Universidade Federal de Pernambuco

Centro de Informática

Mestrado em Ciência da Computação

UM ESQUEMA DE ANONIMATO E AUTENTICAÇÃO MÚTUA PARA SISTEMAS RFID COM PROTOCOLOS ANTICOLISÃO BASEADOS EM ÁRVORE

Bruno Gentilini D'Ambrosio

DISSERTAÇÃO DE MESTRADO

Recife

21 de Fevereiro de 2013

Universidade Federal de Pernambuco

Centro de Informática

Bruno Gentilini D'Ambrosio

UM ESQUEMA DE ANONIMATO E AUTENTICAÇÃO MÚTUA PARA SISTEMAS RFID COM PROTOCOLOS ANTICOLISÃO BASEADOS EM ÁRVORE

Trabalho apresentado ao Programa de Mestrado em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Paulo André da Silva Gonçalves*

Recife

21 de Fevereiro de 2013

Dedico este trabalho primeiramente a Deus por ter me dado forças para sempre seguir em frente. Aos meus pais, por sempre me apoiarem e me instigarem a alcançar patamares cada vez maiores. A minha namorada, por ter servido de inspiração para este trabalho. E por fim ao Professor Paulo Gonçalves, pelas orientações e ensinamentos desde o início de nossa parceria que já dura quase 5 anos.

Gostaria de agradecer primeiramente a Deus. Se não fosse por ele com certeza este trabalho não teria sido concluído.

A minha família, meus pais e todos os outros parentes, que sempre me deram apoio e carinho para que eu seguisse sempre em frente. Agradeço especialmente à minha mãe Regina que durante esse mestrado aguentou todos os meus momentos mais difíceis.

A minha namorada, Suely Barbosa da Silva, que aguentou meus momentos mais complicados e muitas vezes serviu de inspiração para que eu pudesse terminar este trabalho.

Ao meu orientador, o professor *Docteur* Paulo André da Silva Gonçalves, que me ofereceu a oportunidade de trabalhar com pesquisas científicas ainda durante a graduação e sempre acreditou no meu potencial para crescer e fazer trabalhos cada vez melhores.

Aos Professores José Augusto Suruagy Monteiro e Marcial Fernandez que aceitaram participar da banca de avaliação desse trabalho.

Aos meus colegas do grupo de pesquisa Bruno de Jesus, Bruno Almeida, Eduardo, Nivia, Hermano, Marcos, Ivan, Luca, Pedro, José, Felipe e Júlio pelos conselhos fornecidos durante as reuniões do grupo e pelo companheirismo durante essa jornada.

Aos meus amigos do Centro de Informática Igor Ryan, Sylvia Marcela, Carlos Frederico, Diego Cesar, Victor Hugo, Felipe Lapenda, Gabriela Coutinho, Ítalo Macedo e todos os outros que me forneceram momentos de descontração e apoio durante todo o mestrado.

Aos meus grandes amigos Thiago, Fábio, Guilherme, Vanessa, Lara, Andreza, Felype Nery, Márcia Victorino, Jacqueline, Marcelo, Sérgio, Nilson Junior e todos os outros por me lembrarem que existe vida fora do Centro de Informática e pelo imenso apoio.

Agradeço por fim a Deus por sempre ter me dado forças nos momentos mais difíceis e zelado pela minha saúde e segurança durante toda a minha jornada.

"... E eu cavalgo os ventos

De um novo dia

Alto onde as montanhas alcançam

Encontro novamente meu orgulho e esperança

Renascimento de um homem."

—ANGRA (Rebirth - tradução livre)

RESUMO

Dentre as tecnologias desenvolvidas para aplicações que requerem a identificação automática de objetos, uma das mais promissoras é a RFID (*Radio-Frequency IDentification*). Prover autenticação mútua é um dos maiores desafios de sistemas RFID que utilizam etiquetas passivas. Isso ocorre devido às limitações de recursos computacionais e memória inerentes a esse tipo de etiqueta. Os esquemas mais atuais de autenticação mútua etiqueta-leitor para sistemas RFID baseados em etiquetas passivas, o SEAS e o SAMA, são capazes de manter o anonimato das etiquetas. Mas para isso, um requisito mínimo necessário é nunca transmitir em claro o ID real delas durante qualquer troca de mensagem com o(s) leitor(es). Por outro lado, o ID real é comumente utilizado e transmitido em claro durante a execução de protocolos anticolisão baseados em árvore. Essa execução precede o processo de autenticação, fazendo com que o anonimato das etiquetas não possa ser garantido. Este trabalho propõe um esquema para ser utilizado com protocolos anticolisão baseados em árvore que permite a autenticação mútua etiqueta-leitor e preserva o anonimato das etiquetas. O esquema proposto, denominado AMAS (*Anonymous Mutual Authentication Scheme*), é projetado com foco em sistemas com etiquetas passivas, as quais possuem recursos computacionais limitados. A proposta introduz o uso de IDs aleatórios e temporários desde a execução do protocolo anticolisão baseado em árvore, não permitindo a um atacante correlacionar tais IDs com os IDs reais. Este trabalho também avalia os custos do AMAS, em termos de quantidade de portas lógicas e ciclos de relógio. Os resultados demonstram que o AMAS utiliza 1214 portas lógicas e 150 ciclos de relógio, atendendo aos requisitos necessários para que ele possa ser utilizado em sistemas RFID baseados em etiquetas passivas.

Palavras-chave: Segurança, RFID, Autenticação, Anonimato, Ataques, Vulnerabilidades.

ABSTRACT

Amongst the technologies developed for applications which require the automatic identification of objects, one of the most promising is the RFID (Radio-Frequency IDentification). Provide mutual authentication is one of the main challenges of RFID systems that use passive tags. This is due to limitations of computational resources and memory inherent in this type of tag. The most recent tag-reader mutual authentication schemes for RFID systems based on passive tags, SAMA and SEAS, are able to maintain anonymity of the tags. However, a minimum requirement is that the tag's real ID must never be transmitted in clear text during any message exchange with the reader(s). Moreover, the real ID is commonly used and transmitted in plain text during the execution of tree-based anticollision protocols. This execution precedes the authentication process, so that the anonymity of the tags can not be guaranteed. This paper proposes a scheme to be used with tree-based anticollision protocols that allows tag-reader mutual authentication and preserves the anonymity of the tags. The proposed scheme, namely AMAS (Anonymous Mutual Authentication Scheme), is designed with focus on systems that use passive tags, which have limited computing capabilities. The proposal introduces the use of random and temporary IDs since the execution of the tree-based anticollision protocol, not allowing an attacker to correlate these IDs with the real IDs. This work also evaluates AMAS' costs in terms of number of logic gates and clock cycles. The results show that AMAS meets the necessary requirements to be used in RFID systems based on passive tags.

Keywords: Security, RFID, Authentication, Anonymity Attacks, Vulnerabilities.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ACRÔNIMOS

AES *Advanced Encryption Standard*. 24

AMAS *Anonymous Mutual Authentication Scheme*. 34

BS *Binary Search*. 12

CRC *Cyclic Redundancy Check*. 9

DFSA *Dynamic Frame Slotted ALOHA*. 11

EDFSA *Enhanced Dynamic Frame Slotted ALOHA*. 11

EPC *Eletronic Product Code*. 9

EPC Gen 1 *EPCGlobal Generation 1*. 8

EPC Gen 2 *EPCGlobal UHF Class 1 Generation 2*. 9

FSA *Frame Slotted ALOHA*. 11

ID *Identificador*. 1

ISO *International Organization for Standardization*. 10

LCAP *Low-Cost RFID Authenticantion Protocol*. 23

MD5 *Message-Digest algorithm 5*. 21

MIT *Massachusetts Institute of Technology*. 7

NLFSR *Non-Linear Feedback Shift Register*. 30

PA *Pure ALOHA*. 11

PRF Pseudo Random Function. 24

PRNG *Pseudo Random Number Generator*. 24

QT *Query Tree*. 12

RF *Radiofrequência*. 1

RFID *Radio-Frequency IDentification*. 1

SA *Slotted ALOHA*. 11

SAMA *Serverless Anonymous Mutual Authentication*. 30

SEAS *Secure and Efficient Anonymity Scheme*. 28

SHA-1 *Secure Hash Algorithm-1*. 20

SHA-256 *Secure Hash Algorithm-256*. 20

SQL Structed Query Language. 19

TID *Transponder ID*. 9

TS *Tree Splitting*. 12

CAPÍTULO 1

INTRODUÇÃO

Recentemente está ocorrendo o surgimento de uma grande quantidade de aplicações que requerem a identificação, organização e a localização automática de objetos. Dentre as tecnologias desenvolvidas para atender os requisitos desse tipo de aplicação, uma das mais promissoras é a RFID (*Radio-Frequency IDentification* - Identificação por Radio-frequência)[Want 2006].

1.1 MOTIVAÇÃO

Os sistemas RFID têm duas grandes vantagens em relação aos sistemas de identificação tradicionais. A primeira é a atribuição de uma identificação exclusiva para cada objeto pertencente ao sistema, a qual é armazenada na memória da etiqueta RFID acoplada a esse objeto. Assim sendo, esse tipo de sistema consegue garantir que dois ou mais objetos exatamente iguais possam ser diferenciados entre si. A segunda é a realização da leitura dos dados contidos nas etiquetas através de ondas de rádio, o que possibilita a identificação das etiquetas em qualquer direção dentro do alcance do leitor. Graças a essas características, os sistemas RFID vêm sendo utilizados em uma grande quantidade de aplicações, como por exemplo, logística industrial [Wu et al. 2009], sistemas de controle e pagamento automático de estacionamento e pedágio [Wu et al. 2009], aplicações médicas [Tu, Zhou e Piramuthu 2009], sistemas de localização de objetos [Ni et al. 2003], entre outras.

Um sistema RFID pode ser composto por três tipos de componentes [Want 2006]: servidor, leitor e etiquetas. O servidor tem o papel de processar e armazenar informações de acordo com as necessidades da aplicação. Ele se comunica diretamente com o leitor

através de canal de comunicação seguro. Por sua vez, o leitor se comunica com as etiquetas através de sinais de radiofrequência (RF). As principais tarefas de um leitor podem incluir: o controle do acesso das etiquetas ao meio de comunicação sem fio através do uso de um protocolo anticolisão; a mediação ou realização dos processos de identificação e autenticação de etiquetas; e a mediação do tráfego de informações entre as etiquetas e o servidor. As etiquetas armazenam um identificador (ID) único e repondem a requisições feitas pelo leitor. Essas etiquetas podem ser passivas, ativas ou semi-passivas, dependendo da fonte de energia utilizada por cada uma delas. Este trabalho tem um interesse maior nas etiquetas passivas, as quais possuem um menor custo de produção.

Em sistemas RFID, o controle de acesso das etiquetas ao meio é arbitrado pelo leitor através do uso de um protocolo anticolisão [Klair, Chin e Raad 2010]. Nos protocolos anticolisão baseados em árvore, o leitor envia uma requisição às etiquetas. Cada etiqueta que satisfaz à requisição reponde com seu identificador único (ID). Quando duas ou mais etiquetas repondem ao mesmo tempo, ocorre uma colisão. O leitor faz, então, uma série de requisições, dividindo recursivamente essas etiquetas em subgrupos até que apenas uma etiqueta responda. O processo descrito pode ser representado por uma árvore. A raiz representa a população de etiquetas a ser identificada. Os nós intermediários representam subgrupos de etiquetas que colidiram ao responderem a uma mesma requisição do leitor. As folhas representam a resposta de uma única etiqueta a uma requisição do leitor, permitindo sua identificação pelo ID, ou ainda, sua seleção para qualquer outra comunicação exclusiva com essa etiqueta e prevista pela aplicação, como por exemplo, um processo de autenticação mútua etiqueta-leitor.

Prover a autenticação mútua etiqueta-leitor é um dos maiores desafios em sistemas RFID baseados em etiquetas passivas devido à impossibilidade de se utilizar primitivas criptográficas como aquelas baseadas na função SHA-256 e no AES (*Advanced Encryption Standard*) [Feldhofer e Rechberger 2006]. Isso ocorre devido às limitações de recursos computacionais dessas etiquetas, as quais possuem no máximo 4.000 portas lógicas dedicadas aos mecanismos de segurança [Myneni, Misra e Xue 2011, Misra et al. 2009]. Além disso, por limitações de tempo de processamento e de consumo de energia, a

quantidade máxima de ciclos de relógio utilizada por tais mecanismos está limitada em 220 [Myneni, Misra e Xue 2011, Misra et al. 2009].

Diversos esquemas buscam prover autenticação em sistemas RFID com etiquetas passivas [Peris-Lopez et al. 2006, Misra et al. 2009, Myneni, Misra e Xue 2011]. Dentre eles, o SAMA (*Serverless Anonymous Mutual Authentication*) [Myneni, Misra e Xue 2011] e o SEAS (*Secure and Efficient Anonymity Scheme*) [Misra et al. 2009], são os mais atuais para prover autenticação mútua etiqueta-leitor. Esses dois esquemas são capazes de manter o anonimato das etiquetas. Mas para isso, um requisito mínimo necessário é nunca transmitir em claro o ID real dessas etiquetas durante qualquer troca de mensagem com o(s) leitor(es). Por outro lado, o ID real das etiquetas é comumente utilizado e transmitido em claro durante a execução de protocolos anticolisão baseados em árvore. Essa execução precede o processo de autenticação, fazendo com que o anonimato das etiquetas não possa ser garantido.

1.2 OBJETIVOS

O objetivo geral dessa dissertação de mestrado é a proposição de um esquema de autenticação para sistemas RFID. Esse esquema deve manter o anonimato das etiquetas em sistemas que utilizem protocolos anticolisão baseados em árvore e deve ser suficientemente leve para que possa ser implementado em etiquetas passivas. Para alcançar o objetivo geral, são definidos os seguintes objetivos específicos:

1. Estudo sobre conceitos de sistemas RFID e protocolos anticolisão;
2. Estudo de ataques a sistemas RFID;
3. Estudo dos esquemas de autenticação mútua existentes para sistemas RFID;
4. Proposta de um esquema de autenticação mútua;
5. Implementação e avaliação da segurança do esquema proposto;
6. Comparação do esquema proposto com os esquemas mais relevantes do estado da arte;

1.3 CONTRIBUIÇÃO

Este trabalho propõe um mecanismo para ser utilizado com protocolos anticolisão baseados em árvore que permite a autenticação mútua etiqueta-leitor e preserva o anonimato das etiquetas. O esquema proposto, denominado AMAS (*Anonymous Mutual Authentication Scheme*), é projetado com foco em sistemas que utilizam etiquetas passivas. Assim sendo, as limitações de recursos computacionais dessas etiquetas são levadas em consideração. A proposta introduz o uso de IDs aleatórios e temporários desde a execução do protocolo anticolisão baseado em árvore, não permitindo a um atacante correlacionar tais IDs com os IDs reais.

O AMAS utiliza apenas funções de baixo custo em termos de quantidade de portas lógicas e ciclos de relógio, o que é comprovado através de avaliações realizadas por esse trabalho. Assim sendo, o esquema é suficientemente leve para ser aplicado em sistemas RFID que utilizam etiquetas passivas.

O AMAS se diferencia dos esquemas mais atuais presentes no estado da arte, o SAMA e o SEAS, por conseguir manter o anonimato das etiquetas mesmo quando um protocolo anticolisão baseado em árvore é utilizado. Além disso, o esquema proposto consome uma quantidade de portas lógicas menor que os demais esquemas. Contudo, o AMAS consome mais ciclos de relógio do que o SAMA e o SEAS para que diferentemente dessas propostas, proteja o anonimato das etiquetas desde o processo anticolisão.

1.4 ORGANIZAÇÃO

O Capítulo 2 descreve as características principais dos sistemas RFID e detalha o funcionamento dos mecanismos básicos que compoem esses sistemas. O Capítulo 3 apresenta o estado da arte sobre os ataques externos e internos a sistemas RFID. O Capítulo 4 descreve o funcionamento dos principais mecanismos de autenticação mútua para sistemas RFID. Os mecanismos apresentados nesse capítulo estão divididos em três classes: baseados em funções *hash*, baseados em algoritmos criptográficos e baseados em funções lógicas e bit-a-bit. O Capítulo 5 apresenta e avalia o esquema de anonimato e autenticação

mútua proposto neste trabalho. Por fim, o Capítulo 6 conclui o trabalho.

CAPÍTULO 2

RADIO-FREQUENCY IDENTIFICATION (RFID)

Este capítulo apresenta os fundamentos básicos dos sistemas RFID. A Seção 2.1 apresenta a arquitetura desse tipo de sistema, identificando os seus principais componentes. Em seguida, a Seção 2.2 apresenta os padrões de RFID existentes. Por fim, a Seção 2.3 apresenta os principais protocolos anticolisão utilizados na identificação de etiquetas em sistemas RFID.

2.1 ARQUITETURA RFID

Os sistemas RFID são compostos por três componentes principais: o(s) leitor(es), as etiquetas e o(s) servidor(es) [Nekoogar e Dowla 2012, Want 2006]. O servidor é a entidade central de um sistema RFID. Ele tem o papel de processar e armazenar informações de acordo com as necessidades da aplicação. Em geral, o servidor possui um banco de dados no qual ele armazena e acessa as informações de todas as etiquetas e leitores presentes no sistema. O servidor também pode ser responsável por executar partes dos esquemas de segurança dos sistemas RFID. A comunicação entre o servidor e o(s) leitor(es) é realizada através de um canal de comunicação seguro.

A função básica de um leitor em um sistema RFID é consultar as informações contidas nas etiquetas que estiverem no seu alcance de leitura. Ele também pode ser capaz de escrever dados nas etiquetas, mas isso só é possível caso as mesmas possuam memória regravável. Outra função do leitor é realizar o controle de acesso das etiquetas ao meio de comunicação sem fio, o que é feito através da execução de um protocolo anticolisão. O leitor também é responsável pela mediação do tráfego de informações entre as etiquetas e o servidor. A comunicação entre o(s) leitor(es) e as etiquetas é feita através de

radiofrequência.

Os leitores também são responsáveis pela execução dos algoritmos mais complexos que venham a ser utilizados por esquemas de segurança para sistemas RFID. Isso ocorre por eles possuírem um *hardware* com maior poder computacional e utilizarem fontes de energia mais potentes que as etiquetas.

As etiquetas são componentes mais simples que armazenam informações sobre o objeto ao qual estão associadas. A seção a seguir apresenta em maiores detalhes as principais características das etiquetas RFID e as duas principais formas utilizadas para classificá-las.

2.1.1 Etiquetas

As etiquetas são responsáveis por armazenar informações básicas sobre o objeto ao qual estão associadas. Elas, em geral, só são capazes de transmitir informações para o leitor, mas em casos especiais também podem transmitir dados para outras etiquetas, trabalhando de forma semelhante a uma rede de sensores. Em geral a estrutura de uma etiqueta é composta por uma antena, a qual é responsável pelas recepções e transmissões de dados, por um chip de silício, que contém todo o *hardware* da etiqueta, e por material de encapsulamento [Want 2006, Nekoogar e Dowla 2012]. Existe mais de um tipo de etiqueta e nas seções a seguir serão tratadas as duas classificações mais utilizadas na literatura.

2.1.1.1 Classificação de Acordo com a Fonte de Energia

As etiquetas podem ser classificadas de acordo com algumas características, sendo a mais comum se elas utilizam ou não fontes de energia próprias, a qual as divide em três classes: passivas, ativas ou semi-passivas [Nekoogar e Dowla 2012].

