo do Adversário

Em relação a um adversário, o modelo estendido possui a seguinte premissa:

- um nó comprometido durante a etapa de atualização não pode ser comprometido em rodadas adjacentes, como também considerado no trabalho de Herzberg et al. (1995).

Apesar desta premissa evitar um tipo de ataque (NIKOV; NIKOVA, 2005), a escolha de parâmetros do protocolo proposto pode fortalecer o sistema, como é discutido na Seção 3.4.

3.4 Descrição do Protocolo

O protocolo de segurança proposto (SANTOS; MARGI, 2011a) possui duas etapas principais:

1. inicialização e compartilhamento;
2. renovação de quotas.

A primeira etapa é realizada apenas uma vez por dispositivo sensor que obteve dados do ambiente, os quais são considerados como o segredo a ser compartilhado. Também é possível que o segredo seja um material criptográfico (e.g., chave secreta), o que é discutido nesta seção e ilustrado na Figura 8. O objetivo da fase em questão é gerar as quotas a partir do segredo e distribuí-las secretamente para os respectivos nós participantes do esquema. O segredo é apagado do dispositivo e só pode ser recuperado com a agregação de um subconjunto das quotas.

A segunda etapa consiste na troca de mensagens de atualização entre os nós que possuem quotas. Isso permite que os participantes possam renovar os valores de suas quotas. Desta forma, dados que não participaram da etapa são invalidados e tornam-se obsoletos por não terem mais possibilidade de serem combinados para recuperar o segredo. Esta etapa pode ser realizada diversas vezes, mas recomenda-se que ela não aconteça com muita frequência para evitar sobrecusto de comunicação.

Uma recomendação de segurança em relação à segunda etapa é o uso de um mecanismo de renovação de chaves privadas, como nos trabalhos de Herzberg et al. (1995) e Saxena, Tsudik e Yi (2009). Isso porque um adversário pode interceptar a troca de mensagens de atualização das quotas e tentar obter dados suficientes para renovar quotas eventualmente comprometidas em momentos anteriores, já que considera-se a habilidade de armazenar dados comprometidos em um catálogo. A atualização da chave privada evita que o adversário em questão possa decifrar e, portanto, interpretar as mensagens interceptadas.

Antes de descrever o protocolo, apresenta-se um resumo das etapas envolvidas.

Resumo das Etapas

A primeira etapa é baseada no esquema de Yang, Chang e Hwang (2004), o qual se baseia no trabalho de Shamir (1979). Um polinômio h é construído por um nó que realiza a inicialização, o qual usa a identidade de nós sensores como entrada para h com o objetivo de gerar quotas. Ao fim dessa etapa, compartilha-se p segredos com n nós sensores.

A reconstrução do(s) segredo(s) pode ser realizada por pelo menos t nós que agreguem suas quotas, valor chamado de *limiar*. Com o uso da interpolação de Lagrange, com t pontos é possível determinar de forma única o polinômio h de grau $t - 1$. Pode-se usar como segredos todos os coeficientes de h , mas considera-se aqui apenas os coeficientes de termos em variáveis de grau estritamente menor que p . O motivo é habilitar a etapa de renovação de quotas.

Como o esquema de Yang não possui etapa de atualização, é necessário adicionar um protocolo de renovação de chaves privadas, as quais podem ser obtidas pelo $\mathcal{ADV}$. Usamos o esquema sugerido no trabalho de Herzberg et al. (1995).

A renovação de chaves é seguida pelo processo de renovação de quotas. Modificou-se o esquema de Yang para alterar quotas de forma periódica. Após uma sequência de rodadas $l = 1, 2, \dots$, a ideia é criar um novo polinômio $\delta(x)$ de grau $t - 1$ com uma restrição: não há termos com grau menor que p ; no início da rodada l calcula-se $h^l(x) = h^{l-1}(x) + \delta(x)$. Os polinômios $h^{l-1}(x)$ e $\delta(x)$ são apagados. Como resultado, o polinômio $h^l(x)$ terá os mesmos coeficientes de $h^{l-1}(x)$ somente para os termos de variáveis com grau no máximo $p - 1$, os quais são justamente os segredos que poderão ser reconstruídos com a interpolação.

O polinômio $\delta(x)$ é calculado como a soma de n polinômios, em que cada um é gerado de forma independente e pseudo-aleatória por um nó participante, i.e., $\delta(x) = (\delta_1(x) + \delta_2(x) + \dots + \delta_n(x)) \pmod{q}$. Os polinômios $\delta_i(x)$, para $i = 1, 2, \dots, n$, estão sujeitos à regra supracitada, i.e., possui apenas termos de grau maior que $p - 1$.

O processo de renovação apresentado no esquema de Herzberg et al. (1995) possui a propriedade de que somente um segredo é compartilhado entre participantes. Como consideramos múltiplos segredos, o protocolo proposto tem uma abordagem diferente. Por isso utilizou-se o esquema de Yang (YANG; CHANG; HWANG, 2004), com a variação de que não há valores públicos adicionais.

Inicialização e compartilhamento: seja q um número primo. Seja $k_1, k_2, \dots, k_p$ um conjunto de p segredos, cada um em $\mathbb{Z}_q$, a serem compartilhados entre n nós. Define-se ID_i como a identidade do usuário i , para $i = 1, 2, \dots, n$. Seja t o limiar, definido como o número mínimo de quotas que permite a reconstrução do segredo. Seja ainda $a_1, a_2, \dots, a_{t-p}$ valores pseudo-aleatórios em $\mathbb{Z}_q$. Se $p \leq t$, ou seja, o número de segredos a serem compartilhados é menor ou igual ao valor do limiar, então os seguintes passos são executados:

1. Construa um polinômio $h(x) \pmod{q}$ de grau $t - 1$, em que $0 < k_1, k_2, \dots, k_p, a_1, a_2, \dots, a_{t-p} < q$, como a seguir:

$$h(x) = k_1 + k_2x^1 + \dots + k_px^{p-1} + a_1x^p + a_2x^{p+1} + \dots + a_{t-p}x^{t-1} \pmod{q}$$

2. Calcule as quotas $y_i = h(ID_i) \pmod{q}$ para $i = 1, 2, \dots, n$;

3. Transmite secretamente a quota y_i para o nó i .

Renovação de quotas: inicialmente, como já discutido nesta seção, nós devem trocar suas chaves privadas para maior segurança contra adversários móveis, como no trabalho de Herzberg et al. (1995) ou Saxena, Tsudik e Yi (2009). Considerando $p \leq t$, o protocolo de renovação de quotas para cada nó participante P_i , $i \in \{1...n\}$, no início do período de tempo l é:

1. P_i escolhe $t - p$ valores pseudo-aleatórios $\{b_{im}\}_{m \in \{1,2,\dots,t-p\}}$ em $\mathbb{Z}_q$ e define o polinômio $\delta_i(x) = b_{i1}x^p + b_{i2}x^{p+1} + \dots + b_{it-p}x^{t-1}$;
2. Para todos os outros nós P_j , P_i transmite secretamente $u_{ij} = \delta_i(ID_j) \pmod{q}$ para P_j ;
3. Após agregar os novos valores secretos u_{ji} , $j \in \{1...n\}$, P_i calcula sua nova quota $y_i^l = y_i^{l-1} + (u_{1i} + u_{2i} + \dots + u_{ni}) \pmod{q}$;
4. Os valores que foram utilizados para gerar a nova quota são apagados.

O valor $\delta_i(ID_j)$, enviado de P_i para todos os demais nós, é chamado de *quota parcial*.

Reconstrução dos segredos: Pelo menos t participantes agregam suas quotas e, usando a interpolação polinomial de Lagrange (SHAMIR, 1979), com o conhecimento de t pares (ID_i, y_i) , o polinômio $h(x) \pmod{q}$ de grau $t - 1$ pode ser determinado como a seguir:

$$\begin{aligned} h(x) &= \sum_{i=1}^t y_i \prod_{j=1, j \neq i}^t \frac{x - ID_j}{ID_i - ID_j} \pmod{q} \\ &= k_1 + k_2x^1 + \dots + k_px^{p-1} + a_1x^p + a_2x^{p+1} + \dots + a_{t-p}x^{t-1} \pmod{q} \end{aligned}$$

Escolha de parâmetros: o número de nós participantes do protocolo, denotado como n , pode ser menor que o número total de nós da rede. Neste caso, quotas devem ser distribuídas de acordo com a posição geográfica dos nós. Por exemplo, se um adversário concentrar um ataque em uma região de tamanho limitado, uma estratégia razoável seria distribuir quotas uniformemente a todos os nós da rede.

Seja k o número de nós comprometidos pelo adversário. O valor do limiar, denotado como t , tem a seguinte relação de segurança com k : se $k = t - p$ no instante de término de uma fase de atualização de quotas, um adversário precisa somente de uma quota parcial para interpolar o polinômio com coeficientes pseudo-aleatórios que foi utilizado. Para obter tal quota parcial, basta que o adversário compute $\delta_i(0) = 0$. Logo, com o uso de quotas obtidas em rodadas anteriores e quotas parciais computadas, um adversário

poderia obter novas quotas, ou seja, quotas atualizadas a partir da fase de renovação. Isso permite um ataque incremental, ou seja, obter maior vantagem a cada execução da fase de renovação, o que permitiria enfim reconstruir o segredo. Para mais informações sobre como escolher parâmetros de forma que se possa aumentar a segurança de mecanismos de renovação de quotas, indica-se a referência Nikov e Nikova (2005).

Uso de quotas de tamanho reduzido: em esquemas de compartilhamento de segredos, o tamanho de uma quota é geralmente igual ao tamanho do segredo. Um método mais eficiente em consumo de espaço é o *Computational Secret Sharing* (CSS) (KRAWCZYK, 1994). Em um esquema CSS (t, n) , quotas de um segredo S possuem tamanho $\frac{|S|}{t}$ mais uma quantidade de dado cujo tamanho depende de um parâmetro de segurança.

Um esquema CSS combina um algoritmo de dispersão da informação, ou *Information Dispersal Algorithm* (IDA), com um esquema de cifração e um esquema de compartilhamento de segredos. O conceito de IDA foi apresentado inicialmente por Rabin (1989), sendo geralmente implementado através de código corretores de erro.

Os passos para usar o protocolo proposto como um esquema CSS são:

1. Seja **Enc** uma função de cifração, K uma chave secreta pseudo-aleatória e S um segredo. Calcula-se $EncDATA = Enc_K(S)$;
2. Acrescenta-se redundância ao dado criptografado e, com o uso de um IDA, fragmenta-se $EncDATA$ em n partes, $E_1, E_2, \dots, E_n$;
3. Com o uso de um esquema de compartilhamento de segredos **SS**, gera-se n quotas a partir da chave K , obtendo-se $K_1, K_2, \dots, K_n$;
4. Envia-se (K_i, E_i) para cada nó participante, denotado como s_{id_i} , $i = 1, 2, \dots, n$. Em especial, K_i deve ser transmitido com o uso de confidencialidade.

A Figura 8 ilustra o processo supracitado, o qual foi proposto e demonstrado como seguro por Krawczyk (1994). Este processo também foi utilizado de forma similar no contexto da sobrevivência de dados por Ren, Zhao e Ren (2009). Na figura, um algoritmo de cifração recebe um segredo e sua saída é utilizada como entrada em um IDA, o qual produz fragmentos de dados. A chave utilizada para cifração é também entrada para um esquema de compartilhamento de segredos. O resultado (K_i, E_i) é então transmitido para um nó, sendo K_i um dado secreto.

Em relação à chave secreta K , uma alternativa ao uso de um valor pseudo-aleatório, gerado por um nó sensor, é usar a chave privada do sorvedouro. Neste caso, os valores $K_1, K_2, \dots, K_n$ são transmitidos pelo sorvedouro na sua próxima visita, incluindo sua chave pública, denotada como K_{pub} (o par (K, K_{pub}) é sempre gerado pelo sorvedouro). Um canal

Figura 8: Uso eficiente de um mecanismos de compartilhamento de segredos, representado pela caixa **SS**.

seguro para a transmissão de K_i pode ser implementado com o uso de K_{pub} . Se o segredo precisar se revelado a um nó sensor, t valores do par (K_i, E_i) devem ser agregados, o que permite recuperar o segredo cifrado e reconstruir a chave secreta para decifração.

3.5 Segurança do Protocolo Proposto

Esta seção apresenta discussões de segurança relacionadas com o protocolo proposto e com o método adotado para elaboração do protocolo, que é o compartilhamento de segredos. Uma análise detalhada da proposta é apresentada no Capítulo 4.

3.5.1 Reconstrução do Segredo

O problema relacionado com o protocolo é a recuperação do polinômio de grau $t - 1$. Um adversário precisa obter t pontos do polinômio para executar o algoritmo de reconstrução de segredo. O conhecimento de menos que t quotas não pode ser usado para obter o segredo, i.e., um número insuficiente de quotas não fornece informação sobre o segredo.

O protocolo proposto possui uma relação importante entre p e t , i.e., o número de segredos e o valor de limiar. Se $p = t$, os coeficientes do polinômio são exatamente os segredos e não há coeficiente pseudo-aleatório. Neste caso, a fase de atualização não é possível de ser executada porque esta depende da alteração dos coeficientes pseudo-aleatórios. Se $p < t$, é importante observar a quantidade de valores pseudo-aleatórios utilizados não fornece vantagem para um adversário em relação ao número de quotas necessário para recuperação do segredo. Ou seja, é necessário obter t quotas independente dos coeficientes serem parte do segredo ou um valor pseudo-aleatório. Esta é uma implicação do método utilizado, que é chamado de compartilhamento de segredos *perfeito* (KRAWCZYK, 1994; IFTENE, 2007).

