Universidade Federal de Pernambuco

Centro de Informática

Graduação em Engenharia da Computação

UMA ANÁLISE DA SEGURANÇA DE SISTEMAS RFID

Eduardo Henrique de Carvalho Franklin

TRABALHO DE GRADUAÇÃO

Recife

7 de junho de 2010

Universidade Federal de Pernambuco

Centro de Informática

Eduardo Henrique de Carvalho Franklin

UMA ANÁLISE DA SEGURANÇA DE SISTEMAS RFID

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: *Paulo André da Silva Gonçalves*

Recife

7 de junho de 2010

Trabalho de Graduação sob o título *"Uma Análise da Segurança de Sistemas RFID"*, defendida por Eduardo Henrique de Carvalho Franklin e aprovada em 7 de junho de 2010, em Recife, Pernambuco, pela banca examinadora constituída pelos doutores:

Prof. Dr. Paulo Gonçalves

Centro de Informática - UFPE

Orientador

Prof. Dr. Carlos Ferraz

Centro de Informática - UFPE

AGRADECIMENTOS

Dedico meus sinceros agradecimentos para:

– Deus, por ter me dado a honra de conhecer as pessoas que aqui menciono, que influenciaram diretamente minha formação pessoal e profissional;

– João Henrique, meu pai e melhor amigo, com quem sempre pude contar em todos os momentos, desde os mais felizes até aqueles mais tristes, que superamos juntos. Obrigado por sempre ter buscado me oferecer todo conforto possível e por ter me dado as melhores condições de estudo. Mas, acima de tudo, obrigado por estar sempre comigo;

– Ana Maria, minha doce mãe, que sempre esteve ao meu lado, em todas as situações. Obrigado por ter me levado na garupa da bicicleta quando eu ainda estava nos meus primeiros dias de escola. Obrigado por ter dado voltas na cidade quando eu precisava fazer trabalhos escolares, e quando me levava para lanchar;

– Patricia, minha irmã, que sempre me acompanhou. Obrigado por poder sempre contar com você. Siga em frente, acredito muito no seu sucesso;

– Priscilla, minha namorada, o amor da minha vida. Até hoje não acredito como *no meio de tanta gente eu encontrei você*. Desde então, cada segundo que passamos juntos foi de muita felicidade e muito amor. Ficaremos juntos para sempre;

– Prof. Dr. Paulo Gonçalves, meu orientador nesta monografia. Obrigado por acreditar no meu trabalho e por direcionar meus esforços para áreas promissoras de Redes de Computadores;

– a equipe de Sistemas Embarcados do CESAR, que fez parte da minha formação como profissional. Agradeço a todos, em especial a Paulo Urbano, Fábio Maia, Daniel, Igor, Fabinho, Márcio, Dio e Carina;

– os colegas do CIn. Conhecemos nossos verdadeiros amigos nos momentos de dificul-

dades. Durante todo o curso, foram muitos os momentos difíceis. Felizmente conseguimos tomar caminhos de sucesso. Agradeço a todos, em especial a Mamão, Cabelinho, Reaf, Bagaça, Apebão, Borba, Roda, Espanta, Pablo, Breno e Cleunio.

"O homem superior, ao descansar em segurança,

não esquece que o perigo pode vir.

Quando em um estado de segurança,

ele não esquece a possibilidade de ruína.

Assim, sua pessoa não está em perigo,

e seus Estados e todos os seus clãs são preservados."

—CONFÚCIO (Filósofo chinês, 551 aC - 479 aC)

RESUMO

Os sistemas de identificação por radiofrequência (RFID, do inglês *Radio-Frequency IDentification*) serão cada vez mais populares e massivamente utilizados nos próximos anos. Entretanto, a popularização dessa tecnologia exige um conhecimento detalhado a respeitos dos riscos aos quais seus usuários estarão expostos. Este trabalho realiza uma análise de segurança dos sistemas RFID. Os ataques mais tradicionais, capazes de obter informações privilegiadas do sistema RFID alvo, ou mesmo impedir o correto funcionamento deste, serão estudados e analisados. Também serão abordados os ataques direcionados aos algoritmos de consulta, capazes de manipular a comunicação entre os componentes do sistema RFID. Serão estudados o algoritmo de consulta de Duc, o M^2AP e o $Gen2^+$, bem como suas respectivas fragilidades. Há ainda os ataques contra algoritmos anti-colisão, como a *Blocker-Tag*. Será proposta uma nova modalidade de ataque, denominada *Tag Cloner*, que põe em xeque os algoritmos anti-colisão do tipo árvore. Serão discutidos também os requisitos que um algoritmo deve cumprir para ser imune à *Tag Cloner*.

Palavras-chave: RFID, Segurança, Ataques, Protocolos anti-colisão

ABSTRACT

Radio-Frequency Identification systems will become increasingly popular and widely used in the coming years. However, the popularity of this technology requires a detailed knowledge about the risks to which the users will be exposed. This work makes a security analysis of the RFID systems. First, the traditional attacks will be explored. They are able to obtain privileged information from a RFID system, or even prevent it's correct functioning. The attacks against the query algorithms will also be discussed. They are able to exploit the communication between the components of RFID systems. The Duc's query algorithm, the M^2AP and the $Gen2^+$ will be analyzed, such as their respective weaknesses. Moreover, there are attacks against anti-collision algorithms, as the Blocker-Tag. A new model of attack, called Tag Cloner, will be proposed. It puts into question tree-based anti-collision algorithms. The requirements that an algorithm must meet to be immune to Tag Cloner will also be discussed.

Keywords: RFID, Security, Attacks, Anti-collision protocols

SUMÁRIO

GLOSSÁRIO

AES *Advanced Encryption Standard*. 24

ASP *Active Server Page*. 20

CRC *Cyclic Redundancy Code*. 9, 10

CSS *Cross Site Scripting*. 18

DoD *Department of Defense* ou Departamento de Defesa do Estados Unidos. 7

EEPROM *Electrically-Erasable Programmable Read-Only Memory*. 5

EPC *Eletronic Product Code*. 9

GS1 *Global Standards One*. 7

HMAC *Hashed Message Authentication Code*. 10

HTTP *HyperText Transfer Protocol*. 18

IANA *International Article Numbering Associations*. 7

IDS *InDex pSeudonym*. 28

ISO *International Organization for Standardization*. 7, 11

M^{2}AP *Minimalist Mutual-Authentication Protocol*. 28

MAC *Message Authentication Code*. 10, 15

PIN *Personal Identification Number*. 33

QT *Query Tree*. 40

RFID *Radio-Frequency IDentification* ou identificação por radiofrequência. 1

SGBD Sistema de Gerenciamento de Banco de Dados. 19

SQL *Structured Query Language*. 19

UCC *Uniform Code Council*. 7

LISTA DE FIGURAS

CAPÍTULO 1

INTRODUÇÃO

A computação está cada vez mais onipresente no cotidiano das pessoas. O ser humano tem uma necessidade crescente de rastrear, controlar, identificar e automatizar os objetos que o cercam. A tecnologia RFID (*Radio-Frequency IDentification*) surgiu como uma nova realidade que mudará a forma de interação com o meio que ele está inserido.

1.1 MOTIVAÇÃO

Os sistemas de comunicação por radiofrequência, conhecidos como RFID, têm por objetivo realizar a identificação inequívoca de objetos ou pessoas de forma automática. Ele é composto de leitores e etiquetas eletrônicas. Todos os objetos a serem identificados devem receber uma etiqueta, que armazena uma sequência de bits própria. Os leitores, através de consultas, obtêm esse dados, que revelam quais objetos estão presentes na sua área de cobertura. Além disso, um banco de dados pode auxiliar o leitor a armazenar e recuperar informações dos itens identificados. Juels (2006) define os sistemas RFID como um mecanismo que rotula um objeto explicitamente, facilitando a sua percepção através de dispositivos computacionais.

Tradicionalmente, costuma-se identificar tipos de objetos através de códigos de barra. De acordo com estimativas realizadas por Peris-Lopez *et al.* (2006), cerca de 5 bilhões de códigos de barra são lidos diariamente. Essa tecnologia será substituída pelos sistemas RFID, que trazem consigo muitas vantagens.

A primeira grande vantagem dos sistemas RFID em relação aos códigos de barra é a identificação individual de cada objeto. Considere dois itens idênticos em um supermercado, colocados lado a lado em uma prateleira sob o mesmo indicativo de preço. Neles

estão registrados o mesmo código de barra. Entretanto eles podem ter sido fabricado em cidades diferentes, podem ter sido transportados até o supermercado por empresas diferentes e um deles pode estar com o prazo de validade vencido. Apesar de serem semelhantes, eles não são o mesmo item. Os sistemas RFID são capazes de identificar cada item em particular e recuperar suas informações para o usuário.

A segunda vantagem a ser destacada é a automação do processo de leitura. Os códigos de barra precisam ser corretamente posicionados de acordo com um leitor óptico. Essa tarefa é realizada de forma manual por um ser humano. Os leitores de sistemas RFID, por sua vez, criam um ambiente físico onde as etiquetas RFID podem ser identificadas. Para realizar suas compras, os clientes do supermercado só precisam conduzir seus carrinhos para essa região e todas as etiquetas são lidas automaticamente.

As aplicações dos sistemas RFID são diversas. Hoje, já há rodovias privatizadas controladas por leitores de etiquetas, as quais são instaladas em veículos. Os sistemas RFID também já são utilizados para monitorar livros em bibliotecas e identificar passaportes e outros documentos de viagens. Há até mesmo etiquetas subcutâneas implantadas em animais, para acompanhamento de especialistas. No futuro, um alimento portador de etiqueta poderia informar automaticamente a um forno de microondas o tempo necessário para o seu preparo. A geladeira poderia ser avisada que o prazo de validade deste alimento venceu.

Apesar da tecnologia RFID trazer consigo tantas inovações, há também muitos problemas relacionados à segurança desses sistemas. É necessário um controle a respeito de quem tem acesso ao conteúdo das etiquetas. Também é necessário garantir que pessoas mal-intencionadas não interrompam o processo de identificação de etiquetas. Essas e outras questões ainda devem ser muito bem discutidas para que os sistemas RFID possam ser considerados seguros.

1.2 OBJETIVOS

O objetivo geral deste documento é contribuir para o amadurecimento dos sistemas RFID no aspecto de segurança computacional, através da análise dos ataques já existentes e

da proposta de um novo ataque. Serão discutidos os principais algoritmos utilizados em sistemas RFID e as suas vulnerabilidades.

Para alcançar o objetivo geral, os seguintes objetivos específicos foram definidos:

- Estudar a arquitetura e o funcionamento de sistemas RFID;
- Estudar e analisar os principais ataques contra sistemas RFID;
- Estudar e analisar a segurança dos algoritmos de consulta a etiquetas RFID;
- Estudar e analisar a segurança dos algoritmos anti-colisão de sistemas RFID;
- Propor um ataque eficiente contra algoritmos anti-colisão do tipo árvore.

1.3 ORGANIZAÇÃO

Este documento está dividido em seis capítulos. O Capítulo 2 é uma descrição da tecnologia RFID, abordando os aspectos técnicos, seu processo de padronização e as limitações impostas pelo hardware utilizado. O Capítulo 3 é dedicado à abordagem dos problemas de segurança de sistemas RFID. São descritos em detalhes os ataques mais tradicionais, bem como uma nova categoria de ataque baseada no uso de etiquetas com código malicioso. O Capítulo 4 mostra um estudo comparativo dos principais algoritmos de comunicação por radiofrequência recentemente publicados. Eles têm por objetivo permitir o uso da tecnologia de maneira segura. O Capítulo 5 mostra o conceito de algoritmos anti-colisão, bem como uma proposta de ataque desenvolvida neste trabalho, que mostra-se eficaz contra algoritmos anti-colisão do tipo árvore. O Capítulo 6 traz as conclusões do trabalho.

CAPÍTULO 2

FUNDAMENTOS DE RFID

Para os sistemas de identificação por radiofrequência se tornarem viáveis, é preciso que eles sejam eficientes e de baixo custo. A modelagem desse tipo de sistema deve levar em consideração não só a necessidade de componentes capazes de prover uma identificação rápida e segura, mas também as exigências financeiras para a sua implantação em produtos como uma caixa de fósforo. Para a redução dos custos do projeto, é necessária uma política clara de padronização da tecnologia. Além disso, a interoperabilidade entre sistemas RFID de diferentes fabricantes seria possível caso um padrão universal fosse obedecido. Neste Capítulo, serão abordados a arquitetura e o processo de padronização dos sistemas RFID.

2.1 ARQUITETURA DE SISTEMAS RFID

Os sistemas RFID possuem dois componentes principais: etiquetas e leitores. As etiquetas contêm informação digital capaz de identificar o objeto ao qual está associada. Os leitores são capazes de obter essa informação através de comunicação sem fio.