As etiquetas passivas são as mais comuns e baratas. Elas não possuem bateria, obtendo a energia necessária através de um processo conhecido como indução magnética [Want 2006, Nekoogar e Dowla 2012]. Nesse processo a etiqueta absorve a

energia da onda transmitida pelo leitor em um capacitor e a utiliza para realizar suas operações internas e transmitir a reposta. A ausência de bateria faz com que as etiquetas passivas tenham um *hardware* bastante limitado, o que reduz tanto a quantidade de operações que elas podem realizar como o alcance de transmissão das mesmas. O alcance de transmissão de uma etiqueta passiva é de 1 metro de distância [Want 2006].

As etiquetas ativas possuem uma fonte de energia própria que é utilizada para todas as funções que ela realiza. Elas possuem um *hardware* com maior poder computacional que as passivas, o que implica que elas podem realizar uma maior quantidade de operações. O alcance de transmissão desse tipo de etiqueta também é maior, podendo ultrapassar os 6 metros de distância em alguns casos [Want 2006].

As etiquetas semi-passivas utilizam um esquema de energia híbrido. Para realizar suas operações internas essas etiquetas utilizam uma fonte de energia própria [Nekoogar e Dowla 2012]. Já para as transmissões a energia é obtida através do mesmo esquema de indução magnética utilizado pelas etiquetas passivas. Esse tipo de etiqueta tem alcance maior que as passivas, pois a energia absorvida é utilizada somente para transmitir as informações.

2.1.1.2 Classificação do *MIT Auto-ID Center*

O grupo MIT (*Massachusetts Institute of Technology*) *Auto-ID Center* é formado por uma cooperação de sete universidades espalhadas pelo mundo e tem como principal foco de pesquisa os sistemas RFID. Esse grupo e a organização *EPCglobal* são responsáveis pelo desenvolvimento e pela padronização dos sistemas RFID. O *MIT Auto-ID Center* tem uma classificação própria para os tipos de etiqueta. Nessa classificação as etiquetas são divididas em 5 classes distintas: classe 0, classe 1, classe 2, classe 3 e classe 4 [Nekoogar e Dowla 2012].

Classe 0 - As etiquetas dessa classe são as mais simples e tem como única função anunciar a presença do objeto num determinado local. As etiquetas dessa classe são programadas ainda na fase de fabricação e não possuem memória regravável, o que impede que o usuário modifique o conteúdo das mesmas. São frequemente utilizadas em sistemas

anti-furto de diversos tipos de loja. As etiquetas dessa classe são passivas.

Classe 1 - As etiquetas dessa classe possuem um poder computacional um pouco maior e permitem que o usuário grave dados na sua memória uma única vez. Após essa gravação, etiquetas desse tipo passam a permitir apenas operações de leitura. As etiquetas dessa classe também são passivas.

Classe 2 - Etiquetas dessa classe se diferenciam das etiquetas das classes anteriores por possuírem memória regravável. Essa característica permite que elas sejam reutilizadas caso seja necessário trocar a etiqueta entre objetos de diferentes tipos. Também são etiquetas passivas.

Classe 3 - As etiquetas da classe 3 são as primeiras a possuírem sensores capazes de coletar informações sobre o ambiente ao seu redor. Essa informações podem ser gravadas na memória da etiqueta e requisitadas a qualquer momento. Essas etiquetas são semi-passivas.

Classe 4 - As etiquetas da classe 4 são as que possuem a maior quantidade de poder computacional, sendo capazes de realizar comunicações entre etiquetas. Essa característica faz com que sistemas RFID que utilizem etiquetas desse tipo se tornem muito semelhantes às redes de sensores. As etiquetas dessa classe são ativas.

2.2 PADRÕES

Em qualquer tipo de tecnologia, o desenvolvimento de um padrão bem consolidado favorece a popularização e a evolução da mesma. No caso dos sistemas RFID, ainda não existe um padrão unânime, pois diversas empresas e instituições vêm desenvolvendo padrões distintos. Dentre essas empresas, duas têm desenvolvido os padrões mais bem aceitos até o momento: a *EPCglobal Inc.* e a *International Organization for Standardization* (ISO). Nesta seção são apresentados os padrões para sistemas RFID desenvolvidos por essas organizações.

2.2.1 Padrões da EPCGlobal

O EPC Gen 1 (*EPCGlobal Generation 1*) foi um dos primeiros padrões desenvolvidos na tentativa de uniformizar os sistemas RFID. Na década de 1990 ainda não existia nenhum padrão específico para esse tipo de sistema. Por isso, a partir de seu desenvolvimento no início dos anos 2000, ele foi o padrão mais utilizado. No EPC Gen 1, as etiquetas permitem apenas operações de leitura e o leitor tem como único propósito consultar as informações contidas nas mesmas. Assim sendo, as classes de etiquetas 0 e 1 são as únicas suportadas por esse padrão.

O EPC Gen 2 (*EPCGlobal UHF Class 1 Generation 2*) [EPCGlobal 2008] foi aprovado pela EPCGlobal em meados de 2004, mas só foi adotado pelo mercado em larga escala em 2005. A ideia desse protocolo foi se adaptar às necessidades das novas aplicações desenvolvidas com os sistemas RFID. As principais melhorias introduzidas pelo EPC Gen 2 são uma melhora considerável no processo de comunicação e a introdução de mecanismos visando a segurança. Para isso, foi desenvolvida uma arquitetura com duas camadas: uma física e uma para tratar a identificação de etiquetas.

Na camada física é adotada a comunicação *half-duplex*. Assim sendo, tanto etiquetas como leitores podem transmitir e receber dados de forma não simultânea. Isso implica que em um dado momento o leitor age como transmissor e a etiqueta como receptor e num momento seguinte os papéis podem ser invertidos de acordo com a necessidade da aplicação.

A camada física também define a divisão da memória das etiquetas em quatro partes distintitas. Na primeira parte são armazenadas duas chaves de 32 bits: a *kill password*, que é utilizada para desabilitar a etiqueta, e a *access password*, que é utilizada para autorizar qualquer tipo de escrita e leitura no restante da memória da etiqueta. Na segunda parte é armazenado o identificador único da etiqueta de 32 bits conhecido como EPC (*Eletronic Product Code*). Nessa parte também são armazenados 16 bits de controle e mais 16 bits utilizados como parâmetros pelo CRC (*Cyclic Redundancy Check*). Na terceira parte está presente um conjunto de bits conhecidos como memória TID (*Transponder ID*), que tem como objetivo mostrar ao leitor quais comandos e características opcionais a etiqueta

suporta. E a última parte é reservada para dados específicos da aplicação desenvolvida pelo usuário.

A segunda camada da arquitetura trata da identificação das etiquetas. Essa camada define as operações básicas que os leitores podem efetuar em uma etiqueta, sendo elas: *Select*, que permite ao leitor selecionar uma etiqueta; *Inventory*, a qual permite ao leitor identificar uma etiqueta; e, por fim, *Access*, que permite ao leitor acessar e modificar os dados contidos na memória da etiqueta. Outra função dessa camada é lidar com as colisões entre as mensagens de etiquetas e leitores. Caso duas ou mais entidades do mesmo tipo transmitam ao mesmo tempo, pode ocorrer colisão entre as mensagens enviadas, impossibilitando o correto recebimento das mesmas. Para minimizar esse tipo de problema essa camada prevê a utilização de um protocolo anticolisão.

Esse padrão ainda define que o *hardware* das etiquetas contenha um gerador de números pseudo-aleatórios de 16 *bits* que siga as seguintes regras [EPCGlobal 2008]:

- A probabilidade que uma sequência qualquer de 16 bits j seja gerada deve ser maior do que $P_{min} = \frac{0,8}{2^{16}}$ e menor do que $P_{max} = \frac{1,25}{2^{16}}$;
- Em uma população de até 10 mil etiquetas a probabilidade de que duas etiquetas gerem simultaneamente a mesma sequência deve ser menor do que $0,1\%$;
- A probabilidade de um atacante descobrir qual será a próxima sequência gerada deve ser de no máximo $0,025\%$ mesmo que ele tenha acesso a todas as sequências geradas anteriormente.

O padrão também define que as etiquetas possuam um CRC também de 16 bits. O gerador de número pseudoaleatórios e o CRC podem ser utilizados na construção de mecanismos para prover a segurança dos sistemas RFID.

2.2.2 Série de Padrões ISO 18000

A série de padrões ISO 18000 foi desenvolvida pela ISO (*International Organization for Standardization*). Essa série trata apenas da especificação do protocolo de comunicação

sem fio, o qual deve ser implementado pelos sistemas RFID, conhecido como *Air Interface Protocol*. As implementações da parte física e da arquitetura dos sistemas RFID não são tratadas por essa série, deixando essa responsabilidade para as indústrias que produzem os sistemas. Essa restrição do padrão visou torná-lo mais abrangente e compatível com as diferentes gerações de sistemas RFID.

A série 18000 é composta por sete padrões distintos, cada um deles define as normas que o protocolo de comunicação deve seguir para transmitir em uma faixa de frêquencia de transmissão específica. Os 7 padrões são:

1. ISO 18000-1 - Normas gerais para frequências adotadas mundialmente;
2. ISO 18000-2 - Normas para sistemas RFID com frequência abaixo de 125 kHz;
3. ISO 18000-3 - Normas para sistemas RFID com frequência de 13,56 MHz;
4. ISO 18000-4 - Normas para sistemas RFID com frequência de 2,45 GHz;
5. ISO 18000-5 - Normas para sistemas RFID com frequência de 5,8 GHz;
6. ISO 18000-6 - Normas para sistemas RFID com frequência entre 860-930 MHz;
7. ISO 18000-7 - Normas para sistemas RFID com frequência de 433 MHz.

2.3 PROTOCOLOS ANTI-COLISÃO

Duas ou mais etiquetas podem estar no alcance de comunicação do leitor. Quando o leitor envia uma mensagem tentando identificar essas etiquetas, é possível que mais de uma responda no mesmo instante. Logo, é possível ocorrer uma colisão, o que significa que nenhuma das etiquetas que responderam ao mesmo tempo foram identificadas com sucesso.

Para diminuir o número de colisões e facilitar a identificação de etiquetas existe uma quantidade considerável de protocolos anti-colisão [Klair, Chin e Raad 2010] que podem ser utilizados. Esses protocolos estão divididos em duas abordagens: a dos protocolos

baseados em *ALOHA*, os quais são probabilísticos, e os baseados em árvore, os quais são determinísticos.

Os protocolos baseados em *ALOHA* [Klair, Chin e Raad 2010] são probabilísticos, o que implica que existe a probabilidade de que continuem ocorrendo colisões indefinidamente dependendo de fatores aleatórios existentes no funcionamento dos protocolos desse tipo. Isso ocorre porque depois que uma colisão é detectada cada etiqueta seleciona uma quantidade aleatória de unidades de tempo para esperar até que ela possa transmitir novamente. Essas unidades de tempo podem ser divididas em *slots* e/ou *frames* dependendo de qual protocolo desse tipo está sendo utilizado. Caso todas as etiquetas escolham a mesma quantidade de unidades de tempo, o processo tem de ser repetido até que não ocorram colisões.

Existem diversos protocolos baseados em ALOHA, como por exemplo: o PA (*Pure ALOHA*) [Klair, Chin e Raad 2007] , o SA (*Slotted ALOHA*) [Klair, Chin e Raad 2007], o FSA (*Frame Slotted ALOHA*) [Zhena, Kobayashi e Shimizu 2005], o DFSA (*Dynamic Frame Slotted ALOHA*) [Bueno-Delgado, Vales-Alonso e Gonzalez-Castao 2009] e o EDFSA (*Enhanced Dynamic Frame Slotted ALOHA*) [Tong, Zou e Tong 2009]. Esses protocolos não serão descritos detalhadamente uma vez que o foco desse trabalho é nos protocolos anticolisão baseados em árvore.

Os protocolos baseados em árvore [Klair, Chin e Raad 2010] são determinísticos, ou seja, eles obrigatoriamente[1] conseguem identificar todas as etiquetas em colisão ao final do procedimento. Nos protocolos anticolisão baseados em árvore, o leitor envia uma requisição às etiquetas. Cada etiqueta que satisfaz à requisição reponde com seu identificador único (ID). Quando ocorre uma colisão o leitor divide recursivamente essas etiquetas em subgrupos até que apenas uma etiqueta responda. O processo descrito pode ser representado por uma árvore. A raiz representa a primeira rodada de identificação executada pelo protocolo, na qual geralmente ocorre colisão entre toda a população de etiquetas a ser identificada. Os nós intermediários representam as rodadas do protocolo onde ocorrem colisões entre subgrupos de etiquetas ao responderem a uma mesma requisição do

[1]Supondo um canal de comunicação livre de erros

leitor. As folhas representam as rodadas do protocolo nas quais apenas uma única etiqueta enviou uma resposta, permitindo que essa etiqueta seja identificada a partir de seu ID.

Os protocolos baseados em árvore podem ser classificados em categorias distintas, como por exemplo: QT (*Query Tree*), TS (*Tree Splitting*) e BS (*Binary Search*). Nas seções a seguir serão apresentadas as principais características dos protocolos pertencentes a essas três categorias.

2.3.1 Query Tree (QT)

A principal característica dos protocolos da categoria QT é a utilização de prefixos nas consultas, ou seja, as etiquetas só responderão a uma consulta se o prefixo de n *bits* consultado pelo leitor for igual aos n primeiros *bits* do ID da etiqueta. Existem diversos protocolos pertencentes a essa categoria, como por exemplo: o QT básico [Law, Lee e Siu 2000], o *Adaptative* QT [Myung, Lee e Shih 2006], o *Improved* QT [Zhou et al. 2004], o QTR [Choi, Lee e Lee 2007], entre outros [Gou, Jeong e Yoo 2010].

No protocolo QT básico é primeiramente feita uma consulta com um prefixo vazio, à qual todas as etiquetas responderão com seu ID. Em seguida, o leitor vai consultando os ramos da árvore binária recursivamente até que todas as etiquetas sejam identificadas. Para caminhar para os ramos a esquerda de um nó é adicionado um **0** ao final do prefixo de consulta e para os ramos a direta é adicionado um **1** ao final do prefixo. Existem três tipos de situação que o leitor pode encontrar durante a caminhada na árvore:

- Quando mais de uma resposta é recebida para um determinado prefixo significa que houve colisão. Assim sendo, o leitor deve continuar descendo naqueles ramos da árvore para indentificar todas as etiquetas presentes nos mesmos.

- Quando uma resposta única é recebida para um determinado prefixo significa que o leitor chegou a uma folha da árvore. Assim sendo, a etiqueta correspondente ao ID recebido é silenciada e o leitor reinicia o processo a partir do ultimo nó em colisão.

- Quando nenhuma resposta é recebida para uma determinada consulta significa que

o leitor encontrou um ramo vazio e ele deve retornar para o ultimo nó em colisão.

Na Figura 2.1 é apresentado um exemplo do funcionamento do protocolo QT básico. Nesse exemplo, o leitor deve identificar três etiquetas, as quais possuem os IDs **001**, **010** e **100**. Na primeira rodada o leitor realiza uma consulta vazia, o que faz com que as três etiquetas respondam, gerando uma colisão. Em seguida, na segunda rodada, o leitor utiliza o prefixo $\mathbf{q = 0}$ na consulta, o que gera nova colisão entre as etiquetas **001** e **010**. Nas rodadas 3 e 4 o leitor utiliza, respectivamente, os prefixos $\mathbf{q = 00}$ e $\mathbf{q = 01}$, identificando as etiquetas **001** e **010**. Por fim, na rodada 5, o leitor retorna a raiz da árvore e realiza uma consulta com o prefixo $\mathbf{q = 1}$, identificando a última etiqueta.

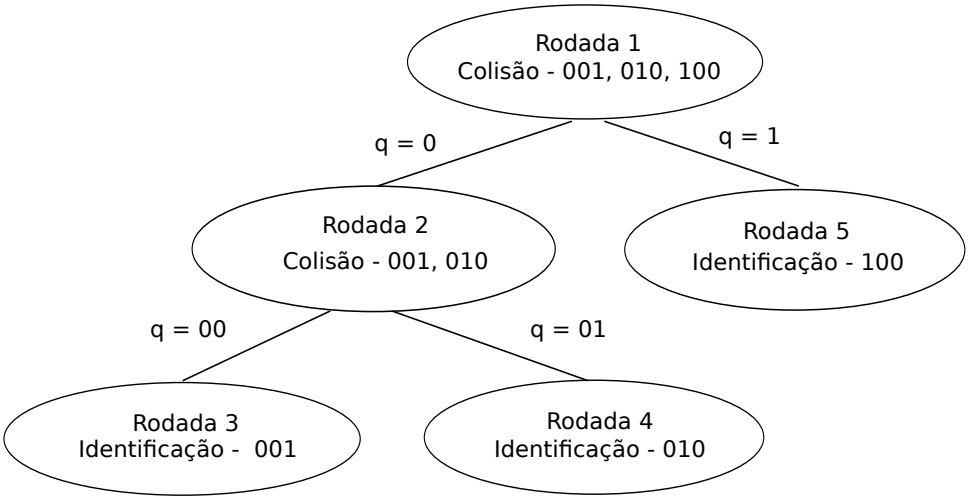


Figura 2.1 Exemplo de Funcionamento do QT.

Os demais protocolos dessa categoria, como o *Adaptative* QT, o *Improved* QT e o QTR, apresentam melhorias em relação ao protocolo QT básico na forma de caminhar na árvore e agilizar o processo de identificação das etiquetas.

2.3.2 Tree Splitting (TS)

Os protocolos da categoria TS realizam a divisão das etiquetas em grupos cada vez menores utilizando um gerador de números binários pseudoaleatórios em cada etiqueta do sistema [Klair, Chin e Raad 2010]. Esse gerador só é capaz de gerar os

valores **0** e **1**. Os protocolos mais conhecidos pertencentes a essa categoria são: o BTS (*Basic Tree Splitting*) [Hush e Wood 1998], o ABTS (*Adaptive Binary Tree Splitting*) [Myung, Lee e Srivastava 2006] e o EBTS (*Enhanced Binary Tree Splitting*) [Chen, Horng e Fan 2007]. No protocolo BTS, as etiquetas são divididas recursivamente em grupos cada vez menores até que cada grupo possua uma única etiqueta. Nesse protocolo as etiquetas possuem um contador de n bits, o qual é utilizado para indicar a posição que elas se encontram na árvore. Esse protocolo é dividido em rodadas, nas quais cada uma das etiquetas podem ou não transmitir dependendo do valor atual do contador. Se o valor do contador for 0 a etiqueta se encontra na raiz do ramo atual da árvore e irá transmitir nessa rodada, caso contrário a etiqueta se encontra em modo de espera e só irá transmitir quando o seu contador chegar a 0. Na primeira rodada, todas as etiquetas iniciam o contador com valor 0. Portanto, se o número de etiquetas for maior que 1, sempre ocorrerá uma colisão. Ao final de cada rodada existem 3 tipos de situação que podem ocorrer:

- Quando o leitor detecta uma colisão, ele avisa todas as etiquetas sobre o ocorrido. Todas as etiquetas que transmitiram na última rodada geram um número binário pseudoaleatório e o somam ao valor do seu contador. Assim sendo, somente as etiquetas que geraram o número **0** continuam em modo de transmissão na rodada seguinte. As etiquetas que estavam em modo de espera somam uma unidade ao valor do seu contador.