3.5.2 Distribuição de Quotas

Uma questão de segurança é a distribuição de quotas. Como cada nó participante do protocolo recebe e armazena uma quota, se nós estiverem em uma mesma vizinhança, um adversário teria vantagens ao comprometer nós em uma região geográfica específica. Além disso, se um nó sensor é comprometido no início da inicialização do protocolo, os dados sigilosos seriam obtidos.

3.5.3 Canal Seguro e Renovação de Chave Secreta

É importante observar que o uso de criptografia simétrica ao invés de esquemas de chave pública é geralmente mais eficiente em consumo de tempo e energia (SANTOS et al., 2010). De fato, trabalhos anteriores (e.g., (REN; ZHAO; REN, Dec. 2009; REN; ZHAO; REN, 2009)) consideram chaves par-a-par estabelecidas entre nós antes da implantação da rede, o que permite uso de canais seguros para distribuição de quotas. Entretanto, este método possui desvantagens como a necessidade de mecanismos de gerenciamento de chaves, pois as chaves deveriam ser atualizadas, e o fato de que um adversário teria esforço reduzido para encontrar chaves na rede, já que cada valor de chave seria utilizado duas vezes. Por outro lado, esquemas assimétricos podem ser utilizados para realizar acordo de chaves, o que permite gerar um valor compartilhado a ser usado por um algoritmo de criptografia simétrica.

O uso de uma chave compartilhada entre o nó sorvedouro e cada nó sensor não se aplica para a implementação de canais seguros, já que o ambiente em questão requer troca de dados segura entre nós da rede, sem intervenção do sorvedouro.

3.5.4 Segurança da Etapa de Atualização

Dado que integridade e autenticação estão fora do escopo da pesquisa de doutorado, enfatiza-se que um esquema de compartilhamento de segredos com proatividade possui, em geral, três tipos de protocolos: distribuição de quotas, renovação de chave secreta e renovação de quotas. O processo de distribuição e atualização de quotas requer um mecanismo criptográfico para proteção de mensagens transmitidas. No modelo adotado, um adversário pode comprometer nós e obter chaves privadas. Logo, tal adversário pode usar dados secretos de nós comprometidos em períodos de tempo anteriores, o que reduz a segurança do sistema. Em especial, um adversário poderia obter quotas parciais, que são trocadas entre nós durante a etapa de atualização (Section 3.4). As quotas parciais obtidas poderiam ser utilizadas em conjunto com dados de nós comprometidos com o objetivo de calcular a nova quota, ou seja, a quota atualizada que é resultante da etapa de renovação.

Como mencionado em (HERZBERG et al., 1995), para garantir segurança proativa, um protocolo de renovação de chave secreta deve ser utilizado antes da etapa de renovação de quotas. O resultado é que um adversário que compromete um nó no período l , mas não controla o nó no período $l + 1$, não pode obter a nova chave. Assim, o ataque mencionado anteriormente não poderia ser aplicado.

Um protocolo de renovação de chaves também aumenta a segurança contra ataques de personificação. Isso porque um adversário que obtém uma chave secreta e personifica um nó sensor não pode persistir em tal atividade se a chave em questão for alterada.

3.5.5 Agregação de Dados

Como mencionado na Seção 2.3.3, a agregação de dados é muito utilizada em RSSFs. Para realizar esse processo deve-se reconstruir o segredo compartilhado para realizar processamento. De forma alternativa, é possível realizar o processamento de quotas sem a reconstrução do segredo.

A reconstrução do segredo é geralmente realizada de uma forma centralizada, i.e., através de um nó sensor. Portanto um adversário poderia comprometer apenas o nó em questão para obter e apagar o segredo. Observa-se que estratégias anteriores possuem a mesma desvantagem, já que a decifração deveria também ser realizada por um nó antes da execução do processo de agregação de dados.

Portanto, agregar dados torna-se um processo de risco para a segurança do sistema, o que poderia ser evitado com o uso de criptografia homomórfica. É importante observar que o nó sorvedouro, o qual é considerado seguro, poderia executar o processo de reconstrução caso esteja disponível.

A criptografia homomórfica fornece ferramentas para o processamento de dados cifrados e permite aumentar a segurança ao evitar que dados sigilosos sejam decifrados para processamento. O projeto de esquemas de criptografia homomórfica que sejam seguros e eficientes ainda é um desafio de pesquisa, já que os tamanhos de chave ainda são impraticáveis para redes de sensores (e.g., (CORON et al., 2011; GENTRY; HALEVI, 2011)). No caso de compartilhamento de segredos, é possível combinar quotas para fazer operações em segredos, como por exemplo soma e multiplicação (BEN-OR; GOLDWASSER; WIGDERSON, 1988).

3.5.6 Outros Modelos de Adversário

Na Seção 2.2.2, foram apresentadas premissas em relação ao modelo de adversário adotado. O único objetivo de um adversário pertencente a este modelo é procurar dados específicos e apagá-los, de forma que o nó sorvedouro não receba informações valiosas

de nós sensores. Portanto o modelo utilizado não considera ataques à integridade de mensagens, como corromper dados armazenados.

Por outro lado, o protocolo proposto pode ser estendido para suportar um modelo de adversário que inclui inconsistências de quotas, ou seja, quotas poderiam ser corrompidas por adversários. O conceito de verificação de compartilhamento de segredos (FELDMAN, 1987) poderia ser adotado em conjunto com o protocolo proposto. Desta forma, nós poderiam detectar quotas inconsistentes ou corrompidas e, caso necessário, executar um processo de restauração ou recuperação de quotas (HERZBERG et al., 1995).

3.6 Considerações do Capítulo

A elaboração de um protocolo de segurança para RSSFs desassistidas com propriedades de redundância e confidencialidade motivou o uso de mecanismos de compartilhamento de segredos. Observou-se que outras propriedades de interesse para o problema da sobrevivência de dados poderiam ser obtidas de tais mecanismos. Em especial, a renovação de quotas foi utilizada para que dados obtidos por adversários se tornem obsoletos, evitando que sejam utilizados no futuro para reconstrução do segredo. Os mecanismos mencionados permitem que o protocolo proposto maximize a sobrevivência de dados, o que é um dos objetivos principais desta tese.

Mostrou-se como utilizar quotas de tamanho reduzido através de algoritmos chamados de dispersão da informação combinados com esquemas de cifração. Além disso, nós sensores podem realizar agregação de dados através da reconstrução do segredo ou, em caso de operações mais simples, através do processamento de quotas, i.e., com o uso de criptografia homomórfica.

Espera-se que o protocolo proposto possa ser utilizado com estratégias baseadas em transmissões periódicas de dados. Dentre as estratégias existentes na literatura, vislumbra-se encontrar as mais promissoras no contexto da sobrevivência de dados. Almeja-se também analisar a eficiência do método proposto com a implementação em dispositivos reais.

4 Avaliação da Proposta

Este capítulo apresenta métodos de avaliação do protocolo de segurança para sobrevivência de dados elaborado no contexto das RSSFs desassistidas. A avaliação é composta por três etapas:

1. implementação e experimentação em dispositivos reais;
2. análise de segurança;
3. modelagem e simulação.

Nas seções que se seguem, descreve-se cada etapa e os resultados obtidos.

4.1 Implementação e Experimentação

Nesta seção, apresenta-se resultados de medições obtidos em dispositivos reais. O método adotado foi similar ao utilizado por Santos et al. (2010), o qual está descrito na Seção 2.4. Foram realizadas medidas diretas sobre o sensor TelosB (POLASTRE; SZEWCZYK; CULLER, 2005). As métricas adotadas foram consumo de energia e tempo de execução.

Em relação aos parâmetros adotados, os coeficientes do polinômio denotado como $h()$ na seção 3.4 foram inicializados com tamanho de 160 bits. Este é também o tamanho de uma quota armazenada por cada nó sensor. Foram coletadas 10 amostras para o cálculo da média – o desvio padrão foi ínfimo (menor que 0,009).

O custo para a inicialização de 80 quotas em um nó sensor é mostrado na Tabela 4. O valor de 80 quotas representa um custo similar à operação fundamental necessária para o processo de recifração, apresentada na Seção 2.4, a qual é a principal estratégia de trabalhos anteriores (PIETRO et al., 2009a). Observa-se que a etapa de inicialização do protocolo proposto deve ser executada apenas uma vez para um conjunto de segredos a serem compartilhados, e portanto esta etapa não é uma desvantagem se comparada com o processo de recifração.

A Tabela 4 também apresenta o custo para a reconstrução de segredos com o valor de limiar 9. Esta operação não depende do número total de nós da rede. O valor de limiar define a sua complexidade. Um custo de menos que 3 segundos seria razoável para muitas

aplicações se comparado com operações fundamentais de propostas anteriores de chave pública como a recifração, como mencionado anteriormente. Geralmente, os nós sensores não deveriam executar a operação de reconstrução, exceto quando operações específicas sobre os dados fossem necessárias, como é o caso da agregação de dados. No cenário das redes desassistidas, o nó sorvedouro, que possui recursos ilimitados, poderia realizar a tarefa de reconstrução no momento de sua disponibilidade.

Tabela 4: Consumo de tempo e energia para a inicialização de 80 quotas e reconstrução do segredo no dispositivo TelosB.

	Tempo (s)	Energia (mJ)
Inicialização de Quotas	8,42	46,12
Reconstrução do Segredo	2,62	14,33

Fonte: (SANTOS; MARGI, 2011a).

Reconstrução do Segredo: a Figura 9 mostra o tempo para reconstrução do segredo em relação a diferentes valores de limiar e tamanhos de coeficiente. Para este experimento, foram coletadas 100 amostras e usado um Pentium 4 3-GHz com 2-Gbyte RAM, sobre o sistema operacional Linux. Observa-se que a reconstrução do segredo geralmente é executada em um nó sorvedouro. Considerou-se valores de limiar entre 20 e 100, para os quais foram observados consumos de tempo na faixa de milissegundos.

Figura 9: Tempo de reconstrução do segredo (*Desktop*).

Fonte: (SANTOS; MARGI, 2011a).

Discussão: a implementação em dispositivos reais foi realizada para os algoritmos de inicialização de quotas e reconstrução de segredos. Mostrou-se que o custo para inicializar

quotas em um nó sensor é similar a uma das operações fundamentais da principal estratégia de trabalhos anteriores e possui a vantagem de ser executada apenas uma vez por conjunto de segredos. O processo de reconstrução do segredo também foi avaliado e apresentou menor tempo que a inicialização para um conjunto de 9 nós sensores combinarem suas quotas. A reconstrução ainda foi avaliada a partir de um *Desktop*, o que seria uma representação de um nó sorvedouro de uma RSSFs. Nesse caso, os resultados ficaram na faixa de milissegundos.

Apresenta-se nas próximas seções um estudo da segurança do sistema com o uso do protocolo proposto. Em especial, avalia-se a probabilidade do adversário de reconstruir o segredo e apagar dados.

4.2 Análise de Segurança

Nesta seção, apresenta-se uma análise de segurança com relação à probabilidade de um adversário obter sucesso em apagar o dado alvo distribuído entre nós da rede. Isto também se relaciona com a probabilidade do sistema poder reconstruir o segredo compartilhado entre nós sensores.

Inicialmente apresenta-se a notação e o cenário utilizados. Em seguida, descreve-se a probabilidade de reconstrução de segredo para um caso geral e uma extensão da análise para o contexto de diferentes estratégias de sobrevivência de dados.

4.2.1 Notação

Considera-se a seguinte notação:

- id_i : identidade de um nó sensor;
- r : índice de rodada;
- p : número de nós participantes do protocolo, i.e., número de nós que possuem uma quota no início da primeira rodada. Este parâmetro também pode ser interpretado como o número total de quotas no início da primeira rodada, já que, para maior segurança, considera-se que um nó não armazena mais de uma quota;
- q : número de nós que não participam do protocolo, i.e., número de nós que não possuem uma quota no início da primeira rodada;
- n : número total de nós, i.e., $n = p + q$;
- s_{id_i} : nó sensor de identidade id_i ;

- k : número de nós comprometidos em uma determinada rodada;
- t : limiar;
- C_r : conjunto de nós comprometidos na rodada r .

A notação supracitada não considera o número de segredos nem o número primo utilizado como módulo do esquema descrito na Seção 3.4. Estes parâmetros foram inicialmente denotados como p e q , respectivamente. Utiliza-se agora p como o número de nós que participam do esquema e q como o número de nós que não participam.

É importante diferenciar os nós que participam dos que não participam do esquema, pois o número de nós da rede pode ser maior que o número de quotas geradas. Nesse caso, alguns nós não recebem quotas durante a fase de inicialização e compartilhamento. O tamanho da rede, denotado como n , tem relação com o sucesso de um adversário. Quanto maior o número de nós da rede, mais rodadas são necessárias para se comprometer todos os nós de forma exaustiva, dado que há um número fixo de nós comprometidos em cada rodada.

4.2.2 Quotas de Tamanho Reduzido e Indistinguibilidade

Uma importante questão para um adversário é a capacidade de distinguir um dado alvo de outras mensagens, já que considera-se que este somente apaga dados específicos que foram identificados. Dado o método CSS (Seção 3.4), desde que um adversário não tenha armazenado pelo menos t quotas em seu catálogo, a chave não seria restaurada, o que implica em não se ter o dado alvo decifrado. Portanto, dados não seriam apagados até a possibilidade de decifração (ou localização) do dado alvo. Este cenário é ilustrado na Figura 10.

Dependendo dos valores de k e t , um adversário não pode apagar dados na primeira rodada. Considerando o método CSS da Figura 8 e o exemplo da Figura 10, e um esquema CSS-(3, 5) com $k = 2$, um adversário precisaria de pelo menos 2 rodadas para poder apagar dados, já que o limiar é maior que o número de nós comprometidos por rodada, i.e., $t > k$. Isso acontece devido à seguinte sequência de passos realizada por um adversário:

1. obtenção de pelo menos 3 quotas para reconstrução da chave utilizada para cifrar o segredo (ou o dado alvo);
2. obtenção de 3 fragmentos adicionais de dados cifrados para decifração e localização do dado alvo a ser apagado.

