UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Extensões ao protocolo de comunicação EPCGlobal para tags Classe 1 utilizando autenticação com criptografia de baixo custo para segurança em Identificação por Radiofreqüência

Rafael Perazzo Barbosa Mota

São Carlos-SP
Maio de 2006

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

M917ep

Mota, Rafael Perazzo Barbosa.
Extensões ao protocolo de comunicação EPCGlobal para tags Classe 1 utilizando autenticação com criptografia de baixo custo para segurança em identificação por radiofreqüência / Rafael Perazzo Barbosa Mota. -- São Carlos : UFSCar, 2006.
78 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2006.

1. Internet (Redes de computação). 2. Tecnologia da informação. 3. Criptografia. 4. Radiofreqüência. 5. Sistemas de comunicação sem fio. I. Título.

CDD: 004.67 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Extensões ao protocolo de comunicação EPCGlobal para tags Classe 1 utilizando autenticação com criptografia de baixo custo para segurança em Identificação por Radiofreqüência"

RAFAEL PERAZZO BARBOSA MOTA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:

Prof. Dr. Sérgio Donizetti Zorzo
(Orientador – DC/UFSCar)

Prof. Dr. Hélio Crestana Guardia
(DC/UFSCar)

Prof. Dr. Omar Carvalho Branquinho
(PUC/Campinas)

São Carlos
Maio/2006

Dedico este trabalho a meus pais
Ocian e Alessandra, responsáveis
diretos pelo meu sucesso na vida.

“Tentar e falhar é, pelo menos,
aprender. Não chegar a tentar é
sofrer a inestimável perda do que
poderia ter sido.”
(Geraldo Eustáquio)

AGRADECIMENTOS

Em primeiro lugar a minha família Ocian, meu pai, Alessandra, minha mãe, Flávio, meu irmão, e Rosa, minha irmã de coração;

Ao meu orientador Prof. Dr. Sérgio Donizetti Zorzo, pelo apoio e disponibilidade;

Aos meus amigos do GSDR Guilherme Somera, Guilherme Torres, Luciano de Paula, Robson de Grande, Leonardo, Luís e Luana pela amizade, ajuda e momentos de descontração;

A todos meus amigos e amigas que residem em Fortaleza-CE e pelo Brasil afora;

Ao Departamento de Computação (DC) pela estrutura oferecida para o correto desenvolvimento dos estudos;

À CAPES pelo apoio financeiro;

À todos os professores do curso de Mestrado em Computação da UFSCar;

À toda comunidade OpenOffice pela excelente suíte Office disponível sem custos;

Enfim, a todos aqueles que, direta ou indiretamente contribuíram para realização deste trabalho.

RESUMO

O protocolo de comunicação para a Identificação por Radiofreqüência (RFID-*Radio Frequency Identification*), definido como padrão pela EPCGlobal, não oferece mecanismos para a garantia de segurança e privacidade aos usuários do sistema. Esta característica possibilita a presença de diversos tipos de problemas no emprego da tecnologia justificando que novos mecanismos de segurança sejam incorporados diretamente ao protocolo de comunicação, preservando os padrões existentes. A proposta deste trabalho baseia-se na utilização de autenticação da comunicação com emprego de criptografia de baixo custo utilizando o algoritmo TEA. O mecanismo de autenticação mútua proposto foi especificado e validado com lógica BAN. Toda a especificação do protocolo considerou o padrão EPCGlobal para tags Classe 1 como base, adicionando extensões visando combater as possibilidades de ataques relacionados à segurança e privacidade na comunicação dos dados. Os resultados obtidos com este trabalho incluem a especificação formal de um protocolo de autenticação com criptografia, permitindo a padronização deste protocolo baseado no padrão para tags Classe 1 e preservando a especificação padrão base.

Palavras-chave: Identificação por radiofreqüência, segurança, RFID, criptografia, autenticação, protocolo de comunicação.