- Quando uma única etiqueta transmite o seu ID em uma dada rodada, significa que o leitor chegou a uma folha da árvore. Assim sendo, a etiqueta correspondente ao ID recebido é silenciada pelo leitor. Além disso, o leitor sinaliza para as demais etiquetas que ocorreu uma identificação. As etiquetas que estavam em modo de espera subtraem uma unidade do valor do seu contador.

- Quando nenhuma etiqueta transmite em uma dada rodada significa, que o leitor encontrou um ramo vazio da árvore. Quando isso ocorre, o leitor sinaliza o ocorrido para as demais etiquetas. Todas as etiquetas em modo de espera subtraem uma

unidade do valor do seu contador.

Na Figura 2.2 é apresentado um exemplo do funcionamento do BTS. Nesse exemplo, o leitor deve identificar as etiquetas **A**, **B** e **C** as quais possuem os IDs **001**, **010** e **100**. Na primeira rodada, as três etiquetas transmitem, pois estão com o contador zerado, o que gera uma colisão. Na segunda rodada, apenas a etiqueta **C** recebe o número aleatório $\mathbf{b = 0}$ de seu gerador, o que faz com que a etiqueta consiga transmitir sozinha o seu ID e seja identificada. Nessa mesma rodada, as demais etiquetas estão em modo de espera pois seus contadores estão com o valor **1**. A identificação de **C** na segunda rodada faz com que os valos dos contadores de **A** e **B** sejam reduzidos em uma unidade, o que faz com que as duas transmitam na rodada 3 gerando nova colisão. Na rodada 4, apenas o gerador da etiqueta **A** obtém o valor **0**, fazendo com que **A** seja identificada e **B** fique em estado de espera. Por fim, na rodada 5, a etiqueta **B** transmite o ID sozinha fazendo com que ela seja a última etiqueta a ser identificada.

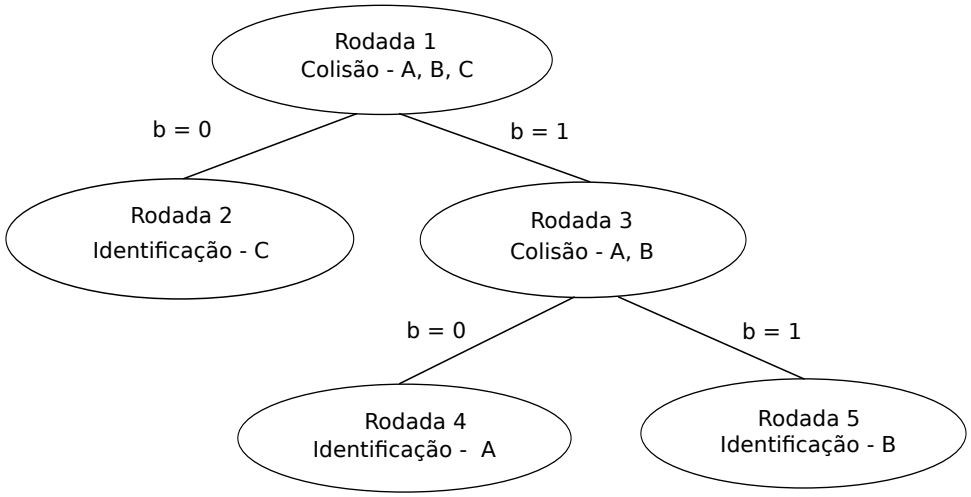


Figura 2.2 Exemplo de Funcionamento do TS.

Os outros dois protocolos dessa categoria, o ABTS e o EBTS, apresentam melhorias em relação ao BTS. Ambos os protocolos conseguem melhorar a velocidade de identificação e reduzir a quantidade de colisões e de ramos vazios da árvore.

2.3.3 Binary Search (BS)

Os protocolos da categoria BS utilizam a codificação Manchester para identificar colisões *bit* a *bit* entre os IDs transmitidos pelas etiquetas. Os protocolos mais conhecidos pertencentes a essa categoria são: o BS básico [Finkenzeller 2003], o EBSA (*Enhanced BS-algorithm*) [Yu et al. 2005] e o DBSA (*Dynamic BS-algorithm*) [Klair, Chin e Raad 2010].

No protocolo BS básico, o leitor transmite uma sequência de *bits*, a qual possui a mesma quantidade de *bits* dos IDs das etiquetas. As etiquetas comparam os seus IDs com a sequência enviada pelo leitor, enviando uma resposta apenas quando o valor de seus IDs forem menores que o valor dessa sequência. As respostas enviadas pelas etiquetas são os próprios IDs. O leitor monitora as respostas recebidas *bit* a *bit* utilizando a codificação Manchester (*Manchester coding*). Quando ocorre uma colisão, o leitor divide as etiquetas em subgrupos de acordo com os *bits* em colisão. Na primeira rodada desse algoritmo, o leitor sempre transmite uma sequência com todos os *bits* iguais a 1. Na rodada seguinte a uma colisão, o leitor modifica a sequência transmitida, mudando o valor do *bit* mais significativo em colisão pelo valor 0. O restante dos *bits* da sequência é mantido com o valor utilizado na rodada anterior.

Na Figura 2.3 é apresentado um exemplo do funcionamento do protocolo BS básico. Nesse exemplo, o leitor deve identificar três etiquetas, as quais possuem os IDs **010**, **011** e **100**. Para iniciar o processo, o leitor envia a sequência **111**, o que gera uma colisão na primeira rodada pois os IDs de todas as etiquetas são menores que **111**. Utilizando a codificação *Manchester*, o leitor identifica que houve colisão nos 3 *bits*, o que faz com que ele substitua o *bit* mais significativo da sequência pelo valor **0**. Na segunda rodada, o leitor transmite a sequência **011**, o que resulta em nova colisão entre as etiquetas **010** e **011**. O leitor identifica que a colisão ocorreu apenas no terceiro *bit*, o substituindo por **0**. Na terceira rodada, o leitor transmite a sequência **010**, o que faz com que apenas a etiqueta **010** responda e seja identificada. Após a identificação de **010**, o leitor reinicia o valor da sequência para **111** e repete os passos anteriores até que as demais etiquetas sejam identificadas, o que ocorre nas rodadas 5 e 6.

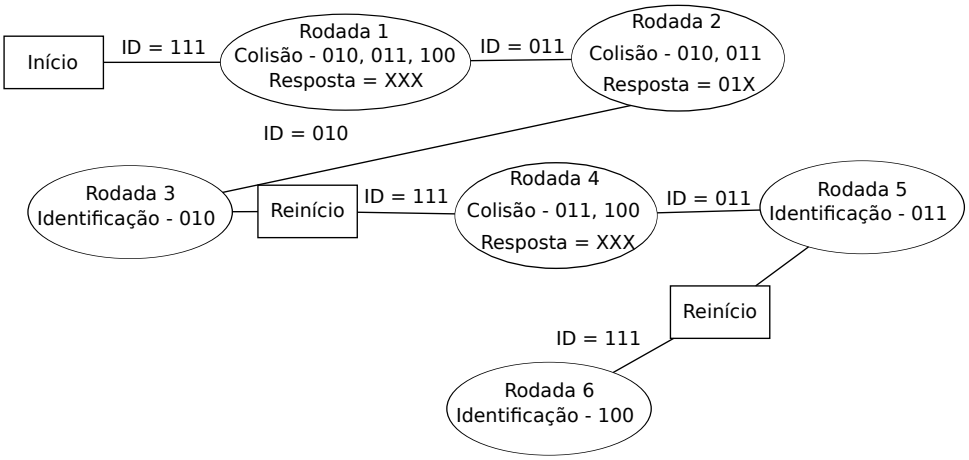


Figura 2.3 Exemplo de Funcionamento do BS.

Os demais protocolos dessa categoria, o EBSA (*Enhanced BS-algorithm*) e o DBSA (*Dynamic BS-algorithm*), apresentam melhorias em relação ao protocolo BS básico. Esses protocolos conseguem reduzir a quantidade de rodadas utilizadas e de *bits* transmitidos durante a identificação das etiquetas.

2.3.4 Diferenças entre Protocolos Anticolisão Baseados em ALOHA e Baseados em Árvore

A primeira diferença que pode ser citada entre os protocolos anticolisão baseados em ALOHA e os baseados em árvore é em relação à completude dos protocolos dessas duas classes. Os baseados em ALOHA são probabilísticos, ou seja, existe a probabilidade de que continuem ocorrendo colisões indefinidamente durante a execução desses protocolos. Já os baseados em árvore são determinísticos, ou seja, se o canal de comunicação for livre de erros, eles sempre conseguem identificar todas as etiquetas ao final do processo. Justificar essa escolha. Outra diferença entre essas duas classes é quanto a necessidade de implementação de um gerador de números aleatórios nas etiquetas. Os protocolos baseados em árvore, em geral, não requerem a utilização de um gerador de números aleatórios. A única exceção nesse caso são os protocolos da categoria TS, os quais requerem um gerador de números aleatórios de apenas 1 *bit*. Por outro lado, todos os protocolos baseados

em ALOHA requerem que as etiquetas implementem geradores de números aleatórios, os quais são utilizados para gerar a quantidade de unidades de tempo ou de *slots* que a etiqueta irá esperar para transmitir após a ocorrência de uma colisão.

A última diferença é referente a transmissão do ID das etiquetas durante a execução do protocolo anticolisão. Os protocolos baseados em ALOHA não requerem que as etiquetas transmitam os seus IDs durante a sua execução. Já os protocolos baseados em árvore, em geral, requerem que as etiquetas transmitam o seu ID. Em alguns protocolos dessa classe, como o QT, os IDs são utilizados na construção da árvore. Essa diferença foi o principal motivo que levou a proposição neste trabalho de um esquema de autenticação mútua para uso com protocolos anticolisão baseados em árvore.

2.4 RESUMO

Esse capítulo apresentou os fundamentos básicos dos sistemas RFID. Foi apresentada inicialmente a arquitetura desse tipo de sistema, a qual é baseada em leitores e etiquetas. Também na parte inicial foram detalhadas as duas principais formas de classificar os diferentes tipos de etiquetas existentes. Em segundo lugar, foram observados os principais padrões criados para tentar unificar os sistemas RFID. Em seguida, foram explicadas as duas abordagens nas quais estão divididos os protocolos anticolisão dos sistemas RFID. Por fim, foram detalhadas as 3 principais categorias de protocolos anticolisão baseados em árvore.

CAPÍTULO 3

ATAQUES A SISTEMAS RFID

Este capítulo analisa os principais ataques aos sistemas RFID. Esses ataques ocorrem devido principalmente a duas caracteristicas básicas dos sistemas RFID: a utilização de transmissões via radiofrequência associadas às limitações de *hardware* das etiquetas. A primeira característica faz com que todos os dados transmitidos entre as etiquetas e os leitores possam ser capturados facilmente por uma entidade maliciosa. A segunda característica torna inviável a utilização de primitivas criptográficas como aquelas baseadas na função SHA-256 e no AES.

Os ataques a sistemas RFID podem ser divididos em duas classes: ataques de origem externa e ataques de origem interna. Nas seções a seguir serão descritas as características dos ataques dessas duas classes e como eles podem impedir o funcionamento correto dos sistemas RFID.

3.1 ATAQUES DE ORIGEM EXTERNA

Os ataques de origem externa se caracterizam por serem realizados por entidades maliciosas que estão fora do sistema RFID. Essas entidades não possuem autorização para realizar nenhum tipo de interação com os dispositivos do sistema. As seções a seguir detalham o funcionamento dos principais ataques externos aos sistemas RFID.

3.1.1 Escuta

Os ataques de escuta (*eavesdroping*) [Hancke 2011] atingem qualquer rede ou sistema que utilize transmissão via radiofrequência. Isso ocorre porque os dados trafegam no canal de comunicação sem fio, bastando um receptor configurado na frequência correta para que

se possa capturar os dados transmidos. No caso dos sistemas RFID, dependendo do tipo de etiqueta utilizada, a potência de transmissão é baixa, o que implica que o receptor precisa estar bem próximo do alvo para conseguir capturar as informações.

O objetivo desse tipo de ataque é a obtenção de informações sigilosas que podem ser utilizadas em diversos outros tipos de ataque, como: rastreamento de etiquetas, clonagem e ataques de repetição. A Figura 3.1 ilustra um exemplo de ataque de escuta. Nesse exemplo, o atacante **C** escuta passivamente a comunicação realizada entre o leitor **B** e a etiqueta **A**, tendo acesso a quaisquer mensagens transmitidas.

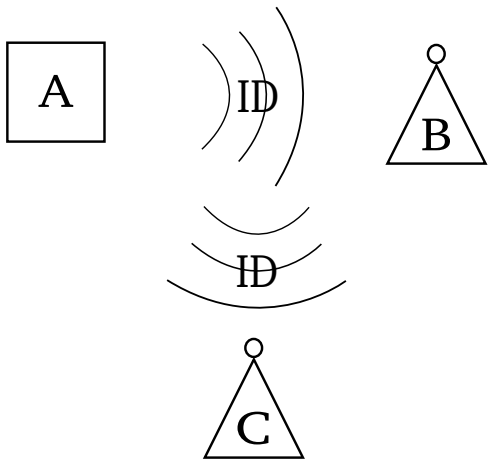


Figura 3.1 Exemplo de ataque de escuta.

3.1.2 Rastreamento de Etiquetas

O ataque de rastreamento (*tracking*) [Chothia e Smirnov 2010] utiliza informações específicas de determinada etiqueta[1], em geral o ID, para identificar sem a devida autorização a presença da mesma um determinado ambiente físico. Em sistemas RFID que utilizam protocolos anticolisão baseados em árvore, o ID das etiquetas é, em geral, transmitido em claro durante a execução desses protocolos [Klair, Chin e Raad 2010]. Assim sendo, a utilização desse tipo de protocolo sem uma proteção específica facilita o rastre-

[1]Dependendo de como o sistema RFID é utilizado, o atacante também pode ter interesse em realizar o rastreamento de um leitor, caso isso traga informações relevantes.

amento das etiquetas [Franklin e Gonçalves 2010].

O rastreamento pode ser feito de forma passiva, na qual o atacante apenas escuta as transmissões entre a etiqueta e o leitor, ou de forma ativa, na qual o atacante finge ser um leitor autorizado, caso isso seja possível. O rastreamento tem como objetivo monitorar o comportamento de uma etiqueta, identificando a trajetória realizada pelo objeto a ela associado. A Figura 3.2 ilustra um exemplo de ataque de rastreamento. Nesse exemplo, as entidades maliciosas **C1** e **C2** conseguem identificar que a etiqueta **A** esteve presente tanto no **local 1** quanto no **local 2**. Isso foi feito através da captura das transmissões entre a etiqueta e os leitores **B** e **D**.

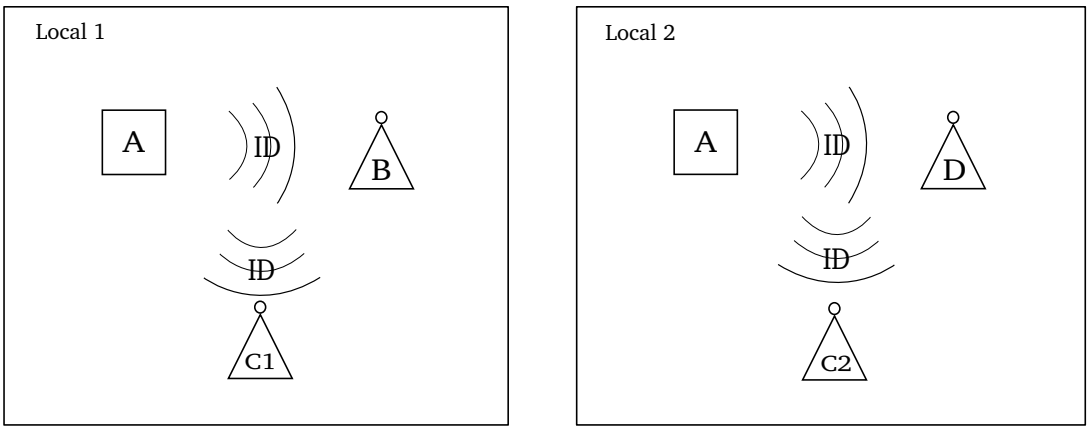


Figura 3.2 Exemplo de ataque de rastreamento

Existe uma variação mais específica desse tipo de ataque conhecida como *hotlisting* [Sun e Ting 2009]. Nessa variação, o rastreamento é feito associado a uma lista de palavras-chave, ou seja, somente o comportamento das etiquetas que esteja associada a uma das palavras-chave presentes na lista será monitorado. Essa lista pode ser composta, por exemplo, por placas de carro de uma locadora de veículos e, através desse ataque, um concorrente poderia identificar quais clientes alugam determinados carros e para onde esses carros se locomovem, invadindo a privacidade dos usuários.

3.1.3 Ataques de Repetição

Os ataques de repetição ou ataques de *replay* [Heydt-Benjamin et al. 2009] também se utilizam de mensagens capturadas pela escuta. Essas mensagens são armazenadas e utilizadas posteriormente pelo atacante como se fossem mensagens autorizadas enviadas por algum dos componentes do sistema. O atacante pode, por exemplo, capturar as mensagens de identificação de uma determinada etiqueta e utilizá-las para tentar conseguir autenticação para acessar o sistema se passando pela etiqueta. A Figura 3.3 ilustra um exemplo de ataque de repetição. Nesse exemplo, o atacante **C** primeiramente captura o **ID** transmitido pela etiqueta **A**. Em seguida, **C** se passa por **A** reenviando o **ID** para o leitor **B**.

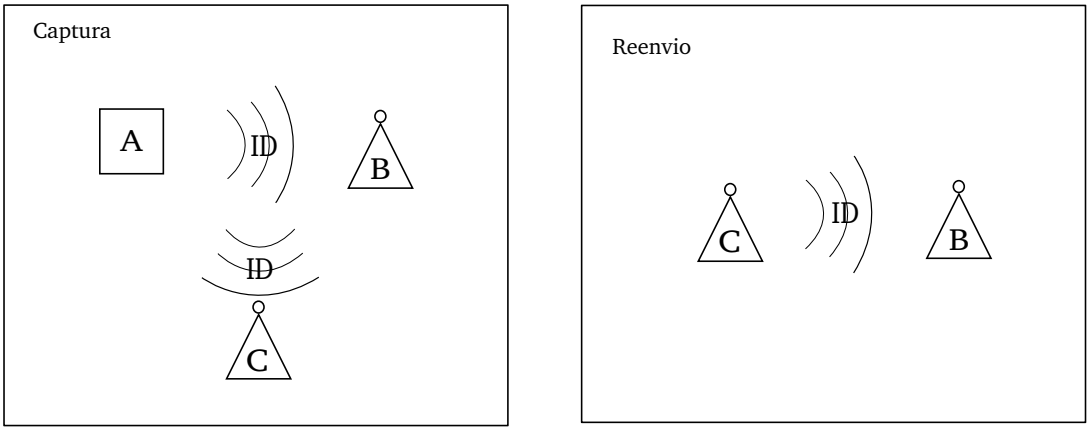


Figura 3.3 Exemplo de ataque de repetição

3.1.4 Clonagem de Dispositivos do Sistema

O ataque de clonagem [Juels 2006, Sun e Ting 2009] pode ser considerado um dos mais prejudiciais a um sistema RFID. Para realizar esse ataque, primeiramente uma entidade maliciosa precisa obter os dados de uma etiqueta ou de um leitor do sistema. Em seguida, essa entidade pode clonar o dispositivo e inserir a cópia no sistema RFID como se fosse uma entidade autorizada.