A análise de segurança da Seção 4.2.3 considera a premissa de que o segundo passo não precisa ser executado, pois fragmentos e quotas são distribuídos em conjunto, como

Figura 10: Exemplo de sequência de passos de um adversário em um esquema CSS-(3,5). Cada nó possui um par (k_i, E_i) . O adversário pode distinguir quotas e iniciar a apagar dados a partir da rodada 2. Na rodada 3, o adversário vence o jogo ao apagar a terceira unidade de dado.

descrito no passo 4 do esquema CSS da Seção 3.4. Portanto a posse de t quotas já permite que um adversário inicie o processo de apagar dados.

Pode-se agora afirmar que um adversário precisa de pelo menos $\lceil \frac{t}{k} \rceil$ rodadas para iniciar o processo de apagar dados.

4.2.3 Probabilidade de Reconstrução do Segredo

De forma similar ao trabalho de Ren, Zhao e Ren (2009), estamos inicialmente interessados na probabilidade de reconstrução dos dados. Inicialmente, considera-se $p = n$, ou seja, o cenário inicial de que cada nó sensor possui uma única quota (esta premissa é alterada nas próximas seções). É razoável considerar $n \geq t$, caso contrário nunca haveria possibilidade de reconstrução do segredo.

Sejam DR_r o evento de requisição de reconstrução de dados pelo sistema na rodada r e DS_{r-1} o evento em que um adversário não vence na rodada $r - 1$, i.e., este não apaga mais que $p - t$ quotas. Seja $P(E_r)$ a fração de quotas apagadas até a rodada r . Os possíveis valores de DR_r condicionado a DS_{r-1} são $\binom{n(1 - P(E_r))}{t}$ de um total de $\binom{n}{t}$. Logo, a probabilidade de reconstrução dos dados na rodada r condicionada ao evento de um adversário não vencer na rodada $r - 1$ é dada por:

$$P(DR_r | DS_{r-1}) = \frac{(n(1 - P(E_r)))! (n - t)!}{(n(1 - P(E_r) - t))! n!} \quad (4.1)$$

A equação 4.1 considera $n(1 - P(E_r)) \geq t$. Seja $r = 1$. Ao fim da rodada 1,

$P(E_1) = k/n$; logo $P(DR_1) = \binom{n-k}{t} / \binom{n}{t}$. Em especial, se $n = p = 100$, $k = 2$, $t = 5$, tem-se que $P(DR_1)$ é 90%. Observa-se que é possível manter o tamanho da rede e distribuir menos quotas, i.e., $p < n$. É interessante investigar diferentes valores para p e n , assim como $P(E_r)$ para cada estratégia adotada entre as listadas na Seção 2.3.

4.2.4 Probabilidade de Apagar Dados

De forma diferente da seção anterior, investiga-se nesta seção a probabilidade de um adversário apagar dados no cenário da combinação de estratégias de defesa e ataque. Considera-se as estratégias apresentadas na Seção 2.3.1 e que um adversário já obteve t quotas (i.e., pior caso para o sistema de defesa), o que é equivalente ao fato de um adversário “quebrar a confidencialidade”.

Seja L_r o evento em que um adversário não apaga dados na rodada r e CL_r o evento em que um adversário não apaga nenhum dado *até* a rodada r . Nota-se que L_r é o evento em que o adversário em questão ainda não “vence o jogo” na rodada r e CL_r a versão cumulativa de L_r .

MOVE ONCE e FRANTIC

Considera-se aqui que a estratégia de defesa MOVE ONCE é combinada com a estratégia FRANTIC de um adversário. Neste caso, as quotas são transmitidas apenas uma vez na rodada 0. O adversário encontra todas as quotas depois de $\lceil \frac{n}{k} \rceil$ rodadas. Logo,

$$P(L_r | CL_{r-1}) = \binom{n-rk}{p} / \binom{n}{p} \quad (4.2)$$

KEEP-MOVING

Adota-se aqui a estratégia de defesa KEEP-MOVING. Se esta for combinada com a estratégia de ataque LAZY, a intuição é que o adversário apenas espera que o dado seja enviado a um nó comprometido, já que este não muda o conjunto de nós sob sua custódia. Apesar disso, como se considera n maior que k , as quotas são movidas a cada rodada para nós escolhidos de forma pseudo-aleatória que podem não estar sob a custódia do adversário. Assim, o adversário não obtém grande vantagem sobre a iteração das rodadas, como acontece na combinação das estratégias apresentadas anteriormente (MOVE ONCE e FRANTIC). Isso é apresentado na equação 4.3, onde ε é um número inteiro positivo limitado pelo valor da rodada atual.

Figura 11: Probabilidade de não se apagar quaisquer dados até a rodada r . Evento L_r condicionado ao evento CL_{r-1} para os parâmetros $n = 100$, $k = 2$, $p = 10$, $t = 5$ e $\varepsilon = r/2$.

$$P(L_r | CL_{r-1}) = \binom{n - \varepsilon k}{p} / \binom{n}{p} \quad (4.3)$$

A equação 4.3 também pode ser aplicada para a estratégia FRANTIC do adversário, já que localização de quotas e conjuntos de nós comprometidos mudam de forma pseudo-aleatória.

Comparação

Uma comparação entre as equações 4.2 e 4.3 mostra que a estratégia MOVE ONCE combinada com FRANTIC (primeira equação) fornece menor segurança. Com o decorrer do tempo, ou seja, quanto maior o número de rodadas, a estratégia KEEP MOVING apresenta mais chance de sobrevivência dos dados que MOVE ONCE, o que é ilustrado na Figura 11. Adicionalmente, se o valor de p é próximo de n , um adversário possui maior chance de apagar os dados, deixando menos quotas para a fase de atualização.

O gráfico da Figura 11 utilizou uma estimativa para o parâmetro ε , o qual é limitado pelo valor da rodada atual. Se ε é próximo de r , não há diferença entre as equações 4.3 e 4.2. Com a estimativa utilizada tem-se o valor médio.

Com a simulação do comportamento do adversário e do sistema de defesa, apresentados na seção seguinte, espera-se obter resultados que possam ser comparados com o estudo teórico. Em especial, vislumbra-se confirmar quais estratégias são mais seguras no contexto do método de segurança proposto neste projeto de pesquisa de doutorado.

É importante observar que os resultados apresentados na Figura 11 não são considerados como probabilidade de sobrevivência de dados, como será o caso dos resultados das próximas seções. Isso porque a figura em questão não inclui a propriedade de redundância oferecida pelo protocolo proposto, ou seja, a análise teórica é limitada pelo pior caso já que não tolera que um adversário apague qualquer unidade de dado.

4.3 Modelagem e Simulação com Redes de Petri Coloridas

As redes de Petri são uma ferramenta para modelagem gráfica e matemática para estudo e descrição de sistemas (MURATA, 1989). Elas fornecem recursos como ferramentas visuais, que podem ajudar a simular sequências de eventos do sistema, e a base formal para análise do comportamento de sistemas.

O conceito das redes de Petri foi apresentado inicialmente por Carl Adam Petri em 1962. Desde então, a teoria das redes de Petri evoluiu por possuir diversas aplicações, incluindo sua integração com linguagens de programação.

As redes de Petri são geralmente divididas em redes clássicas ou de baixo nível (DESEL; REISIG, 1998) e redes de alto nível (ISO/IEC 15909, 2000). As redes de Petri *elementares* e as redes *lugar-transição* são exemplos de redes clássicas, enquanto as redes *predicado-transição* e *coloridas* são exemplos de redes de alto nível.

A principal diferença entre as classes supracitadas são a distinguibilidade de *marcas* e *inscrições de arcos*. Já que estas propriedades não estão descritas nas redes clássicas, a modelagem pode se tornar um processo complexo devido ao grande número de elementos da representação gráfica. Por isso neste trabalho se utiliza as redes coloridas (JENSEN, 1987).

Para uma definição clara das redes coloridas usa-se os termos tipo, valor, operação, expressão, variável e atribuição exatamente como se usa em linguagens de programação funcionais. A notação que se segue é similar ao trabalho de Jensen (1994). Antes de apresentar a definição de redes coloridas, considera-se a seguinte notação:

- o *tipo de uma variável*, digamos v , é denotado por **Tipo**(v);
- o *tipo de uma expressão*, digamos expr , é denotado por **Tipo**(expr);
- o *conjunto de variáveis* em uma expressão é denotado por **Var**(expr);
- o tipo *booleano* (contendo os elementos {falso, verdadeiro}) é denotado por **Bool**.

Defini-se redes de Petri coloridas como:

Definição 1. Uma rede de Petri colorida é uma ênupla $(\Sigma, P, T, A, N, C, G, E, I)$ que satisfaz os seguintes requisitos (JENSEN, 1994):

1. Σ é um conjunto finito de tipos não vazios, chamados *colour sets* ou **cores**.
2. P é um conjunto finito de **lugares**.
3. T é um conjunto finito de **transições**.
4. A é um conjunto finito de **arcos** tal que:
 - $P \cap T = P \cap A = T \cap A = \emptyset$.
5. N é uma função de **nó**, definida de A em $P \times T \cup T \times P$.
6. C é uma função de **cor**, definida de P em Σ .
7. G é uma função de **guarda**, definida de T em expressões tal que:
 - $\forall t \in T: [\text{Tipo}(G(t)) = \text{Bool} \wedge \text{Tipo}(\text{Var}(G(t))) \subseteq \Sigma]$.
8. E é uma função de **expressão de arco**, definida de A em expressões tal que:
 - $\forall a \in A: [\text{Tipo}(E(a)) = C(p) \wedge \text{Tipo}(\text{Var}(E(a))) \subseteq \Sigma]$, em que p é um lugar de $N(a)$.
9. I é uma função de **inicialização**, definida de P em expressões fechadas tal que:
 - $\forall p \in P: [\text{Tipo}(I(p)) = C(p)]$.

Neste documento chama-se cada elemento do conjunto Σ (item 1 da definição) de *marca*¹, sendo que a função de cor (item 6) define qual lugar pode estar associado a determinada marca². Exemplos de marca são nós da rede, quotas ou ainda uma combinação entre nó e quota. O termo expressão é usado de forma equivalente ao termo inscrição. Uma **inscrição de arco** (item 8) é uma função ou procedimento que, ao ser calculado, resulta em uma ou mais marcas de uma mesma cor. O posicionamento de marcas em lugares individuais é chamado de *estado da rede*³ ou *marcação*, a qual pode mudar de acordo com o *disparo* de uma transição, que acontece de acordo com *regras de disparo*.

Observa-se que lugares são representados graficamente como círculos ou elipses e transições como retângulos, como será ilustrado nas próximas seções. O objetivo deste documento não é apresentar detalhes sobre a teoria das redes de Petri. Uma abordagem

¹ Do inglês, *token*.

² O termo cor é mais preciso que o termo marca, já que marcas se diferenciam por cores e, portanto, qualquer cor é uma marca.

³ É implícito que “estado da rede” significa estado da rede de Petri. Da mesma forma, “redes coloridas” significa redes de Petri coloridas.

completa no contexto das redes coloridas pode ser encontrada no trabalho de Jensen (1994). Recomenda-se ainda o trabalho de Murata (1989), este no contexto das redes clássicas.

4.3.1 Escopo do Modelo

O modelo projetado através do uso das redes de Petri coloridas é composto pelo comportamento do adversário e do *sistema da rede* ou *sistema de defesa*, o qual fornece segurança contra o adversário em questão. Portanto o modelo tem a característica de um *jogo de marcas* com atividades de *defesa* e *ataque*. O adversário sempre *vence* o jogo, ou seja, os dados não sobrevivem (são apagados) em um determinado momento. O importante é qual o momento exato em que o jogo terminou, ou seja, quanto tempo o sistema de defesa suportou os ataques do adversário. Como mencionado anteriormente, o tempo é dividido em rodadas para simplificar a análise, sendo que em cada rodada são permitidas algumas atividades específicas.

Observa-se que o protocolo proposto é parte do sistema de defesa. Algumas estratégias de defesa e ataque no contexto da rede podem ser usadas em conjunto com o protocolo proposto, como mencionado na Seção 2.3. Tais estratégias também foram utilizadas em trabalhos anteriores (PIETRO et al., 2008; PIETRO et al., 2009a) para observar o comportamento de algoritmos adotados para maximizar a sobrevivência de dados.

4.3.2 Modelo

Antes de apresentar a representação gráfica do modelo construído em redes coloridas, descreve-se uma síntese do projeto deste modelo.

Marcas e seus Tipos

As marcas definidas para a rede colorida foram nós e quotas, representados como NODE e SHARE, respectivamente. Define-se nó como um dispositivo sensor, o qual possui identificação e pode ter em sua memória não mais que uma quota. Se fosse possível armazenar mais que uma quota o adversário poderia obter mais dados ao comprometer apenas um nó, facilitando o ataque. Define-se como nó *participante* um nó que possui uma quota armazenada em sua memória. Caso um nó não tenha nenhuma quota este é chamado de nó *não-participante*. Seja p o número inicial de participantes e q de não-participantes, sendo $t = p + q$. As marcas definidas são:

- colset NODE = index n with $1..t$, i.e., $n(1), n(2), \dots, n(t)$
- colset SHARE = index s with $1..p$, i.e., $s(1), s(2), \dots, s(p)$

Os tipos criados facilitam o uso de parâmetros como número de nós e quotas, tornando o modelo flexível e escalável. A flexibilidade está relacionada com o fato de existir diferentes relações entre número de nós e quotas, sendo que a escalabilidade vem do fato de se poder definir um grande número de nós e quotas.