ABSTRACT

EPCGlobal communication protocol for RFID (*Radio Frequency Identification*) does not ensure security and privacy for its system users. This fact makes possible several kinds of security problems on RFID technology usage justifying that security mechanisms should be added to communication protocol preserving the existing standards. This work proposal is based on communication authentication usage making use of low-cost cryptography with the TEA algorithm. The proposed mutual authentication mechanism was specified and proved using BAN logic. All protocol specification has considered the EPCGlobal standard for Class 1 tags as base adding security and privacy extensions to data communication to get protection against several attacks. The results from this work include an authentication protocol formal specification with cryptography allowing this way the protocol standardization using as base the current Class 1 protocol.

Keywords: Radio frequency identification, security, RFID, cryptography, authentication, communication protocols.

LISTA DE FIGURAS

Figura 1. Componentes do Sistema RFID. 5
Figura 2. Tag RFID. 6
Figura 3. Organização interna de uma tag RFID. 7
Figura 4. Posicionamento e fluxo de dados entre os componentes do sistema RFID. 10
Figura 5. Seqüência de envio de quadros entre leitor e tag. 14
Figura 6. Formato de quadros de requisição e resposta..... 15
Figura 7. Memória da tag..... 17
Figura 8. Algoritmo para singularização das tags "Percurso em árvore binária"..... 21
Figura 9. Comportamentos de um intruso em relação às posições de origem e destino. 23
Figura 10. Ataque do tipo interceptação em sistemas RFID..... 25
Figura 11. Percurso silencioso em árvore binária (Silent Binary Tree Walking)..... 28
Figura 12. Funcionamento do emulador de tag..... 29
Figura 13. Trava por Hash (Hash Lock)..... 35
Figura 14. Trava por hash aleatório..... 36
Figura 15. Autenticação para RFID utilizando "Zero-Knowledge Protocol". 39
Figura 16. Autenticação da tag 41
Figura 17. Autenticação do leitor. 41
Figura 18. Comunicação com autenticação mútua..... 55

LISTA DE TABELAS

Tabela 1. Frequências de operações definidas pela EPCGlobal para tags RFID..... 8
Tabela 2. Comandos obrigatórios do protocolo de comunicação. 15
Tabela 3. Parâmetros dos canais de comunicação. 16
Tabela 4. Formato e campos dos quadros de dados. 18
Tabela 5. Comandos obrigatórios..... 19
Tabela 6. Formato do quadro ScrollID Reply..... 20
Tabela 7. Formato do quadro PingID Reply..... 20
Tabela 8. Principais tipos de problemas de segurança em sistemas RFID..... 26
Tabela 9. Objetivos da proposta de segurança..... 42
Tabela 10. Características do microcontrolador utilizado..... 47
Tabela 11. Algoritmo para envio de informação..... 48
Tabela 12. Algoritmo para recepção de informação..... 48
Tabela 13. Recursos necessários para implementação do algoritmo TEA..... 49
Tabela 14. Metodologia utilizada na pesquisa..... 52
Tabela 15. Notação utilizada..... 53
Tabela 16. Tipos possíveis de comunicações..... 54
Tabela 17. Notação básica da lógica BAN..... 58
Tabela 18. Protocolo formalizado..... 60
Tabela 19. Suposições para análise do protocolo..... 60
Tabela 20. Objetivos do protocolo..... 60
Tabela 21. Prova do protocolo. 61
Tabela 22. O comando AuthTagRequest..... 62
Tabela 23. O comando de resposta AuthTagReply..... 63
Tabela 24. O comando ScrollEncID..... 63
Tabela 25. O comando de resposta ScrollEncIDReply..... 63
Tabela 26. O comando AuthRequest..... 64
Tabela 27. O comando de resposta AuthReply..... 64
Tabela 28. O comando Auth..... 64
Tabela 29. Comandos personalizados para disponibilização de segurança..... 65

Tabela 30. Comparação entre protocolos no aspecto segurança..... 66
Tabela 31. Comparação entre protocolos no aspecto desempenho..... 68
Tabela 32. Proteções oferecidas pelas extensões de segurança..... 69