As etiquetas são microchips que devem ser acoplados a objetos físicos. Elas podem ser classificadas em três grupos: passivas, semi-passivas e ativas. As etiquetas passivas não possuem baterias. Elas capturam a energia emitida pelos leitores, quando estes enviam mensagens, através de um efeito de indução magnética. Want (2004) apresenta uma descrição interessante do processo de obtenção de energia em etiquetas passivas. O autor cita que a corrente alternada na bobina do leitor induz uma corrente na bobina da etiqueta, permitindo o carregamento de um capacitor. Essa energia armazenada será utilizada no funcionamento do circuito da etiqueta. Outras etiquetas possuem baterias

em seu esquema elétrico. Elas são classificadas como ativas quando retiram toda a energia que necessitam de suas baterias. Assim, geralmente possuem um maior raio de alcance na emissão de seus dados. Finalmente, etiquetas semi-passivas são aquelas que utilizam a energia emitida pelos leitores para a comunicação, mas possuem uma bateria que fornece energia para outros módulos de seu circuito.

A estrutura das etiquetas deve possuir três componentes:

- Antena;
- Chip de silício;
- Material de encapsulamento.

No caso de etiquetas passivas, a bobina funciona como antena. O *chip* de silício deve possuir uma memória *on-board* capaz de armazenar o código identificador. Muitos fabricantes de sistemas RFID adotam uma memória do tipo EEPROM, utilizada para leitura e escrita. Isso permite que algumas aplicações sejam desenvolvidas para a etiqueta, sobretudo protocolos de autenticação.

Os leitores são responsáveis por consultar as informações das etiquetas. Em geral, eles possuem uma conexão segura com um banco de dados. Uma vez capturado o conteúdo das etiquetas, os leitores buscam no banco de dados maiores informações a respeito do objeto em questão. Essas informações podem conter o histórico de vendas, número de série, o prazo de validade, a data de fabricação, dentre várias outras características particulares do produto.

A Figura 2.1 exemplifica a infra-estrutura de um sistema RFID. Nela estão representados as etiquetas, o leitor e um banco de dados com uma conexão segura.

2.2 CLASSES DE SISTEMAS RFID

O MIT Auto-ID Center é um grupo de pesquisa da área de sistemas de identificação por radiofrequência. Ele é formado por uma cooperação entre sete universidades, distribuídas entre Europa, Ásia, Austrália e América do Norte. Essas instituições possuem a responsabilidade de, juntamente com a EPCglobal, desenvolver a arquitetura dos sistemas RFID.

Figura 2.1 Sistema RFID

O MIT Auto-ID Center definiu quatro classes de etiquetas RFID.

- Classe 0

  A primeira classe de etiquetas RFID possui somente a capacidade de anunciar sua presença. O conteúdo das etiquetas é programado ainda na fase de fabricação, não sendo possível ao usuário realizar operações de gravação de dados. As etiquetas classe 0 podem ser encontradas em livrarias e bibliotecas. Frequentemente são instaladas na entrada desses estabelecimentos, anunciando a passagem de livros e outros itens, mas sem identificá-los. As etiquetas classe 0 são passivas.

- Classe 1

  As etiquetas classe 1 permitem que o usuário grave dados na memória uma única vez. Após esses dados serem armazenados na etiqueta, eles não podem sofrer modificações, somente sendo possíveis operações de leitura. As etiquetas classe 1 são passivas.

- Classe 2

  As etiquetas classe 2 podem ter seu conteúdo modificado, de acordo com a necessidade do usuário do sistema. Para isso, possuem uma memória regravável, com capacidade de escrita e leitura. Assim, podem ser associadas a diferentes objetos, de acordo com suas configurações. As etiquetas classe 2 são passivas.

- Classe 3

  As etiquetas classe 3 possuem sensores embarcados em sua arquitetura. Dessa forma, são capazes de obter informações do meio onde estão inseridas, armazená-las em sua memória e transmiti-las aos leitores. Essas etiquetas são semi-passivas e possuem memória regravável.

- Classe 4

  As etiquetas classe 4 podem se comunicar umas com as outras. Dessa forma, é possível a construção de uma rede sem fio de etiquetas. Essas etiquetas são ativas e possuem memória regravável.

2.3 PADRONIZAÇÃO DA TECNOLOGIA

A popularização de sistemas RFID e a evolução de seus aplicativos dependem, em grande parte, da padronização desta tecnologia. Há alguns grupos definindo padrões e regulando o uso de RFID, como a EPCglobal Inc. e a *International Organization for Standardization* (ISO). Diversas outras organizações vêm cooperando com o desenvolvimento da tecnologia e o estabelecimento de padrões. Dentre elas, podemos citar Wal-Mart, Tesco (UK) e Departamento de Defesa dos Estados Unidos (DoD).

2.3.1 O Padrão EPCGen1

A EPCglobal faz parte da *Global Standards One* (GS1), uma instituição internacional dedicada ao desenvolvimento e implementação de padrões globais. A GS1 surgiu da união entre a *International Article Numbering Association* (IANA) e do *Uniform Code Council* (UCC), entidades que regulavam o uso de códigos de barra nos Estados Unidos e no resto do mundo.

A definição do padrão EPCglobal Geração 1 foi o primeiro esforço na tentativa de padronizar os protocolos utilizados em sistemas RFID. Nos anos 90, não havia regras específicas para a comunicação entre etiquetas e leitores. Entre 2002 e 2005, as indústrias

passaram a utilizar o chamado EPCGen1.

Segundo o Dr. Chris Diorio, um dos arquitetos-chefes da EPCglobal, a mentalidade que orientava o desenvolvimento do EPCGen1 era simplesmente a substituição dos códigos de barra. A comunicação era unidirecional, ou seja, somente as etiquetas podiam ser lidas. Não era possível enviar comandos a partir dos leitores. As etiquetas utilizadas correspondiam as classes 0 e 1. Assim, a tecnologia não permitia que leitores escrevessem dados nas etiquetas.

O padrão EPCGen1, de acordo com Diorio, foi utilizado como piloto, permitindo que os usuários tivessem contato com a nova tecnologia RFID. Esse padrão continuou no mercado até meados de 2005, quando a segunda geração do padrão da EPCglobal foi definitivamente adotada.

2.3.2 O Padrão EPCGen2

O padrão EPCglobal UHF Classe 1 Geração 2, também chamado de EPCGen2, foi aprovado pela EPCglobal em 2004. Ele busca conciliar as limitações de custo atuais da tecnologia RFID com a necessidade de uma comunicação eficiente e com alguma segurança. A maior parte dos protocolos de comunicação de sistemas RFID propostos por pesquisadores baseiam-se no padrão EPCGen2.

O EPCGlobal define um protocolo de duas camadas: uma física e uma responsável pela identificação de etiquetas. Juntas, elas definem as interações físicas, os comandos e operações necessários para a comunicação e o tratamento de colisões.

A camada física estabelece que toda comunicação é do tipo *half-duplex*. Assim, leitores e etiquetas podem transmitir e receber dados, porém não simultaneamente. Em um determinado instante um dos dispositivos ocupa o papel de transmissor e o outro será receptor. Em outro instante os papéis podem se inverter. Os leitores devem realizar a transmissão de dados para as etiquetas através da modulação de sinais de radiofrequência. As etiquetas devem ser passivas, recebendo dos leitores a energia necessária para seu funcionamento.

A memória das etiquetas deve ser logicamente separada em quatro regiões. A primeira

8 bits	28 bits	24 bits	36 bits
Header	Fabricante	Classe do Objeto	Número de Série

Figura 2.2 Eletronic Product Code (EPC)

região, a memória reservada, contém duas senhas de 32 bits. Uma delas, denominada *kill password*, é utilizada por leitores para desabilitar a etiqueta. A outra senha, denominada *access password*, é utilizada na autorização da escrita e leitura na memória. A segunda região é a memória EPC. Ela contém os parâmetros de um CRC de 16 bits, além de 16 bits de controle de protocolo e o identificador único da etiqueta, de 32 bits, denominado *Eletronic Product Code* (EPC). A Figura 2.2 apresenta o identificador EPC, que é composto por um cabeçalho (*header*), o fabricante da etiqueta, a classe do objeto e o número de série. A terceira região é a memória TID, que possui recursos para exibir as funcionalidades da etiqueta para os leitores. A região final de memória consiste em um espaço de armazenamento de dados específicos do usuário.

A camada de identificação de etiquetas define que os leitores devem interagir com as etiquetas através de três operações básicas: *Select* (para escolher uma etiqueta), *Inventory* (para identificar a etiqueta) e *Access* (para escrever ou ler na memória da etiqueta). Essa camada trata ainda do problema de colisão entre as mensagens de etiquetas. Quando um leitor recebe respostas de várias etiquetas simultaneamente, ocorre interferência entre os sinais, impossibilitando a correta identificação dos objetos. Para evitar esse problema, a camada de identificação especifica um algoritmo anti-colisão a ser utilizado. No Capítulo 5 os algoritmos anti-colisão serão analisados com maiores detalhes.

O *harware* das etiquetas devem incluir um gerador de números pseudo-randômico e um CRC, ambos de 16 bits. Esses módulos serão utilizados para prover segurança à comunicação entre etiquetas e leitores, auxiliando na verificação da integridade das mensagens e na autenticação dos remetentes.

2.3.2.1 O Gerador de Números Pseudo-Randômico

Segundo Duc *et al.* (2006), um número randômico é aquele que é retirado de um conjunto de n números com uma probabilidade de exatamente n^{-1}. A geração de números pseudo-aleatórios é um recurso computacional que oferece uma sequência de números aproximadamente randômicos. Ela se baseia em uma função que utiliza um número raiz. À medida que a probabilidade de escolha de um dado número do domínio da função aproxima-se de n^{-1}, a função aproxima-se de uma geração de números verdadeiramente aleatórios.

Burmester *et al.* (2009) apresentam três critérios definidos pela EPCglobal para que uma função seja utilizada na geração de números aleatórios do EPCGen2. São eles:

- Probabilidade de escolha

  A probabilidade de um número pseudo-randômico RN ter um valor conhecido RN_0 é limitada por:

  $0,8/2^{16} < Prob(RN = RN_0) < 1,25/2^{16}$.

- Geração de sequências idênticas

  Dada uma população de 10.000 etiquetas, a probabilidade de duas ou mais delas gerarem a mesma sequência de RN16s é inferior a 0,1%.

- Previsão do número seguinte

  A probabilidade de prever-se o número seguinte de uma sequência, dados os números anteriores, é inferior a 0,025%.

2.3.2.2 *Cyclic Redundancy Code*

Na segurança computacional tradicional, é comum a utilização de códigos *checksum* na verificação da integridade dos dados transmitidos. São exemplos de códigos *checksum* a função *hash*, o MAC e o HMAC. No caso do EPCGen2, o código *checksum* utilizado é o *Cyclic Redundancy Code*, conhecido por CRC. Ele deve ser calculado e anexado ao

quadro a ser transmitido. Após a etapa de recepção, deve ser verificado para que seja confirmada a ausência de alterações.

De acordo com Tanenbaum (2003), o cálculo do CRC se baseia no tratamento de strings de bits como representações de polinômios com coeficientes 0 e 1 apenas. Por exemplo, 110001 tem seis bits, portanto representa um polinômio de seis termos com os coeficientes 1, 1, 0, 0, 0 e 1: $x^5 + x^4 + x^0$. O autor explica que, antecipadamente, o transmissor e o receptor devem concordar em relação a um polinômio gerador, $G(x)$. O transmissor deve, então, acrescentar um total de verificação ao final do quadro a ser enviado, de forma a torná-lo divisível por $G(x)$. Quando obtiver o quadro, o receptor tentará realizar a divisão. A existência de resto indica que houve um erro de transmissão. Para o padrão EPCGen2, utiliza-se um *checksum* de 16 bits.

2.3.3 O Padrão ISO 18000

A *International Organization for Standardization*, conhecida como ISO, é a mais respeitada entidade mundial de determinação de padrões. Ela também se envolveu no processo de padronização da tecnologia RFID, publicando a chamada série ISO 18000. Segundo Zebra (2003), trata-se de um conjunto de documentos que descrevem um padrão de identificação e gerenciamento de itens, utilizando sistemas que operam em diversas frequências.

O padrão ISO 18000 se restringe à especificação do protocolo de comunicação sem fio, denominado *Air Interface Protocol*. Ao contrário dos padrões estabelecidos pela EPCglobal, no padrão da ISO não há nenhuma consideração a respeito da arquitetura dos componentes dos sistemas RFID. Toda a implementação física é deixada para a indústria. Ao restringir seu padrão à comunicação sem fio, a ISO tentou torná-lo mais abrangente e coerente com futuras evoluções da tecnologia RFID.