A forma definida para se modelar um nó que possui uma quota foi usar um par $(NODE, SHARE)$. A Figura 12 ilustra as marcas. Observa-se que os lugares da figura possuem tipos, o que, como já mencionado, restringe o posicionamento das marcas.

Figura 12: Marcas definidas para o modelo da rede de Petri colorida: nós, quotas e pares (nó,quota).

Resolução de Conflitos

Não há um lugar exclusivo para uma marca que representa um nó, já que um lugar do tipo `NODE` pode possuir marcas que representam quaisquer nós (e.g., $n(1)$, $n(2)$, ..., $n(t)$). O mesmo vale para marcas que representam quotas e pares (nó,quota), pois o modelo projetado possui essa característica. A desvantagem é que isso pode gerar conflitos como o ilustrado na Figura 13.

Para evitar conflitos e manter a abstração do modelo optou-se por usar listas, que são sequências de marcas do mesmo tipo. Exemplos de listas de quotas, nós e pares (nó,quota) são: $[s(1), s(2)]$; $[n(1), n(2)]$; $[(n(1),s(1)), (n(2),s(2))]$.

As listas permitem que marcas sejam consumidas simultaneamente com apenas um evento, ou seja, o disparo de uma transição. É possível realizar diversos tipos de operações sobre listas, como inserir e remover elementos, obter o tamanho e comparar listas.

Funções de Guarda

Uma função de guarda pode ser usada em transições para testar valores de variáveis, as quais estão presentes em inscrições de arcos de entrada. Isso permite o emprego de

Figura 13: Exemplo de conflito. Não há como prever exatamente quais marcas devem se posicionar no lugar NodesAndShares após o disparo da transição.

restrições, como por exemplo evitar que uma transição dispare se os valores das variáveis x e z são diferentes. Ainda neste exemplo, a variável x pode assumir o valor de um nó participante e z de um nó comprometido; a igualdade dessas variáveis significa que um nó sob custódia do adversário possui uma quota e, portanto, uma ação poderia ser tomada (e.g., quota apagada). A ação pode ainda depender de outro teste, ou seja, outra função de guarda para a mesma transição.

Observa-se que neste trabalho não usamos funções pseudo-aleatórias em transições, ou seja, em funções de guarda. Isso pode causar comportamentos inesperados com a simulação do modelo da rede, pois não poderia ser determinado se uma transição deve ou não disparar. Por outro lado, usa-se pseudo-aleatoriedade em inscrições de arcos (e.g., gerar as marcas que representam os nós comprometidos na rodada atual).

Prioridades

Se mais de uma transição está habilitada em um dado instante, é possível definir qual delas irá disparar primeiro. Isso permite impor restrições à dinâmica da rede, que é justamente a mudança do posicionamento das marcas causado por eventos (disparos de transições).

As prioridades de transição são usadas neste trabalho principalmente para permitir que um adversário possa reagir imediatamente quando se detecta que dados suficientes foram apagados, ou seja, não há mais necessidade de continuar com procedimentos de ataque. Em geral, o modelo projetado considera que o adversário tem prioridade em executar eventos. Isso porque em segurança deve-se considerar o pior caso. Um exemplo é o caso em que o adversário tenta apagar dados que podem ser transmitidos para outro dispositivo no momento do ataque, ou seja, pode haver vantagem do sistema de defesa. Para que a prioridade de ação do adversário não influencie nas estratégias de defesa, há condições no modelo que evitam a ação repetitiva do adversário em uma mesma rodada.

Tais condições não tem relação com o recurso de prioridade de transições.

Cuidados com Atribuições

Uma marca pode ser atribuída a uma variável. Isso pode ser entendido como a atribuição de valores em linguagens de programação. O modelo foi projetado de forma que se tenha extremo cuidado ao se atribuir valores, pois, se isso não for observado, a atribuição pode ser não-determinística, como ilustrado na Figura 14. O mesmo vale para o caso da atribuição em funções de guarda, que acontece em transições, o que é ilustrado na Figura 15. Neste último caso há um agravante: casos em que uma variável que não está em arcos de entrada é atribuída na função de guarda, o que não seria detectado pela ferramenta de *software* utilizada, o CPN Tools (JENSEN; KRISTENSEN, 2009).

Figura 14: Atribuição de uma marca para a variável y não é determinística. Neste exemplo, pode-se ter $y=1$ ou $y=2$.

Fonte: CPN Tools.

Figura 15: Variável i não está no arco de entrada, mas é atribuída à variável j na função de guarda. Adicionalmente, variável b pode ser *true* ou *false*, pois também não tem associação em arco de entrada.

Fonte: CPN Tools.

O uso de listas evita este tipo de situação, já que a atribuição passa a ser através de conjuntos de dados, o que minimiza o uso de variáveis na atribuição.

Modelo Hierárquico

O modelo projetado através das redes de Petri coloridas deve ser uma representação fiel do cenário da Figura 16, o qual descreve a combinação das estratégias KEEP MOVING e FRANTIC com o protocolo proposto. Outras estratégias de defesa e ataque, mencionadas na seção 2.3.1, podem ser facilmente derivadas a partir da figura em questão.

Figura 16: Cenário em que as estratégias KEEP-MOVING (defesa) e FRANTIC (adversário) são combinadas com o protocolo proposto.

Inicialização: a partir da leitura de p valores, nó sensor calcula p quotas e distribui de forma exclusiva para $s_{id_1}, s_{id_2}, \dots, s_{id_p}$. Os nós $s_{id_{p+1}}, s_{id_{p+2}}, \dots, s_{id_n}$ não recebem quotas.

Início da Rodada 0

$\mathcal{ADV}$ tem conhecimento dos parâmetros do sistema e inicializa C_r ($C_r \leftarrow \emptyset$). Nós participantes escolhem de forma aleatória nós destino para transmissão de quotas. Cada nó recebe no máximo uma quota.
 $r \leftarrow 1$;

Início da Rodada r

1. $\mathcal{ADV}$ escolhe aleatoriamente o conjunto C_r com k nós comprometidos nós comprometidos, dado que $C_r \cap C_{r-1} = \emptyset$;
2. **se** $\mathcal{ADV}$ está habilitado a distinguir quotas, **então**
para cada nó comprometido que possui uma quota;
 $\mathcal{ADV}$ armazena a quota em um catálogo e a apaga do nó sensor;
3. cada participante escolhe aleatoriamente um nó destinatário entre $s_{id_1}, s_{id_2}, \dots, s_{id_n}$ com a restrição de que um nó somente pode receber até uma quota;
4. nós transmitem suas quotas para destinatários escolhidos;
5. cada nó recebe uma única quota;
6. **se** $\mathcal{ADV}$ possui pelo menos t quotas, **então**
confidencialidade é *quebrada*;
7. **se** $\mathcal{ADV}$ possui pelo menos $p - t + 1$ quotas, **então**
dados não sobreviveram e jogo é terminado;
8. **senão**
 $r \leftarrow r + 1$;
reiniciar no passo 1;

A Figura 17 apresenta o modelo do sistema de ataque e defesa, que inclui o comportamento do protocolo proposto. Observa-se que as transições com barras duplas são *subredes*, já que se trata de um modelo hierárquico.

Considera-se que o protocolo já foi inicializado, ou seja, que as quotas foram criadas (e.g., a partir do dado alvo ou de uma chave criptográfica) e distribuídas para os nós. A etapa de renovação de quotas não foi considerada no modelo em questão. Observa-se que o processo de renovação de quotas tem como objetivo tornar obsoletos os dados de nós sensores comprometidos, com potencial para maximizar a sobrevivência de dados.

Uma característica do modelo é que a rodada atual sempre é incrementada quando se recebe todas as quotas, ou seja, uma vez que todas as quotas transmitidas foram recebidas em seus respectivos destinatários há o incremento da rodada.

A transição ADV-Deleted Enough Shares é disparada se o adversário conseguiu apagar quotas suficientes para evitar a possibilidade de recuperação do segredo, ou seja, caso em que o adversário vence o jogo. A transição ADV Win foi inserida para

Figura 17: Modelo da rede de Petri que inclui o jogo de defesa e ataque. O protocolo é parte do sistema de defesa.

facilitar o entendimento do modelo. Ela também é utilizada para indicar o fim da simulação através da ferramenta de *software* utilizada, o CPN Tools (JENSEN; KRISTENSEN, 2009).

Comportamento do Sistema de Defesa

O sistema de defesa inclui o protocolo proposto e uma estratégia de sobrevivência de dados (Seção 4.3.1). As figuras 18 e 19 ilustram o modelo para recepção e envio de quotas, respectivamente. Observa-se que esses modelos são subredes do modelo principal.

No caso da recepção de quotas (Figura 18), observa-se que há escolha pseudo-aleatória dos nós que serão destinatários de mensagens (transição *Randomly Choose Nodes*). Após o disparo da transição *Receive Share*, todos os nós que são não-participantes (lugar *NonParticipant-Nodes*) e recebem atribuições de quotas (lugar *Randomly Chosen Destination Nodes*) tornam-se nós participantes, com posicionamento das marcas no lugar *List of Participant Nodes*. Adicionalmente, as quotas que estavam sendo transmitidas são removidas do lugar *SharesInTransit*, pois são associadas com seus nós de destino.

Como já se pode perceber pela descrição anterior, a transmissão de quotas (Figura 19) possui o seguinte objetivo: mover quotas de nós participantes para o lugar *SharesInTransit*, sendo que todos os nós ficam sem quotas e suas marcas são po-

4.3.3 Simulação e Resultados

A modelagem do sistema, que inclui o protocolo proposto e o adversário, foi implementada na ferramenta de *software* CPN Tools (JENSEN; KRISTENSEN, 2009). Não foi considerada a etapa de renovação de quotas e, como mencionado anteriormente, o tempo foi dividido em rodadas para evitar o uso de *marcas com tempo*, simplificando o processo. As figuras 21a e 21b mostram os resultados da sobrevivência de dados em termos de distribuição cumulativa. A diferença entre as figuras é que na segunda não há o uso de confidencialidade, ou seja, considera-se que o adversário *quebrou* a confidencialidade dos dados a partir da primeira rodada. Isso significa que o adversário sabe distinguir qualquer quota de um valor pseudo-aleatório e, portanto, já é possível apagar o dado em questão. Este último processo pode ser comparado ao uso de replicação.

Figura 21: Probabilidade de sobrevivência dos dados para comprometimento de dois nós por rodada, dada uma rede de 100 nós com 10 quotas e limiar de valor 5.

É importante observar que foram obtidas 500 amostras e a simulação não é determinística, ou seja, há o uso de pseudo-aleatoriedade para definir quais nós devem ser comprometidos e quais quotas devem ser alocadas para os nós. Para cada simulação anotou-se a rodada em que o adversário venceu o jogo (ou que o dado não sobreviveu). Os resultados foram agrupados a cada 5 rodadas. Por exemplo, até a rodada x o adversário venceu o jogo i vezes de um total de 500 simulações, e até a rodada $x + 5$ o adversário venceu o jogo j vezes somado com as i vezes das rodadas anteriores (i.e., $i + j$) e assim por diante. Desta forma obtém-se a rodada máxima em que o adversário vence o jogo entre todas as simulações executadas, quando a probabilidade do adversário em vencer o jogo é 100%. Observa-se que os resultados apresentados nesta seção são o complemento do evento em que o adversário vence o jogo, ou seja, a sobrevivência de dados.

Como pode ser visto na Figura 21a, a combinação das estratégias MOVE ONCE/FRANTIC apresenta maior número de rodadas com chance máxima (100%) de so-

brevivência dos dados, como já esperado após o estudo teórico. Apesar disso, a partir de aproximadamente 50 rodadas há queda brusca neste quesito, com grandes chances para o adversário apagar o dado alvo. As outras estratégias apresentam maior estabilidade a partir deste mesmo ponto (50 rodadas) e não apresentam diferença significativa entre si.

A Figura 21b mostra que há desvantagem em relação ao caso em que o adversário deve vencer a barreira da confidencialidade antes de iniciar o processo de apagar o dado alvo. Neste caso, as combinação das estratégias MOVE ONCE/FRANTIC apresentou menos resistência ao adversário, sendo que as outras estratégias se mostraram seguras até pelo menos 30 rodadas, já que nestes casos apresentam mais de 80% de probabilidade de sobrevivência.

Observa-se que a combinação das estratégias MOVE-ONCE/FRANTIC apresenta queda brusca de segurança porque o adversário sempre encontra o dado alvo após esgotarem as possibilidades de nós procurados. Isso porque cada unidade de dado fica armazenado em apenas um único lugar (ou nó sensor), sem ser movido para outros nós possíveis.

Uma análise do resultado apresentado nas figuras 21a e 21b indica quando seria importante iniciar o processo de renovação de quotas. Um parâmetro pode ser criado para que o processo de renovação seja iniciado automaticamente na rodada em que se observa uma diminuição da segurança do sistema, ou seja, quando a probabilidade de sobrevivência de dados é menor que um valor estabelecido pelo parâmetro.

4.4 Considerações do Capítulo

A análise do método proposto para o problema da sobrevivência de dados em RSSFs desassistidas foi feita com base em três etapas: implementação, estudo teórico e simulação.

A implementação em dispositivos reais foi realizada para os algoritmos de inicialização de quotas e reconstrução de segredos. Mostrou-se que o custo para inicializar 80 quotas em um nó sensor é similar a uma das operações fundamentais da principal estratégia de trabalhos anteriores. Nota-se que 80 quotas é um valor relevante em muitas aplicações, já que uma quota seria enviada de forma exclusiva para cada nó sensor da rede. Em relação ao processo de reconstrução, utilizou-se um valor de limiar que equivale a 9 nós sensores agregarem suas quotas para obter o segredo. O custo de menos que 3 segundos foi obtido a partir de um nó sensor, o que é mais eficiente que as principais operações fundamentais do principal mecanismos de chave pública de propostas anteriores, a recifração.