SUMÁRIO

1. INTRODUÇÃO.....1
1.1 MOTIVAÇÕES 1
1.2. ORGANIZAÇÃO 4
2. REVISÃO DE LITERATURA..... 5
2.1. O SISTEMA RFID.....5
2.1.1. TAGS 6
2.1.2. LEITORES..... 9
2.1.3. BANCO DE DADOS 11
2.1.4. APLICAÇÕES RFID 11
2.1.5. CÓDIGO ELETRÔNICO DE PRODUTO E A REDE EPCGLOBAL 12
2.1.6. PADRÕES PARA RFID 13
2.2. SEGURANÇA 21
2.2.1. SEGURANÇA EM RFID.....24
2.2.2. SEGURANÇA E ASSIMÉTRIA DOS CANAIS..... 27
2.3. CRIPTOGRAFIA 31
2.4. AUTENTICAÇÃO.....37
3. EXTENSÕES DE SEGURANÇA PARA O PROTOCOLO EPCGLOBAL PARA TAGS CLASSE 1 46
3.1. AVALIAÇÃO DO ALGORITMO TEA.....46
3.2. RESULTADO DA AVALIAÇÃO 48
3.3. ESPECIFICAÇÃO DO PROTOCOLO DE AUTENTICAÇÃO E SEGURANÇA PROPOSTO 51
3.3.1. CONSIDERAÇÕES INICIAIS 52
3.3.2. CONFIGURAÇÕES INICIAIS.....54
3.4. ESPECIFICAÇÃO FORMAL DO PROTOCOLO.....57
3.5. PADRONIZAÇÃO DO PROTOCOLO DE AUTENTICAÇÃO 62
3.6. DISCUSSÃO.....65

3.7. CONTRIBUIÇÕES..... 69
4. CONCLUSÕES E TRABALHOS FUTUROS..... 71
REFERÊNCIAS BIBLIOGRÁFICAS..... 73

1. INTRODUÇÃO

Identificação por Radiofreqüência (RFID – *Radio Frequency Identification*) é uma tecnologia que utiliza ondas de radiofreqüência para transmissão de dados.

O recurso surgiu há muito tempo, como uma forma de leitura remota de dados (de identificação), armazenados em pequenos objetos anexados a bens ou seres vivos. Sua primeira grande aplicação deu-se durante a Segunda Guerra Mundial, quando foi usada pelas forças britânicas para identificar inimigos e amigos respondendo ou não a pedidos de identificação por meio de ondas de rádio [Weis 2004] [Landt 2001].

Foram necessários mais de trinta anos de evolução da eletrônica, levando à associação da RFID a técnicas digitais de tratamento da informação, até que se chegasse à possibilidade da sua ampla disseminação. Isso envolve componentes eletrônicos, microeletrônicos e softwares especializados, compondo um sistema de identificação digital.

1.1 MOTIVAÇÕES

A tecnologia RFID já é utilizada em algumas aplicações familiares como o controle de acesso a prédios e ambientes corporativos e o ingresso em meios de transporte, ambos por meio de cartões de aproximação sem tarjas magnéticas, códigos de barras ou fendas para leitura direta de *chips*. São igualmente conhecidas as etiquetas em livros e os pequenos objetos plásticos presos em bens no comércio para evitar o furto à saída das lojas. No Brasil podemos verificar também sua utilização em pedágios das rodovias do estado de São Paulo assim como no destravamento de automóveis. Entretanto, uma gama infinitamente maior de aplicações é possível, sendo objeto de projetos pilotos em diferentes lugares no mundo.

Foi o *Wal-Mart*, o supermercadista líder mundial, quem colocou a identificação RFID sob o foco das atenções ao exigir que cem de seus fornecedores passassem a fazer uso obrigatório de etiquetas inteligentes em suas entregas, e dando-lhes um prazo para adequação

de seus processos e sistemas ao novo padrão EPCGlobal[1] [EPCGlobal 2005]. Tendo em vista as dificuldades enfrentadas na implantação das novas soluções, o prazo, inicialmente fixado em janeiro de 2005, acabou sendo estendido para 2006. A exigência gerou reações nas grandes organizações do comércio e nas indústrias que as atendem [Gutierrez et al 2005].