Na verdade, há uma grande discussão quanto à adoção do padrão EPCGen2 e o padrão ISO 18000. Tuner (2003) discute essa guerra de padrões. Devido à abrangência do padrão ISO, é possível que algumas implementações obedeçam o *Air Interface Protocolo*, mas ignorem completamente as especificações da EPCglobal. Essas diferenças entre as padronizações podem gerar incompatibilidades entre diferentes desenvolvedores. O ideal

seria que as duas instituições entrassem em um comum acordo, em prol da popularização da tecnologia.

2.4 RESUMO

Os sistemas RFID possuem uma arquitetura bastante simples, baseada na interação entre etiquetas e leitores. As etiquetas devem armazenar informações capazes de identificar inequivocamente os itens aos quais estão acopladas. Os leitores devem realizar consultas às etiquetas e buscar em um banco de dados outras informações a respeito delas.

A energia necessária para o funcionamento das etiquetas passivas é capturada dos leitores, quando estes enviam mensagens. Já as etiquetas ativas possuem em seu esquema elétrico uma bateria, que fornece toda a energia necessária. Há ainda as etiquetas semi-passivas, que utilizam a energia emitida pelos leitores para a comunicação e a energia de uma bateria para os demais fins. O MIT Auto-ID Center definiu uma classificação das etiquetas. Assim, elas são distribuídas em cinco classes, de acordo com suas funcionalidades.

A padronização dos sistemas RFID é fundamental para a popularização desta tecnologia. A EPCglobal foi pioneira nesse processo, definindo o padrão EPCGen1. Neste primeiro momento, os sistemas RFID eram meros substitutos dos códigos de barra. Não era possível obter maiores informações a respeito dos itens portadores de etiquetas. A geração seguinte, denominada EPCGen2, trouxe consigo novas funcionalidades para etiquetas e leitores. A comunicação passou a ser bidirecional, os leitores passaram a escrever em etiquetas e adotou-se o conceito de acompanhamento dos produtos rotulados.

A ISO também se envolveu no processo de padronização da tecnologia RFID. Entretanto, limitou-se a definir regras para a comunicação sem fio. Assim, não foi estabelecido nenhum requisito para a arquitetura do sistema. Como consequência, criou-se uma disputa entre os padrões ISO e EPCGen2. A adoção definitiva de um padrão único representaria um grande avanço para a tecnologia RFID.

CAPÍTULO 3

SEGURANÇA DE SISTEMAS RFID

A popularização da tecnologia RFID pode trazer consigo, além de seus benefícios, uma série de riscos relacionados à segurança de seus usuários. As ameaças podem surgir de terceiros que observam ou afetam as consultas por radiofrequência, mas também podem surgir das próprias etiquetas RFID, preparadas para atacar leitores vulneráveis. Neste Capítulo, são descritas as principais técnicas de ataque à segurança de sistemas RFID. Muitas delas são variações de ataques conhecidos da segurança computacional tradicional. Também são mostrados os ataques baseados em conteúdo malicioso de etiquetas que podem transformar qualquer objeto que as possua em uma verdadeira fonte de risco a todo o sistema.

3.1 ATAQUES A UM SISTEMA RFID

Há uma grande preocupação à respeito da segurança de sistemas RFID como um todo. Nesta seção, os ataques serão divididos de acordo com sua origem: externa e interna. Os ataques de origem externa são realizados por terceiros que observam o sistema RFID e tentam explorá-lo. Os ataques de origem interna são realizados por elementos do próprio sistema, tipicamente etiquetas com conteúdo malicioso.

3.1.1 Ataques de Origem Externa

Em sistemas RFID, podemos classificar como ataques de origem externa aqueles que têm como objetivo a obtenção de informações, por terceiros, de forma não-autorizada durante a comunicação entre etiqueta e leitor. A seguir, serão abordados os principais ataques de origem externa.

3.1.1.1 Escuta

O ataque de escuta (*eavesdropping*) consiste em interceptar a comunicação entre uma etiqueta RFID e um leitor autorizado. Para efetuar o ataque, é necessário obedecer as restrições de proximidade impostas pela potência do sinal de radiofrequência.

De acordo com Hancke (2008), o sucesso do ataque pode ser alcançado com um conjunto de *hardware* de baixo custo. Segundo o autor, o ataque pode ser efetuado com um receptor RFID associado a uma antena e um módulo de armazenamento de dados, os quais podem ser projetados por cerca de 150 libras.

Os ataques de escuta são bem conhecidos pelos especialistas de segurança e, geralmente, são solucionados com a utilização de algoritmos criptográficos. Entretanto, as limitações de *hardware* da maioria das etiquetas RFID tornam essa alternativa inviável.

3.1.1.2 *Hotlisting*

Assim como na escuta, o *hotlisting* consiste em capturar a transmissão de dados entre uma etiqueta e um leitor autorizado. No *hotlisting*, conforme citado por Sun e Ting (2009), o atacante possui uma lista de palavras-chave e está interessado em dados a ela relacionados. Essa lista pode constituir, por exemplo, de um subconjunto de livros de uma biblioteca. Assim, o autor pode ser avisado quando algum dessas livros estiver sendo alugado.

Através do *hotlisting*, o atacante consegue progressivamente descrever o comportamento de um usuário do sistema RFID, obervando a forma como interage com as etiquetas que constam na lista de palavras-chave. Esse processo pode representar uma ameaça à privacidade dos usuários de etiquetas RFID.

3.1.1.3 Ataque de Repetição

Após realizar uma escuta e capturar uma mensagem entre a etiqueta e um leitor, um atacante pode decidir reproduzi-la, reenviando-a ao destinatário em um momento posterior.

Esse procedimento é conhecido como ataque de repetição e está descrito em Peris-Lopez *et al.* (2006).

De forma geral, os ataques de repetição são solucionados com técnicas criptográficas como MAC (*Message Authentication Code*). Outra alternativa é a aplicação de uma técnica de sincronização e registro de tempo, conhecida como *Timestamp*. Essas duas técnicas estão descritas em detalhes em Stallings (2008), entretanto não podem ser aplicadas às etiquetas RFID devido às limitações de *hardware*.

3.1.1.4 Clonagem

A clonagem é uma das maiores ameaças a sistemas RFID. Ela é citada em Juels (2006), Duc *et al.* (2006) e Sun e Ting (2008). Caso um atacante consiga capturar o conteúdo de uma etiqueta, ele pode realizar uma cópia deste e armazená-lo em um segunda etiqueta, vazia. Assim, realizará um ataque de clonagem que resulta na perda de unicidade na identificação da etiqueta da vítima.

Uma vez que o sistema de identificação RFID se baseia na identificação inequívoca de suas etiquetas, o ataque de clonagem torna não-confiável qualquer registro da vítima posterior à sua ocorrência.

3.1.1.5 Rastreamento de Etiqueta

O ataque denominado rastreamento de etiqueta, descrito em Sun e Ting (2009), é um dos ataques mais comuns quando se trata de sistemas RFID. Ele consiste em identificar, sem qualquer permissão, a presença de uma etiqueta específica em um dado ambiente físico.

Através do rastreamento de etiqueta, o atacante é capaz de registrar e posteriormente gerenciar a localização de etiquetas. Dessa forma, é possível rastrear a trajetória da etiqueta, monitorando os locais por onde o objeto a ela associado esteve. O rastreamento pode ser realizado através de um ataque passivo, que observa as mensagens trocadas entre etiquetas e leitores, ou de um ataque ativo, manipulando o conteúdo das próprias etiquetas.

3.1.1.6 Forjamento de Dados

O ataque de forjamento de dados pode ser efetuado em etiquetas que possuem memória com capacidade de escrita e leitura. O atacante pode conseguir permissão para realizar escritas na memória, forjando seus dados e modificando o conteúdo da etiqueta. Sun e Ting (2009) citam o ataque como uma uma das ameaças a sistemas RFID.

Suponha, por exemplo, que em um estabelecimento comercial, além da identificação do objeto, a etiqueta contenha outras informações, como o preço. O forjamento desses dados pode resultar em grandes prejuízos financeiros para o proprietário desse estabelecimento.

3.1.1.7 Negação de Serviço

O ataque de negação de serviço consiste em evitar que usuários legítimos tenham acesso a um dado serviço. Tradicionalmente, baseia-se no envio de grandes quantidades de pacotes inúteis ao servidor do serviço, consumindo os recursos computacionais deste e impedindo o acesso dos demais usuários.

Juels *et al.* (2003) propuseram uma alternativa para evitar a leitura de etiquetas utilizando um dispositivo denominado *Blocker Tag*. Ele funciona criando uma região física na qual os leitores não conseguem completar consultas às etiquetas. O uso mal-intencionado da *Blocker Tag* pode representar um ataque de negação de serviço, evitando a identificação das etiquetas pelo leitor. No Capítulo 5, a *Blocker Tag* e as vulnerabilidades por ela exploradas serão estudadas com maiores detalhes.

3.1.1.8 *Spoofing*

O ataque de *spoofing* está descrito em Tanenbaum (2003) no contexto de segurança da informação. Ele é semelhante ao ataque de rastreamento, mas possui um objetivo diferente. Nesse ataque, o atacante pretende enviar mensagens de forma que estas indiquem uma etiqueta válida como remetente. Ou seja, o atacante constrói uma mensagem e a envia, como se na realizada fosse uma outra etiqueta.

3.1.2 Ataques de Origem Interna

Um outro tipo de ataque foi pouco estudado no meio acadêmico: a utilização do conteúdo das etiquetas como origem de ataques. Ameaças desse tipo demonstram que a obtenção de informações de etiquetas podem constituir um risco a leitores e bancos de dados que compõem o sistema de identificação. Todos os portadores de etiquetas, como embalagens, animais de estimação ou objetos pessoais, passam a representar possíveis fontes de ataque e disseminação de código malicioso.

A utilização de etiquetas com conteúdo malicioso é uma forma de ataque inicialmente exposta por Rieback *et al.* (2006). A seguir, serão abordados as principais formas de ataque baseados em etiquetas e o surgimento de vírus de RFID.

3.1.2.1 *Buffer Overflow*

O *Buffer Overflow* é considerado por Cowan *et al.* (2000) a vulnerabilidade da década. Essa modalidade de ataque é a mais comum quando se trata de penetração de redes de computadores. Entre os códigos que se mostraram vulneráveis ao *Buffer Overflow* estão: *syslog*, *splitvt*, *sendmail* 8.7.5, *Linux/FreeBSD*, *Xt library*, além de uma infinidade de códigos legados, sem as proteções necessárias para evitar esse ataque. Além disso, o *Buffer Overflow* foi usado em ataques como o Vírus de Morris (1988), *Code Red* (2001) e *SQL Slammer* (2003).

Tipicamente objetivo desse ataque é injetar e executar código malicioso em uma máquina alvo. Em geral, o código injetado contém uma chamada ao *shell* do sistema. O resultado é o controle total ou parcial da vítima.

O ataque se baseia na injeção de uma quantidade de dados maior que a esperada em um *buffer* de um sistema. Algumas linguagens possuem funções que realizam cópias de dados sem verificar os limites do *buffer* de destino.

Um exemplo típico é a função *strcpy*, da linguagem C. Quando ela é utilizada na tentativa de armazenar uma cópia de uma *string* maior que aquela suportada pela variável destino, os dados em excesso estouram o *buffer*. Toda a informação que não foi arma-

zenada no espaço correto da memória passa a ocupar os endereços seguintes, alterando o conteúdo da pilha de chamadas de função do sistema. Dessa forma, o atacante pode direcionar o fluxo de execução para o endereço desejado. Se entre os dados inseridos estiver um comando de chamada ao *shell* do sistema, este pode ser executado. Outra opção é realizar chamadas a variáveis do sistemas que contenham o endereço da *shell*.

Toda a implementação de um *exploit* baseado em *Buffer Overflow* pode ser encontrada em One (1996). O *Buffer Overflow* também é descrito em detalhes em Cowan (2000).

Rieback *et al.* (2006) sugerem que etiquetas RFID podem conter códigos maliciosos baseados em *Buffer Overflow*, projetados para corromper o *middleware* da arquitetura de sistemas de RFID. Caso a etiqueta envie um grande volume de mensagens, ela seria capaz de provocar um estouro do *buffer* no nível de aplicação do *middleware* RFID. Isso provocaria um estouro da pilha de chamadas da aplicação e uma execução imprevisível do sistema.

3.1.2.2 Injeção de Código

Uma outra categoria de ataques se caracteriza pela inserção de código malicioso em aplicações, utilizando linguagem de *script*, como *Javascript* e *Perl*. O objetivo desses ataques é roubar *cookies* de um usuário, capazes de realizar sua autenticação. Assim o atacante pode agir, passando-se pelo usuário na rede, realizando outras operações indevidas.