A análise de segurança foi realizada através de dois métodos: estudo teórico e

simulação com o uso das redes de Petri. É possível comparar o resultado do modelo teórico com a simulação, o que permite observar que a estratégia MOVE ONCE combinada com a estratégia FRANTIC apresentaram menor segurança que as outras estratégias analisadas. Observa-se que as redes de Petri também permitem fazer uma descrição do sistema, auxiliando na análise da aplicabilidade do método proposto. Isso é feito através de um jogo de marcas do modelo da rede de Petri.

5 Aplicação em Redes Heterogêneas

Neste capítulo apresenta-se um cenário de aplicação para o protocolo descrito na Seção 3.4. Em particular, considera-se as Redes de Segurança Pública¹, ou RSPs, que podem ser consideradas como um tipo de rede heterogênea. O estudo do protocolo no cenário de RSP resultou na seguinte publicação:

- **Santos, M. A. S.**; Margi, Cíntia B.. *A Secure Multi-Party Protocol for Sharing Valuable Information in Public Safety Networks*. In: Mobile Ad Hoc Networks for Public Safety Systems; Techniques and Challenges in On-Going Work - co-located with IEEE MASS 2011., 2011, Valencia. In Proceedings: 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (**IEEE MASS**). Washington, DC, USA : IEEE Computer Society, 2011. p. 935-940.

Apresenta-se também um estudo de caso sobre a implantação de RSPs com o uso das redes definidas por *software*. Em particular, propõe-se protocolos para estender a conectividade de dispositivos, incluindo a segurança como parte integrante da solução. As seguintes publicações foram decorrentes do trabalho realizado:

- Nunes, B., **Santos, M. A. S.**, Margi, C.B., Oliveira, B.T., Obraczka, K., Turletti, T., *Software-Defined Networking Enabled Capacity Sharing in User Centric Networks*, IEEE Communications Magazine, 2014 (aceito para publicação);
- **Santos, M. A. S.**, Nunes, B., Obraczka, K., Turletti, T., Margi C.B., Oliveira, B.T. *Decentralizing SDN's Control Plane*, IEEE Local Computer Networks (**IEEE LCN**), 2014. (*short paper* aceito para publicação);
- **Santos, M. A. S.**; de Oliveira, B.T.; Margi, C.B.; Nunes, B.AA; Turletti, T.; Obraczka, K., *Software-defined networking based capacity sharing in hybrid networks*, Network Protocols (**IEEE ICNP**), 2013 21st IEEE International Conference on , vol., no., pp.1,6, 7-10 Oct. 2013.

¹ Do inglês, Public Safety Networks.

5.1 Redes de Segurança Pública

As redes de segurança pública (SANTOS; MARGI, 2011b), ou RSPs, são utilizadas por autoridades para alertar a população sobre situações de emergência como é o caso de desastres naturais ou ataques terroristas. As RSPs podem ser consideradas como redes heterogêneas, pois geralmente possuem diversos tipos de dispositivo (celulares, computadores, etc.) e, portanto, requerem interoperabilidade. Há também a característica de heterogeneidade de rede, pois RSPs geralmente operam sobre diferentes padrões de comunicação (e.g. padrões IEEE 802.11 (IEEE, 2007) e 802.16 (IEEE, 2004)²). Além disso, é necessário que se tenha mecanismos de resiliência a falhas (e.g., falta de energia, danos físicos).

No contexto do protocolo apresentado neste trabalho (Seção 3.4), a interoperabilidade pode ser atendida através da agregação dos dados de diferentes dispositivos para reconstrução do segredo. No caso da resiliência a falhas, existe um mecanismo de redundância que pode ser utilizado para evitar a perda de dados por problemas com dispositivos da rede.

Considera-se a arquitetura CHORIST (EADS Defence and Security Systems, 2009) para implantar segurança em RSPs. O acrônimo CHORIST (*integrating Communications for enHanced envirOnmental RISK management and citizens safeTy*) é também referência a um projeto europeu que propõe soluções para aumentar a eficiência e a agilidade de intervenções na presença de um desastre natural ou industrial, com o propósito de fornecer segurança a pessoas atingidas e comunicação entre agentes de resgate (e.g., bombeiros, policiais).

A Figura 22 ilustra a arquitetura CHORIST, a qual possui três módulos principais:

1. avaliação de risco;
2. disseminação de alertas;
3. implantação eficiente de sistemas.

O módulo 1 define que autoridades possam avaliar a situação da área geográfica afetada e tomar decisões de como se proceder (e.g., implantação de equipes de resgate, aviso da população afetada). Neste contexto, dados valiosos devem ser trocados entre autoridades ou oficiais de segurança pública durante esta fase, de modo que uma ação apropriada possa ser tomada.

Propõe-se como um dos resultados da presente pesquisa de doutorado, e no trabalho de Santos e Margi (2011b), que o módulo 1 da arquitetura CHORIST tenha o serviço

² Os padrões 802.11 e 802.16 também são conhecidos como WiFi e WiMAX, respectivamente.

Figura 22: A arquitetura CHORIST.

Fonte: (EADS Defence and Security Systems, 2009)

de confidencialidade com interoperabilidade e resiliência a falhas através do protocolo proposto em Santos e Margi (2011a) e também descrito na Seção 3.4. Adicionalmente, dados valiosos não seriam consolidados em apenas uma agência de segurança pública, inserindo um mecanismo de confiança entre as agências.

A aplicação ao cenário de segurança pública também inclui a etapa de renovação de quotas do protocolo proposto. Agências de segurança poderiam agregar suas quotas para um processo de atualização. Agências que não fossem autorizadas a participar desse processo teriam suas quotas invalidadas e não poderiam mais agregar seus valores com outras autoridades para reconstrução do segredo.

Simulação e resultados

O objetivo da simulação é demonstrar o processo de renovação de quotas com uma biblioteca criptográfica e também avaliar o protocolo no contexto do cenário adotado. Realizamos as simulações com o uso do simulador Qualnet³ em conjunto com a biblioteca RELIC⁴. Também modificamos a aplicação *Constant Bit Rate* (CBR) para reproduzir o procedimento de renovação de quotas. Consideramos que n nós participantes já tenham suas quotas recebidas, o que equivale à etapa de inicialização e compartilhamento. A

³ <http://www.scalable-networks.com>

⁴ <http://code.google.com/p/relic-toolkit/>

aplicação executada por cada nó participante calcula e transmite uma quota parcial aos demais nós. Cada nó participante que recebe $n - 1$ quotas parciais na camada de aplicação realiza a soma dos valores recebidos com sua quota, o que resulta na nova quota. O procedimento de renovação é incompleto para um nó da rede se pelo menos uma quota parcial não for recebida.

A simulação diferencia roteadores de nós participantes. Utilizamos 25 roteadores *mesh* dispostos em grade para cobrir toda a área geográfica. Cada roteador está na área de transmissão de seus vizinhos. Implantamos de 10 a 50 nós participantes do protocolo de segurança em distribuição uniforme para uma região geográfica de $2000 \times 2000 \text{ m}^2$. A configuração do cenário é similar ao trabalho de Karaoglan, Levi e Savas (2010). Observa-se ainda que o cenário de redes *mesh* também foi usado no contexto de redes de segurança pública no trabalho de Bruno, Conti e Gregori (2005).

Para evitar colisões cada nó espera até 1 segundo antes de transmitir um pacote. Optamos por este procedimento para evitar possíveis deficiências do modelo de simulação de protocolos da camada de enlace. Coletamos 50 amostras e as barras de erro representam um intervalo de confiança de 95%. Utilizamos pacotes de 256 ou 1024 bytes, sendo que a aplicação CBR possui 28 bytes de cabeçalho. O cenário de simulação é resumido na Tabela 5.

Tabela 5: Cenário de simulação

Dimensão	2000x2000 m^2
Número de nós	10 a 50
Tamanho de pacote	256 ou 1024 bytes
Camada de transporte	<i>User Datagram Protocol</i> (UDP)
Protocolo de roteamento	<i>Dynamic Source Routing</i> (DSR)
Camada de enlace	802.11
Taxa de transmissão	11 Mbps
Potência de transmissão (nós)	15 dBm
Potência de transmissão (roteadores)	22 dBm
Fator de ruído	0 dB

O cenário adotado não utiliza serviços orientados a conexão. O protocolo UDP foi escolhido para evitar o custo de estabelecer conexões fim-a-fim. Para evitar sucessivas retransmissões e problemas de interferência na camada de enlace, considerou-se a inexistência de fatores de ruído.

As principais métricas adotadas foram:

- **tempo da renovação de quotas:** definido como o tempo para cálculo de uma nova quota para todos os participantes do procedimento;

- **taxa de sucesso da renovação de quotas:** definida como o número de nós que podem calcular a nova quota com sucesso dividido pelo número total de participantes.

Os resultados de simulação são mostrados nas figuras 23a e 23b. O número de nós apresentou grande impacto no tempo de renovação de quotas (Figura 23a). Isso acontece devido a mensagens trocadas entre nós para atualização dos valores das quotas, já que se um nó muda seu estado os outros também devem atualizar seus valores em um processo de sincronia. Apesar disso, a renovação não é um processo que precisa ser executado com muita frequência, já que se trata de uma atualização de segurança.

Em relação à taxa de sucesso (Figura 23b), há impacto devido a qualquer quota parcial perdida. Isso porque as quotas parciais de nós participantes devem ser transmitidas e agregadas, como descrito na Seção 3.4. O uso de roteadores *mesh* para cobertura da área geográfica foi importante para evitar perdas. Pelo menos 66% dos nós estariam aptos a renovar suas quotas no cenário considerado.

Figura 23: Resultados de simulação em relação à renovação de quotas com o uso do simulador Qualnet.

(a) Tempo da renovação de quotas.

(b) Taxa de sucesso da renovação de quotas.

5.2 Implantação Segura de RSPs

A arquitetura CHORIST, apresentada na Figura 22, possui um módulo relacionado com a implantação de RSPs. O módulo 3 destaca a importância de fornecer serviços de comunicação para dispositivos de usuário que atuam em campo (EADS Defence and Security Systems, 2009). No contexto de uma única agência de segurança (e.g., bombeiros), times de resgate devem ser capaz de se comunicar com suas respectivas bases. Além disso, é importante fornecer serviços de rede para diferentes agências de segurança.

A Figura 24 apresenta um exemplo de implantação de RSPs, em que veículos são capazes de atuar como *gateways* entre diferentes agências e como *pontos de acesso* para dispositivos em campo. Observa-se na figura que um dispositivo de usuário poderia estar no limiar do alcance do sinal de um ponto de acesso. Seria possível que um dispositivo ainda esteja fora do alcance por questões de mobilidade. Portanto seria vantajoso que a conectividade de um dispositivo seja estendida através de outros dispositivos, os quais estariam por sua vez dentro do alcance do ponto de acesso. Isso pode ser realizado através das redes de compartilhamento de capacidade.

Figura 24: Exemplo de implantação de RSPs.

Fonte: (EADS Defence and Security Systems, 2009).

5.2.1 Redes de Compartilhamento de Capacidade

A necessidade de conectar dispositivos de rede de forma eficiente, permitindo que um dispositivo possa também compartilhar seus recursos para oferecer conectividade a outros dispositivos, motivou a criação das redes de compartilhamento de capacidade (NUNES et al., 2014b; SANTOS et al., 2013). Nesse tipo de rede, também chamado de redes centradas no usuário⁵ (NUNES et al., 2014b), dispositivos de usuário podem atuar como provedores de recursos.

A Figura 25 ilustra uma rede de compartilhamento de capacidade em que dispositivos de usuário (DUs) podem obter conectividade através de pontos de acesso ou de outros DUs, chamados de *gateways* (GW). Neste exemplo, dispositivos de usuário podem também formar uma rede ad hoc. Além disso, há uma entidade que gerencia parte da rede (e.g., autoriza ou bloqueia fluxos de dados, distribui chaves criptográficas), chamada de controlador (NUNES et al., 2014b).

⁵ Do inglês, *user centric networks*.

Figura 25: Exemplo de uma rede de compartilhamento de capacidade. Dispositivos de usuário (DUs) podem ser conectados via pontos de acesso ou via outros DUs.

Fonte: (NUNES et al., 2014b).

5.2.2 Aplicação das Redes de Compartilhamento de Capacidade em RSPs

As redes de compartilhamento de capacidade podem ser utilizadas no contexto do cenário da Figura 24. Veículos de diferentes agências de segurança formam uma rede ad hoc. DUs que atuam em campo podem se conectar em pontos de acesso sem fio instalados em veículos. DUs podem atuar como *gateways* e estender a conectividade para outros DUs, nos moldes do exemplo da Figura 25. Observa-se ainda que usuários podem formar uma rede ad hoc com seus dispositivos, a qual poderia estar conectada com a rede infraestruturada a partir de DUs habilitados como *gateways*.

No contexto da segurança, vislumbra-se serviços adicionais ao mecanismo de compartilhamento de dados apresentado na Seção 5.1. Propõe-se aqui, através da elaboração de um protótipo de RSPs, uma arquitetura para redes de compartilhamento de capacidade, a qual possui a segurança como parte integrante de seus protocolos.

5.2.3 Arquitetura para Redes de Compartilhamento de Capacidade

A arquitetura proposta nesta seção, também descrita em (SANTOS et al., 2013; NUNES et al., 2014b; SANTOS et al., 2014), utiliza o paradigma das Redes Definidas por Software (RDS) (NUNES et al., 2014a) com base no padrão OpenFlow (MCKEOWN et

al., 2008). Nesse paradigma, o plano de controle é dissociado do plano de dados, facilitando a implantação de novos serviços e protocolos. O plano de dados é representado pelos dispositivos de encaminhamento de pacotes, enquanto o plano de controle passa a ser produto de *software* (e.g., cálculo de rotas, geração e distribuição de chaves criptográficas) executado em um computador central chamado de controlador.