Os ataques de clonagem se diferenciam dos ataques de repetição por utilizarem os

dados de dispositivos reais para criarem dispositivos falsos, os quais são inseridos no sistema e tentam mimetizar o comportamento dos dispositivos verdadeiros. Nos ataques de repetição, não existe a criação de nenhum dispositivo falso, apenas a reutilização de mensagens capturadas anteriormente.

A Figura 3.4 ilustra um exemplo de um ataque de clonagem. Nesse exemplo, o atacante **C** primeiramente captura o ID transmitido pela etiqueta **A**. Em seguida, ele usa o ID para clonar a etiqueta **A**, criando uma etiqueta A'.

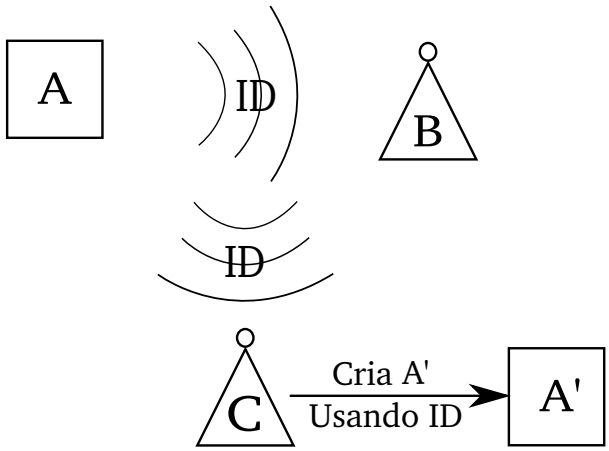


Figura 3.4 Exemplo de ataque de clonagem.

Caso uma etiqueta seja clonada os protocolos anticolisão baseados em árvore poderão ter problemas para identificar a mesma. Isso ocorre, pois caso a etiqueta real e a clonada estejam dentro do alcance de leitura do leitor ao mesmo tempo sempre ocorrerá colisão de identificadores entre as duas. Além disso, a etiqueta clonada pode ser utilizada para a realização de ataques internos, nos quais códigos maliciosos são inseridos por dispositivos da própria rede. Os ataques internos serão detalhados posteriormente. No caso da clonagem de um leitor, o atacante pode utilizar o clone para acessar os dados de qualquer etiqueta a que o mesmo possua acesso.

3.1.5 Negação de Serviço

Os ataques de negação de serviço ou *denial of service* (DoS) [Mitrokotsa, Rieback e Tanenbaum 2010] têm como objetivo impedir que os

usuários legítimos consigam utilizar o sistema. No caso dos sistemas RFID existem pelo menos 2 ataques distintos que podem ser considerados ataques de negação de serviço.

O primeiro ataque é conhecido como *RF jamming* [Fu, Zhang e Wang 2010]. Essa forma de realizar negação de serviço é inerente a qualquer rede ou sistema que utilize transmissão via radiofrequência. Para realizar esse ataque basta que uma entidade maliciosa insira ruído no canal de comunicação com potência suficiente para atrapalhar ou impedir transmissões entre os dispositivos legítimos do sistema.

O segundo ataque utiliza um mecanismo chamado *blocker tag* [Juels, Rivest e Szydlo 2003]. Esse mecanismo é uma etiqueta especial que cria uma região física na qual mesmo leitores autorizados não conseguem realizar consultas a etiquetas do sistema. Originalmente a *blocker tag* foi criada para impedir que uma ou mais etiquetas fossem lidas sem a autorização do usuário. No entanto, o mal uso desse mecanismo pode bloquear a leitura de qualquer etiqueta que se encontre num dado ambiente, configurando um ataque de negação de serviço. A Figura 3.5 ilustra um exemplo de um ataque de negação de serviço utilizando a *blocker tag*. Nesse exemplo, a *blocker tag* **C** é utilizada por um atacante para criar uma área de bloqueio, a qual impede todas as tentativas de comunicação entre a etiqueta **A** e o Leitor **B**.

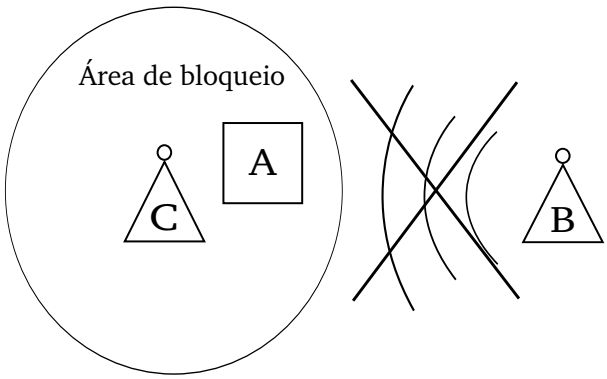


Figura 3.5 Exemplo de ataque de negação de serviço utilizando a blocker tag.

3.1.6 Ataque de Homem no Meio (Man-in-the-middle)

Outro ataque inerente a todas redes ou sistemas que usem transmissão via radiofrequência é o ataque de homem no meio (ou *man-in-the-middle*) [Galluccio, Morabito e Catania 2011, Dimitriou 2005]. Nos sistemas RFID, o atacante se posiciona numa posição intermediária entre um leitor e uma etiqueta que esteja fora do alcance de leitura do mesmo. Dessa posição, o atacante consegue capturar e repassar as informações transmitidas tanto pelo leitor como pela etiqueta. Com isso, ele consegue enganar ambos os componentes e ter acesso a informações privilegiadas, incluindo até acesso válido ao sistema. A Figura 3.5 ilustra um exemplo de um ataque de homem no meio. Nesse exemplo, o atacante **C** se posiciona entre a etiqueta **A** e o leitor **B**, capturando e retransmitindo as informações de ambos os componentes do sistema.

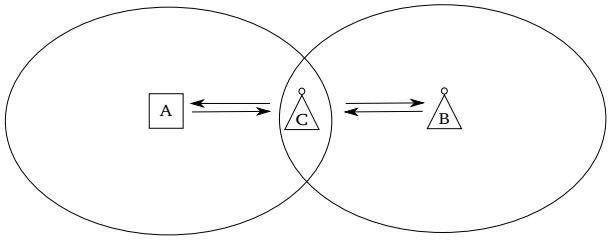


Figura 3.6 Exemplo de ataque de homem no meio.

3.1.7 Ataque de Dessincronização

Ataques de dessincronização [Deursen e Radomirovic 2008] têm como alvo sistemas RFID que utilizem mecanismos de autenticação que exijam a atualização sincronizada do valor de determinadas variáveis presentes tanto nos leitores como nas etiquetas. Nesse tipo de ataque, a entidade maliciosa realiza a dessincronização do sistema bloqueando ou adulterando o conteúdo das mensagens responsáveis pela atualização do valor das variáveis sincronizadas. Para conseguir realizar o bloqueio e a adulteração das mensagens esse tipo de ataque deve utilizar um ataque de negação de serviço ou um ataque de homem no meio [Deursen e Radomirovic 2008].

3.1.8 Acesso Físico aos Dispositivos do Sistema

Um atacante pode acessar fisicamente um dispositivo do sistema com o objetivo de obter as informações privilegiadas que aquele dispositivos possui [Myneni, Misra e Xue 2011]. As etiquetas RFID são mais vulneráveis a esse tipo de ataque por possuírem um *hardware* mais limitado. Essa limitação torna inviável a inclusão de algum tipo de mecanismo que proteja as informações contidas nas mesmas caso esse tipo de ataque seja realizado. Um algoritmo criptográfico é um exemplo de mecanismo que poderia ser utilizado para proteger as informações, caso houvesse poder computacional suficiente. As informações obtidas através desse ataque permitem a realização de outros ataques como, por exemplo, o de clonagem.

3.2 ATAQUES DE ORIGEM INTERNA

Os ataques de origem interna são um subgrupo de ataques aos sistemas RFID que são realizados por dispositivos da própria rede, diferentemente dos ataques anteriores que são realizados por entidade externas. Os atacantes internos são, geralmente, dispositivos falsos que foram criados a partir das informações obtidas num ataque externo e posteriormente inseridos no sistema.

Os ataques de origem interna a sistemas RFID são variações de ataques já bastante difundidos nas redes cabeadas, como [Mitrokotsa, Rieback e Tanenbaum 2010]: *buffer overflow*, injeção de código e injeção SQL. Em [Rieback, Crispo e Tanenbaum 2006] é demonstrado como esses ataques podem ser utilizados na construção de alguns tipos vírus e *worms* para sistemas RFID.

Apesar de alguns desses ataques serem considerados ultrapassados, ainda é possível utilizá-los contra sistemas RFID. Isso ocorre, porque nesse tipo de sistema o maior receio é em relação à segurança contra ataques externos e em geral não se espera que dispositivos internos sejam capazes de prejudicar o sistema. Nas seções a seguir é detalhado o funcionamento dos principais ataques internos contra os sistemas RFID.

3.2.1 Buffer Overflow

O *buffer overflow* [Bishop et al. 2012] é a fonte de falhas mais comum em códigos de programação. Isso ocorre porque esse tipo de vulnerabilidade está presente no mal uso de códigos escritos em linguagens de programação que não possuem proteção de memória adequada, como é o caso de C e C++ [Rieback, Crispo e Tanenbaum 2006]. Além disso, a popularização do *buffer overflow* se deu graças à sua utilização na disseminação de pragas conhecidas como o Vírus de *Morris* (1998), o *Code Red* (2001) e o *SQL Slammer* (2003) [Rieback, Crispo e Tanenbaum 2006].

Para realizar o *buffer overflow*, o atacante precisa estourar o limite de memória de um dos *buffers* utilizados no código. Isso é alcançado explorando códigos desprotegidos que utilizem funções que não fazem checagem de memória, como *strcpy, strlen, strcat, sprintf* e *gets* da linguagem C. A partir do estouro do *buffer*, o atacante consegue injetar comandos que são executados sem o conhecimento do usuário. Através desses comandos, é possível até mesmo conseguir acesso a um *shell* e obter controle de partes do sistema [Bishop et al. 2012].

No caso dos sistemas RFID, a limitação de memória das etiquetas em 1024 *bits* ou menos, intuitivamente impediria um ataque desse tipo a *buffers* maiores que esse tamanho. No entanto, existem comandos como o "*write multiple blocks*" do padrão ISO-15963 que permitem a etiqueta enviar cópias do mesmo bloco várias vezes, o que poderia acarretar em um *buffer overflow* no nível de aplicação. Além disso, também seria possível realizar o ataque em um *buffer* do *middleware* RFID utilizando um dispositivo de simulação de etiquetas, como o *RFID Guardian* [Mitrokotsa, Rieback e Tanenbaum 2010].

3.2.2 Injeção de Código

A injeção de código malicioso [Mitrokotsa, Rieback e Tanenbaum 2010] é realizada geralmente contra aplicações baseadas em linguagens de *script*, como *java*, *javascript* e *Perl* [Rieback, Crispo e Tanenbaum 2006]. Esse tipo de ataque exige uma preparação prévia, com o atacante forjando uma URL falsa e utilizando engenharia social para ten-

tar convencer o usuário a clicar no *link* forjado. Caso o *script* seja ativado ele injetará o código desejado com o objetivo de tomar o controle da máquina atingida.

No caso dos sistemas RFID, esse tipo de ataque só é útil caso os dados das etiquetas sejam escritos em linguagens de *script*. Além disso, o *middleware* dos leitores deve ser capaz de interpretá-los, pois caso contrário não será possível realizar a injeção de código. Caso essa condição seja cumprida, o ataque funciona da mesma forma que em um navegador de Internet convencional.

3.2.3 Injeção SQL

O ataque de injeção SQL (*Structured Query Language*) [Kieyzun et al. 2009] tem como principal alvo os bancos de dados das aplicações que utilizem a linguagem de consulta SQL. A falta de proteção contra consultas mal feitas e o mal uso da linguagem SQL são as principais falhas que permitem a realização desse tipo de ataque.

Para que o ataque tenha sucesso, basta identificar um sistema de consultas desprotegido e realizar consultas mal formadas que incluam, por exemplo, a pontuação ";" no início da consulta, como em "*;shutdown–*" [Rieback, Crispo e Tanenbaum 2006]. As consequências de um ataque desse tipo vão desde o acesso a dados sigilosos do banco de dados até a inutilização do banco inteiro.

Para que esse tipo de ataque funcione em sistemas RFID, o *middleware* do banco de dados do sistema deve suportar a realização de consultas SQL e o sistema de consultas deve ser identificado como desprotegido. Caso esses requerimentos sejam cumpridos, o atacante precisa apenas incluir consultas mal formadas no conteúdo de uma das etiquetas para injetar códigos SQL maliciosos no sistema.

3.3 RESUMO

Nesse capítulo foram apresentados os principais ataques que exploram as vulnerabilidades dos sistemas RFID. Os ataques a esse tipo de sistema podem ser classificados em dois grandes grupos: externos e internos. Os externos partem de entidades que não fazem

parte do sistema e que, em geral, estão tentando obter acesso ou informações sobre esse sistema. Os internos partem de dispositivos que fazem parte do próprio sistema. Os ataques internos, em geral, têm como objetivo obter informações do banco de dados do sistema ou danificá-lo de alguma forma.

CAPÍTULO 4

ESQUEMAS DE AUTENTICAÇÃO PARA SISTEMAS RFID

Este capítulo apresenta as principais propostas de esquemas que lidam com requisitos de autenticação e anonimato em sistemas RFID. Essas propostas podem ser divididas em três classes distintas [Misra et al. 2009]: as baseadas em funções *hash* [Dimitriou 2005, Lee et al. 2005, Weis et al. 2004, Yang et al. 2005]; as baseadas em algoritmos criptográficos [Dimitriou 2006, Dominikus, Oswald e Feldhofer 2005, Feldhofer, Dominikus e Wolkerstorfer 2004, Feldhofer e Rechberger 2006] e; as baseadas em operações lógicas e bit a bit [Peris-Lopez et al. 2006, Misra et al. 2009, Myneni, Misra e Xue 2011].

As propostas da primeira classe se baseiam em funções *hash* como: SHA-1, SHA-256, MD4 e MD5. Um problema das propostas dessa classe é a utilização de um número significativo de portas lógicas e de ciclos de relógio. Em particular, a função MD4 é a menos custosa entre as citadas e requer 7.350 portas lógicas e 456 ciclos de relógio para que possa ser implementada [Feldhofer e Rechberger 2006]. As propostas da segunda classe se baseiam em algoritmos criptográficos como o AES, o qual é capaz de prover um nível adequado de segurança ao sistema. As propostas dessa classe também consomem uma quantidade significativa de postas lógicas e ciclos de relógio. Em [Feldhofer e Rechberger 2006], por exemplo, a implementação mais otimizada do AES requer 3.400 portas lógicas e 1.032 ciclos de relógio.

Os mecanismos baseados em funções lógicas e *bit-a-bit* apresentam um custo computacional mais modesto, pois as operações e algoritmos por eles utilizados são mais simples. Essa classe de mecanismos é a que melhor se adapta em sistemas que utilizam etiquetas

passivas.

Nas seções a seguir são detalhadas as características e os esquemas de autenticação mais relevantes de cada uma dessas classes.

4.1 ESQUEMAS BASEADOS EM FUNÇÕES HASH

As soluções baseadas em funções *hash* propostas em [Dimitriou 2005, Lee et al. 2005, Weis et al. 2004, Yang et al. 2005] requerem que as etiquetas sejam capazes de utilizar funções como a SHA-1 (*Secure Hash Algorithm-1*), a SHA-256 (*Secure Hash Algorithm-256*), e a MD5 (*Message-Digest algorithm 5*). Esse tipo de função recebe um bloco de tamanho fixo como entrada e quebra esse bloco em partes menores de tamanhos iguais. As partes menores são comprimidas, gerando uma saída aleatória de tamanho fixo. No caso da *SHA-1*, a saída é de 160 *bits*. A saída da *SHA-256* é de 256 *bits* e a saída do *MD5* é de 128 *bits*.

Em [Feldhofer e Rechberger 2006] é demonstrado que as funções *hash* existentes exigem entre 7350 e 10868 portas lógicas para funcionar. Assim sendo, o uso desse tipo de função em etiquetas passivas, as quais possuem no máximo 4000 portas lógicas para essa finalidade, é até o presente momento, inviável. No entanto, nada impede o uso desse tipo de solução em sistemas RFID com etiquetas semi-passivas e ativas, pois esses dois tipos de etiquetas possuem um poder computacional mais elevado.

Nas seções a seguir são apresentadas as propostas mais relevantes de esquemas de autenticação para RFID baseados nesse tipo de função.

4.1.1 Notação

Para facilitar as explicações sobre os esquemas baseados em funções *hash* serão utilizadas duas notações básicas $h(M)$ e $h_k(M)$. A notação $h(M)$ indica a aplicação da função *hash* $h()$ a um valor qualquer M. Já a notação $h_k(M)$ indica a aplicação a um valor qualquer M da função *hash* chaveada $h_k()$, que produz a saída de acordo com valor da chave k.

4.1.2 Hash-Based Access Control

Uma das primeiras ideias para o uso de funções *hash* na segurança de sistemas RFID é o esquema de controle de acesso baseado em funções *hash* [Weis et al. 2004]. Esse esquema permite que o administrador do sistema utilize uma chave aleatória para colocar suas etiquetas em estado *locked* (trancado). Nesse estado, as etiquetas passam a responder a todas as requisições com um *metaID* e só voltam ao estado original caso recebam a chave correta. Esse não é um esquema de autenticação propriamente dito, mas o leitor realiza uma espécie de autenticação perante a etiqueta para colocá-la de volta ao estado *unlocked* (destrancado).

Para trancar uma etiqueta com esse esquema, o leitor seleciona uma chave aleatória x e fornece essa chave como entrada para uma função *hash*. O valor da saída dessa função é enviado para a etiqueta e é utilizado como o *metaID*. A partir do recebimento do *metaID*, a etiqueta entra em estado trancado.

Para destrancar a etiqueta, o leitor primeiramente envia uma mensagem qualquer para a mesma e recebe como resposta o *metaID* atual. Esse *metaID* é enviado para o banco de dados, o qual retorna a chave x correspondente ao leitor. O leitor envia x para a etiqueta que calcula o valor de $hash(x)$ e verifica se ele é correspondente ao seu *metaID* atual. Em caso afirmativo, a etiqueta volta para o estado *unlocked* (destrancado). Caso contrário, a etiqueta responde novamente com o seu *metaID*.

Esse esquema possui uma desvantagem, a qual ocorre enquanto a etiqueta está destrancada. Nesse momento, um atacante pode simplesmente consultar a etiqueta e obter o seu ID sem nenhuma dificuldade. Outro problema é que os autores desse esquema o desenvolveram para que ele fosse utilizado em etiquetas passivas. No entanto, posteriormente foi comprovado que a utilização de esquemas baseados em funções *hash* não podem ser utilizados nesse tipo de etiqueta devido ao seu custo alto em termo de portas lógicas e ciclos de relógio [Feldhofer e Rechberger 2006] como visto na Seção 4.1.

4.1.3 Dimitriou's Lightweight Hash-based Scheme

Em [Dimitriou 2005] é proposto um esquema de autenticação mútua para sistemas RFID baseado em funções *hash*. Nesse esquema, a cada rodada de leitura, a etiqueta atualiza o seu ID em sincronia com o servidor, o qual é utilizado como um segredo pré-compartilhado. Essa característica permite que o sistema se mantenha seguro mesmo que o atacante consiga em uma determinada rodada i ter acesso ao ID_i de uma etiqueta, pois na próxima rodada essa informação não poderá ser mais utilizada.