Dentre os ataques de Injeção de Código, o *Cross Site Scripting* (CSS) é especialmente interessante. Segundo Klein (2002), o CSS é um dos tipos de ataque mais comuns realizados por hackers no intuito de obter informações através de aplicações web.

O ataque CSS se baseia na inserção de parâmetros maliciosos passados a um *site* vunerável. Geralmente, um *script* é responsável pela recepção das requisições HTTP, repassando as variáveis do usuário ao *site* em questão. Considere a seguinte requisição a um *site* vunerável, cujo *script* é denominado welcome.cgi:

```
GET /welcome.cgi?name=<script>window.open("http://www.atacante.site/collec
```

```
    t.cgi?cookie="%2Bdocument.cookie)</script>
```

A página de resposta seria da forma:

```
<html>
<title>Bem-vindo! </title>
Olá
<script>window.open("http://www.atacante.site/collect.cgi?cookie=
    "%Bdocument.cookie)</script>
<br>
Bem-vindo ao nosso sistema!
</html>
```

Assim, ao invés de seu nome, o atacante forneceu um comando *Javascript* capaz de enviar os *cookies* do usuário para seu *site* particular. Analogamente, etiquetas RFID cujo conteúdo consiste em comandos de *script*, podem efetuar ataques de injeção de código no *middleware* de sistemas RFID. Segundo Rieback *et al.* (2006), caso as aplicações RFID utilizem protocolos web para realizar consultas ao banco de dados, é possível que seu *middleware* interprete comandos de *scripts*. Nesse caso, o ataque CSS seria realizável.

3.1.2.3 Injeção SQL

A linguagem SQL (*Structured Query Language*), implementada pela IBM Research como uma interface para um sistema experimental de banco de dados, tornou-se um padrão para o desenvolvimento de banco de dados relacionais. Dentre os Sistemas de Gerenciamento de Banco de Dados (SGBD) comerciais que adotaram a SQL, ou suas variações, estão Oracle, DB2 e SQL/DS da IBM, SQL Server e ACCESS da Microsoft, INGRES, INFORMIX e SYBASE. A facilidade de uso e as fácil adaptação dos sistemas SQL a diferentes SGBD foram os principais motivos dessa popularização da linguagem.

As consultas SQL consistem em uma série de declarações que representam restrições. Os elementos do banco de dados que obedecerem às especificações da consulta constituem um conjunto único, que é retornado ao usuário. Segundo Anley (2002), a Injeção SQL

ocorre quando o atacante é capaz de inserir uma série de declarações SQL em uma consulta, através da manipulação dos dados de entrada de uma aplicação.

Considere a seguinte consulta SQL:

```
SELECT id, nome, sobrenome, endereco from EMPREGADO
WHERE nome = ‘John’ AND senha=‘abc’
```

Essa consulta retornará os campos *id*, *nome*, *sobrenome* e *endereco* de um elemento da tabela EMPREGADO cujo nome e senha são 'John' e 'abc', respectivamente. Note que as cadeias de caracteres deve ser limitadas por aspas simples. É comum a construção de consultas SQL a partir de formulários de páginas web, utilizando ASP, por exemplo. O exemplo abaixo demonstra uma consulta SQL construída a partir de um formulário:

```
var sql = ‘‘SELECT id, nome, sobrenome, endereco from EMPREGADO
WHERE nome = ‘ ’’ + username + ‘‘ ’ and senha = ‘ ’’ + password + ‘‘ ’ ’’
```

Dessa forma, o nome e a senha tornam-se variáveis, a serem definidas pelo usuário. Entretanto, o usuário pode se aproveitar dos campos do formulário para inserir código malicioso no banco de dados. Considere que o usuário forneça os seguintes dados:

Username: '; DROP table EMPREGADO−−

Password:

Assim, o conteúdo da variável *var* seria:

```
SELECT id, nome, sobrenome, endereco from EMPREGADO
WHERE nome = ‘’ ; DROP table EMPREGADO--  ’ and password = ‘’
```

O código SQL possui agora um novo comando, que resultará na eliminação da tabela EMPREGADO, transformando o restante do código em comentário com o uso do comando −−.

Etiquetas RFID podem armazenar código malioso capaz de executar um ataque de Injeção SQL, exatamente como o ataque tradicional acima descrito. Uma leitura desse conteúdo poderia corromper o banco de dados utilizado pelo sistema RFID. Como a porção de código necessário para efetuar esse ataque é bastante reduzida, a pequena

memória das etiquetas já seria suficiente para interromper o funcionamento do servidor e de toda a rede RFID.

3.1.2.4 Vírus de RFID

Uma aplicação interessante das técnicas de ataque expostas nesta seção é o desenvolvimento de vírus de RFID. Segundo Stallings (2008), um vírus é um *software* que pode infectar outros programas, modificando-os. A modificação inclui uma cópia do programa de vírus, que pode então prosseguir para infectar outros programas. No caso de sistemas RFID, a disseminação de vírus acontece através da alteração do conteúdo das etiquetas. Portanto, pode-se dizer que uma etiqueta está infectada com um vírus quando esta possui um conteúdo capaz de se replicar em outras etiquetas semelhantes.

Suponha uma situação em que um banco de dados deve atualizar o conteúdo de um conjunto de etiquetas periodicamente. O objetivo dessa operação poderia ser, por exemplo, substituir o produto portador da etiqueta, em um supermercado. Para isso, o banco de dados deve atuar permitindo que o leitor adquira permissão de escrita nas etiquetas e fornecendo um novo conteúdo para cada uma delas. Nessa situação, suponha que uma das etiquetas esteja contaminada por um vírus de RFID. Essa etiqueta particular poderia estar preparada para realizar um ataque de Injeção SQL, capaz de modificar os registros do banco de dados. Assim, o banco de dados, ao invés de armazenar em suas tabelas possíveis valores de identificadores, poderia passar a armazenar réplicas do conteúdo da etiqueta contaminada. Quando fosse necessário realizar uma atualização de etiqueta, o leitor receberia um código virulento para ser inserido nas demais etiquetas, espalhando o vírus.

Há ainda algumas alternativas para otimizar esse modelo de vírus. Ao invés de alterar o conteúdo das tabelas do banco de dados, o ataque de Injeção SQL poderia criar procedimentos específicos. Assim, o conteúdo malicioso não seria exibido diretamente nas tabelas, dificultando sua identificação. Também é possível utilizar parâmetros na operação de replicação de código. Uma opção seria uma marca temporal, que gerasse um binário em particular que fosse dependente do instante da operação. Assim, o vírus

passaria a ser representado por códigos que, apesar de realizarem as mesmas operações, possuem padrões binários diferentes. Esse tipo de vírus é chamado de polimórfico.

3.2 RESUMO

A preocupação com a segurança de sistemas RFID é crescente. Ao passo em que a tecnologia evolui, os atacantes tornam-se cada vez mais criativos e eficazes. Há várias estratégias de ataque contra sistemas RFID.

Neste Capítulo, os ataques foram classificados de acordo com a suas origens: externa e interna.

Os ataques de origem externa são realizados por terceiros que, sem qualquer permissão, obtêm informações das etiquetas ou alteram seus conteúdos. Entre esses ataques, estão a escuta, o *hotlisting*, os ataques de repetição, a clonagem, o rastreamento de etiquetas, o forjamento de dados, a negação de serviço e o *spoofing*.

Os ataques de origem interna representam uma ameaça que surge de uma fonte inesperada: as próprias etiquetas do sistema. Ao serem lidas, elas transmitem ao leitor o código malicioso que possuem, efetuando o ataque. Ou seja, a simples leitura das etiquetas representam um risco a todo o sistema RFID. Todos os portadores de etiquetas, como objetos ou animais, passam a ser considerados potenciais atacantes. Os principais ataques de origem interna são adaptações de ataques bem conhecidos da segurança computacional. São eles: *Buffer Overflow*, Injeção de Código e Injeção SQL.

Uma aplicação interessante dos ataques acima citados é o surgimento de vírus de RFID. Esse tipo de ameaça é capaz de, a partir de uma etiqueta contaminada, difundir o conteúdo malicioso para as demais etiquetas. A tendência é que os vírus de RFID tornem-se cada vez mais complexos. As alternativas para otimização do modelo de vírus podem torná-los cada vez mais difíceis de serem identificados.

CAPÍTULO 4

ANÁLISE DOS ALGORITMOS DE CONSULTA

Os algoritmos de consulta constituem a mais importante funcionalidade dos sistemas RFID. Eles devem permitir que um leitor autorizado obtenha o conteúdo de uma determinada etiqueta. Atualmente, a maior parte dos pesquisadores se dedica ao estudo de algoritmos de consulta de etiquetas desenvolvidas segundo o padrão EPCGen2. Neste Capítulo, será realizada uma análise de alguns dos principais algoritmos de consulta propostos no meio acadêmico.

4.1 INTRODUÇÃO AOS ALGORITMOS DE CONSULTA

O procedimento de consulta a etiquetas de um sistemas RFID é um tópico que vem sendo bastante discutido no meio acadêmico. Ainda não há um consenso a respeito do melhor protocolo para essa operação. Na verdade, muitos pesquisadores têm divulgado suas próprias propostas de algoritmos consultas. Entretanto, a cada nova idéia, surgem também novos possíveis ataques.

É de extrema importância a análise das propostas dos algoritmos de consulta, pois a discussão a respeito de suas vulnerabilidades resultam no amadurecimento da teoria de comunicação via radiofrequência. Neste Capítulo, serão estudados alguns dos principais algoritmos de consulta de etiquetas RFID. Foram selecionados algoritmos que contribuíram para a evolução da teoria RFID. O algoritmo de Duc (2006) valorizou os conceitos de criptografia minimalista, apresentados por Juels (2006). O algoritmo de Peris-Lopez *et al.* (2006) trouxe consigo o conceito de autenticação mútua. Por sua vez, o algoritmo de Sun e Ting (2009) é uma proposta recente um pouco mais complexa.

A seguir, serão analisados o funcionamento de cada um desses algoritmos, bem como

as diretrizes em que se basearam.

4.2 CRIPTOGRAFIA MINIMALISTA

Atualmente, as etiquetas RFID de baixo custo não dispõem de recursos computacionais suficientes para executar algoritmos de criptografia moderna. Segundo Peris-Lopez *et al.* (2006), as etiquetas RFID de baixo custo possuem somente entre 5 a 10 mil portas lógicas. Entretanto, o número de portas dedicadas à execução de funções de segurança é ainda menor, variando entre 250 a 3000. Para efeito de comparação, considere o *Advanced Encryption Standard* (AES), um algoritmo popular de criptografia baseada no uso de chave simétrica, adotado como padrão criptográfico pelo governo norte americano. A implementação padrão do AES exige entre 20 a 30 mil portas lógicas.

Diante das restrições de recursos computacionais das etiquetas RFID, Juels (2006) definiu um novo conceito de criptografia para sistemas RFID, denominado Criptografia Minimalista. O objetivo desse modelo é possibilitar a garantia de segurança desses sistemas sem utilizar padrões criptográficos modernos.

A Criptografia Minimalista de Juels baseia-se na utilização de pseudônimos, que devem ser armazenados nas etiquetas em forma de lista circular. A cada consulta, realizada por leitores, as etiquetas respondem com o próximo pseudônimo da lista. Somente um leitor autorizado seria capaz de relacionar os pseudônimos à verdadeira identidade do objeto consultado. Leitores não-autorizados permaneceriam incapazes de identificar as etiquetas.

Os ataques considerados no modelo de Criptografia Minimalista possuem configurações mais realísticas, a exemplo da proximidade física às etiquetas. Juels sugere que as restrições físicas associadas ao modelo de sistemas RFID constituem uma vantagem ao desenvolvedor da arquitetura do sistema. Para fortalecer essa vantagem, ele propõe um mecanismo de retardo no fornecimento de informações por parte das etiquetas, com o objetivo de forçar o atacante a permanecer fisicamente próximo ao objeto cuja etiqueta será lida. Esse mecanismo é chamado *throttling*.

A utilização de lista de pseudônimos traz consigo uma limitação: a memória reduzida

de etiquetas RFID seria capaz de armazenar somente uma pequena lista de pseudônimos, limitando as garantias de segurança deste modelo. Para minimizar esse problema, Juels sugere que os leitores, após uma etapa de autenticação, renovem a lista de pseudônimos das etiquetas.

O modelo de Criptografia Minimalista vem recebendo grande aceitação na comunidade de pesquisa de tecnologia RFID. Na busca de um protocolo seguro que exija pouca capacidade computacional por parte das etiquetas, diversos pesquisadores adotaram o conceito de Criptografia Minimalista em suas propostas. Entretanto não há qualquer prova que essa abordagem irá satisfazer as necessidades de segurança dos usuários.