Como ilustrado no exemplo da Figura 25, os componentes da arquitetura proposta incluem *gateways* e DUs, os quais são descritos a seguir.

- **Gateway (GW):** dispositivos que oferecem serviços através dos quais usuários podem se conectar a uma infraestrutura de rede (e.g., pontos de acesso). Devem possuir um módulo de *software* que implementa a arquitetura proposta nesta seção. Além disso, um GW utiliza um *switch* habilitado para RDS (e.g., Open vSwitch (PFAFF et al., 2009)) com o objetivo principal de fazer o repasse de pacotes entre diferentes interfaces de rede. Um módulo de *software* atuando como plano de controle da rede pode estar instalado no próprio GW, o que habilita este dispositivo a atuar como um controlador local, autorizado pelo controlador principal da rede (SANTOS et al., 2014);
- **Dispositivo de Usuário (DU):** dispositivos que utilizam um GW como provedor de conectividade e de serviços. Tipicamente, um DU se conecta a um GW via rede sem fio. Um DU pode não ser habilitado para RDS, mas deve possuir um módulo de *software* para comunicação segura com GWs e módulos relacionados com demais serviços utilizados (e.g., QoS). Observa-se que é possível ainda que um DU não utilize serviços de extensão de conectividade, permitindo uma implantação incremental da arquitetura proposta.

O benefício de se adotar o paradigma das RDS não é apenas a flexibilidade gerada pela separação entre os planos de dados e de controle. Uma outra vantagem é a possibilidade de se identificar usuários que se conectam via *gateways* e fornecer serviços de forma personalizada. Como não se usa um Network Address Translator (NAT), um provedor de serviços poderia por exemplo vender um plano de acesso temporário para um DU e incentivar um DU a se tornar um *gateway* e compartilhar seus recursos.

As figuras 26a e 26b ilustram as arquiteturas de GWs e DUs, nos quais implementa-se um conjunto de serviços fornecidos a partir da arquitetura proposta, chamados de *arcabouço*. Por exemplo, um DU conectado a um GW poderia passar por um processo de *handover* de forma que ao final desse processo estaria conectado a outro GW (SANTOS et al., 2014). O *handover* poderia ser gerenciado pelo controlador da rede (NUNES et al., 2014b). Outro exemplo é a implantação de mecanismos de tolerância a falhas (SANTOS et al., 2014) e serviços de QoS (NUNES et al., 2014b).

Figura 26: Arquiteturas de GWs (a) e DUs (b).

Fonte: (NUNES et al., 2014b).

Requisitos

Os requisitos da arquitetura proposta são:

- **Compartilhamento de recursos:** como mencionado anteriormente, um DU poderá optar por oferecer serviços de extensão de conectividade, ou seja, atuar como um GW;
- **Controle de acesso:** DUs que requisitam a extensão de conectividade devem passar por um processo de autenticação mútua com um GW. Após o processo de autenticação, o GW requisita o acesso ao usuário para um controlador da rede, que poderá consultar seu banco de dados e autorizar ou negar o acesso ao serviço de acordo com políticas que estão fora do escopo deste trabalho;
- **Confidencialidade e autenticidade de dados:** a interação entre GW e DU permite que chaves criptográficas sejam estabelecidas para comunicação segura.

Descrição Geral da Arquitetura

Os protocolos da arquitetura proposta, ilustrados na Figura 27, são:

- **Gateway discovery:** com o uso do arcabouço ilustrado na Figura 26a, nós GW enviam mensagens periódicas para anunciar seus serviços; um DU, ao recebê-las, escolhe um GW disponível através do envio da mensagem Request, habilitada pelo arcabouço ilustrado na Figura 26b;
- **Handshaking:** um nó GW responde a uma requisição de usuário e inicia o procedimento de *handshaking* para autenticação mútua entre DU e GW;

- **DU *check-in*:** um nó GW requisita autorização a um controlador da rede, o qual aprova ou rejeita o acesso ao usuário informado pelo GW.

Quando um DU é autorizado, o controlador da rede cria entradas apropriadas na tabela de fluxos do GW envolvido e também nas tabelas de fluxos dos dispositivos da rede habilitados para RDS, permitindo que o DU tenha acesso à rede através do GW.

Figura 27: Etapas para o compartilhamento de capacidade: requisição do dispositivo de usuário, *handshaking* e *check-in*.

Fonte: (SANTOS et al., 2013).

Implementação do *Handshaking*

Um exemplo de implementação do procedimento de *handshaking*, descrito no trabalho de Santos et al. (2013), utiliza a Criptografia Baseada em Identidade (CBI) (BONEH; FRANKLIN, 2001). A CBI permite que um usuário calcule uma chave pública a partir de uma cadeia de caracteres. O uso de CBI facilita o projeto de protocolos simples e eficientes, como o que será mostrado nesta seção. O uso de uma identidade de usuário como chave pública também permite que não se tenha a verificação de chave por uma entidade confiável em tempo real.

Como uma chave pública possui valor arbitrário, já que depende da identidade de um usuário, a chave privada deve ser gerada por uma entidade confiável ou *Trusted Third Party* (TTP). A geração da chave privada é feita através de um algoritmo que recebe a

chave privada da entidade confiável, chamada de *chave privada mestre*, e a chave pública do usuário.

Apesar de todas as chaves privadas serem de conhecimento da TTP, pode-se explorar o fato de que, no paradigma das redes definidas por *software*, o controlador é responsável por aplicações que gerenciam a rede. Isso mostra a sinergia entre controlador e entidade confiável. Logo, um controlador pode ser utilizado como TTP para gerar e distribuir chaves criptográficas.

Protocolos que utilizam o paradigma da CBI podem ser implementados com o uso de emparelhamentos. Como mencionado na Seção 2.1, um emparelhamento associa pares de pontos em curvas elípticas a um elemento de um corpo finito. Com o uso de CBI e emparelhamentos é possível implementar protocolos de acordo de chaves autenticados (BONEH; FRANKLIN, 2001). Isso evita o uso de esquemas complexos como cifração de chave pública, já que compartilha-se uma chave para uso de criptografia simétrica ao se utilizar acordos de chave.

O *handshaking* entre DU e GW apresentado nesta seção é composto de duas etapas: inicialização e autenticação mútua. Esta última inclui o processo de acordo de chaves. Antes de descrever as etapas, apresenta-se a notação utilizada através da Tabela 6.

Tabela 6: Notação.

ID_X	identidade do nó X
s	chave privada mestre (chave privada do controlador)
S_X	chave privada do nó X
P_X	chave pública do nó X (derivada de ID_X)
$K_{X,Y}$	chave acordada entre nós X e Y
ctr	contador
$authenc(msg, k)$	cifração autenticada de msg usando a chave k
$dec(msg, k)$	decifração de msg usando a chave k
mac	valor gerado pelo algoritmo de autenticação
$e(\cdot, \cdot)$	emparelhamento

Inicialização: como chaves públicas são derivadas de identidades, a TTP (i.e., o controlador) mapeia a identidade de um nó, ID_X , para um ponto na curva elíptica, P_X . O mapeamento é uma função pública, já que um nó pode gerar qualquer chave pública. A TTP gera a chave secreta s e calcula a chave privada de um nó da rede como $S_X = sP_X$. Esse valor deve ser transmitido com o uso de um canal seguro ou inserido em nós antes da implantação da rede.

Autenticação mútua: o DU é requisitado a responder um “desafio” (ou *challenge*) para que o GW possa verificar sua identidade. GW e DU devem conhecer as identidades envolvidas para calcularem uma chave compartilhada entre eles, a qual é usada para cifração autenticada do desafio. Assim, o GW pode cifrar o valor do desafio para o DU, o

qual realiza decifração e cifração para transmitir o valor obtido de volta para o GW. Por fim, o GW pode verificar se o valor cifrado recebido é realmente um resultado do valor do desafio transmitido no início desse processo.

A Figura 28 mostra em detalhes a autenticação mútua entre GW e DU.

Figura 28: Exemplo de implementação do procedimento de *handshaking* entre DU e GW.

Fonte: (SANTOS et al., 2013).

Um protótipo foi elaborado a partir dos conceitos apresentados neste capítulo, o que resultou na publicação de um trabalho em uma conferência da área de redes de computadores (SANTOS et al., 2014). Foi demonstrado o procedimento de *handover* de um DU entre dois *gateways*, como ilustrado na Figura 29.

Para que o *handover* aconteça do GW_1 para o GW_2 (Figura 29), o DU conectado a GW_1 é habilitado a requisitar serviços do GW_2 , o que resulta em um novo procedimento de *handshaking*. Isso permite que o controlador remova os fluxos instalados em GW_1 e adicione os fluxos correspondentes em GW_2 , o que equivale ao procedimento de *check-in* da arquitetura proposta. Considera-se que o *handover* é seguro devido aos serviços de confidencialidade e autenticidade implementados.

Em relação à interface entre dispositivos habilitados para RDS e controlador da rede, utilizamos comandos da ferramenta Open vSwitch (PFAFF et al., 2009) ao invés de adotar um controlador disponível em código aberto. Isso facilitou a implementação de um protótipo, já que a *testbed* criada possui no máximo dois *gateways*, além de um usuário adicional e de um controlador.

Figura 29: Exemplo de *handover* de um DU entre dois GWs.

5.3 Considerações do Capítulo

O protocolo de sobrevivência de dados elaborado neste projeto de pesquisa foi utilizado para fornecer serviços de segurança em RSPs. Em especial, foi proposto o compartilhamento de dados entre autoridades de segurança pública com o uso da arquitetura CHORIST.

A etapa de renovação de quotas do protocolo foi demonstrada e avaliada em um cenário de RSPs, o qual foi inspirado em trabalhos anteriores. Utilizou-se uma biblioteca criptográfica em conjunto com um simulador de redes para implementação da etapa de renovação de quotas. Resultados mostram que pode haver atraso significativo para renovar quotas quando se usa uma determinada quantidade de nós. Isso acontece devido à troca de quotas parciais entre os nós participantes do protocolo, o que enfatiza o uso do processo de renovação em períodos de tempo não muito frequentes.

Mostrou-se uma forma de implantar RSPs ainda no modelo da arquitetura CHORIST, em que DUs podem se conectar a *gateways* instalados em carros de bombeiros. Como a conectividade da rede é importante em RSPs, DUs também foram utilizados como *gateways* para ampliar a comunicação entre dispositivos móveis. As redes de compartilhamento de capacidade foram adotadas para a elaboração de uma arquitetura que fornece extensão de conectividade e serviços de segurança como confidencialidade e autenticidade. O paradigma das redes definidas por *software* foi utilizado na implantação da rede, a qual considerou o controlador como uma entidade responsável por gerar e distribuir material criptográfico. Foi apresentado ainda um exemplo de como implementar o procedimento de *handshaking* da arquitetura proposta, o que é parte de um protótipo desenvolvido.

6 Considerações Finais

A implantação segura de RSSFs com o uso de mecanismos criptográficos tem sido alvo de muitos projetos de pesquisa nos últimos anos. A elaboração de mecanismos eficientes de segurança tornou-se um desafio na área de RSSFs, já que os dispositivos sensores possuem recursos de processamento e memória limitados. O uso de nós sorvedouros itinerantes, como é o caso das RSSFs desassistidas, torna a rede ainda mais exposta a ataques, já que dados armazenados em dispositivos poderiam ser obtidos e apagados por adversários.

A presente pesquisa de doutorado abordou o problema da sobrevivência de dados em RSSFs. Como descrito no Capítulo 2, a partir de um estudo e da implementação dos principais mecanismos utilizados para sobrevivência de dados, mostrou-se as principais limitações de propostas anteriores. Em especial, os métodos de criptografia utilizados tornam inviável a agregação de dados segura. Além disso, uma operação fundamental do principal mecanismo, que é a recifração, possui alto custo de processamento, sendo que sua otimização requer alto consumo de memória.

Com motivação nas deficiências das propostas anteriores, foi elaborado um protocolo de segurança que fornece compartilhamento de dados e redundância. Como descrito no Capítulo 3, o protocolo proposto permite que nós realizem um procedimento de atualização para evitar ataques de um adversário que utiliza dados de nós sensores comprometidos. No caso da agregação de dados segura, com o uso de criptografia homomórfica, o método utilizado permite operações básicas como soma e multiplicação. Outras operações ainda seriam possíveis com o uso da reconstrução do segredo.

Como descrito no Capítulo 4, foram escolhidos três métodos para avaliação do protocolo proposto, sendo eles a implementação e experimentação em dispositivos reais, análise de segurança através de um modelo analítico e a modelagem com o uso das redes de Petri. Mostrou-se que o tempo de apenas uma operação fundamental necessária para a recifração, principal mecanismo de trabalhos relacionados, é suficiente para a inicialização de 80 quotas com o uso do protocolo proposto. Através de análise de segurança e de modelagem, mostrou-se quais estratégias seriam mais eficazes com o uso do protocolo proposto, incluindo cenários de aplicação eficientes como o uso de quotas de tamanho reduzido.

Identificou-se que o protocolo proposto pode ser utilizado no contexto das redes heterogêneas, em especial as redes de segurança pública, como descrito no Capítulo 5.

Foi explorado como fornecer o serviço de compartilhamento de dados entre agências de segurança pública. A implementação do protocolo em um simulador de redes mostrou que, para o cenário adotado, o processo de renovação de quotas possui limitações se um grande número de nós for utilizado. Apesar disso, pode-se utilizar compartilhamento de dados entre vizinhos, reduzindo o número de nós do protocolo.

O Capítulo 5 também apresentou um método para estender a conectividade de dispositivos de usuário, os quais tipicamente são utilizados por agentes de segurança em campo (e.g., bombeiros). As redes de compartilhamento de capacidade foram utilizadas com mecanismos de segurança tradicionais para elaboração de um protótipo para redes de segurança pública. É importante observar que o Capítulo 5 possui resultados de pesquisa que foram apoiados pelo CNPq através de uma bolsa de doutorado sanduíche no exterior (processo número 245588/2012-4).