O processo de autenticação é iniciado pelo leitor R que gera e transmite para a etiqueta T um *Nonce* N_R, o qual é um número aleatório que identifica o leitor durante essa rodada. A etiqueta gera um outro *Nonce* N_T, calcula $h(ID_i)$ e $h_{ID_i}(N_T, N_R)$ e envia esses três valores para o leitor. O leitor finaliza essa etapa inicial enviando N_T, N_R, $h(ID_i)$ e $h_{ID_i}(N_T, N_R)$ para o servidor S.

O servidor utiliza $h(ID_i)$ para encontrar o valor armazenado de ID_i e utiliza N_T, N_R e ID_i para recalcular o valor de $h_{ID_i}(N_T, N_R)$. Caso os valores sejam iguais, a etiqueta é autenticada e o servidor atualiza ID_i para ID_{i+1}. Esse valor é utilizado para calcular $h_{ID_{i+1}}(N_T, N_R)$ que é enviado para o leitor e repassado para a etiqueta. A etiqueta utiliza a mesma operação feita pelo servidor para atualizar ID_i para ID_{i+1} e recalcula o valor de $h_{ID_{i+1}}(N_T, N_R)$. Caso o valor esteja correto o processo de autenticação está concluído.

Esse esquema é vulnerável a um ataque de dessincronização. Nesse ataque, caso o atacante consiga de alguma forma bloquear o recebimento de $h_{ID_{i+1}}(N_T, N_R)$, no momento da transmissão entre o leitor e a etiqueta ocorre a dessincronização do ID_i, pois o servidor já terá atualizado o ID_i da etiqueta no seu banco de dados e a etiqueta ainda não terá atualizado o seu ID_i. Assim sendo, em uma próxima rodada o servidor não irá conseguir autenticar essa etiqueta. Além disso, esse mecanismo exige que a etiqueta realize dois cálculos de função *hash*, os quais consomem uma quantidade grande de portas lógicas e ciclos de relógio como visto na Seção refsec:hash. Essa exigência pode inviabilizar o uso desse mecanismo até mesmo em etiquetas com maior poder computacional.

4.1.4 LCAP: Low-Cost RFID Authentication Protocol

O LCAP (*Low-Cost RFID Authenticantion Protocol*) [Lee et al. 2005] também é um esquema de autenticação mútua para sistemas RFID baseado em funções *hash*. O LCAP se difere do esquema de *Dimitriou* por possuir um mecanismo específico para evitar a dessincronização no caso da perda de mensagens. Esse mecanismo armazena tanto o ID anterior quanto o atual, assim caso o servidor atualize o seu ID e a etiqueta não receba a mensagem de atualização, os dois podem se ressincronizar.

No LCAP o processo de autenticação é iniciado pelo leitor R que envia um valor aleatório r para a etiqueta T. A etiqueta, por sua vez, calcula os valores de $h(ID)$, o qual é armazenado numa váriavel chamada $HaID$, e $h(ID||r)$. $HaID$ e a metade mais a esquerda de $h(ID||r)$, a qual é denominada $h_L(ID||r)$, são enviados de volta para o leitor que os repassa juntamente com r para o servidor S.

O servidor procura no banco de dados o ID da etiqueta que possua o $HaID$ igual ao valor recebido e calcula novamente $h(ID||r)$. Em seguida, ele verifica se o valor $h_L(ID||r)$ recebido está correto, e em caso afirmativo a etiqueta é autenticada. Após a autenticação, o servidor atualiza os valores atuais do ID para $ID = ID \oplus r$ e de $HaID$ para $HaID = h(ID \oplus r)$. Por fim, ele envia para o leitor o valor da metade mais à direita de $h(ID||r)$, o qual é denominado $h_R(ID||r)$.

Para finalizar a autenticação, o leitor repassa o valor de $h_R(ID||r)$ para a etiqueta, que compara esse valor com aquele que ela havia calculado no início do processo. Caso eles sejam iguais, a etiqueta atualiza seu ID e seu $HaID$ para os mesmos valores que o servidor atualizou, mantendo o sistema sincronizado.

Apesar de afirmar que o LCAP poderia suportar ressincronização, o trabalho não explica de que forma seria feito esse processo. Além disso, o LCAP também exige que as etiquetas efetuem duas operações com funções *hash*, o que o torna tão custoso quanto o esquema de Dimitriou em termos de portas lógicas e ciclos de relógio.

4.2 ESQUEMAS BASEADOS EM ALGORITMOS CRIPTOGRÁFICOS

Os esquemas de autenticação baseados em algoritmos criptográficos para RFID são, em sua grande maioria, adaptações simplificadas de mecanismos de segurança amplamente utilizados em outros sistemas computacionais. As diferenças das versões originais para as simplificadas se restringem a modificações e otimizações no funcionamento interno dos algoritmos criptográficos. Assim sendo, a troca de mensagens dos esquemas simplificados é praticamente a mesma dos originais. O esquema de criptografia simétrica é o que tem sido mais utilizado nessas adaptações [Dimitriou 2006, Dominikus, Oswald e Feldhofer 2005, Feldhofer 2004, Feldhofer, Dominikus e Wolkerstorfer 2004, Feldhofer e Rechberger 2006].

Em [Dimitriou 2006] é proposto um esquema de autenticação mútua baseado em criptografia simétrica. Esse esquema utiliza um PRNG (*Pseudo Random Number Generator*) e uma PRF (*Pseudo Random Function*) para realizar o processo de cifragem. O esquema é seguro, mas exige que as etiquetas realizem operações que consomem no mínimo 3.400 portas lógicas e 1.032 ciclos de relógio [Feldhofer e Rechberger 2006]. Assim sendo, o seu uso em sistemas RFID que utilizam etiquetas passivas é praticamente inviável.

Em [Dominikus, Oswald e Feldhofer 2005, Feldhofer 2004, Feldhofer, Dominikus e Wolkerstorfer 2004] são proposto esquemas de segurança, que também utilizam autenticação baseada em criptografia simétrica. Nesses esquemas, as operações para manter a segurança nas etiquetas são feitas pelo AES (*Advanced Encryption Standard*) [Daemen e Rijmen 1998]. O AES foi implementado por esses esquemas na sua forma pura, sem simplificações. Isso significa que esses esquemas também consomem uma grande quantidade de recursos computacionais e a aplicação dos mesmos na prática só é possível em sistemas RFID com etiquetas ativas na melhor das hipóteses.

Já em [Feldhofer e Rechberger 2006] é apresentada uma proposta que utiliza uma versão mais simplificada do AES. Ainda assim, são necessárias no mínimo 3400 portas lógicas para utilizar essa implementação, o qual ainda é um custo computacional elevado para sistemas RFID com etiquetas passivas. Além disso, a quantidade de ciclos de

relógio utilizada por esse esquema é de 1034 ciclos, a qual é muito superior aos 220 ciclos suportados por etiquetas passivas [Myneni, Misra e Xue 2011].

4.3 ESQUEMAS BASEADOS EM FUNÇÕES LÓGICAS E BIT-A-BIT

Como dito anteriormente, as etiquetas passivas possuem no máximo 4.000 portas lógicas para uso dedicado dos mecanismos de segurança e a quantidade máxima de ciclos de relógio utilizada por tais mecanismos está limitada em 220 [Myneni, Misra e Xue 2011, Misra et al. 2009]. Por causa disso, surgiram propostas baseadas em operações lógicas e bit a bit [Peris-Lopez et al. 2006, Misra et al. 2009, Myneni, Misra e Xue 2011]. Essas propostas buscam prover anonimato e autenticação utilizando mecanismos menos custosos em termos de portas lógicas e ciclos de relógio para viabilidade de uso em etiquetas passivas. As seções a seguir detalham o funcionamento dos principais esquemas de segurança dessa classe.

4.3.1 M^2AP: Minimalist Mutual-Authentication Protocol

O M^2AP (*Minimalist Mutual-Authentication Protocol*) [Peris-Lopez et al. 2006] é um protocolo de autenticação mútua etiqueta-leitor que busca preservar o anonimato das etiquetas. Nesse esquema, tanto a etiqueta se autentica perante o leitor quanto o leitor se autentica perante a etiqueta. No M^2AP, a etiqueta possui dois *IDs*: o primeiro é o ID real e o segundo é conhecido como IDS (*index-pseudonym*). Este último é um pseudônimo que funciona como um índice e permite ao leitor encontrar numa tabela do banco de dados onde estão armazenadas as informações da etiqueta correspondente. Além disso, a etiqueta também compartilha com o leitor um conjunto de quatro chaves secretas ($K1$, $K2$, $K3$ e $K4$), as quais são utilizadas para fornecer segurança ao processo de autenticação mútua etiqueta-leitor.

O M^2AP considera que qualquer comunicação entre leitores e etiquetas pode ser capturada, mas que o canal entre os leitores e o banco de dados é seguro. Além disso, esse mecanismo segue o conceito de criptografia minimalista criado por *Juels* [Juels 2006].

Assim sendo são utilizadas somente as seguintes operações bit-a-bit: XOR ($\oplus$), OR ($\vee$), AND ($\wedge$), e soma módulo m ($+$).

4.3.1.1 Funcionamento do M^2AP

O M^2AP pode ser dividido em quatro estágios: identificação da etiqueta, autenticação mútua, atualização do IDS e atualização das chaves de segurança.

Identificação da Etiqueta

Nesse primeiro estágio, o leitor envia uma mensagem de *hello* para a etiqueta e recebe como resposta o IDS atual da mesma. O leitor utiliza o IDS para buscar no banco de dados as quatro chaves de segurança daquela etiqueta. Caso o IDS não corresponda a algum conjunto de chaves no banco de dados, o M^2AP encerra o procedimento.

Autenticação Mútua

Primeiramente, é realizada a autenticação do leitor perante a etiqueta. Para iniciar esse processo, o leitor gera dois números aleatórios $n1$ e $n2$ que são utilizados juntamente com o IDS e as chaves de segurança para construir as mensagens A, B e C. Essa mensagens são contruídas, respectivamente, através das Equações (4.1), (4.2) e (4.3):

$$A = IDS_{tag(i)}^{(n)} \oplus K1_{tag(i)}^{(n)} \oplus n1, \tag{4.1}$$

$$B = (IDS_{tag(i)}^{(n)} \wedge K2_{tag(i)}^{(n)}) \vee n1, \tag{4.2}$$

$$C = IDS_{tag(i)}^{(n)} + K3_{tag(i)}^{(n)} + n2. \tag{4.3}$$

A etiqueta recebe as três mensagens e utiliza A e B para realizar a autenticação do leitor. Primeiramente, a etiqueta calcula $n1$ a partir de A. Em seguida, a etiqueta utiliza o valor obtido para $n1$ e recalcula B, comparando o valor obtido com o recebido anteriormente. Caso os valores sejam iguais, o leitor é autenticado.

A partir da mensagem C, a etiqueta é capaz de obter o valor de $n2$. Em seguida, a etiqueta constrói as mensagens D e E, respectivamente, através das Equações (4.4) e (4.5):

$$D = (IDS_{tag(i)}^{(n)} \vee K4_{tag(i)}^{(n)}) \wedge n2, \tag{4.4}$$

$$E = (IDS_{tag(i)}^{(n)} + ID_{tag(i)}) \oplus n1, \tag{4.5}$$

essas mensagens são enviadas para o leitor com o objetivo de autenticar a etiqueta.

O leitor recebe as mensagens D e E e recalcula os valores das duas, verificando através de D se a etiqueta está realmente autorizada. Caso os valores correspondam, o estágio de autenticação mútua é concluído e o leitor obtém o ID da etiqueta através da mensagem E.

Atualização do IDS

A cada rodada de autenticação, é necessário que tanto a etiqueta quanto o banco de dados atualizem os valores de seu IDS para evitar o rastreamento da etiqueta. Para esse fim, é utilizada a equação:

$$IDS_{tag(i)}^{(n+1)} = (IDS_{tag(i)}^{(n)} + (n2 \oplus n1)) \oplus ID_{tag(i)}. \tag{4.6}$$

Atualização das Chaves de Segurança

O M^2AP também prevê a atualização das quatro chaves de segurança a cada rodada de autenticação. Isso é feito através das seguintes equações:

$$K1_{tag(i)}^{(n+1)} = K1_{tag(i)}^{(n)} \oplus n2 \oplus (K3_{tag(i)}^{(n)} + ID_{tag(i)}), \tag{4.7}$$

$$K2_{tag(i)}^{(n+1)} = K2_{tag(i)}^{(n)} \oplus n2 \oplus (K4_{tag(i)}^{(n)} + ID_{tag(i)}), \tag{4.8}$$

$$K3_{tag(i)}^{(n+1)} = (K3_{tag(i)}^{(n)} \oplus n1) + (K1_{tag(i)}^{(n)} \oplus ID_{tag(i)}), \tag{4.9}$$

$$K4_{tag(i)}^{(n+1)} = (K4_{tag(i)}^{(n)} \oplus n1) + (K2_{tag(i)}^{(n)} + ID_{tag(i)}). \quad (4.10)$$

4.3.1.2 Considerações Finais Sobre o M^2AP

Em [Barasz 2007] é mostrado um ataque contra o M^2AP. Esse ataque permite a uma entidade maliciosa ter acesso às chaves de segurança utilizadas e ao ID real da etiqueta. Para isso, basta que o atacante consiga capturar poucas rodadas de trocas de mensagens entre uma etiqueta-alvo e o leitor durante a execução do M^2AP. O acesso a essas informações permite a um atacante clonar essa etiqueta.

4.3.2 SEAS: A Secure and Efficient Anonymity Scheme

Em [Misra et al. 2009] é apresentado um esquema de autenticação mútua etiqueta-leitor conhecido como SEAS (*Secure and Efficient Anonymity Scheme*). Nesse esquema, cada etiqueta T_j possui, além do identificador X_j, uma chave S_j exclusiva, a qual é pré-compartilhada com o servidor central S. Todo o processo de autenticação das etiquetas feito pelo SEAS é baseado nessas chaves. Além disso, o esquema utiliza operações bit-a-bit de deslocamento à esquerda ($<<$) e a geração de números pseudo-aleatórios nas etiquetas no processo de autenticação.

No SEAS, cada leitor R_i do sistema também possui uma chave pré-compartilhada com o servidor S, o qual também é conhecido como R_i. Essa chave é utilizada tanto no processo de autenticação das etiquetas quanto na verificação de autenticidade do leitor perante o servidor S.

Assim como o M^2AP, o modelo de ameaça do SEAS também considera que o canal entre os leitores e o servidor S é seguro. Esse modelo também considera que o canal entre os leitores e etiquetas é inseguro, o que implica que qualquer mensagem transmitida entre um leitor e uma etiqueta pode ser capturada.

4.3.2.1 Funcionamento do SEAS

Todo o processo de autenticação realizado pelo SEAS é iniciado por um leitor R_i tentando estabelecer comunicação com uma etiqueta T_j. Para isso, o leitor primeiramente gera um número pseudo-aleatório r e o usa para calcular R_1 de acordo com a equação:

$$R_1 = r \oplus R_i. \tag{4.11}$$

Em seguida, R_1 é transmitido para a etiqueta T_j. A etiqueta gera outros dois números pseudo-aleatórios R_2 e R_3 e utiliza R_1, R_2, R_3, o seu identificador X_j e a sua chave pré-compartilhada S_j para calcular R_4, C, D e E e construir a mensagem M de acordo com as equações:

$$R_4 = R_2 << R_3, \tag{4.12}$$

$$C = X_j \oplus R_1 \oplus R_4, \tag{4.13}$$

$$D = R_2 \oplus S_j, \tag{4.14}$$

$$E = R_3 \oplus S_j, \tag{4.15}$$

$$M = C||D||E. \tag{4.16}$$

A mensagem M é enviada para o leitor que a concatena com a identidade do próprio leitor R_i e com r para criar $M^{'}$, a qual é enviada para o servidor S. A primeira operação realizada pelo servidor é verificar se o leitor de identidade R_i é um leitor autorizado. Em caso positivo, o servidor calcula $R_1^{'}$ de acordo com a equação:

$$R_1^{'} r \oplus R_i. \tag{4.17}$$

Para finalizar a autenticação o servidor busca em sua base de dados uma chave $S_j^{'}$, que gere um resultado correto para as seguintes equações:

$$R_2^{'} = D \oplus S_j^{'}, \tag{4.18}$$

$$R_3^{'} = E \oplus S_j^{'}, \tag{4.19}$$

$$R_4^{'} = R_2^{'} << R_3^{'}, \tag{4.20}$$

$$X^{'} = C \oplus R_1^{'} \oplus R_4^{'}, \tag{4.21}$$

onde $X^{'}$ é um ID válido na base dados.

Caso encontre uma chave $S_j^{'}$ correta, o servidor envia uma mensagem de autenticação M_{auth} para o leitor autorizando a troca de mensagens com a etiqueta.

4.3.2.2 Considerações Finais Sobre o SEAS

O SEAS apresenta novos conceitos de segurança de sistemas RFID utilizando a ideia básica de criptografia minimalista. As principais novidades são a utilização de um segredo fixo pré-compartilhado e a geração de números pseudo-aleatórios nas etiquetas. Além disso, ele possui um baixo custo computacional necessitando de apenas 2220 portas lógicas e 44 *clock cycles* para ser utilizado [Misra et al. 2009].

Até o presente momento, não existem ataques que explorem possíveis vulnerabilidades do SEAS. No entanto, o esquema assume que a etiqueta já tenha passado pelo processo anticolisão para iniciar o processo de autenticação e que seu ID real nunca seja transmitido em claro a fim de manter o anonimato da mesma.

4.3.3 SAMA: Serverless Anonymous Mutual Authentication

Em [Myneni, Misra e Xue 2011] é proposto o SAMA (*Serverless Anonymous Mutual Authentication*). O SAMA se propõe a oferecer um esquema de autenticação mútua etiqueta-leitor sem a necessidade de uso de um servidor conectado ao leitor. A vantagem alegada é a não necessidade de haver conexões persistentes entre o leitor e o servidor para a realização do processo de autenticação. Os contras da eliminação do servidor incluem: 1) o aumento do custo do leitor, já que o mesmo precisa armazenar toda a base de informações da aplicação; 2) a necessidade de um mecanismo de sincronização da base de dados em cada leitor do sistema caso informações sobre etiquetas precisem ser adicionadas ou removidas da base; 3) o aumento do consumo de energia do leitor devido

à necessidade de se realizar mais processamento.

O processo de autenticação do SAMA utiliza dois componentes: o NLFSR (*Non-Linear Feedback Shift Register*)[Dubrova, Teslenko e Tenhunen 2008] e uma função de perturbação. O NLFSR é um registrador que realiza deslocamentos e que possui uma função de transição composta por uma quantidade pré-determinada de portas lógicas do tipo XOR e AND. Essa função de transição faz com que cada NLFSR modifique a sua entrada de uma maneira única. No processo de autenticação da etiqueta perante o leitor, a etiqueta utiliza o NLFSR para gerar uma espécie de assinatura que é transmitida para o leitor. Essa assinatura muda toda vez que o processo de autenticação é executado, uma vez que números pseudoaleatórios compõem a entrada para o NLFSR. A assinatura é utilizada pelo leitor para identificar e autenticar a etiqueta sem a necessidade de transmissão do ID real. Maiores detalhes sobre o NLFSR estão disponíveis na Seção 5.2 do Capítulo 5. A função de perturbação é um mecanismo auxiliar utilizado em conjunto com o NLFSR para manter anônimas as mensagens transmitidas durante o processo de autenticação, evitando que um atacante consiga associar as mensagens a uma etiqueta.