4.3 O PROTOCOLO DE DUC

O protocolo proposto por Duc *et al.* (2006) utiliza os recursos de geração de número pseudo-aleatório e código CRC das etiquetas. Na tentativa de prevenir a captura dos dados por terceiros, o protocolo procura utilizar criptografia baseada em operações XOR. Uma chave deve ser compartilhada pela etiqueta e pelo banco de dados. Essa chave deve ser de uso único, a cada sessão, de forma semelhante ao tradicional *One-time Pad*[1]. Assim, uma chave compartilhada diferente deve ser utilizada na sessão seguinte. Para isso, a etiqueta e o banco de dados devem realizar a geração de números pseudo-aleatórios através de um mesmo valor raiz.

O protocolo de Duc dispensa a autenticação do leitor perante a etiqueta. Ao invés disso, o leitor deve receber os dados criptografados e autenticar-se perante o banco de dados. Dependendo de seu nível de privilégio, o leitor poderá receber do banco de dados as informações da etiqueta.

[1] O *One-time Pad* é um algoritmo de criptografia baseado em uma chave descartável. Claude Shannon provou que o One-time Pad possui uma propriedade denominada segredo perfeito, por não revelar nenhuma informação a respeito do texto claro. Maiores informações em Stallings (2008).

4.3.1 O Funcionamento do Protocolo de Duc

O protocolo de Duc pode ser dividido em três etapas: consulta da etiqueta, acesso à etiqueta e atualização de chave.

4.3.1.1 Consulta da etiqueta

Na primeira etapa, é realizada a leitura do conteúdo da etiqueta. Para isso o leitor inicia a consulta, enviando um valor r' à etiqueta. Esse valor r' e um desafio r gerado pela etiqueta são utilizados em uma criptografia baseada em operações XOR. O valor r e o resultado da operação, M_1, são retornados para o leitor, que autentica-se com o banco de dados e envia-lhe M_1, r e r'. O banco de dados deve possuir a chave da criptografia realizada, efetuando a decifragem e devolvendo ao leitor o conteúdo da etiqueta. Esse processo é descrito pelos passos a seguir:

Passo 1 R $\rightarrow$ T: *Pedido de consulta*, r'

Passo 2 T $\rightarrow$ R: $M_1 = \text{CRC}(1||\text{EPC}\oplus\text{r}\oplus\text{r'}) \oplus K_i$, r

Passo 3 R $\leftrightarrow$ S: *Autenticação mútua*

Passo 4 R $\rightarrow$ S: M_1, r, r'

Passo 5 S $\rightarrow$ R: *Informação do objeto*

4.3.1.2 Acesso à etiqueta

Quando o leitor deseja obter permissão para escrever na etiqueta, a autenticação perante o banco de dados não é suficiente. O leitor deve se autenticar perante a etiqueta. Para isso, o banco de dados gera o código CRC utilizando o identificador da etiqueta, o código PIN de segurança e o valor r. O código CRC e a chave compartilhada participam então de uma operação XOR. O resultado é enviado ao leitor que, por sua vez, encaminha à etiqueta. Esta verificará se o CRC foi corretamente calculado. Em caso positivo, o leitor recebe autorização de escrita. Esse processo é representado pelos passos a seguir:

Passo 6 S → R: $M_2 = \text{CRC}(1||EPC||PIN||r) \oplus K_i$

Passo 7 R → T: M_2

Passo 8 T: Verificar se $M_2 \oplus K_i = \text{CRC}(1||EPC||PIN||r)$

4.3.1.3 Atualização de chave

Na etapa final, o leitor envia um sinal de encerramento de sessão para a etiqueta e para o banco de dados. Estes, por sua vez, ao receberem o sinal, utilizam a função de geração de números pseudo-aleatórios *f(.)* para atualizar a chave em comum. A raiz dessa função e a forma de geração dos valores devem ser previamente acordados entre etiqueta e banco de dados, de forma que suas chaves sempre sejam coincidentes. Esse processo é descrito conforme os passos a seguir:

Passo 9 R → T, S: *Fim de Sessão*

Passo 10 T: $K_{i+1} = \text{f}(K_i)$

Passo 11 S: $K_{i+1} = \text{f}(K_i)$

4.3.2 O Ponto Fraco do Protocolo de Duc

O protocolo de Duc apresenta alguns pontos de vulnerabilidade, cmo mostrados em Chien e Chen (2006).

Suponha que, na etapa final, um dos comandos de encerramento de sessão sejam interceptados. O resultado é que o compartilhamento de chaves entre a etiqueta e o banco de dados é desincronizado. A partir de então, a etiqueta e o leitor não poderão mais se autenticar. Assim o ataque de negação de serviço foi bem-sucedido.

Além disso, se ambas as mensagem de encerramento de sessão forem interceptadas, não haverá mudança da chave. Assim, os dados (M_1, r, r') ainda são válidos e podem ser utilizados por uma etiqueta maliciosa para simular a etiqueta original.

4.4 M^2AP: UM PROTOCOLO DE AUTENTICAÇÃO MÚTUA

Peris-Lopez *et al.* (2006) propuseram um protocolo para aplicações RFID que se baseia em uma autenticação mútua entre a etiqueta e o leitor de radiofrequência. Esse protocolo foi denominado Minimalist Mutual-Authentication Protocol (M^2AP).

O M^2AP utiliza o conceito de pseudônimos. Dessa forma, cada etiqueta compartilha com o leitor um pseudônimo variável, denominado *InDex-pSeudonym* (IDS), além de quatro chaves secretas K_1, K_2, K_3 e K_4. Cada etiqueta possui também um identificador único (ID), que a identifica inequivocamente. ID, IDS, K_1, K_2, K_3 e K_4 possuem um tamanho de 96 bits cada um.

A ideia de criptografia minimalista, proposta por Juels, também é utilizada no M^2AP. Todas as etapas do protocolo se baseiam nas seguintes operações aritméticas simples: XOR bit-a-bit ($\oplus$), OR bit-a-bit ($\vee$), AND bit-a-bit ($\wedge$) e soma modular ($+$).

Considera-se que toda a comunicação entre a etiqueta e o leitor de radiofrequência pode ser capturada por um atacante. Já a comunicação do leitor com o banco de dados é considerada segura, uma vez que estes possuem capacidade computacional suficiente para utilizar técnicas de criptografia mais complexas.

4.4.1 O Funcionamento do M^2AP

O protocolo M^2AP pode ser dividido em quatro etapas: identificação da etiqueta, autenticação do leitor de radiofrequência, autenticação da etiqueta e, por fim, atualização das chaves e do pseudônimo.

4.4.1.1 Identificação da etiqueta

Na etapa inicial do M^2AP, o leitor envia uma mensagem de *hello* e recebe, como resposta da etiqueta, o pseudônimo atual (IDS) desta. Uma vez de posse do IDS, o leitor deve ser capaz de obter no banco de dados o conjunto de chaves secretas (K_1, K_2, K_3 e K_4). O processo é descrito pelos seguintes passos:

Passo 1 R → T: *hello*

Passo 2 T → R: IDS

Passo 3 R → S: *autenticação*, IDS

Passo 4 S → R: K_1, K_2, K_3 e K_4

4.4.1.2 Autenticação do leitor de radiofrequência

Para cada autenticação, o leitor deve gerar dois números aleatórios, n_1 e n_2. A partir da mensagem A, descrita abaixo, a etiqueta é capaz de calcular o valor de n_1. Uma vez de posse de n_1, a etiqueta pode calcular o valor de B e verificar se este corresponde ao valor enviado pelo leitor. O valor de n_2 é obtido a partir da mensagem C. Essa etapa é representada pelos seguintes passos:

Passo 5 R → T: IDS $\oplus$ K_1 $\oplus$ n_1 (A)

Passo 6 R → T: (IDS $\wedge$ K_2) $\vee$ n_1 (B)

Passo 7 R → T: IDS + K_3 + n_2 (C)

4.4.1.3 Autenticação da etiqueta

Nesta etapa, a etiqueta também é autenticada, e o valor de seu identificador único é enviado para o leitor, conforme os passos a seguir:

Passo 8 T → R: (IDS $\vee$ K_4) $\wedge$ n_2 (D)

Passo 9 T → R: (IDS + ID) $\oplus$ n_1 (E)

4.4.1.4 Atualização das chaves e do pseudônimo

Por fim, a cada iteração do protocolo, deve-se realizar a atualização das chaves K_1, K_2, K_3 e K_4, além do pseudônimo IDS. Os passos a seguir reprepresentam o processo de atualização das chaves e do pseudônimo:

Passo 10 IDS = (IDS + $(n_1 \oplus n_2)$) $\oplus$ ID

Passo 11 $K_1 = K_1 \oplus n_2 \oplus (K_3 + \text{ID})$

Passo 12 $K_2 = K_2 \oplus n_2 \oplus (K_4 + \text{ID})$

Passo 13 $K_3 = (K_3 \oplus n_1) + (K_1 \oplus \text{ID})$

Passo 14 $K_4 = (K_4 \oplus n_1) + (K_2 \oplus \text{ID})$

4.4.2 Um Ataque Passivo ao M^2AP

O M^2AP foi modelado com o objetivo de permitir o acesso ao identificador da etiqueta apenas a leitores autorizados. A segurança na transmissão do ID é baseada no compartilhamento das quatro chaves secretas entre as partes comunicantes, bem como na utilização de Criptografia Minimalista na troca de mensagens. Bárász et. al. (2007) apresentou um ataque passivo ao M^2AP capaz de obter o identificador da tag (ID) e as quatro chaves secretas (K_1, K_2, K_3 e K_4).

A vulnerabilidade do M^2AP reside no fato de, a cada operação do protocolo, cada bit influenciar somente os bits a sua esquerda. Isso ocorre devido à natureza das operações aritméticas utilizadas no protocolo. Bárász *et al.* (2007) também chama a atenção para as vulnerabilidades no uso de pseudônimos. Segundo os autores, há aplicações cujos requisitos de segurança não são satisfeitos caso a etiqueta emita pseudônimos a qualquer leitor. Nesses casos, as etiquetas sequer deveriam responder aos leitores não-autorizados.

4.5 O PROTOCOLO $Gen2^+$

A ideia fundamental que embasou o protocolo proposto por Sun e Ting (2009) foi a utilização de valores randômicos em cada sessão. Além disso, os autores seguiram à risca a especificação original do padrão EPCGen2, permitindo compatibilidade com os sistemas RFID que não utilizem seu protocolo, mas obedeçam ao padrão da EPCglobal. Por isso, este protocolo foi denominado $Gen2^+$.

Segundo o protocolo $Gen2^+$, cada etiqueta compartilha com o banco de dados uma string randômica denominada *keypool*. O banco de dados deve conter registros que as-

sociem o *keypool* das etiquetas com seus respectivos identificadores. Assim, as etiquetas podem ser identificadas revelando informações a respeito do valor de sua *string keypool*. Além disso, após 14 autenticações bem-sucedidas, uma etapa extra deve ser acrescentada ao protocolo, com o objetivo de atualizar a *string keypool* da etiqueta, prevenindo ataques de clonagem. Essas etapas serão descritas a seguir.

4.5.1 O Funcionamento do Protocolo $Gen2^+$

O processo de autenticação do protocolo de Sun e Ting é composto de quatro etapas: Consulta inicial, Autenticação da etiqueta, Envio da *centralkey* e Autenticação do leitor.