No Brasil, uma outra aplicação da RFID vem sendo debatida em função de suas implicações econômicas para o País. Trata-se da rastreabilidade animal, em especial a do gado bovino. A regulamentação específica foi decretada pelo Ministério da Agricultura, Pecuária e Abastecimento em janeiro de 2002[Gutierrez et al 2005]. Todavia, o uso da nova tecnologia eletrônica pode não apenas prover o atendimento das questões legais como impulsionar fortemente o processo brasileiro de produção da carne, por meio de um aumento da produtividade, da melhoria da qualidade do produto e da ocupação de nichos de mercado consumidor.

O sistema é formado basicamente por três componentes chaves que são o leitor, as tags (ou etiquetas) que são fabricadas nos mais diversos formatos e custos de fabricação) e uma aplicação externa. O leitor faz requisições às tags que, por sua vez, enviam a resposta. Recebida a informação desejada, o leitor pode ainda interagir com uma aplicação externa, geralmente uma base de dados.

O custo para a confecção das tags apresenta-se como fator limitante principal para a disponibilização de segurança para o sistema RFID. A utilização de mecanismos de criptografia exige muitos recursos de hardware, o que pode aumentar muito o custo de fabricação. O padrão EPCGlobal para tags Classe 1 (etiquetas que permitem várias leituras e uma única gravação) define o protocolo de comunicação bi-direcional entre o leitor das tags e as próprias tags RFID [EPCGlobal 2005], no entanto não define nenhum mecanismo de segurança para o sistema. Embora estejam sendo desenvolvidas novos tipos de tags classificadas como Classe 1-Gen 2 (ou geração 2) estas ainda não dispõem de segurança adequada no protocolo [Duc et al 2006].

A característica implícita de ubiqüidade das tags RFID gera potenciais problemas para a segurança e privacidade dos usuários da tecnologia [Juels 2004]. Uma

1 O propósito do EPCGlobal é padronizar as aplicações e produtos RFID com o objetivo de massificar a utilização e comercialização destas soluções no mundo

simples tag RFID envia seu código EPC[2][EPCGlobal 2004] (ao ser solicitado) para algum leitor próximo de sua área de alcance. Apenas esta característica já nos mostra uma potencial violação de privacidade do usuário. Por exemplo, pode-se com uma leitura saber informações sobre um objeto carregado pela pessoa, ou até mesmo as medicações por ela utilizadas, entre várias outras informações pessoais que podem ser obtidas.

Para diversas aplicações como as citadas anteriormente, um esquema que utilize criptografia e autenticação pode ser desejável, caso contrário pode tornar o sistema vulnerável a diversos tipos de ataques. Como exemplo, pode-se citar um supermercado que empregue a tecnologia para identificação de seus produtos, evitando que usuários mal-intencionados possam levar sua própria tag, substituir a tag original de um produto com o fim de obter um preço mais barato para a compra. Outro exemplo é que um leitor não autorizado ao consultar as tags de um caminhão de carga, poderá obter informações indevidas e ter idéia do valor da mercadoria transportada e deixando o veículo vulnerável a possíveis roubos.

Dessa forma, o objetivo desta dissertação está na análise dos diversos problemas de segurança que envolvem a utilização de um sistema de baixo custo e na proposta de extensões de segurança ao protocolo EPCGlobal. O foco principal é aplicar as propostas no protocolo para tags Classe 1, que é o padrão utilizado em larga escala atualmente; no entanto, as extensões podem ser aplicadas também ao protocolo para tags Classe 1 Gen-2 (Geração 2) [EPCGlobal 2005b]. As extensões visam manter total compatibilidade com o padrão existente e atacar os problemas de uma gama de aplicações sensíveis aos diversos tipos de ataques existentes, objetivando assim a utilização de todos os recursos e vantagens oferecidas pela tecnologia sem que pessoas mal-intencionadas possam tirar vantagens ou mesmo prejudicar os utilizadores do sistema. O algoritmo de criptografia simétrica TEA (*Tiny Encryption Algorithm*) [Wheeler et al 1995] foi avaliado e utilizado nas extensões de segurança.

2 EPC - Código eletrônico de produto (*Eletronic Product Code*) é simplesmente um número de identificação único no mundo inteiro para um objeto específico definido pela EPCGlobal [EPCGlobal 2004].