A função de perturbação divide a sua entrada em duas metades. Uma das metades é dividida em blocos de 8 *bits* e é enviada para as *s-boxes*. A outra metade passa por uma permutação pré-estabelecida. No final, o resultado das duas metades é concatenado.

4.3.3.1 Funcionamento do SAMA

O SAMA é um mecanismo que funciona em duas etapas. Na primeira, a etiqueta T_i se autentica perante o leitor R_j. Na segunda etapa, o leitor se autentica perante a etiqueta. Essas etapas serão analisadas com mais detalhes nas seções a seguir.

Autenticação da Etiqueta Perante o Leitor

O processo de autenticação da etiqueta T_i é iniciado pelo leitor R_j. Para isso ele gera uma *string* aleatória r_{ji}^1 e a transmite para a etiqueta. A etiqueta T_i recebe r_{ji}^1 e a utiliza como entrada para o seu NLFSR, no qual a *string* passará por uma sequência de

k deslocamentos. A saída do NLFSR é a *string* R^1_{ji}.

A etiqueta gera um número aleatório r^2_{ij} e faz um XOR desse número com a saída do NLFSR R^1_{ji} resultando em um valor x^1_{ij}. Esse valor é utilizado como entrada da função de perturbação que gera como saída a *string* m^1_{ij}. Os valores de m^1_{ij} e r^2_{ij} são concatenados, gerando a mensagem M^1_{ij} que é enviada para o leitor.

O leitor verifica se o valor de M^1_{ij} está correto utilizando r^1_{ji} e r^2_{ij}. Para isso, ele testa todos os NLFSR que ele possui armazenados até que ele encontre o único capaz de gerar um valor de m^1_{ij} que seja igual ao contido na mensagem M^1_{ij}. Quando isso ocorre, o leitor dá início a próxima etapa do processo de autenticação. Caso nenhum NLFSR correspondente seja encontrado, o leitor retransmite r^1_{ji} indicando para a etiqueta que ela não foi autenticada e encerra a comunicação.

Autenticação do Leitor Perante a Etiqueta

O leitor inicia a sua etapa da autenticação utilizando r^2_{ij} como entrada para o NLFSR da etiqueta ao qual ele está se autenticando. Após realizar os k deslocamentos o NLFSR gera a saída R^2_{ji}. O leitor gera um novo número aleatório r^3_{ji} e faz um XOR desse número com R^2_{ji} gerando x^2_{ji}.

O valor x^2_{ji} é utilizado como entrada para a função de perturbação que gera m^2_{ji}. Esse valor é concatenado com r^3_{ji} formando a mensagem M^2_{ji}. Essa mensagem é enviada para etiqueta que verifica a sua validade utilizando r^2_{ij} e r^3_{ji}. Caso M^2_{ji} seja válido, o processo de autenticação do leitor é finalizado. Caso contrário, a etiqueta para de se comunicar com o leitor.

4.3.3.2 Considerações finais sobre o SAMA

O SAMA é um dos mais recentes esquemas de autenticação mútua para sistemas RFID. Apesar de usar novos conceitos que fornecem mais segurança como o NLFSR e a função perturbação, ele ainda consegue ser um esquema leve, utilizando apenas 1393 portas lógicas e 70 *clock cycles*. Ainda não existem ataques conhecidos contra o SAMA. No

entanto, o esquema não leva em consideração como é realizado processo anticolisão, mas requer o sigilo do ID real para manutenção do anonimato das etiquetas.

4.4 RESUMO

Nesse capítulo foram apresentados diversas propostas de esquemas que lidam com requisitos de autenticação e anonimato em sistemas RFID. Essas propostas podem ser classificadas em três classes: baseadas em funções *hash*, baseadas em algoritmos criptográficos e baseadas em funções lógicas e bit-a-bit. Os sistemas RFID que utilizam etiquetas passivas só são capazes de utilizar os esquemas baseados em funções lógicas e bit-a-bit por conta das limitações em termos de portas lógicas e ciclos de relógio inerentes a esse tipo de etiqueta.

CAPÍTULO 5

AMAS - ANONYMOUS MUTUAL AUTHENTICATION SCHEME

Este capítulo apresenta o esquema proposto nesse trabalho, denominado AMAS (*Anonymous Mutual Authentication Scheme*). Este esquema se diferencia dos trabalhos relacionados por poder ser utilizado em conjunto com protocolos anticolisão baseados em árvore[1] e ao mesmo tempo preservar o anonimato das etiquetas. O AMAS, reutiliza mecanismos empregados no processo de autenticação para que as etiquetas gerem IDs aleatórios e temporários a serem utilizados durante a execução do protocolo anticolisão baseado em árvore. Essa abordagem busca garantir o anonimato das etiquetas desde o processo de anticolisão, não permitindo a um atacante correlacionar os IDs aleatórios e temporários com os IDs reais. O AMAS também é projetado com foco em sistemas que utilizam etiquetas passivas. Nas seções a seguir, são apresentados: os modelos de sistema e de ameaça adotados neste trabalho, uma descrição detalhada do funcionamento do AMAS e, por fim, uma análise da segurança e do custo do AMAS.

5.1 MODELO DO SISTEMA E MODELO DE AMEAÇA

Esta seção detalha o modelo do sistema RFID adotado neste trabalho e o modelo de ameaça contra o esquema de autenticação e anonimato proposto.

[1] O AMAS também pode ser utilizado com protocolos anticolisão baseados em ALOHA, desde que sejam feitas as adaptações necessárias para isso.

5.1.1 Modelo do Sistema

O sistema RFID considerado para a proposta desta dissertação é composto por três componentes: um servidor S, leitores e etiquetas. As etiquetas são passivas e possuem memória necessária para armazenar as informações utilizadas pelo esquema proposto. Adicionalmente, as etiquetas possuem um gerador de números pseudoaleatórios que funciona dentro das especificações estabelecidas no padrão EPC Gen 2 apresentadas no Capítulo 2. Toda comunicação entre o leitor e o servidor é feita através de uma conexão segura. A comunicação entre leitor e etiquetas é realizada através de radiofrequência (RF) e pode ser facilmente capturada. O leitor utiliza um protocolo anticolisão baseado em árvore para controle de acesso das etiquetas ao meio de comunicação.

Cada etiqueta T_i possui espaço de memória reservado para um identificador aleatório e temporário (IDT_i) de n *bits*. Esse identificador substitui seu identificador real (IDR_i) de n *bits* durante a execução do protocolo anticolisão baseado em árvore. Para a geração do IDT_i e a realização do processo de autenticação, cada etiqueta possui uma chave atualizável (K_i) e um $NLFSR_i$ único, sendo ambos de n *bits*. É importante ressaltar cada etiqueta do sistema possui um $NLFSR_i$ diferente, com uma função de transição exclusiva.

5.1.2 Modelo de Ameaça

Um modelo de ameaça tem como objetivo definir o que uma entidade maliciosa pode fazer para tentar quebrar a segurança de um sistema. Neste trabalho, é assumido que uma entidade maliciosa pode ameaçar o sistema RFID das seguintes formas:

1. Capturando quaisquer mensagens trocadas entre o leitor e as etiquetas durante a execução do protocolo anticolisão;

2. Capturando quaisquer mensagens trocadas entre as etiquetas e o leitor durante o processo de autenticação;

3. Realizando ataques de *replay*;

4. Utilizando dados capturados para tentar obter informações sigilosas;

5. Realizando ataques de dessincronização;

6. Realizando ataques internos.

Na *ameaça 1*, o atacante realiza um ataque de escuta durante a execução do protocolo anticolisão baseado em árvore tendo acesso aos identificadores que estejam sendo utilizados pelas etiquetas nesse processo. Essa ameaça também permite a um atacante que ele tente se passar por um leitor e realize o processo de anticolisão de etiquetas. Isso significa que o atacante pode obter os identificadores aleatórios e temporários utilizados pelas etiquetas a qualquer momento.

Na *ameaça 2*, o atacante realiza um ataque de escuta durante a troca de mensagens do processo de autenticação. Isso permite que ele tenha acesso as mensagens transmitidas durante o processo de autenticação e que as utilize em algum outro tipo de ataque.

As informações capturadas nas *ameaças 1* e *2* podem ser utilizadas para que o atacante tente realizar os seguintes ataques:

- Rastreamento de uma ou mais etiquetas através dos identificadores utilizados por elas durante a execução do protocolo anticolisão baseado em árvore;

- Clonagem de uma ou mais etiquetas através do identificador e de outras informações obtidas em ambas as escutas;

A *ameaça 3* permite a uma entidade maliciosa reutilizar as mensagens capturadas nas *ameaças 1* e *2* para tentar se passar por uma etiqueta autêntica do sistema. A *ameaça 4* permite a um atacante utilizar as mensagens capturadas nas *ameaças 1* e *2* para tentar obter o identificador real, a chave atualizável ou o $NLFSR_i$ das etiquetas (vide Seção 5.1.1).

Na *ameaça 5*, a entidade maliciosa pode bloquear a recepção de uma ou mais mensagens trocadas entre qualquer etiqueta e o leitor. Esse bloqueio é feito através de um ataque de negação de serviço configurado especificamente para essa finalidade. Isso pode impedir uma etiqueta de atualizar sua chave (K_i) em sincronia com o servidor.

Na *ameaça 6*, a entidade maliciosa pode tentar realizar um dos ataques internos citados no Capítulo 3 para prejudicar o sistema RFID. É importante ressaltar que esse tipo de ameaça só pode ser realizada caso o atacante consiga efetuar um ataque externo que tenha a capacidade de inserir uma entidade falsa no sistema. Portanto, o atacante deve conseguir clonar uma etiqueta real ou autenticar uma etiqueta falsa com sucesso através de um ataque de *replay* para conseguir realizar um ataque interno.

Alguns ataques citados no Capítulo 3 não são tratados neste modelo de ameaça, como por exemplo: comprometer uma etiqueta RFID fisicamente; ataques de homem no meio (*man-in-the-middle*) e; outros tipos de negação de serviço (DoS - *Denial of Service*) diferentes da dessincronização. Proteger um sistema RFID de forma efetiva contra essas ameaças ainda é um desafio atualmente [Myneni, Misra e Xue 2011].

5.2 DETALHAMENTO DO AMAS

O AMAS é um esquema para ser utilizado com protocolos anticolisão baseados em árvore. Esse esquema permite a atenticação mútua etiqueta-leitor e preserva o anonimato das etiquetas. O AMAS utiliza IDs aleatórios e temporários para a execução do protocolo anticolisão, não permitindo a um atacante correlacionar esses IDs com os IDs reais. O ID aleatório e temporário de cada etiqueta é referenciado como IDT_i.

O AMAS é realizado em duas etapas: a primeira é responsável pela geração do IDT_i e a segunda é reponsável pela autenticação mútua etiqueta-leitor. No AMAS, cada etiqueta possui um $NLFSR_i$ único pré-compartilhado com o servidor S. Esse NLFSR é utilizado no processo de geração do IDT_i para garantir que apenas um leitor autorizado consiga identificar a etiqueta. O NLFSR também é utilizado no processo de autenticação mútua etiqueta-leitor, garantindo que tal processo só possa ser feito por etiquetas e leitores autorizados. Todas as operações realizadas durante a execução do AMAS utilizam números de n bits, onde n é um múltiplo de 8. Em geral, o n utilizado é de 32 *bits*.

O esquema de autenticação mútua etiqueta-leitor do AMAS é baseado no do SAMA. As diferenças entre o AMAS e o SAMA são discutidas no final dessa seção. A seguir, são apresentados: uma descrição detalhada do funcionamento do NLFSR, o processo de

geração do IDT_i, o processo de autenticação mútua etiqueta-leitor e, por fim, as diferenças entre o AMAS, o SEAS e o SAMA.

5.2.1 NLFSR (Non-Linear Feedback Shift Register)

Os NLFSRs (*Non-Linear Feedback Shift Registers*) são blocos comumente utilizados na construção de algoritmos de cifra de fluxo (ou *stream cipher*) [Dubrova, Teslenko e Tenhunen 2008, Gammel, Gottfert e Kniffler 2006]. O NLFSR é um registrador de x *bits* de tamanho que realiza y rodadas de deslocamento à direita, onde y é o tamanho da saída do NLFSR.

Cada NLFSR possui uma função de transição única composta por uma quantidade aleatória de portas lógicas do tipo XOR ou AND. Cada uma das portas lógicas pode receber como entrada o valor do *bit* que se encontra em uma das posições do registrador ou mesmo a saída de outra porta lógica. Antes de cada rodada de deslocamento, o NLFSR substitui o valor de cada *bit* do registrador pelo resultado de um XOR entre o valor atual do *bit* e o resultado da função de transição. A cada rodada de deslocamento, o *bit* mais à direita do registrador é concatenado aos *bits* de saída do NLFSR. É importante ressaltar que quaisquer modificações realizadas na função de transição como, por exemplo, a alteração no número de portas lógicas, faz com que a saída do NLFSR seja completamente modificada.

A Figura 5.1 apresenta um exemplo de *NLFSR* de 32 *bits* que possui uma função de transição composta por sete portas lógicas, sendo quatro do tipo XOR e três do tipo AND.

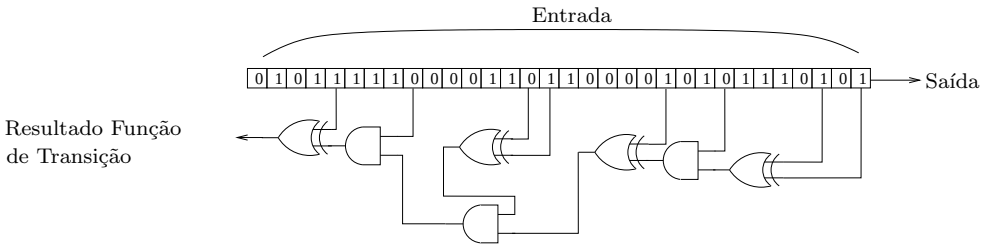


Figura 5.1 Exemplo de NLFSR de 32 bits com sete portas lógicas.

5.2.2 Geração do IDT_i

A Figura 5.2 apresenta o processo de geração do IDT_i realizado por cada uma das etiquetas a serem identificadas. O leitor R_j precisa que as etiquetas gerem os seus respectivos IDT_i antes do início da execução do protocolo anticolisão. Para isso, ele gera um número aleatório r_j^1 e o transmite em *broadcast* para as etiquetas. Cada uma das etiquetas recebe r_j^1 e o utiliza como entrada para o seu respectivo $NLFSR_i$. A saída desse processo é um número r_j^1. Em seguida, cada etiqueta gera um número aleatório r_{ij}^2 e faz um XOR entre r_j^1, r_{ij}^2, o seu identificador real IDR_i e sua chave K_i, gerando o número p_{ij}^1. Esse número é fornecido como entrada para o $NLFSR_i$ que, após os n deslocamentos, gera o IDT_i.

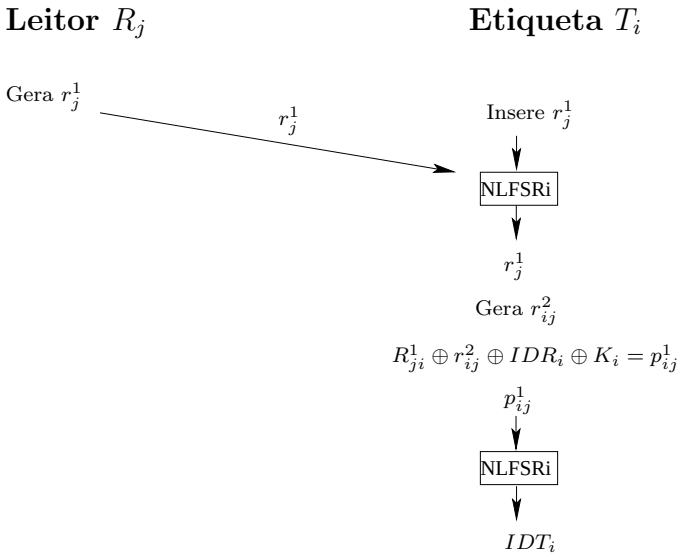


Figura 5.2 Geração do IDT_i por uma etiqueta T_i.

Após o fim da primeira etapa do AMAS, o leitor executa o protocolo anticolisão. É importante observar que existe a possibilidade de que duas etiquetas gerem o mesmo IDT_i. No entanto, dado que até 500 etiquetas com IDs de 32 *bits* estejam sendo identificadas ao mesmo tempo, a probabilidade de que pelo menos duas gerem dois IDT_i iguais é de aproximadamente $2,91 \times 10^{-5}$ conforme uma aproximação do paradoxo do aniversário [Tsaban 2003]. Essa aproximação é calculada através da seguinte equação:

$$1 - e^{\frac{(n-1)n}{2m}}, \tag{5.1}$$

onde e é a constante de exponenciação, n é quantidade de etiquetas que estão sendo identificadas e m é o número de IDs distintos que podem ser construídos com 32 *bits*. O procedimento a ser adotado pelo protocolo anticolisão para contornar esse problema deve ser tratado em trabalhos futuros.

5.2.3 Autenticação da Etiqueta perante o Leitor

A Figura 5.3 apresenta o processo de autenticação de uma etiqueta T_i perante o leitor R_j. O processo se inicia assim que a mesma é selecionada durante a execução do protocolo anticolisão baseado em árvore. O leitor R_j armazena o IDT_i da etiqueta e inicia a verificação da autenticidade da mesma. Para isso, o leitor requisita o r_{ij}^2 gerado pela etiqueta durante o processo de geração de seu IDT_i. Após o recebimento de r_{ij}^2, o leitor envia IDT_i, r_j^1 e r_{ji}^2 para o servidor S. O servidor busca no seu banco de dados uma terna (IDR_i, K_i, $NLFSR_i$) que consiga gerar o IDT_i quando combinada com r_j^1 e r_{ij}^2. Em outras palavras, o servidor verifica um a um os dados das etiquetas presentes no banco de dados até que a terna correta seja encontrada. Esse processo de busca é custoso, mas um servidor com uma boa capacidade computacional consegue evitar que o desempenho do sistema seja afetado. Quando o servidor encontra a terna correta, ele a envia para o leitor que, por sua vez, autentica a etiqueta. Caso nenhuma terna correspondente seja encontrada, a etiqueta é considerada falsa e a comunicação com essa etiqueta é encerrada.

Para garantir uma maior segurança do mecanismo de geração do IDT_i, a chave K_i é atualizada pelo servidor após a autenticação da etiqueta. Para isso, é feito um XOR entre K_i, r_j^1 e R_{ij}^2. Este último é o valor de r_{ij}^2 após ele ser modificado pelo $NLFSR_i$. O resultado do XOR é fornecido como entrada para o $NLFSR_i$, o qual gera como saída o valor de K_i que será utilizado na próxima geração do IDT_i. O servidor guarda o valor antigo de K_i para evitar que ataques de dessincronização possam afetar o sistema RFID. O detalhamento de como o AMAS evita ataques dessincronização é apresentado na Seção 5.3.