4.5.1.1 Consulta inicial

Na primeira etapa, o leitor envia um comando de consulta a uma etiqueta específica. A etiqueta, por sua vez, gera um valor pseudo-aleatório de 16 bits que serão utilizados como dois endereços de 8 bits, a e b. Esses dois endereços determinam um segmento do *keypool* armazenado na etiqueta. Observe que os bits do *keypool* são divididos como uma lista circular. Isso significa que, para $a \leq b$, o segmento será k[a:b]. Entretanto, para $a > b$, deve-se utilizar o segmento k[a:l-1]||k[0:b]. A etiqueta também calcula o CRC do referido segmento. O Passo 1, representado a seguir, representa esse processo:

Passo 1 R $\rightarrow$ T: *Query*

4.5.1.2 Cálculo da *centralkey*

Na segunda etapa, a etiqueta envia para o leitor o valor de 16 bits que determina o segmento do *keypool*. O leitor, ao receber tal valor, simplesmente o encaminha ao servidor de banco de dados. Este, por sua vez, procura o registro da etiqueta em sua base. A seguir, o banco de dados gera de um número de 16 bits, chamado *centralkey*, calculando o CRC do segmento do keypool desta etiqueta. Essa etapa é representada no Passo 2, a seguir:

Passo 2 T → R → S: (a,b)

4.5.1.3 Envio da *centralkey*

Na terceira etapa, o banco de dados envia a *centralkey* para o leitor, que a encaminha para a etiqueta, conforme o Passo 3 a seguir:

Passo 3 S → R → T: *centralkey*

4.5.1.4 Autenticação do leitor

Na quarta etapa, a etiqueta compara o CRC por ela calculado com a *centralkey* recebida. A comparação é realizada utilizando o conceito de distância de *Hamming*, descrito em Tanenbaum (2003) como o número de posições que duas sequências de bits diferem entre si. Se a distância de *Hamming* entre esse dois valores for inferior a um limite pré-definido, a autenticação do leitor é realizada com sucesso e a etiqueta responde com seu identificador. Caso contrário, a etiqueta não envia nenhuma mensagem. O uso da distância de *Hamming* neste algoritmo é útil para realizar um equilíbrio entre segurança e eficiência. O Passo 4, a seguir, representa a quarta etapa do processo:

Passo 4 T → R: ID ou Silêncio

4.5.2 Um Ataque do Protocolo $Gen2^+$

Uma análise do $Gen2^+$ é realizada em Burmester *et al.* (2009). Os autores mostram que a vulnerabilidade desse protocolo reside no fato de que o único ponto de aleatoriedade está na etiqueta. Uma vez que ela define o segmento do *keypool* a ser utilizado e calcula o seu CRC, toda a troca de mensagens restante é previsível.

Assim, basta o atacante capturar uma única consulta e observar a troca de mensagens para tornar-se capaz de simular uma etiqueta honesta. A sequência de mensagens continua válida até que o *keypool* seja atualizado. Nesse intervalo de tempo, o ataque poderia ser realizado.

Tabela 4.1 Complexidades de resistência a *spoofing* e clonagem

	Algoritmos		
Ataque	**Gen2$^+$**	**Duc**	**M^{2}AP**
Spoofing	O(1)	O(1)	O(1)
Clonagem	O(2^{32})	O(2^{32})	O(1)

Vale mencionar que a redução da frequência de atualização das etiquetas não eliminaria essa vulnerabilidade, a não ser que fosse realizada a cada consulta. Nesse caso, o protocolo passaria a ser vulnerável a desincronizações, da mesma forma que o protocolo de Duc, anteriormente explicado.

4.6 UM ESTUDO COMPARATIVO

Quando se trata de definir um algoritmo padrão para realizar uma determinada tarefa, dois procedimentos devem ser realiados. Primeiro, é necessário que pesquisadores se dediquem a buscar um protocolo que cumpra os requisitos necessários. A segunda tarefa, não menos importante, é realizar uma análise dessas sugestões, de forma imparcial, identificando as vantagens e desvantagens de cada uma delas.

Cada um dos protocolos das seções anteriores possui qualidades e defeitos. Além disso, eles servem como base para futuras propostas que solucionem os problemas que neles constam. Vale a pena realizar um estudo comparativo desses protocolos e entender suas diferenças.

Será realizada uma análise do ponto de vista da resistência a dois ataques bem conhecidos. Trata-se da clonagem e do ataque de *spoofing*. Esses ataques já foram descritos no Capítulo 3. A Tabela 4.1 resume as complexidades para resistência a cada um deles.

Primeiramente, será discutida resistência a um ataque de clonagem. Tanto o protocolo $Gen2^+$ quanto o protocolo de Duc utilizam um PIN de 32 bits (no primeiro caso,

denominado *keypool*). Esse valor é utilizado para realizar a autenticação da etiqueta perante o leitor. Assim, a complexidade de tempo para resitir a um ataque de clonagem é $O(2^{32})$. Já no caso do M^2AP, o ataque de Bárász *et al.* (2007) realiza a clonagem com duas rodadas de escuta. Assim, a complexidade de tempo é constante.

A seguir, trataremos do custo para o ataque de *spoofing*. No caso do protocolo $Gen2^+$, não há nenhuma checagem da autenticidade da etiqueta. O atacante pode simplesmente aceitar uma mensagem do leitor contendo a *centralkey* e retornar o identificador da etiqueta alvo. Além disso, o identificador é transmitido em texto plano. Assim, o custo para realizar o ataque de *spoofing* no $Gen2^+$ é constante. O protocolo de Duc também não oferece nenhuma autenticação da etiqueta. O atacante pode simplesmente montar a resposta $M_1 = \text{CRC}(1||\text{EPC}\oplus r\oplus r') \oplus K_i$ com um valor EPC alvo, um valor r qualquer e o valor r' fornecido pelo atacante. No caso do M^2AP, o fato de haver um tempo constante para clonar a etiqueta já indica que o custo para realizar um *spoofing* também é constante, utilizando o mesmo algoritmo e enviando uma mensagem maliciosamento construída.

Como vimos anteriormente, todos os algoritmos citados neste documento foram alvos de ataques bem-sucedidos. A utilização desses algoritmos em um sistema crítico representa um alto risco para os usuários com também para o responsável pela rede. Entretanto, dependendo do contexto em que é aplicado, os desenvolvedores de sistemas RFID podem se sujeitar ao riscos atuais de ataque. Neste ponto, deve-se avaliar os potenciais incidentes resultantes dos principais ataques e escolher o algoritmo que evita maiores danos no caso de falha. Nesta seção, percebemos que o custo de resistência dos algoritmos varia de acordo com o ataque ao qual está sujeito. Quando um ataque de clonagem é inadimissível ao uma aplicação, a escolha do M^2AP seria desastrosa. Entretanto se a aplicação não pode permitir um ataque de spoofing, nenhum dos algoritmos aqui descritos seria eficiente.

4.7 RESUMO

Os algoritmos de consulta podem ser considerados o coração dos sistemas RFID. Eles realizam todo o processo de obtenção da informação das etiquetas. Para desenvolvê-los, os pesquisadores precisam aliar a limitação de recursos de *hardware* das etiquetas com a resistência a ataques.

A complexidade dos recursos de criptografia modernos torna inviável seu uso em sistemas RFID. Entretanto, as características desses sistemas permitiram a aplicação de um novo conceito, baseado na utilização de pseudônimos. Trata-se da Criptografia Minimalista. Essa idéia foi aplicada em muitos dos algoritmos de consulta já existentes.

Neste capítulo, estudamos três algoritmos de consulta que tornaram-se populares no meio acadêmico por utilizarem idéias interessantes, que podem servir de base para futuras propostas. O algoritmo de Duc utiliza uma chave diferente para cada sessão. O algoritmo M^2AP traz a necessidade de uma autenticação mútua. Por sua vez, o $Gen2^+$ utiliza somente uma parte de uma chave compartilhada, definida por dois endereços aleatórios.

Um olhar crítico revela que todos os algoritmos aqui propostos são vulneráveis a algum tipo de ataque. Na verdade, ainda não há nenhum algoritmo de consulta reconhecidamente seguro. Entretanto, dependendo da aplicação, pode-se optar por utilizar um sistema RFID, escolhendo-se um desses algoritmos de forma criteriosa e sabendo exatamente quais os riscos de ocorrência de ataques.

CAPÍTULO 5

ALGORITMOS ANTI-COLISÃO: UMA PROPOSTA DE ATAQUE

Todo o funcionamento de um sistema RFID depende da correta comunicação entre etiquetas e leitores. Quando várias etiquetas tentam enviar mensagens simultaneamente, o leitor fica impossibilitado de identificá-las. Neste Capítulo, serão apresentados os algoritmos capazes de controlar o acesso ao canal de comunicação entre etiquetas e leitores. Será mostrada uma alternativa para promover um ataque de negação de serviço baseado em algoritmo anti-colisão. A seguir, será proposto um modelo de ataque capaz de realizar uma clonagem de etiqueta através da exploração de algoritmos anti-colisão.

5.1 INTRODUÇÃO AOS ALGORITMOS ANTI-COLISÃO

Quando um leitor envia um comando de consulta à uma etiqueta, esta recebe energia suficiente para fornecer uma resposta, que contém o seu identificador (ID). O leitor então consulta um banco de dados para, a partir do ID recebido, identificar a etiqueta inequivocamente. Entretanto, caso o comando de consulta do leitor for respondido por mais de uma etiqueta ao mesmo tempo, os sinais de resposta interferirão entre si, tornando difícil o correto recebimento da informação. Nesse caso, o leitor não consegue identificar as etiquetas. Esse fenômeno é chamado de colisão.

Para evitar o problema descrito acima, foram desenvolvidos os algoritmos anti-colisão. Eles estabelecem regras na troca de informações entre etiquetas e leitores, reduzindo o número de colisões e permitindo a identificação de múltiplas etiquetas na mesma área de cobertura do leitor. Basicamente, há dois tipos de algoritmos anti-colisão: os algoritmos baseados no protocolo ALOHA e os algoritmos baseados em árvore.

Segundo Tanenbaum (2003), o protocolo ALOHA foi desenvolvido por Norman Abramson e seus colegas da Universidade do Havaí, na década de 1970. Trata-se de um método para resolver o problema de alocação de canais em uma rede local. Sempre que dois quadros tentarem ocupar um canal ao mesmo tempo, haverá uma colisão e ambos serão danificados. De acordo com protocolo ALOHA, quando um transmissor percebe que seu quadro foi destruído, este é reenviado após um período de tempo aleatório. Os algoritmos anti-colisão de sistemas RFID baseados em ALOHA seguem este mesmo princípio. Quando ocorre uma colisão nas respostas das etiquetas, estas esperam um período de tempo aleatório para então reenviar a informação. Esse procedimento é repetido até que todas as etiquetas consigam enviar suas respostas sem interferências das demais.

Os algoritmos anti-colisão de sistemas RFID baseados em árvore constroem uma árvore binária, dividindo recursivamente o grupo de etiquetas que participaram de uma colisão em dois subgrupos. A divisão é realizada com base nos prefixos dos ID das etiquetas. Esse procedimento é repetido até que cada folha da árvore seja formada por uma única etiqueta. Assim, os leitores são capazes de enviar consultas para etiquetas com um dado prefixo, permitindo que somente ela responda e identifique-se.

Este Capítulo irá focar nos algoritmos anti-colisão baseados em árvore.

5.2 A BLOCKER-TAG

Juels *et al.* (2003) demonstraram uma alternativa viável para impedir a leitura de grupos de etiquetas, através da exploração do algoritmo anti-colisão utilizado. Trata-se da *Blocker Tag*, uma etiqueta capaz de criar uma região física na qual os leitores são incapazes de identificar inequivocamente as etiquetas.

O funcionamento da *Blocker Tag* é bastante simples. Ela provoca uma colisão intencional a cada consulta realizada pelo leitor, simulando a existência de todas as 2^k etiquetas possíveis, onde k é o tamanho de seus IDs. Para isso, em todas as consultas realizadas pelo leitor, a etiqueta responde indicando seu próximo bit como 0 e como 1, simultaneamente. Esse procedimento força o leitor continuar as interações, buscando sempre identificar as etiquetas inequivocamente, explorando toda a árvore de possibilidades de

identificadores. O resultado é que nenhuma das reais etiquetas próximas à *Blocker Tag* consegue responder sozinha às consultas e, portanto, nunca é identificada.

A *Blocker Tag* que impede a leitura de qualquer etiqueta é chamada de bloqueadora universal ou total. Também é possível selecionar quais subconjuntos de etiquetas serão impedidas de serem lidas. Nesse caso, as etiquetas que não possuem os identificadores escolhidos continuarão sendo lidas normalmente. A *Blocker Tag* que funciona dessa forma é chamada de bloqueadora parcial ou seletiva.

A alternativa de Juels, conforme descrita no seu artigo, pode ser utilizada em dois contextos. No primeiro caso, ela pode ser utilizada como um mecanismo de proteção à privacidade de consumidores de produtos rotulados com etiquetas RFID. Ao manter consigo a *Blocker Tag*, um consumidor pode evitar a leitura indesejada da etiqueta de seus produtos por terceiros. No segundo caso, a *Blocker Tag* é utilizada como uma ferramenta maliciosa, capaz de realizar um ataque de negação de serviço em um sistema RFID alvo.

Ainda não existem mecanismos capazes de evitar um ataque de negação de serviço baseado na *Blocker Tag*. Teoricamente, é possível desenvolver leitores especiais, que sejam capazes de estimar uma região física onde a *Blocker Tag* está instalada. Esses leitores poderiam se basear, por exemplo, na potência do sinal recebido. Entretanto tal leitor nunca foi implementado.

Em certas situações, apesar da impossibilidade de evitar o ataque, é possível identificar a sua ocorrência. Tais situações ocorrem quando o número de etiquetas a serem lidas é previamente conhecido, ou ao menos possui um limite máximo. Nesses casos, o leitor pode identificar que tal valor foi superado diante das colisões ocorridas, e registrar a ocorrência de um ataque de negação de serviço.