1.2. ORGANIZAÇÃO

A dissertação está estruturada como segue: O Capítulo 2 explora as características de um sistema RFID com todas suas características, o funcionamento do protocolo de comunicação, aspectos de segurança envolvidos e trabalhos relacionados a segurança e privacidade em RFID, detalhando assim o estado da arte. Trata também de criptografia e autenticação e sua importância no processo de segurança da informação e como pode ser utilizada em Identificação por radiofreqüência. O Capítulo 3 apresenta a especificação e validação de um protocolo de autenticação e sua padronização como extensões de segurança para o protocolo de comunicação EPCGlobal para tags Classe 1. As justificativas de utilização do algoritmo TEA e cenários de aplicações beneficiadas também são abordadas no Capítulo 3. O Capítulo 4 apresenta as conclusões e pesquisas futuras da proposta apresentada neste trabalho.

2. REVISÃO DE LITERATURA

"A potencial redução de custos com o uso do sistema RFID provavelmente fará das tags RFID um dos chips mais fabricados da história." [Weis et al 2003b]

" O sistema de identificação por radiofreqüência (RFID) permite recuperar, à distância, sem fio (wireless), informações armazenadas em um pequeno objeto preso ou incorporado a bens, produtos ou seres vivos. O objeto identificador é capaz de reconhecer e responder a um sinal recebido do sistema de identificação." [Gutierrez et al 2005]

2.1. O SISTEMA RFID

Sistemas RFID são compostos por 3 componentes principais: tag RFID, leitor RFID e banco de dados [Weis et al 2003b]. Alguns autores, no entanto, afirmam que apenas com leitores e tags pode-se considerar também um sistema RFID [Finkenzeller 2003] [Knospe 2004]. Entretanto todos eles falam da importância do banco de dados para considerarmos o sistema completo do ponto de vista prático. A Figura 1 [Finkenzeller 2003] ilustra os componentes do sistema RFID.

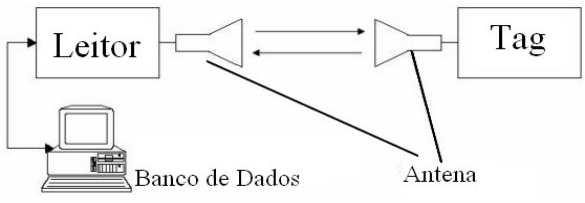


Figura 1. Componentes do Sistema RFID.

A Tag RFID, ou *transponder*, transporta as informações que identificam o objeto. O Leitor RFID, ou *transceiver*, lê e escreve uma informação na tag. O Banco de Dados armazena registros associados aos conteúdos das tags.

O Leitor e a Tag possuem uma antena de comunicação utilizadas para o envio de requisições para a tag e o envio de respostas para o leitor. O Leitor, por sua vez, pode comunicar-se com um banco de dados externo através de uma interface adicional, permitindo assim o encaminhamento dos dados para outro sistema. Os três componentes mostrados na Figura 1 (tags, leitores e banco de dados) serão detalhados a seguir.

2.1.1. TAGS

As tags são tipicamente compostas externamente por um *microchip*, para armazenamento e computação, e uma antena para comunicação [Ranasinghe et al 2004]. A memória da tag pode ser de leitura apenas, de uma escrita e várias leituras, ou totalmente regravável. A Figura 2 [Finkenzeller 2003] ilustra um exemplo de tag RFID.

Figura 2. Tag RFID.

No centro da Tag em cor escura e circulado tem-se o *microchip*. Ao redor do *chip* temos a antena de comunicação, ilustrada por linhas claras em volta do centro e por toda a área da Tag.

As tags internamente são formadas por três componentes básicos e um componente de criptografia (não padronizado) ilustrados na Figura 3 [Aigner e Feldhofer 2005]: *front-end*, *controlador*, *EEPROM e Hardware de Criptografia*. O *front-end* é responsável pela modulação e demodulação dos dados e pelo fornecimento de energia para a tag. O *controlador* é responsável pela implementação dos recursos de software necessários como codificação, comandos do protocolo, mecanismos anti-colisão e detecção de erros. A

EEPROM armazena dados específicos da tag como o identificador único (ID ou código EPC) [EPCGlobal 2004] e a chave de criptografia. Para a utilização do protocolo de autenticação e segurança outro componente de hardware deve ser utilizado. Chamamos de *Hardware de Criptografia* o local onde serão implementadas as rotinas de segurança.