Conclui-se que os objetivos do presente projeto de pesquisa foram atingidos, pois o protocolo elaborado para sobrevivência de dados em RSSFs apresentou segurança e aplicabilidade no cenário adotado. Também observou-se que o protocolo é eficiente em consumo de tempo e energia em relação a trabalhos já existentes na literatura.

6.1 Trabalhos Futuros

Resultados de simulação foram obtidos com a modelagem do protocolo proposto através do uso das redes de Petri e da ferramenta CPN Tools. Apesar da flexibilidade em se variar parâmetros como número de nós e quantidade de nós comprometidos, foram utilizados valores baseados em trabalhos anteriores. Como trabalho futuro vislumbra-se a análise para encontrar parâmetros com resultados em diferentes escalas. Além disso, vislumbra-se incluir outros modelos de adversário possíveis (e.g., SMART (PIETRO et al., 2008)) e a fase de renovação de quotas, apesar desta já ter sido prototipada em um simulador de redes. Ainda é possível estimar o custo da renovação através do número de interações entre os nós da rede, tendo em vista o custo de cada interação (e.g., tempo ou consumo de energia para transmissão ou processamento de dados).

O conjunto de premissas utilizadas não inclui a mobilidade de dispositivos sensores. As premissas incluem apenas a mobilidade de nós sorvedouros. A mobilidade de nós sensores pode ser explorada com os mecanismos utilizados neste trabalho para fornecer disponibilidade dos dados, o que foi feito no trabalho de Pietro e Guarino (2013). Como exemplo de cenário considera-se um nó sensor móvel que possui dados armazenados e foi movido para uma região geográfica sem conectividade com outros nós. Neste caso, com o uso do protocolo proposto, ainda seria possível restaurar o dado armazenado pelo nó sensor sem conectividade, já que outros nós poderiam agregar suas quotas e executar o

processo de reconstrução descrito na Seção 3.4.

A aplicabilidade desta pesquisa de doutorado em redes heterogêneas motiva a elaboração de novos métodos de inicialização de segurança. No caso das redes de compartilhamento de capacidade, utilizadas na implantação de redes de segurança pública, considerou-se o paradigma das redes definidas por *software*. A presença de uma entidade confiável, chamada de controlador da rede, permite que se distribua material criptográfico de forma segura (SANTOS et al., 2013), de forma similar a um nó sorvedouro de uma RSSFs. Um controlador pode ainda fornecer serviços adicionais de rede (NUNES et al., 2014b), já que é capaz de gerenciar dispositivos de encaminhamento (e.g., escrever em tabelas de fluxo).

6.2 Publicações

As publicações relacionadas com esta pesquisa de doutorado são:

- **Santos, M. A. S.**; Margi, Cíntia B.;Simplicio, M.A.;Geovandro, C.C.F.P.;de Oliveira, B.T.. *Implementation of data survival in unattended Wireless Sensor Networks using cryptography*. In: 5th IEEE International Workshop on Practical Issues in Building Sensor Network Applications. Co-located with 35th IEEE Local Computer Networks, 2010, Denver, CO, USA. In Proceedings: 35th IEEE Local Computer Networks (**IEEE LCN**), 2010. p. 961-967;
- **Santos, M. A. S.**; Margi, Cíntia B.. *Design and Implementation of Data Survival in Unattended Wireless Sensor Networks*. In: The 2011 International Workshop on Hot Topics on Wireless Network Security and Privacy. Co-located with 30th IEEE International Performance Computing and Communications Conference (**IEEE IPCCC**), 2011, Orlando, FL, USA. In Proceedings: 30th IEEE International Performance Computing and Communications Conference (IPCCC 2011), 2011. p. 1-6;
- **Santos, M. A. S.**; Margi, Cíntia B.. *TinySharing: Uma Ferramenta para Compartilhamento de Segredos em Redes de Sensores Sem Fio*. In: XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (**SBRC**) - Salão de Ferramentas, 2011, Campo Grande, MS. Anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2011. v. 1. p. 949-956;
- **Santos, M. A. S.**; Margi, Cíntia B.. *A Secure Multi-Party Protocol for Sharing Valuable Information in Public Safety Networks*. In: Mobile Ad Hoc Networks for Public Safety Systems; Techniques and Challenges in On-Going Work - co-located with IEEE MASS 2011., 2011, Valencia. In Proceedings: 8th IEEE International

Conference on Mobile Ad-hoc and Sensor Systems (**IEEE MASS**). Washington, DC, USA : IEEE Computer Society, 2011. p. 935-940;

- **Santos, M. A. S.**; de Oliveira, B.T.; Margi, C.B.; Nunes, B.AA; Turletti, T.; Obraczka, K., *Software-defined networking based capacity sharing in hybrid networks*, In: Capacity Sharing Workshop - co-located with IEEE International Conference on Network Protocols. In Proceedings: 21st IEEE International Conference on Network Protocols(**IEEE ICNP**), 2013, pp.1,6, 7-10 Oct. 2013;
- Nunes, B., **Santos, M. A. S.**, Margi, C.B., Oliveira, B.T., Obraczka, K., Turletti, T., *Software-Defined Networking Enabled Capacity Sharing in User Centric Networks*, IEEE Communications Magazine, v. 52, p. 28-36, 2014.;
- **Santos, M. A. S.**, Nunes, B., Obraczka, K., Turletti, T., Margi C.B., Oliveira, B.T. *Decentralizing SDN's Control Plane*, IEEE Local Computer Networks (**IEEE LCN**), 2014. (*short paper* aceito para publicação).

Outras publicações que não são resultados diretos deste projeto de pesquisa, mas que foram obtidas durante o período de doutoramento são:

- Geovandro C.C.F. Pereira, **Mateus A.S. Santos**, Bruno T. de Oliveira, Marcos A. Simplicio Jr., Paulo S.L.M. Barreto, Cíntia B. Margi, Wilson V. Ruggiero, *SMS-Crypto: A lightweight cryptographic framework for secure SMS transmission*, Journal of Systems and Software, Volume 86, Issue 3, March 2013, Pages 698-706, ISSN 0164-1212;
- Leal, R. R., Simplicio Jr, M. A., **Santos, Mateus A. S.**, GOMES, M. A. L., GOYA, W. A. *Cheating detection in P2P online trading card games* In: XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2013, Manaus. Anais do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. Sociedade Brasileira de Computação (SBC), 2013. p.170 - 183;
- Marcos A. Simplicio Jr., **Mateus A.S. Santos**, Rodrigo R. Leal, Marco A.L. Gomes and Walter A. Goya. *SecureTCG: a lightweight cheating-detection protocol for P2P multiplayer online trading card games*. Security and Communication Networks, Wiley, 2014. DOI 10.1002/sec.952 (publicado *online* em 17/01/2014).
- **Mateus A. S. Santos**, Deivid E. T. Porras, Regina M. Silveira, Cíntia B. Margi. *Multipath source routing strategies for video transmission in ad hoc wireless networks*. Wireless Networks, Springer, 2014. DOI 10.1007/s11276-014-0823-x (publicado *online* em 30/09/2014).

Por fim, há ainda um artigo em elaboração que está diretamente relacionado com a pesquisa de doutorado.

- **Santos, M. A. S.**; Margi, Cíntia B. *Design and Analysis of a Proactive Secret Sharing Scheme for Data Survival in UWSNs* (artigo em elaboração para submissão em periódico internacional);

Referências

- AGILENT. ***Agilent 34401A Multimeter***. 2007. Acesso em: 21 ago. 2014. Disponível em: <<http://cp.literature.agilent.com/litweb/pdf/5968-0162EN.pdf>>.
- AGILENT. ***E363xA Series Programmable DC Power Supplies***. 2009. Acesso em: 21 ago. 2014. Disponível em: <<http://cp.literature.agilent.com/litweb/pdf/5968-9726EN.pdf>>.
- ATENIESE, G. et al. **Improved proxy re-encryption schemes with applications to secure distributed storage**. *ACM Transactions on Information and System Security*, v. 9, p. 1–30, February 2006. ISSN 1094-9224.
- BEN-OR, M.; GOLDWASSER, S.; WIGDERSON, A. **Completeness theorems for non-cryptographic fault-tolerant distributed computation**. In: PROCEEDINGS OF THE TWENTIETH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING. New York: ACM, 1988. (STOC '88), p. 1–10. ISBN 0-89791-264-0.
- BLAKLEY, G. **Safeguarding cryptographic keys**. In: AFIPS 1979 NATIONAL COMPUTER CONFERENCE, 1979. Japan: CiNii, 1979. v. 48, p. 313–317.
- BONEH, D.; FRANKLIN, M. **Identity-based encryption from the weil pairing**. In: KILIAN, J. (Ed.). *CRYPTO 2001*. Santa Barbara: Heidelberg: Springer, 2001, (LNCS, v. 2139). p. 213–229. ISBN 978-3-540-42456-7.
- BRUNO, R.; CONTI, M.; GREGORI, E. **Mesh networks: commodity multihop ad hoc networks**. *Communications Magazine, IEEE*, v. 43, n. 3, p. 123 – 131, March 2005. ISSN 0163-6804.
- CORON, J.-S. et al. **Fully homomorphic encryption over the integers with shorter public keys**. In: ROGAWAY, P. (Ed.). *Advances in Cryptology CRYPTO 2011*. Santa Barbara: Heidelberg: Springer, 2011, (Lecture Notes in Computer Science, v. 6841). p. 487–504. ISBN 978-3-642-22791-2.
- CULLER, D.; ESTRIN, D.; SRIVASTAVA, M. **Overview of sensor networks**. *Computer Magazine, IEEE Computer Society*, v. 37, n. 8, p. 41–49, 2004.
- DESEL, J.; REISIG, W. **Place/transition petri nets**. In: REISIG, W.; ROZENBERG, G. (Ed.). Heidelberg: Springer, 1998, (Lecture Notes in Computer Science, v. 1491). p. 122–173. ISBN 978-3-540-65306-6.
- EADS Defence and Security Systems. ***CHORIST Final report SP0.R7***. 2009. Acesso em: 21 ago. 2014. Disponível em: <<http://www.chorist.eu/>>.
- FASOLO, E. et al. **In-network aggregation techniques for wireless sensor networks: a survey**. *Wireless Communications, IEEE*, v. 14, n. 2, p. 70–87, April 2007. ISSN 1536-1284.

- FELDMAN, P. **A practical scheme for non-interactive verifiable secret sharing**. In: SFCS '87: PROCEEDINGS OF THE 28TH ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE. Washington: IEEE Computer Society, 1987. p. 427–438. ISBN 0-8186-0807-2.
- GAY, D. et al. **The nesC language: A holistic approach to networked embedded systems**. In: IN PROCEEDINGS OF PROGRAMMING LANGUAGE DESIGN AND IMPLEMENTATION. San Diego: ACM, 2003. p. 1–11.
- GENTRY, C.; HALEVI, S. **Implementing gentry fully-homomorphic encryption scheme**. In: PATERSON, K. (Ed.). *EUROCRYPT 2011*. Berlin: Heidelberg: Springer, 2011, (Lecture Notes in Computer Science, v. 6632). p. 129–148. ISBN 978-3-642-20464-7.
- HERZBERG, A. et al. **Proactive secret sharing or: How to cope with perpetual leakage**. In: CRYPTO '95: PROCEEDINGS OF THE 15TH ANNUAL INTERNATIONAL CONFERENCE ON ADVANCES IN CRYPTOLOGY. Santa Barbara: London: Springer-Verlag, 1995. p. 339–352. ISBN 3-540-60221-6.
- HONG, Y.-W. P. et al. **Wireless sensor networks: Signal processing and communications perspectives (edited book)**. John-Wiley & Sons, 2007.
- IEEE. **IEEE 802.16 standard for Local and Metropolitan Area Networks -Part 16: Air Interface for Fixed Broadband Wireless Access Systems**. New York, 2004.
- IEEE. **Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**. New York, 2007.
- IFTENE, S. **Secret Sharing Schemes with Applications in Security Protocols**. Tese (Doutorado) — Universitatea Alexandru Ioan Cuza, Iasi, 2007.
- INTANAGONWIWAT, C.; GOVINDAN, R.; ESTRIN, D. **Directed diffusion: A scalable and robust communication paradigm for sensor networks**. In: PROCEEDINGS OF THE 6TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING. New York: ACM, 2000. (MobiCom '00), p. 56–67. ISBN 1-58113-197-6.
- ISO/IEC 15909, I. S. **High-level Petri nets - concepts, definitions and graphical notation**. New York: ISO, 2000.
- JENSEN, K. **Coloured petri nets**. In: BRAUER, W.; REISIG, W.; ROZENBERG, G. (Ed.). *Petri Nets: central models and their properties*. Heidelberg: Springer, 1987, (Lecture Notes in Computer Science, v. 254). p. 248–299. ISBN 978-3-540-17905-4.
- JENSEN, K. **An introduction to the theoretical aspects of coloured petri nets**. In: BAKKER, J. de; ROEVER, W. de; ROZENBERG, G. (Ed.). *A decade of concurrency reflections and perspectives*. Heidelberg: Springer, 1994, (Lecture Notes in Computer Science, v. 803). p. 230–272. ISBN 978-3-540-58043-0.
- JENSEN, K.; KRISTENSEN, L. M. **Coloured Petri nets modelling and validation of concurrent systems**. Berlin: Verlag: Springer, 2009.