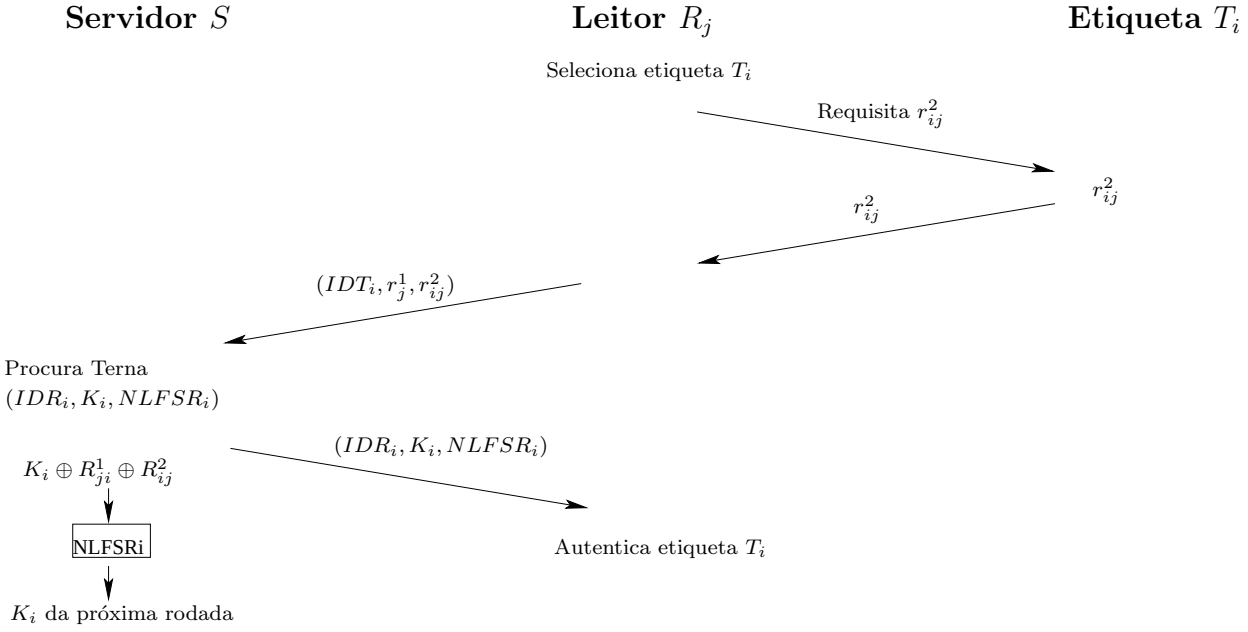


Figura 5.3 Autenticação ben sucedida de uma etiqueta T_i perante o leitor R_j.

5.2.4 Autenticação do Leitor perante à Etiqueta

A Figura 5.4 apresenta o processo de autenticação do leitor R_j por uma etiqueta T_i. Para isso, o leitor fornece r_{ij}^2 como entrada para o $NLFSR_i$, o qual gera a saída R_{ij}^2. Em seguida, o leitor gera um novo número aleatório r_{ji}^3 e faz um XOR desse número com R_{ji}^2, gerando x_{ji}^1. Este último número é fornecido como entrada ao $NLFSR_i$, o qual gera a saída m_{ji}^1. Os valores de m_{ji}^1 e r_{ji}^3 são concatenados, gerando a mensagem M_{ji}^1. Essa mensagem é enviada para a etiqueta que verifica a sua validade utilizando r_{ij}^2, r_{ji}^3 e o $NLFSR_i$. Caso M_{ji}^1 seja válida, o processo de autenticação do leitor é finalizado.

Após autenticar o leitor, a chave K_i é atualizada na etiqueta da mesma forma que foi feita no servidor S. Em seguida, T_i faz um XOR entre R_{ij}^2, r_j^1 e $r_j^3 i$ e fornece o resultado dessa operação como entrada para o $NLFSR_i$, gerando como saída o número m_{ij}^2. Este último número é enviado para o leitor e serve como confirmação para o servidor de que o processo foi concluído corretamente. Após o recebimento, o servidor utiliza R_{ij}^2, r_j^1 e $r_j^3 i$ para autenticar o valor de m_{ij}^2, verificando se foi realmente a etiqueta que enviou o número.

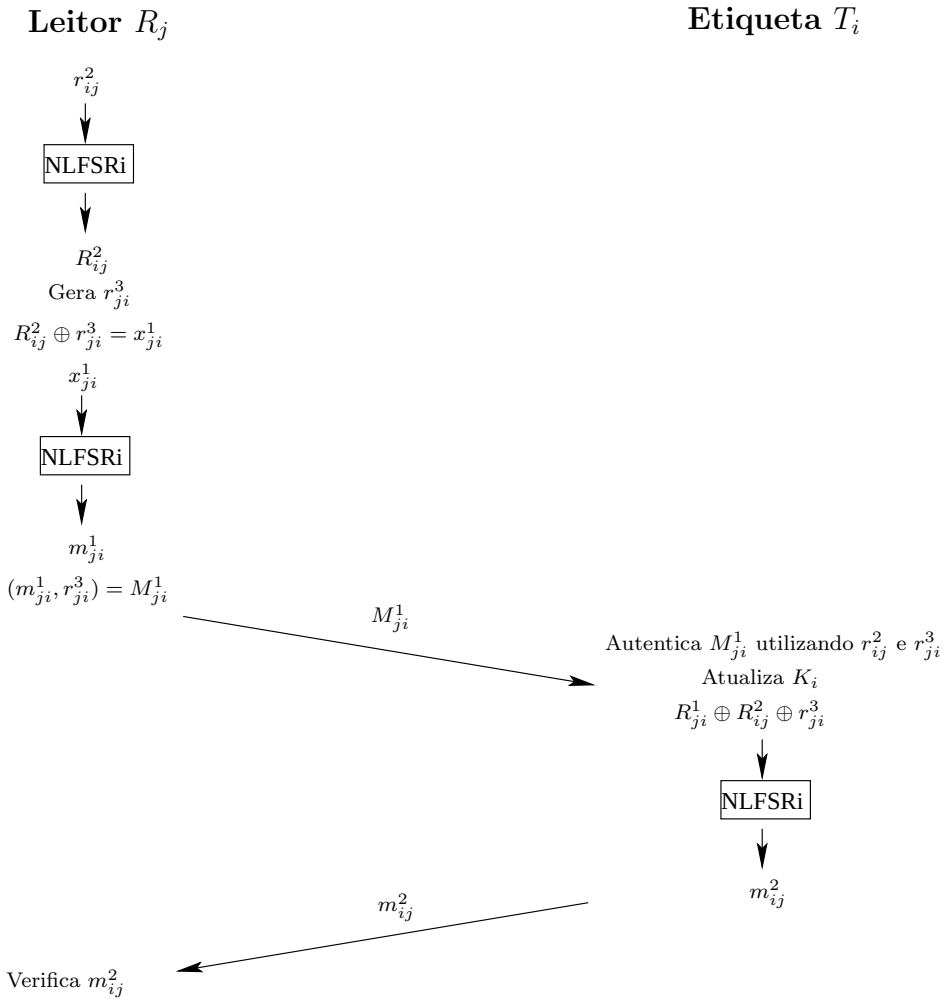


Figura 5.4 Autenticação bem sucedida de um leitor R_j perante uma etiqueta T_i.

5.2.5 Diferenças entre o AMAS, o SEAS e o SAMA

O AMAS e o SEAS são esquemas diferentes. O AMAS utiliza uma chave de segurança atualizável para cada etiqueta e baseia sua segurança no NLFSR. Já o SEAS utiliza uma chave de segurança fixa para cada etiqueta e baseia sua segurança apenas em operações XOR e em uma função linear de deslocamento à esquerda. O SEAS não leva em conta como o processo anticolisão é feito nem as vulnerabilidades desse processo.

O esquema de autenticação do AMAS é uma versão modificada daquele usado pelo SAMA [Myneni, Misra e Xue 2011]. Primeiramente, o esquema utilizado pelo SAMA se difere pela não utilização de um servidor no processo de autenticação, fazendo com que

cada leitor tenha que armazenar os dados de todas as etiquetas do sistema para realizar tal processo. No AMAS, o servidor é utilizado para armazenar e buscar todas as informações relativas às etiquetas. Isso reduz significativamente a quantidade de memória que cada leitor precisa ter disponível para realizar o processo de autenticação.

A segunda diferença entre o AMAS e o SAMA ocorre no processo de autenticação da etiqueta perante o leitor. No AMAS, esse processo inclui a utilização do IDT_i, do ID real da etiqueta (IDR_i) e da chave atualizável (K_i). Essa modificação permite que o IDT_i seja utilizado tanto pelo processo anticolisão baseado em árvore como na autenticação da etiqueta perante o leitor, recebendo a segurança adicional fornecida pela chave atualizável (K_i).

Outra diferença entre o AMAS e o SAMA é a não utilização da função de perturbação pelo AMAS, sendo substituída, quando necessário, pelo $NLFSR_i$. Isso é feito porque a função de perturbação pode ser invertida facilmente. Assim sendo, a utilização dessa função não traria nenhum ganho para a segurança do esquema de autenticação do AMAS e resultaria em um consumo desnecessário de recursos para sua implementação e execução. Assim como o SEAS, o SAMA não leva em conta como o processo anticolisão é realizado nem as fraquezas desse processo.

5.3 ANÁLISE DA SEGURANÇA E DE CUSTOS DO AMAS

Essa seção apresenta, inicialmente, uma criptoanálise do NLFSR e uma análise da segurança do AMAS perante as ameaças definidas na Seção 5.1. Em seguida, é apresentada uma avaliação de seu custo em termos de quantidade de portas lógicas e ciclos de relógio, comparando-os com os quantitativos obtidos pelo SEAS, SAMA, SHA-1, MD4 e AES.

5.3.1 Análise da Segurança

5.3.1.1 Criptoanálise do NLFSR

O $NLFSR$ é o dispositivo de segurança básico utilizado pelo AMAS. Assim sendo, um atacante precisa primeiramente identificar o $NLFSR_i$ da etiqueta T_i que ele pretende

atacar para conseguir burlar o AMAS. No entanto, de acordo com o *Corolário 1*, cuja prova pode ser encontrada em [Myneni, Misra e Xue 2011], a probabilidade de que um atacante consiga identificar um NLFSR é desprezível mesmo para um NLFSR de 32 bits.

Corolário 1: Assumindo que o NLFSR é de x bits, o número de portas lógicas é l, m é a quantidade de tipos de portas lógicas utilizados, $P_{x,l}$ é uma permutação que representa todas as possíveis formas de escolher as saídas das l portas lógicas e C_{2l}^{x-l} é uma combinação que representa todas as possíveis formas de escolher as entradas das l portas lógicas, a probabilidade de um atacante conseguir identificar corretamente o $NLFSR$ é dada pela seguinte equação:

$$\frac{1}{(m^l \cdot l! \cdot P_{x,l} \cdot C_{2l}^{x-l})}. \tag{5.2}$$

Assim sendo, a probabilidade de um atacante identificar um NLFSR de 32 bits com 9 portas lógicas é igual a $\frac{1}{6{,}36X10^{25}}$, sendo um valor desprezível. Outra análise importante é qual a complexidade computacional para se identificar o NLFSR e a entrada y utilizados para gerar uma saída y' através de um ataque de força bruta. Para isso, um atacante precisaria testar todas as entradas em todos os NLFSR possíveis até encontrar a combinação correta. O número total de entradas que podem ser utilizadas em um NLFSR de 32 bits é de 2^{32}, valor que pode ser aproximado para 10^{10}. O número total de NLFSRs de 32 bits e 9 portas lógicas é $6,36X10^{25}$ de acordo com o Corolário 1. Portanto, o esforço computacional para se identificar o NLFSR e a entrada y utilizados para gerar uma saída y' através da força bruta é de aproximadamente $6,36X10^{35}$.

5.3.1.2 Rastreamento

Uma etiqueta T_i não é rastreável se um atacante não conseguir o seguinte:

- **caso 1 -** correlacionar seus identificadores aleatórios e temporários, gerados em rodadas distintas, como pertencentes à etiqueta T_i;
- **caso 2 -** relacionar seu identificador aleatório e temporário atual com o seu identificador real.

Em ambos os casos, um atacante poderia tentar rastrear uma etiqueta T_i utilizando dois métodos. No primeiro, o atacante apenas escuta mensagens trocadas entre a etiqueta T_i e o leitor. No segundo, o atacante tenta se passar por um leitor legítimo e iniciar o processo anticolisão posteriormente ao envio por ele de um r_j^1 para que a etiqueta gere seu IDT_i.

O primeiro método não funciona no **caso 1** pelo seguinte: um atacante precisa identificar o $NLFSR_i$ e a chave atualizável (K_i) da etiqueta T_i que foram utilizados no processo de geração de dois ou mais identificadores aleatórios e temporários. Como apresentado na Seção 5.3.1.1, a probabilidade dele identificar o $NLFSR_i$ é desprezível. Além disso, a chave K_i é mantida em sigilo durante todo o processo e é atualizada a cada rodada de autenticação.

O segundo método não funciona no **caso 1** pelo seguinte: as etiquetas utilizam um número aleatório r_{ij}^2 para gerar o IDT_i. Toda vez que o processo anticolisão for rodado, a etiqueta gerará um novo número aleatório r_{ij}^2 e, consequentemente, um novo IDT_i. Para conseguir identificar que dois ou mais identificadores IDT_i foram gerados por uma mesma etiqueta T_i, um atacante precisaria obter o seu $NLFSR_i$ e sua chave atualizável (K_i), aos quais ele não tem acesso.

Ambos os métodos não funcionam para o **caso 2**. Isso ocorre, porque o atacante precisa obter o $NLFSR_i$ e a chave atualizável (K_i) da etiqueta T_i, os quais ele não tem acesso, para conseguir relacionar o IDT_i atualmente utilizado pela etiqueta com o seu identificador real.

5.3.1.3 Clonagem

Para que um atacante consiga clonar corretamente uma etiqueta, ele precisa conseguir burlar o mecanismo de autenticação do AMAS. Para isso, a etiqueta falsa T_i' teria que conseguir gerar corretamente o IDT_i a partir de um r_j^1 enviado pelo leitor. A única forma de gerar IDT_i corretamente é utilizando o $NLFSR_i$ e a chave atualizável K_i da etiqueta T_i. Assim sendo, T_i' não consegue burlar esse mecanismo, pois ela não consegue identificar qual é o $NLFSR_i$ utilizado por T_i e não tem acesso à chave K_i. É importante observar

que se um atacante conseguir acessar fisicamente uma etiqueta, ele terá acesso a todos os segredos contidos dentro da mesma e poderá cloná-la com facilidade. No entanto, essa ameaça não foi considerada no modelo de ataque pelo motivo já citado na Seção 5.1.

5.3.1.4 Ataque de Replay

Mesmo que um atacante consiga capturar diversas mensagens de autenticação transmitidas entre um leitor e uma etiqueta, ele não conseguirá reutilizá-las posteriormente em um ataque de *replay*. Isso ocorre devido à utilização do IDT_i e dos números r_j^1, r_{ij}^2 e r_{ji}^3, os quais são gerados aleatoriamente a cada rodada da autenticação.

5.3.1.5 Acesso a Informações Sigilosas

As únicas informações sigilosas mantidas pelo AMAS são a chave K_i, o ID real IDR_i e o $NLFSR_i$. No entanto, o $NLFSR_i$ nunca é transmitido e a chave K_i e o IDR_i não são transmitidos sem terem sido combinados com outras variáveis e modificados pelo $NLFSR_i$. Portanto, um atacante não consegue obtê-los somente capturando mensagens trocadas entre a etiqueta e o leitor. Mesmo que ele conseguisse adivinhar a chave K_i de uma determinada rodada por força bruta, a mesma já não teria mais valor na rodada seguinte devido à atualização da chave no processo de autenticação.

5.3.1.6 Ataques de Dessincronização

O único parâmetro atualizado pelo AMAS a cada rodada é a chave K_i. Ele consegue garantir que um ataque de dessincronização simples não seja capaz de afetar a atualização sincronizada desse parâmetro pela etiqueta e pelo servidor. Isso ocorre porque o servidor utiliza o número m_{ij}^2 como garantia de que uma etiqueta T_i tenha conseguido atualizar K_i corretamente. Se o atacante conseguir evitar que o processo seja concluído com algum ataque de dessincronização mais elaborado, isso não afetará a geração do IDT_i e o processo de autenticação mútua etiqueta-leitor. Isso ocorre porque o servidor armazena

para cada etiqueta do sistema, tanto a K_i atualizada quanto o valor de K_i utilizado na rodada anterior do processo autenticação. Dessa forma, ele garante que o leitor consegue autenticar a etiqueta em uma certa rodada mesmo que a chave K_i não seja atualizada corretamente na rodada anterior. A aleatoriedade de IDT_i também não é afetada por conta da utilização de um novo r_{ij}^2 a cada vez que ele é gerado.

5.3.1.7 Ataques Internos

Como dito na Seção 5.1 a realização de ataques internos só é possível caso o atacante seja capaz de realizar um dos seguintes ataques externos: clonar uma etiqueta ou autenticar uma etiqueta falsa com sucesso perante o leitor através de um ataque de *replay*. Como visto nas seções anteriores, o AMAS impede a realização tanto do ataque de *replay* quanto do ataque de clonagem. Assim sendo, o esquema evita que um atacante consiga obter as condições necessárias para realizar um ataque interno.

5.3.2 Avaliação de Custos

Esta seção avalia os custos do AMAS em termos de quantidade de portas lógicas e ciclos de relógio, comparando-os com os custos do SEAS, do SAMA, da SHA-1, da MD4 e do AES. O AMAS foi implementado como uma máquina de estados combinacional na linguagem de *hardware System Verilog*[1]. Essa implementação foi utilizada para calcular seus custos através do programa de síntese de *hardware ALTERA Quartus II*[2]. A implementação utilizou n igual a 32, ou seja, todos os parâmetros e mecanismos utilizados foram de 32 bits. Isso foi feito para que fosse possível comparar o AMAS com os trabalhos relacionados, os quais também utilizam 32 bits.

O SAMA também foi implementado no ambiente utilizado para a implementação do AMAS. Os custos obtidos para o SAMA através dessa implementação foram semelhantes

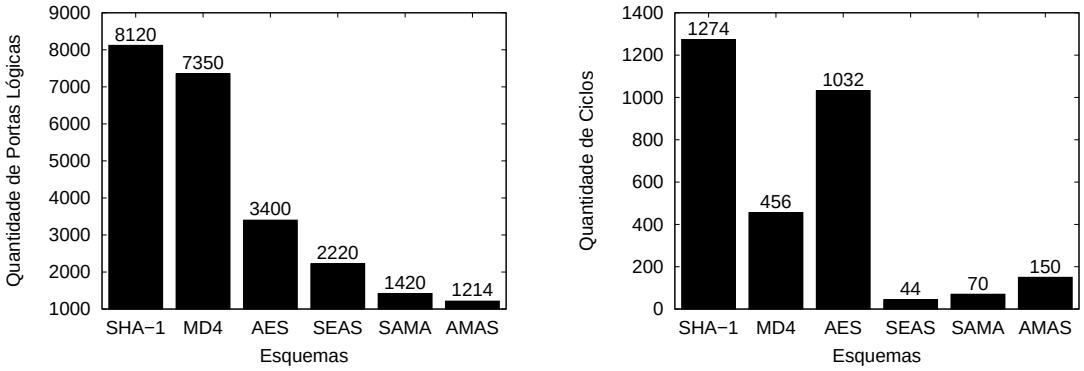
[1] mais informações em http://www.systemverilog.org/

[2] mais informações em http://www.altera.com/products/software/quartus-ii/about/qts-performance-productivity.html

aos obtidos em [Myneni, Misra e Xue 2011]. A versão da implementação do SAMA no *ALTERA Quartus II* utiliza 1.420 portas lógicas e 70 ciclos de relógio, ao passo que, os autores do SAMA informam que o esquema utiliza 1.393 portas lógicas e 70 ciclos de relógio. A pequena diferença se deve ao fato dos resultados serem dependentes da forma como o algoritmo é codificado. Para fins de comparação, este artigo adota os custos do SAMA obtidos no ambiente *ALTERA Quartus II*. Os custos da SHA-1, da MD4 e do AES foram obtidos em [Feldhofer e Rechberger 2006] enquanto os custos do SEAS foram obtido em [Misra et al. 2009]. Esses custos devem ser vistos como uma aproximação para comparação com o SAMA e o AMAS. A implementação no mesmo ambiente do AMAS não traria potencialmente mudanças significativas a ponto de alterar as conclusões deste trabalho.