5.3 TAG CLONER: UMA PROPOSTA DE ATAQUE A ALGORITMOS ANTI-COLISÃO

Quando a intenção é impedir o funcionamento de um sistema RFID, a *Blocker Tag* de Juels pode ser bastante eficiente. Entretanto, em algumas situações é mais interessante para o atacante, ao invés de evitar qualquer leitura, conhecer o conteúdo do conjunto de

etiquetas em questão. Há pouca literatura a respeito de ataques a sistemas RFID que exploram os algoritmos anti-colisão.

Nesta seção, é proposta a *Tag Cloner*, um modelo de etiqueta maliciosa que explora algoritmos anti-colisão do tipo árvore. Os ataques baseados em *Tag Cloner* consistem em uma etiqueta maliciosa que acompanha a execução de um algoritmo anti-colisão e efetua um ataque de clonagem. A ideia básica do modelo dessa etiqueta é identificar, a partir das consultas recebidas, quais foram as respostas das demais etiquetas da região.

De forma geral, os algoritmos anti-colisão baseados em árvore apresentam um padrão de consultas que deve ser obedecido até que uma etiqueta seja inequivocamente identificada. Quando isso acontece, o leitor passa a procurar uma nova etiqueta. Nesse ponto, é comum a ocorrência de uma mudança no padrão de consultas. A etiqueta maliciosa recebe as consultas da mesma forma que as etiquetas honestas presentes na área de cobertura do leitor. Assim, quando o padrão de consultas é modificado, pode-se presumir informações a respeito da etiqueta que foi identificada na consulta anterior. Outros algoritmos anti-colisão do tipo árvore podem revelar informações a respeito do conteúdo das etiquetas no momento de seu encerramento, como por exemplo um prefixo do identificador. A observação da consulta final pode ser suficiente para a identificação de padrões de bits da última etiqueta identificada.

Em todo caso, pode ser necessário um período de aprendizagem, no qual a etiqueta maliciosa captura prefixos dos identificadores de etiquetas até que todo o conteúdo seja clonado. Também é necessário o conhecimento prévio do algoritmo anti-colisão que será utilizado, para que as mudanças nos padrões de consulta sejam interpretadas corretamente. Caso seja possível utilizar maiores recursos de memória, deve-se armazenar algumas sequências de consultas na etiqueta maliciosa, afim de identificar o algoritmo anti-colisão em execução.

O modelo *Tag Cloner* traz consigo uma nova estratégia de ataque, que coloca em xeque os algoritmos anti-colisão do tipo árvore. A seguir, serão apresentados ataques baseados em *Tag Cloner* para três algoritmos desse tipo.

Tabela 5.1 Algoritmo QT

	Consultas						
ID	**0**	**1**	**01**	**(...)**	**010**	**011**	
0110	110		10			0	
1000		000					
0101	101		01		1		
Etiqueta Maliciosa							Armazena 011x

5.3.1 Atacando o Algoritmo QT

O algoritmo de consulta em árvore, ou *Query Tree* (QT), foi apresentado por Ching Law *et al.* (2000). Ele consiste em uma série de rodadas de consultas às etiquetas. A cada rodada, o leitor consulta o conjunto de etiquetas, perguntando quais possuem um certo prefixo. Se mais de uma etiqueta responder, o leitor não é capaz de identificá-las, porém saberá que ao menos duas delas possuem o prefixo apresentado. Assim, são apresentados valores binários em sequência, até que todas as etiquetas respondam individualmente e sejam identificadas.

Suponha que uma etiqueta maliciosa seja inserida ao conjunto de etiquetas a serem identificadas. Assim como as demais etiquetas, ela deverá receber as consultas do leitor, que obedecerá o algoritmo QT. A etiqueta maliciosa somente observa a execução das consultas. Como os prefixos são apresentados de forma sequencial, ao final da consulta a última etiqueta honesta terá sido identificada e a etiqueta maliciosa conhecerá seu prefixo.

A seguir apresentamos um exemplo do ataque. Nesse caso, as etiquetas a serem identificadas possuem os IDs 0110, 1000 e 0101. A Tabela 5.1 retrata a primeira interação das consultas. As colunas representam as consultas realizadas pelo leitor. As linhas correspondem ao comportamento de cada etiqueta, representada pelo seu ID.

Inicialmente, o leitor realiza uma consulta por etiquetas com o prefixo 0. Neste caso, as etiquetas 0110 e 0101 respondem à chamada, enviando o restante da sequência de bits que as identifica (110 e 101, respectivamente). Entretanto uma colisão ocorre, pois não é possível identificar as duas etiquetas simultaneamente. O leitor prossegue com suas consultas, enviando o prefixo 1. Desta vez, somente a etiqueta 1000 responde, enviando a sequência de bits 000 restante. Assim, essa etiqueta é a primeira a ser identificada com sucesso. A sequência de consultas continua até que todas as etiquetas sejam identificadas.

A etiqueta cujo ID é 0110 foi a última a ser identificada, por possuir o prefixo 011. Uma vez que o leitor encerrou suas consultas, a etiqueta maliciosa descobre que há uma etiqueta com tal prefixo, pois ele foi utilizado na consulta final. Então a etiqueta maliciosa captura esse prefixo. Como neste caso resta somente um bit desconhecido, a etiqueta maliciosa já é capaz de realizar um ataque de clonagem com 50% de probabilidade de sucesso.

5.3.2 Atacando o Algoritmo de Zhou

O algoritmo publicado por Zhou *et al.* (2004) realiza um estudo que estabelece funções de custo para protocolos anti-colisão. Seu objetivo é reduzir o consumo de energia de sistemas RFID. Como parte de seu trabalho, o autor apresenta uma melhoria do algoritmo anti-colisão QT. Segundo o algoritmo de Zhou, uma vez que o leitor detecta a ocorrência de uma colisão, este deve enviar comandos que interrompam as respostas das etiquetas. Além disso, a sequência de consultas utilizadas nesse artigo é ligeiramente diferente daquela utilizada no algoritmo QT. A partir das respostas das etiquetas, o leitor é capaz de identificar em qual bit ocorre a colisão. Assim, na consulta seguinte, o leitor repete os bits que não provocam colisões e acrescenta um novo bit, evitando a colisão anterior.

O algoritmo de Zhou é vulnerável à *Tag Cloner*. Novamente, para realizar o ataque, basta que uma etiqueta maliciosa observe as consultas do leitor. Quando o prefixo utilizado em uma consulta for menor que o prefixo da consulta seguinte, significa que uma etiqueta foi identificada. Isso acontece devido à estrutura das consultas do algoritmo de Zhou, similar a uma busca em profundidade. Além disso, quando o algoritmo é encerrado,

Tabela 5.2 Algoritmo Zhou

	Consultas							
ID	ϵ	**0**	**000**	**001**	**1**	**10**	**11**	
0001	0001	001	1					
0011	0011	011		1				
1000	1000				000	00		
1100	1100				100		00	
Etiqueta Maliciosa					Armazena 001x			Armazena 11xx

ele revela o prefixo da última etiqueta identificada.

No exemplo a seguir, as etiquetas a serem identificadas possuem os ID 0001, 0011, 1000 e 1100. A Tabela 5.2 retrata uma leitura que utiliza o algoritmo de Zhou.

Em uma primeira consulta, o leitor espera que todas as etiquetas presentes respondam, enviando um prefixo vazio. Uma vez que houve colisões, o leitor consulta as etiquetas com prefixo 0. As etiquetas 0001 e 0011 respondem com as sequências 001 e 011, respectivamente. O algoritmo de Zhou considera que o leitor é capaz de perceber que ambas as sequências respondidas 001 e 011 iniciam com o bit 0. Assim, o leitor entende que o prefixo 00 resultaria em uma nova colisão. A próximo consulta é, então, sobre o prefixo 000. É interessante notar que, diferentemente do algoritmo de Duc que obedece à sequência numérica natural, o algoritmo de Zhou tenta identificar primeiramente aquelas etiquetas que sofreram colisão. Assim, a etiqueta 0001 responde a esta consulta com seu bit restante, 1. A etiqueta 0011 também é identificada na consulta seguinte, com a consulta do prefixo 001.

Note que a consulta do prefixo 001 antecede uma consulta de um prefixo menor, no caso, 1. Esse fato revela à etiqueta maliciosa que 001 é um prefixo válido para uma das etiquetas honestas do sistema. Além disso, a consulta final revela que 11 também é um prefixo válido. Assim, a etiqueta maliciosa consegue obter informações a respeito

do conteúdo das etiquetas. Em futuras interações, a etiqueta maliciosa pode passar a responder a consultas do leitor, provocando novas colisões e obtendo prefixos maiores das demais etiquetas, até realizar uma clonagem completa.

5.3.3 Atacando o Algoritmo de Backtracking

O algoritmo de *Backtracking* proposto por Shi *et al.* (2008) também é baseado na busca em árvore. Porém, nesse caso, as consultas não são sequenciais. Os autores utilizam a codificação *Manchester*, de forma que quando ocorre uma colisão, o leitor é capaz de identificar quais bits dos IDs são semelhantes e quais são diferentes. Essas informações são utilizadas como parâmetros para a nova consulta. Além disso, quando uma etiqueta é identificada, o leitor a envia um comando UNSELECT, solicitando que não mais responda às suas consultas.

Inicialmente o leitor pede para que todas as etiquetas respondam. Isso é feito enviando o prefixo vazio. Então, o leitor identifica se há um padrão nos prefixos das etiquetas, evitando assim a realização de consultas desnecessárias. Uma vez identificado o primeiro bit a conter valores diferentes nas etiquetas, o leitor realiza uma busca em profundidade, de forma recursiva, até que uma etiqueta seja identificada isoladamente. Nesse ponto, o leitor envia o comando UNSELECT a esta etiqueta e realiza novamente a consulta que resultou na última colisão. Dessa vez, a etiqueta identificada não participará desta colisão.

O algoritmo de *Backtracking* também é vulnerável a um ataque da *Tag Cloner*. Nesse caso, a vulnerabilidade está no retorno à última colisão. Sempre que for realizada uma consulta repetida em uma mesma interação, sabe-se que uma etiqueta foi identificada e seu prefixo pode ser armazenado.

No exemplo a seguir, as etiquetas a serem identificadas possuem os ID 10110011 (etiqueta 1), 10100011 (etiqueta 2), 10110111 (etiqueta 3) e 11100011 (etiqueta 4). A Tabela 5.3 retrata uma leitura que utiliza o algoritmo de *Backtracking*.

Note que, após a consulta de prefixo vazio, ocorre a primeiro consulta de prefixo 10. Nesta consulta, as etiqueta 1, 2 e 3 respondem. Entretranto todas elas iniciam suas

Tabela 5.3 Algoritmo Backtracking

	Consultas						
ID	ϵ	**10**	**1010**	**10**	**101100**	**10**	ϵ
10110011	10110011	110011		110011	11		
10100011	10100011	100011	0011				
10110111	10110111	110111		110111		110111	
11100011	11100011						11100011
Etiq. Malic.				Arm. 1010xxxx		Arm. 101100xx	

respostas com o bit 1. Assim, a próxima consulta já considera este bit identificado, e tenta um prefixo 1010. Neste momento ocorre a identificação da etiqueta 2. O leitor envia o comando de UNSELECT e realiza novamente a consulta de prefixo 10. A etiqueta maliciosa, percebendo a consulta repetida, percebe que há uma etiqueta cujo prefixo é 1010. Uma situação análoga acontece quando a consulta 10 é realizada pela terceira vez. Neste caso, é o prefixo 101100 que é capturado.

Dessa forma, a etiqueta maliciosa é capaz de capturar informações sobre as etiquetas que a rodeiam. Ela poderia, por exemplo, simular o comportamento dessas etiquetas, até que todos os ID fossem capturados.

5.3.4 Considerações sobre Defesa Contra a Tag Cloner

Até a publicação deste documento, não foi encontrada nenhuma técnica eficaz capaz de evitar o modelo de ataque da *Tag Cloner*. Um protocolo anti-colisão pode ser considerado imune à *Tag Cloner* se, uma vez identificada uma etiqueta, a confidencialidade de seu conteúdo não é afetada pela próxima consulta. Além disso, o encerramento do protocolo de consulta não deve revelar informações a respeito do conteúdo de nenhuma etiqueta. Uma estratégia do tipo força bruta seria realizar a consulta sequencial de todo o universo de etiquetas, do início ao fim. Assim, o leitor seria capaz de capturar o conteúdo das etiquetas sem deixar vazar qualquer informação para eventuais atacantes. Entretanto,

essa estratégia possui um custo de tempo muito elevado e não é interessante do ponto de vista prático.