- KARAOGLAN, D.; LEVI, A.; SAVAS, E. **A distributed key establishment scheme for wireless mesh networks using identity-based cryptography.** In: PROC. OF THE 6TH ACM WORKSHOP ON QOS AND SECURITY FOR WIRELESS AND MOBILE NETWORKS. Bodrum, Turkey: ACM, 2010. p. 11–18. ISBN 978-1-4503-0275-3.
- KRAWCZYK, H. **Secret sharing made short.** In: STINSON, D. (Ed.). *Advances in Cryptology 94*. Santa Barbara: Heidelberg: Springer, 1994, (Lecture Notes in Computer Science, v. 773). p. 136–146. ISBN 978-3-540-57766-9.
- KRISHNAMACHARI, B.; ESTRIN, D.; WICKER, S. **The impact of data aggregation in wireless sensor networks.** In: DISTRIBUTED COMPUTING SYSTEMS WORKSHOPS, 2002. PROCEEDINGS. 22ND INTERNATIONAL CONFERENCE ON. Washington: IEEE, 2002. p. 575–578.
- MA, D.; SORIENTE, C.; TSUDIK, G. **New adversary and new threats: security in unattended sensor networks.** *Network, IEEE*, v. 23, n. 2, p. 43–48, March 2009. ISSN 0890-8044.
- MA, D.; TSUDIK, G. **Dish: Distributed self-healing.** In: KULKARNI, S.; SCHIPER, A. (Ed.). *Stabilization, safety, and security of distributed Systems*. Berlin: Heidelberg: Springer, 2008, (Lecture Notes in Computer Science, v. 5340). p. 47–62. ISBN 978-3-540-89334-9.
- Margi, C. B. et al. **Impact of operating systems on wireless sensor networks (security) applications and testbeds.** In: 4TH WORKSHOP ON WIRELESS MESH AND AD HOC NETWORKS (WIMAN) IN CONJUNCTION WITH IEEE ICCN. Zurich: IEEE, 2010. p. 1–6.
- MCKEOWN, N. et al. **Openflow: enabling innovation in campus networks.** *SIGCOMM Comput. Commun. Rev.*, ACM, New York, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833.
- MURATA, T. **Petri nets: properties, analysis and applications.** *Proceedings of the IEEE*, v. 77, n. 4, p. 541–580, apr 1989. ISSN 0018-9219.
- NIKOV, V.; NIKOVA, S. **On proactive secret sharing schemes.** In: HANDSCHUH, H.; HASAN, M. (Ed.). *Selected areas in cryptography*. Kingston: Heidelberg: Springer, 2005, (Lecture Notes in Computer Science, v. 3357). p. 308–325. ISBN 978-3-540-24327-4.
- NUNES, B. A. A. et al. **A survey of software-defined networking: Past, present, and future of programmable networks.** *Communications Surveys Tutorials, IEEE*, v. 16, n. 3, p. 1617–1634, Third 2014. ISSN 1553-877X.
- NUNES, B. A. A. et al. **Software-Defined Networking Enabled Capacity Sharing in User Centric Networks.** 2014. Aceito em IEEE Communications Magazine.
- OLIVEIRA, B. T. de. **Arquitetura de Segurança fim-a-fim para Redes de Sensores sem Fio.** Dissertação (Mestrado) — Universidade de São Paulo, São Paulo, 2012.
- OLIVEIRA, L. B. et al. **TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks.** *Computer Communications*, 2010. To appear.

- OZDEMIR, S.; XIAO, Y. **Secure data aggregation in wireless sensor networks: A comprehensive overview**. *Computer Networks*, v. 53, n. 12, p. 2022 – 2037, 2009. ISSN 1389-1286.
- PFAFF, B. et al. **Extending networking into the virtualization layer**. In: 8TH ACM WORKSHOP ON HOT TOPICS IN NETWORKS (HotNets). New York: ACM, 2009.
- PIETRO, R. D.; GUARINO, S. **Confidentiality and availability issues in mobile unattended wireless sensor networks**. In: WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS (WOWMOM), 2013 IEEE 14TH INTERNATIONAL SYMPOSIUM AND WORKSHOPS ON A. Madrid: IEEE, 2013. p. 1–6.
- PIETRO, R. D. et al. **Data security in unattended wireless sensor networks**. *Computers, IEEE Transactions on*, v. 58, n. 11, p. 1500 –1511, nov. 2009. ISSN 0018-9340.
- PIETRO, R. D. et al. **Catch me (if you can): Data survival in unattended sensor networks**. In: PERCOM '08: PROCEEDINGS OF THE 2008 SIXTH ANNUAL IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS. Washington: IEEE Computer Society, 2008. p. 185–194. ISBN 978-0-7695-3113-7.
- PIETRO, R. D. et al. **Playing hide-and-seek with a focused mobile adversary in unattended wireless sensor networks**. *Ad Hoc Networks*, v. 7, n. 8, p. 1463 – 1475, 2009. ISSN 1570-8705. Privacy and Security in Wireless Sensor and Ad Hoc Networks.
- PIETRO, R. D. et al. **Collaborative authentication in unattended WSNs**. In: WISEC '09: PROCEEDINGS OF THE SECOND ACM CONFERENCE ON WIRELESS NETWORK SECURITY. Zurich: ACM, 2009. p. 237–244. ISBN 978-1-60558-460-7.
- POLASTRE, J.; SZEWCZYK, R.; CULLER, D. **Telos: enabling ultra-low power wireless research**. In: INFORMATION PROCESSING IN SENSOR NETWORKS. FOURTH INTERNATIONAL SYMPOSIUM ON. Los Angeles: ACM, 2005. p. 364 – 369.
- RABIN, M. O. **Efficient dispersal of information for security, load balancing, and fault tolerance**. *Journal of the ACM*, ACM, New York, v. 36, n. 2, p. 335–348, April 1989. ISSN 0004-5411.
- RE, G. L.; MILAZZO, F.; ORTOLANI, M. **Secure random number generation in wireless sensor networks**. *Concurrency and Computation: Practice and Experience*, p. n/a–n/a, 2014. ISSN 1532-0634.
- REN, W.; ZHAO, J.; REN, Y. **Mss: A multi-level data placement scheme for data survival in wireless sensor networks**. In: WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING, 2009. WICOM '09. 5TH INTERNATIONAL CONFERENCE ON. Beijing: IEEE, 2009. p. 1 –4.
- REN, W.; ZHAO, J.; REN, Y. **Network coding based dependable and efficient data survival in unattended wireless sensor networks**. *Journal of Communications*, v. 4, n. 11, Dec. 2009.

- REN, Y.; OLESHCHUK, V.; LI, F. **A scheme for secure and reliable distributed data storage in unattended WSNs**. In: GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM 2010), 2010 IEEE. New York: IEEE, 2010. p. 1 –6. ISSN 1930-529X.
- REN, Y.; OLESHCHUK, V. A.; LI, F. Y. **Optimized secure and reliable distributed data storage scheme and performance evaluation in unattended WSNs**. *Computer Communications*, v. 36, n. 9, p. 1067 – 1077, May, 2013. ISSN 0140-3664. Reactive wireless sensor networks.
- SAKAI, R.; OHGISHI, K.; KASAHARA, M. **Cryptosystems based on pairing**. In: SYMPOSIUM ON CRYPTOGRAPHY AND INFORMATION SECURITY (SCIS00). Okinawa: SCIS, 2000. p. 135–148.
- SANTOS, M.; MARGI, C. **Design and implementation of data survival in unattended wireless sensor networks**. In: PERFORMANCE COMPUTING AND COMMUNICATIONS CONFERENCE (IPCCC WORKSHOPS), 2011 IEEE 30TH INTERNATIONAL. Orlando: IEEE, 2011. p. 1 –6. ISSN 1097-2641.
- SANTOS, M.; MARGI, C. **A secure multi-party protocol for sharing valuable information in public safety networks**. In: MOBILE ADHOC AND SENSOR SYSTEMS (MASS WORKSHOPS), 2011 IEEE 8TH INTERNATIONAL CONFERENCE ON. Valencia: IEEE, 2011. p. 935 –940. ISSN 2155-6806.
- SANTOS, M. A. S. et al. **Implementation of data survival in unattended wireless sensor networks using cryptography**. In: IEEE LOCAL COMPUTER NETWORKS SENSEAPP 2010. Denver: IEEE Computer Society, 2010.
- SANTOS, M. A. S. et al. **Decentralizing SDNs Control Plane**. 2014. Aceito como *short paper* em IEEE Local Computer Networks (LCN).
- SANTOS, M. A. S. et al. **Software-defined networking based capacity sharing in hybrid networks**. In: IEEE INTERNATIONAL CONFERENCE ON NETWORK PROTOCOLS WORKSHOPS. Gottingen: IEEE, 2013.
- SAXENA, N.; TSUDIK, G.; YI, J. H. **Efficient node admission and certificateless secure communication in short-lived manets**. *Parallel and Distributed Systems, IEEE Transactions on*, v. 20, n. 2, p. 158 –170, Feb. 2009. ISSN 1045-9219.
- SHAMIR, A. **How to share a secret**. *Communications*, ACM, New York, v. 22, n. 11, p. 612–613, nov. 1979. ISSN 0001-0782.
- STALLINGS, W. **Segurança de Computadores**. São Paulo: Elsevier, 2008.
- VITALI, D. et al. **Mhrep: Multi-hop replication scheme for data survival in unattended wireless sensor networks**. In: RELIABLE DISTRIBUTED SYSTEMS WORKSHOPS (SRDSW), 2011 30TH IEEE SYMPOSIUM ON. Madrid: IEEE Computer Society, 2011. p. 42 –51.
- YANG, C.-C.; CHANG, T.-Y.; HWANG, M.-S. **A (t,n) multi-secret sharing scheme**. *Applied Mathematics and Computation*, v. 151, n. 2, p. 483–490, 2004. ISSN 0096-3003.
- ZHAO, F.; GUIBAS, L. **Wireless sensor networks: an information processing approach**. Boston, MA: Elsevier-Morgan Kaufmann, 2004.

Apêndices

APÊNDICE A – Declarações dos Modelos em Redes de Petri

Nesta seção apresenta-se as declarações em CPN ML utilizadas para contruir os modelos das redes de Petri, descritos na Seção 4.3.2.

As variáveis e coleções utilizadas foram:

```

1    val P_ULTRA = 50;
    val P_HIGH = 100;
3    val P_HIGH_NEXT = 500;
    val P_NORMAL = 1000;
5    val P_LOW = 10000;

7    colset UNIT = unit;
    colset INT = int;
9    colset BOOL = bool;
    colset STRING = string;

11   val tHold; #limiar
13   val compNodes; #dispositivos comprometidos
    val q; #dispositivos nao participantes do protocolo
15   val p; #dispositivos participantes do protocolo
    val t = q + p; #total de participantes
17   var i,j,k,c,x,y,z : INT; #itens auxiliares
    var round: INT; #rodada atual

19   colset NODE = index n with 1..t; #modelagem de dispositivos
21   colset SHARE = index s with 1..p; #modelagem de quotas

23   colset NODExSHARE = product NODE * SHARE; #par (dispositivo ,quota)

25   colset LSHARE = list SHARE; #lista de quotas
    colset LNODE = list NODE; #lista de dispositivos
27   colset LPACK = list NODExSHARE; #lista de pares (dispositivo ,quota)

```

As funções utilizadas foram:

- `CreatePNodes`: cria lista inicial de nós participantes do protocolo, formando o par (nó, quota);
- `CreateNonPNodes`: cria lista inicial de nós não participantes do protocolo;
- `CreateCompNodes`: cria lista inicial de nós comprometidos;
- `ConcatLists`: recebe duas listas de entrada e devolve a concatenação dessas listas;
- `AllNodesInList`: cria uma lista com todos os nós existentes;
- `ChooseOne`: escolhe um nó de uma lista de forma pseudo-aleatória;
- `ChooseNodes`: escolhe um nó de uma lista de forma pseudo-aleatória com o requisito de que nós em uma lista de entrada não serão escolhidos.

As funções supracitadas são detalhadas a seguir.

```

1 fun CreatePNodes a list =
2   if (a=1) then
3     ins list ((n(1),s(1)))
4   else
5     ins list ((n(a),s(a))++)
6   CreatePNodes (a-1) list;
```

```

1 fun CreateNonPNodes a list =
2   if (a>t) then
3     empty
4   else
5     ins list (n(a))++
6     CreateNonPNodes(a+1) list;
```

```

1 fun CreateCompNodes(i: int) =
2   let
3     val cd = NODE.ran();
4   in
5     if (i=1) then
6       1'cd
7     else
8       1'cd++
9     CreateCompNodes(i-1)
10  end;
```



```
1 fun concatLists l1 l2 lres =  
2   if (List.length l1 = List.length l2) then  
3     if (List.length l1 = 0) then empty  
4     else  
5       ins lres (List.nth(l1,0), List.nth(l2,0))++  
6       concatLists (List.drop(l1,1)) (List.drop(l2,1)) lres  
7   else empty;
```

```
1 fun allNodesInList a mylist =  
2   let  
3   val cd = (n(a))  
4   in  
5   if (a=0) then empty  
6   else  
7     cd::mylist++  
8     allNodesInList (a-1) mylist  
9 end;
```

```
1 fun chooseOne l1 =  
2   let  
3   val b = List.length l1  
4   in  
5   if (b=1) then List.nth(l1,0)  
6   else  
7     if (b=0) then n(1)  
8     else  
9       List.nth (l1, discrete(0,b) )  
10  end;
```

```
fun chooseNodes a lavoid lchosen =  
2 let  
  val ltodos = allNodesInList t []  
4 val lresta = listsub ltodos lavoid  
  val aNode = chooseOne lresta  
6 in  
  if (a=0) then lchosen  
8 else  
  if (List.length lavoid + 1 = List.length ltodos) then  
10   (aNode::lchosen)  
  else  
12   if (List.length lavoid = List.length ltodos) then  
    []  
14   else  
    chooseNodes (a-1) (aNode::lavoid) (aNode::lchosen)  
16 end;
```