A Figura 5.5(a) apresenta uma comparação do número necessário de portas lógicas para a implementação dos esquemas estudados. Note que a quantidade de portas lógicas do AMAS (1214 portas lógicas) é a menor de todas. A diferença de portas lógicas entre o AMAS e o SAMA ocorre por dois motivos. O primeiro é a não utilização pelo AMAS da função de perturbação do SAMA, a qual utiliza aproximadamente 300 portas lógicas em sua implementação. O segundo é o fato de que os mecanismos de geração do IDT_i e de atualização da chave K_i do AMAS reutilizam o mesmo $NLFSR$ do esquema de autenticação para prover IDs para o processo anticolisão. Com a reutilização, esses mecanismos contribuem na quantidade adicional de aproximadamente 100 portas lógicas para serem implementados.

A Figura 5.5(b) apresenta o resultado de ciclos de relógio para os esquemas estudados. O custo do AMAS (150 ciclos) ficou abaixo do limite de 220 ciclos para etiquetas passivas embora tenha sido mais elevado do que o ciclo de relógio do SAMA (70 ciclos) e do SEAS (44 ciclos). A necessidade de se utilizar mais ciclos de relógio no AMAS em relação ao SAMA se deve ao mecanismo de geração do IDT_i e ao uso, por mais vezes, do $NLFSR$, gerando o consumo de 80 ciclos de relógio a mais do que o SAMA. No entanto, quando comparado com os mecanismos tradicionais de segurança como o AES (1032 ciclos), o AMAS possui um custo significativamente menor.



(a) (b)

Figura 5.5 Comparação entre os diversos esquemas.

5.4 RESUMO

Nesse capítulo foi apresentado o AMAS, o qual é um esquema de autenticação mútua etiqueta-leitor para ser utilizado com protocolos anticolisão baseados em árvore. Primeiramente, foram descritos os modelos de sistema e de ameaça adotados pelo esquema. Em segundo lugar, foi feita a apresentação do funcionamento do AMAS. Por fim, foram apresentadas as análises da segurança e do custo do esquema.

CAPÍTULO 6

CONCLUSÃO

Dentre as tecnologias desenvolvidas para atender os requisitos de aplicações de identificação, organização e localização automática de objetos, uma das mais promissoras é a RFID. A capacidade de armazenamento de informações nas etiquetas e a leitura das informações através de ondas de rádio são as duas principais características que tornam esse tipo de sistema bastante versátil. Graças a essas características, uma infinidade de aplicações das mais diversas áreas utilizam esse tipo de sistema. Dentre os diversos tipos de sistemas RFID existentes, este trabalho se interessou mais pelos que utilizam etiquetas passivas devido ao seu baixo custo de produção.

Prover autenticação mútua é um dos maiores desafios de sistemas RFID que utilizam etiquetas passivas. Isso ocorre devido às limitações de recursos computacionais e memória inerentes a esse tipo de etiqueta. O SAMA e o SEAS são as propostas mais atuais para prover autenticação mútua etiqueta-leitor em sistemas RFID baseados em etiquetas passivas. Esses dois esquemas são capazes de manter o anonimato das etiquetas. Mas, para isso, um requisito mínimo necessário é nunca transmitir em claro o ID real dessas etiquetas durante qualquer troca de mensagem com o(s) leitor(es). Por outro lado, o ID real das etiquetas é comumente utilizado e transmitido em claro durante a execução de protocolos anticolisão baseados em árvore. Essa execução precede o processo de autenticação, fazendo com que o anonimato das etiquetas não possa ser garantido.

Este trabalho propôs um esquema de autenticação mútua etiqueta-leitor para ser utilizado em conjunto com protocolos anticolisão baseados em árvore e ao mesmo tempo preservar o anonimato das etiquetas. Essa é uma das principais contribuições deste trabalho. O esquema proposto, denominado AMAS, reutilizou mecanismos empregados no processo de autenticação para que as etiquetas gerassem IDs aleatórios e temporários

a serem utilizados durante a execução do protocolo anticolisão baseado em árvore. Essa abordagem buscou garantir o anonimato das etiquetas desde o processo anticolisão, não permitindo a um atacante correlacionar os IDs aleatórios e temporários com os IDs reais.

Este trabalho também apresentou uma análise da segurança do AMAS, demonstrando que o esquema consegue defender o sistema de todos os ataques presentes no modelo de ameaça adotado. Adicionalmente, os custos do AMAS, em termos de quantidade de portas lógicas e ciclos de relógio, foram avaliados. Os resultados demonstraram que o AMAS necessita de apenas 1214 portas lógicas e 150 ciclos de relógio para ser implementado. Assim sendo, o esquema proposto atende aos requisitos necessários para que possa ser utilizado em sistemas RFID baseados em etiquetas passivas.

O AMAS se diferencia das propostas mais atuais para prover autenticação mútua em sistemas RFID baseados em etiquetas passivas, o SAMA e o SEAS, por conseguir fornecer anonimicidade aos sistemas RFID que utilizam protocolos anticolisão baseados em árvore. O esquema proposto mantém o anonimato das etiquetas utilizando mecanismos de baixo custo computacional, o que permite o seu uso em etiquetas passivas. O AMAS também se destaca por possuir o menor custo em termos de portas lógicas (1214 portas) quando comparado com o SAMA (1420 portas) e com SEAS (2220 portas).

A única desvantagem do AMAS em relação ao SAMA e ao SEAS é o número de ciclos de relógio que ele utiliza. Enquanto o AMAS utiliza 150 ciclos de relógio, o SAMA e o SEAS utilizam apenas, respectivamente, 70 e 42 ciclos. Assim sendo, como trabalhos futuros, devem ser encontradas formas de otimizar a quantidade de ciclos de relógio utilizada pelos mecanismos do AMAS e, consequentemente, obter uma redução no consumo de energia. Nessa linha de desenvolvimento, um primeiro passo seria encontrar um substituto para o NLFSR que consuma uma quantidade menor de ciclos de relógio e mantenha o esquema de autenticação seguro.

REFERÊNCIAS BIBLIOGRÁFICAS

[Barasz 2007]BARASZ, M. Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags. In: *Proceedings of the First Int'l Workshop RFID Technology (EURASIP)*. [S.l.: s.n.], 2007.

[Bishop et al. 2012]BISHOP, M. et al. A Taxonomy of Buffer Overflow Characteristics. *IEEE Transactions on Dependable and Secure Computing*, 2012. v. 9, n. 3, p. 305–317, june 2012.

[Bueno-Delgado, Vales-Alonso e Gonzalez-Castao 2009]BUENO-DELGADO, M.; VALES-ALONSO, J.; GONZALEZ-CASTAO, F. Analysis of DFSA Anti-Collision Protocols in Passive RFID Environments. In: *Proceedings of the IEEE Industrial Electronics 35th Annual Conference (IECON)*. [S.l.: s.n.], 2009. p. 2610–2617.

[Chen, Horng e Fan 2007]CHEN, W.-C.; HORNG, S.-J.; FAN, P. An Enhanced Anti-collision Algorithm in RFID Based on Counter and Stack. In: *Proceedings of the Second International Conference on Systems and Networks Communications (ICSNC)*. [S.l.: s.n.], 2007. p. 21–24.

[Choi, Lee e Lee 2007]CHOI, J. H.; LEE, D.; LEE, H. Query Tree-Based Reservation for Efficient RFID Tag Anti-Collision. *IEEE Communications Letters*, 2007. v. 11, n. 1, p. 85–87, January 2007.

[Chothia e Smirnov 2010]CHOTHIA, T.; SMIRNOV, V. A Traceability Attack Against e-Passports. In: *Proceedings of the 14th International Conference on Financial Cryptography and Data Security*. [S.l.: s.n.], 2010.

[Daemen e Rijmen 1998]DAEMEN, J.; RIJMEN, V. *AES Proposal: Rijndael*. June 1998. Disponível em: <http://www.comms.scitech.susx.ac.uk/fft/crypto/rijndael.pdf>.

[Deursen e Radomirovic 2008]DEURSEN, T.; RADOMIROVIC, S. Security of RFID Protocols - A Case Study. In: *Proceedings of the 4th International Workshop on Security and Trust Management (STM)*. [S.l.: s.n.], 2008. p. 41–52.

[Dimitriou 2005]DIMITRIOU, T. A Lightweight RFID Protocol to Protect Against Traceability and Cloning Attacks. In: *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*. [S.l.: s.n.], 2005. p. 59–66.

[Dimitriou 2006]DIMITRIOU, T. A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete. In: *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM)*. [S.l.: s.n.], 2006. p. 269–275.

[Dominikus, Oswald e Feldhofer 2005]DOMINIKUS, S.; OSWALD, E.; FELDHOFER, M. Symmetric Authentication for RFID Systems in Practice. In: *Proceedings of the Ecrypt Workshop on RFID and Lightweight Cryptography*. [S.l.: s.n.], 2005. p. 14–15.

[Dubrova, Teslenko e Tenhunen 2008]DUBROVA, E.; TESLENKO, M.; TENHUNEN, H. On Analysis and Synthesis of (n, k)-Non-linear Feedback Shift Registers. In: *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*. [S.l.: s.n.], 2008. p. 1286–1291.

[EPCGlobal 2008]EPCGLOBAL. *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860-960 MHz version 1.2.0*. October 2008.

[Feldhofer 2004]FELDHOFER, M. An Authentication Protocol in a Security Layer for RFID Smart Tags. In: *Proceedings of the IEEE Mediterranean Electrotechnical Conference (MELECON)*. [S.l.: s.n.], 2004. p. 759–762.

[Feldhofer, Dominikus e Wolkerstorfer 2004]FELDHOFER, M.; DOMINIKUS, S.; WOLKERSTORFER, J. Strong Authentication for RFID Systems Using the AES Algorithm. *Lecture Notes In Computer Science*, 2004. p. 357–370, 2004.

[Feldhofer e Rechberger 2006]FELDHOFER, M.; RECHBERGER, C. A Case Against Currently Used Hash Functions in RFID Protocols. *Lecture Notes In Computer Science*, 2006. v. 4278, p. 372–381, 2006.

[Finkenzeller 2003]FINKENZELLER, K. *RFID Handbook*. [S.l.]: F. John Wiley and Sons Ltd., 2003. (Fundamentals and Applications in Contactless Smart Cards and Identification).

[Franklin e Gonçalves 2010]FRANKLIN, E. H. C.; GONÇALVES, P. A. S. Uma Análise da Segurança de Sistemas RFID. *Trabalho de Graduação*, 2010. July 2010.

[Fu, Zhang e Wang 2010]FU, Y.; ZHANG, C.; WANG, J. A Research on Denial of Service Attack in Passive RFID System. In: *Proceedings of the International Conference on Anti-Counterfeiting Security and Identification in Communication (ASID)*. [S.l.: s.n.], 2010. p. 24 –28.

[Galluccio, Morabito e Catania 2011]GALLUCCIO, L.; MORABITO, G.; CATANIA, M. Facing Man-in-the-middle and Route Diversion Attacks in Energy-Limited RFID Systems Based on Mobile Readers. In: *Proceedings of the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. [S.l.: s.n.], 2011. p. 58 –64.

[Gammel, Gottfert e Kniffler 2006]GAMMEL, B.; GOTTFERT, R.; KNIFFLER, O. An NLFSR-based Stream Cipher. In: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. [S.l.: s.n.], 2006. p. 4.

[Gou, Jeong e Yoo 2010]GOU, H.; JEONG, H. cheol; YOO, Y. A Bit Collision Detection Based Query Tree Protocol for Anti-Collision in RFID System. In: *Proceedings of the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. [S.l.: s.n.], 2010. p. 421–428.

[Hancke 2011]HANCKE, G. P. Practical Eavesdropping and Skimming Attacks on High-Frequency RFID Tokens. *Journal of Computer Security*, 2011. v. 19, n. 2, p. 259–288, 2011.

[Heydt-Benjamin et al. 2009]HEYDT-BENJAMIN, T. S. et al. Vulnerabilities in First-Generation RFID-Enabled Credit Cards. *Economic Perspectives*, 2009. v. 33, n. 1, 2009.

[Hush e Wood 1998]HUSH, D.; WOOD, C. Analysis of Tree Algorithms for RFID Arbitration. In: *Proceedings of the IEEE International Symposium on Information Theory.* [S.l.: s.n.], 1998. p. 107–107.

[Juels 2006]JUELS, A. Minimalist cryptography for low-cost RFID tags. In: *Proceedings of the Fourth Int'l Conf. Computational Intelligence and Security (CIS).* [S.l.: s.n.], 2006.

[Juels 2006]JUELS, A. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communication*, 2006. v. 24, n. 2, p. 381–392, February 2006.

[Juels, Rivest e Szydlo 2003]JUELS, A.; RIVEST, R.; SZYDLO, M. The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy. In: *Proceedings of the 8th ACM Conference on Computer and Communications Security.* [S.l.: s.n.], 2003. p. 103–111.

[Kieyzun et al. 2009]KIEYZUN, A. et al. Automatic Creation of SQL Injection and Cross-site Scripting Attacks. In: *Proceedings of the 31st IEEE International Conference on Software Engineering (ICSE).* [S.l.: s.n.], 2009. p. 199–209.

[Klair, Chin e Raad 2010]KLAIR, D.; CHIN, K.-W.; RAAD, R. A Survey and Tutorial of RFID Anti-Collision Protocols. *IEEE Communications Surveys Tutorials*, 2010. v. 12, n. 3, p. 400–421, 2010.

[Klair, Chin e Raad 2007]KLAIR, D. K.; CHIN, K.-W.; RAAD, R. An Investigation Into the Energy Efficiency of Pure and Slotted Aloha based RFID Anti-Collision Protocols. In: *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (IEEE WoWMoM).* [S.l.: s.n.], 2007.

[Law, Lee e Siu 2000]LAW, C.; LEE, K.; SIU, K.-Y. Efficient Memoryless Protocol for Tag Identification (Extended Abstract). In: *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*. [S.l.: s.n.], 2000. p. 75–84.

[Lee et al. 2005]LEE, S. et al. Efficient Authentication for Low-Cost RFID Systems. In: *Proceedings of the International Conference on Computational Science and its Applications (ICCSA)*. [S.l.: s.n.], 2005. p. 619–627.

[Misra et al. 2009]MISRA, S. et al. SEAS: A Secure and Efficient Anonymity Scheme for Low-Cost RFID tags. In: *Proceedings of the IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2009. p. 1–6.

[Mitrokotsa, Rieback e Tanenbaum 2010]MITROKOTSA, A.; RIEBACK, M. R.; TANENBAUM, A. S. Classifying RFID Attacks and Defenses. *Information Systems Frontiers*, 2010. v. 12, n. 5, p. 491–505, 2010.

[Myneni, Misra e Xue 2011]MYNENI, S.; MISRA, S.; XUE, G. SAMA: Serverless Anonymous Mutual Authentication for Low-Cost RFID Tags. In: *Proceedings of the IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2011. p. 1–5.

[Myung, Lee e Shih 2006]MYUNG, J.; LEE, W.; SHIH, T. An Adaptive Memoryless Protocol for RFID Tag Collision Arbitration. *IEEE Transactions on Multimedia*, 2006. v. 8, n. 5, p. 1096–1101, October 2006.

[Myung, Lee e Srivastava 2006]MYUNG, J.; LEE, W.; SRIVASTAVA, J. Adaptive Binary Splitting for Efficient RFID Tag Anti-Collision. *IEEE Communications Letters*, 2006. v. 10, n. 3, p. 144–146, March 2006.

[Nekoogar e Dowla 2012]NEKOOGAR, F.; DOWLA, F. *Basics of Radio Frequency Identification (RFID) Systems*. [S.l.]: Springer US, 2012. 1–23 p. (Ultra-Wideband Radio Frequency Identification Systems).

[Ni et al. 2003]NI, L. et al. LANDMARC: Indoor Location Sensing Using Active RFID. In: *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)*. [S.l.: s.n.], 2003. p. 407–415.

[Peris-Lopez et al. 2006]PERIS-LOPEZ, P. et al. M2AP: A Minimalist Mutual-Authentication Protocol for low-cost RFID tags. In: *Proceedings of the Third Int'l Conf. Ubiquitous Intelligence and Computing (UIC)*. [S.l.: s.n.], 2006.

[Rieback, Crispo e Tanenbaum 2006]RIEBACK, M.; CRISPO, B.; TANENBAUM, S. Is Your Cat Infected with a Computer Virus? In: *Proceedings of the Fourth Annual IEEE Internacional Conf. Pervasive Computing and Communications (PerComp)*. [S.l.: s.n.], 2006.

[Sun e Ting 2009]SUN, H.; TING, W. A Gen2-Based RFID Authentication Protocol for Security and Privacy. In: *Proceedings of the IEEE Transactions on Mobile Computing*. [S.l.: s.n.], 2009. p. 1053–1062.

[Tong, Zou e Tong 2009]TONG, Q.; ZOU, X.; TONG, H. Dynamic Framed Slotted ALOHA Algorithm Based on Bayesian Estimation in RFID System. In: *Proceedings of the WRI World Congress on Computer Science and Information Engineering*. [S.l.: s.n.], 2009. p. 384–388.

[Tsaban 2003]TSABAN, B. Bernoulli numbers and the probability of a birthday surprise. *Discrete Applied Mathematics*, 2003. v. 127, n. 3, p. 657–663, 2003.

[Tu, Zhou e Piramuthu 2009]TU, Y.-J.; ZHOU, W.; PIRAMUTHU, S. Identifying RFID-embedded Objects in Pervasive Healthcare Applications. *Decision Support Systems*, 2009. v. 46, n. 2, p. 586–593, 2009.

[Want 2006]WANT, R. An Introduction to RFID Technology. *IEEE Pervasive Computing*, 2006. v. 5, n. 1, p. 25–33, 2006.

[Weis et al. 2004]WEIS, S. et al. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. *Lecture Notes In Computer Science*, 2004. v. 2802, p. 201–212, 2004.

[Wu et al. 2009]WU, D.-L. et al. A Brief Survey on Current RFID Applications. In: *Proceedings of the International Conference on Machine Learning and Cybernetics*. [S.l.: s.n.], 2009. v. 4, p. 2330–2335.

[Yang et al. 2005]YANG, J. et al. Mutual Authentication for Low-Cost RFID Systems. In: *Proceedings of the Ecrypt Workshop on RFID and Lightweight Cryptography*. [S.l.: s.n.], 2005. p. 17–24.

[Yu et al. 2005]YU, S. et al. Anti-Collision Algorithm Based on Jumping and Dynamic Searching and its Analysis. *Computer Engineering*, 2005. v. 31, p. 19–20, 2005.

[Zhena, Kobayashi e Shimizu 2005]ZHENA, B.; KOBAYASHI, M.; SHIMIZU, M. Framed Aloha for Multiple RFID Objects Identification. *IEICE-Transactions on Communications*, 2005. v. 88, p. 991–999, 2005.

[Zhou et al. 2004]ZHOU, F. et al. Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems. In: *Proceedings of the 2004 International Symposium on Low Power Electronics and Design (ISLPED)*. [S.l.: s.n.], 2004. p. 357–362.