5.4 RESUMO

Quando várias etiquetas respondem simultaneamente a uma consulta de um leitor, não é possível identificar nenhuma delas. Assim, faz-se necessária a utilização de algoritmos anti-colisão para controlar o acesso ao canal de transmissão entre etiquetas e leitores. Os algoritmos anti-colisão são classificados em dois grupos: baseados no protocolo ALOHA e baseados em árvore.

Os algoritmos baseados no protocolo ALOHA utilizam os mesmos princípios do tradicional protocolo de Norman Abramson. Quando duas ou mais etiquetas enviam informações ao meio de forma simultânea, ocorre uma colisão. A seguir, cada uma delas reenvia seus dado após um período de tempo aleatório. Esse processo se repete até que todas as etiquetas sejam identificadas.

Os algoritmos baseados em árvore utilizam uma árvore binária para dividir as etiquetas de acordo com seus identificadores. Cada uma delas passa a ser representada como folhas da árvore. O leitor percorre cada nó dessa estrutura. Caso haja uma colisão, os nós filhos também serão consultados individualmente, através de um prefixo mais específico. Esse processo se repete até que todas as etiquetas sejam identificadas.

A *Blocker-Tag* é uma alternativa interessante para realizar um ataque de negação de serviço, explorando os algoritmos anti-colisão tipo árvore. Esse dispositivo força a ocorrência de colisões, indicando seu próximo bit como 0 e 1 simultaneamente. Esse procedimento evita que qualquer etiqueta seja corretamente identificada.

A *Tag Cloner* é uma proposta de ataque que realiza a clonagem de etiquetas honestas. Através da observação das consultas realizadas pelos leitores, é possível identificar o conteúdo das etiquetas do ambiente. Esse ataque mostrou-se eficaz contra os algoritmos QT de Ching Law *et al.* (2000), o algoritmo de Zhou *et al.* (2004) e o algoritmo de *Backtracking* proposto por Shi *et al.* (2008). Assim, a *Tag Cloner* é uma nova ameaça contra algoritmos anti-colisão baseados em árvore. Uma possível estratégia de defesa seria

consultar de forma sequencial todo o universo de etiquetas, entretanto essa alternativa não é interessante do ponto de vista prático.

CAPÍTULO 6

CONCLUSÕES

Há uma grande tendência de popularização dos sistemas RFID. Em breve, as pessoas estejam rodeadas de inúmeras etiquetas, anexadas ao mais variados objetos. Essa tecnologia substituirá os códigos de barras, com duas principais vantagens: a identificação inequívoca e a leitura automatizada.

As aplicações da tecnologia RFID são bastante interessantes e inovadoras. As etiquetas podem ser anexadas a itens de um supermercado, automatizando o processo de compra e evitando filas. Também seria possível obter maiores informações sobre os produtos, como seu histórico, validade ou o modo de preparo. De forma análoga, etiquetas subcutâneas implantadas em animais poderiam auxiliar no acompanhamento das espécies. Há também sistemas RFID que gerenciam o tráfego em rodovias e os empréstimos de livros em bibliotecas.

A arquitetura dos sistemas RFID é bastante simples. Dessa forma, torna-se possível a aplicação em objetos de baixo custo. Essencialmente, os sistemas RFID possuem dois componentes principais: os leitores e as etiquetas. Há também um banco de dados, capaz de se comunicar com os leitores e oferecer maiores informações a respeito dos itens lidos.

O processo de padronização da tecnologia ainda está em andamento. A EPCglobal saiu na frente, com o padrão EPCGen1. Entretanto os conceitos da utilização dos sistemas RFID evoluiu, e surgiu o novo padrão EPCGen2. Ele define toda a arquitetura do sistema e a comunicação entre seus módulos. A ISO também propôs o seu padrão, conhecido como ISO 18000. Entretanto, diferentemente do EPCGen2, o padrão da ISO limita-se a definir a comunicação entre etiquetas e leitores, permitindo que os projetistas definam a arquitetura livremente. Isso gerou uma divergência entre padrões, criando um obstáculo político para a evolução da tecnologia RFID.

O modelo de sistemas RFID prioriza a simplicidade da arquitetura, restando poucos recursos destinados à segurança computacional. Assim, há diversas estratégias para realizar ataques. Estes podem ser de origem externa, quando terceiros tentam obter ou manipular informações do sistema acompanhando a comunicação entre etiquetas e leitores. Outros ataques podem ter origem interna, quando as próprias etiquetas do sistema disparam um ataque contra o restante dos componentes. É interessante notar que é possível a implementação de um vírus de RFID que contamine todo o sistema partindo de uma única etiqueta com conteúdo malicioso.

Os algoritmos de consulta permitem que leitores capturem o conteúdo das etiquetas. Eles possuem uma importância destacada, pois devem proporcionar ao leitor um mecanismo de consulta que evite o vazamento de informações a terceiros. Apesar das propostas que vêm surgindo no meio acadêmico, ainda não há um algoritmo de consulta reconhecidamente seguro.

A Criptografia Minimalista, proposta por Juels, traz consigo um novo conceito de segurança, destinado a sistemas RFID. Como a limitação do hardware das etiquetas impede a utilização de técnicas modernas de RFID, a Criptografia Minimalista propõe que se considere o ambiente da aplicação como uma restrição ao atacante. Além disso, é sugerido o uso de pseudônimos, que evitam que o conteúdo das etiquetas transite de forma desprotegida. A Criptografia Minimalista vem sendo adotada em diversos protocolos de consulta. Entretanto, não há qualquer prova que esta abordagem irá satisfazer as necessidades de segurança dos usuários.

Entre os protocolos de consulta que ganharam destaque no meio acadêmico estão o Protocolo de Duc, o M^2AP e o $Gen2^+$. Todos eles se baseiam em operações ariméticas simples, capazes de serem executadas por etiquetas do padrão EPCGen2. Eles também possuem vulnerabilidades já identificadas. Considerando que ainda não há nenhum protocolo de consulta imune a ataques, a utilização da tecnologia RFID em sistemas críticos deve ser evitada. Entretanto, há sistemas que admitem certos modelos de ataques sem resultar em grandes perdas. Afinal, os ataques acontecem até mesmo nos sistemas computacionais mais robustos. Assim, a escolha do algoritmo de consulta deve depender de

qual modelo de ataque o projetista tem mais disposição a admitir. Supondo que um ataque de clonagem resulta em graves danos, o algoritmo de Duc deve ser evitado. Seria mais coerente, neste caso, a escolha dos algoritmos M^2AP ou $Gen2^+$. Entretanto, se um ataque de *spoofing* for inadimissível, nenhuma destas alternativas seria interessante.

Quando várias etiquetas respondem a uma consulta simultaneamente, ocorre interferência entre seus sinais de resposta. Assim, para que sejam identificadas corretamente, é necessário um algoritmo anti-colisão que controle o acesso ao meio. A exploração da segurança desses algoritmos ainda é um assunto pouco estudado no meio acadêmico.

Uma alternativa para atacar algoritmos anti-colisão baseados em árvore é a *Blocker-Tag* de Juels. Quando o leitor solicita a resposta de etiquetas que possuem um dado prefixo, a *Blocker-Tag* responde e envia como seu próximo bit como 0 e 1, simultaneamente. Assim, é realizado um ataque de negação de serviço no sistema RFID em questão.

Este documento propôs um novo modelo de ataque contra algoritmos anti-colisão baseados em árvore. Trata-se da *Tag Cloner*. Seu funcionamento se baseia em uma etiqueta maliciosa que acompanha a execução do algoritmo anti-colisão. A partir da sequência de consultas realizadas, é possível identificar as etiquetas que estão presentes na área de cobertura do leitor. Esse modelo de ataque se mostrou eficaz contra o algoritmos QT, o algoritmo de Zhou e o algoritmo de *Backtracking* de Shi. Como os algoritmos baseados em árvore possuem pequenas variações no seu funcionamento, pode-se esperar que o ataque de *Tag Cloner* tenha sucesso em uma ampla gama de sistemas RFID.

Trabalhos Futuros

A tecnologia RFID está em constante evolução. Assim, este documento certamente pode servir como base para futuras pesquisas na área de segurança de sistemas RFID. Alguns temas merecem destaque. Por exemplo, seria interessante propor um algoritmo anti-colisão que seja eficaz contra a *Tag Cloner* e não reduzisse a performance dos sistemas RFID. Além disso, a busca por um algoritmo de consulta continua, e a proposta de um algoritmo que tenha uma alta resistência a ataques seria um avanço na área. Este documento deu mais um passo no estudo da segurança de sistemas RFID, mas muito deste caminho ainda precisa ser percorrido.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] C. Anley. Advanced SQL injection in SQL server aplications. Technical report, Next Generation Security Software Ltd, 2002.

[2] M. Barasz. Passive attack against the M^2AP mutual authentication protocol for RFID tags. *In Proceedings oh the First Int'l Workshop RFID Technology (EURASIP)*, September 2007.

[3] Mike Burmester, Breno de Medeiros, Jorge Munilla, and Alberto Peinado. Secure EPC Gen2 compliant radio frequency identification. *Lecture Notes in Computer Notes*, 5037:227–240, 2009.

[4] Hung-Yu Chien and Che-Hao Chen. Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards. *Computer Standards and Interfaces*, 29, February 2007.

[5] Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole. Buffer overflows: Attacks and defenses for the vulnerability of the decade. *In Proceedings of DARPA Information Survivability Conference and Exposition*, January 2000.

[6] Dang Nguyen Duc, Hyunrok Lee, and Kwangjo Kim. Enhancing security of EPC-Global Gen-2 RFID against traceability and cloning. *In Proceedings of Third Conf. Soft Computing and Intelligent Systems (SCIS'06)*, January 2006.

[7] Simson L. Garfinkel. Adopting fair information pratices to low cost RFID systems. *In Proceedings of Ubiquitous Computing Privacy Workshop*, 2002.

[8] G. P. Hancke. Eavesdropping attacks on high-frequency RFID tokens. *In Proceedings of Workshop on RFID Security RFIDSec'08*, July 2008.

[9] Ari Juels. Minimalist cryptography for low-cost RFID tags. *In Proceedings of Fourth Int'l Conf. Computational Intelligence and Security (CIS'06)*, November 2006.

[10] Ari Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communication*, 24(2):381–392, February 2006.

[11] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The Blocker Tag: Selective blocking of RFID tags for consumer privacy. *In Proceedings of 8th ACM Conference on Computer and Communications Security*, pages 103 – 111, 2003.

[12] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. *In Proceedings of Fifth Annual IEEE Int'l Conf. Pervasive Computing and Communications (PerComp'07)*, 2006.

[13] Amit Klein. Cross site scripting explained. Technical report, Sanctum Security Group, June 2002.

[14] Ching Law, Kayi Lee, and Kai-Yeung Siu. Efficient memoryless protocol for tag identification. *In Proceedings of 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, pages 75–84, August 2000.

[15] Aleph One. Smashing the stack for fun and profit. *Phrack Magazine*, 49, 1996.

[16] Pedro Peris-Lopez, Julio Hernandez-Castro, Juan M. Estevez-Tapiador, and Arturo Ribagorda. M^2AP: A Minimalist Mutual-Authentication Protocol for low-cost RFID tags. *In Proceedings of Third Int'l Conf. Ubiquitous Intelligence and Computing (UIC'06)*, September 2006.

[17] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum. Is your cat infected with a computer virus? *In Proceedings of Fourth Annual IEEE Int'l Conf. Pervasive Computing and Communications (PerComp'06)*, 2006.

[18] X. L. Shi, F. Wei, Q. L. Huang, L. Wang, and X. W. Shi. Novel binary search algorithm of backtracking for RFID tag anti-collision. *Progress In Electromagnetics Research B*, 9:97–104, 2008.

[19] William Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice-Hall, 2007.

[20] Hung-Min Sun and Wei-Chih Ting. A Gen2-Based RFID authentication protocol for security and privacy. *In Proceedings of IEEE Transactions on Mobile Computing*, 8:1053–1062, August 2009.

[21] Andrew S. Tanenbaum. *Redes de Computadores*. Campus, 2003.

[22] Zebra Technologies. RFID: The next generation of AIDC. Technical report, 2005.

[23] Chris Turner. EPC and ISO 18000-6. *RFID Journal*, March 2003.

[24] Roy Want. The magic of RFID. *QUEUE*, pages 40–48, October 2004.

[25] S. Weiss. Security and privacy in radio-frequency identification devices. Master's thesis, Massachusetts Institute of Technology (MIT), May 2003.

[26] Feng Zhou, Chunhong Chen, Dawei Jin, Chenling Huang, and Hao Min. Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems. *In Proceedings of ACM ISLPED'04*, pages 357–362, 2004.