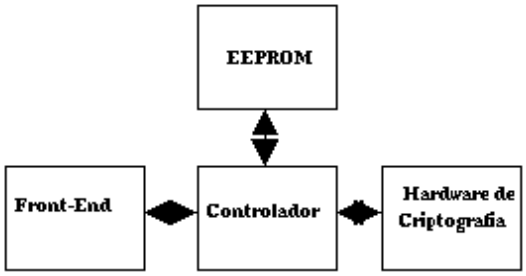


Figura 3. Organização interna de uma tag RFID.

As tags RFID podem ser classificadas pela característica de sua fonte de energia que pode ser ativa, semi-passiva e passiva [Knospe 2004]. Tags ativas contêm uma fonte de energia própria, como uma bateria, possuindo a habilidade de iniciar sua própria comunicação com o leitor como também com outras tags. Tags semi-passivas possuem bateria, mas podem apenas responder à transmissões que cheguem até elas. Tags passivas recebem toda energia através do leitor e não podem iniciar nenhuma comunicação por iniciativa própria. Para oferecer uma analogia de como o processo de energização passiva funciona, devemos pensar que os leitores "gritam" para as tags passivas, e depois extraem os dados resultantes do "eco". Tags passivas são completamente inativas na ausência de um leitor [Weis 2003].

Outra forma de se classificar as tags é através da capacidade de ler e escrever dados. A EPCGlobal [EPCGlobal 2004] define cinco classes descritas a seguir.

Tags Classe 0 são etiquetas somente de leitura, pré-programadas pelo fabricante. Os dados são apenas uma *string* alfanumérica identificadora e sua função é apenas indicar sua presença, não contendo um código EPC gravado. Geralmente são utilizadas como etiquetas anti-roubo utilizadas em bibliotecas e lojas.

Tags Classe 1, abordadas nesta dissertação, são etiquetas onde grava-se uma vez e lê-se várias (*Write once – read many*). Podem ser programadas pelo fabricante ou pelo

usuário. A idéia principal é que os dados são gravados apenas uma vez. São geralmente usadas para a simples identificação de produtos ou em cartões de acesso. São o tipo de etiquetas mais utilizadas no mundo inteiro devido a sua especificação estar completa e operacional. Encontra-se em fase de desenvolvimento pela EPCGlobal uma categoria de tags Classe 1 chamadas de Classe 1 Gen 2, ou seja, Geração 2. Estas são mais modernas, possuem de fábrica uma função geradora de números aleatórios entre outros pequenos avanços na área de segurança [Duc et al 2006].

Tags Classe 2 permitem tanto leitura quanto gravação (várias vezes). Tipicamente possuem mais memória e capacidade de gerar logs. Assim como as tags Classe 3 e Classe 4 não estão disponíveis no mercado e estão em fase de especificação.

Tags Classe 3 também possuem capacidade de leitura e escrita mas também possuem sensores acoplados. Possuem circuitos para registrar temperatura, pressão entre outros.

Finalmente as tags Classe 4 são as mais sofisticadas que além de possuírem todos os recursos oferecidos pelas Classes anteriores, estas possuem capacidade de iniciar comunicação com outras tags ou dispositivos com ou sem a presença de um leitor. Para operar independentemente do leitor estas tags precisam possuir bateria própria.

Por último, também classificamos as tags através de sua freqüência de operação. A Tabela 1 ilustrada adiante mostra as freqüências de operação definidas pela EPCGlobal.

Tabela 1. Frequências de operações definidas pela EPCGlobal para tags RFID.

Freqüência	<135Khz	13.56Mhz	860-930Mhz	2.45Ghz
Fonte de energia	Passiva	Geralmente Passiva	Ativa ou passiva	Ativa ou passiva
Características	Baixo custo, baixa velocidade de leitura, níveis baixos de energia, sensível a ruídos.	Baixo custo, velocidade média de leitura e menos sensível a ruídos.	Mais cara e alta velocidade de leitura.	A mais cara de todas, altíssima velocidade de leitura.
Exemplos	Imobilizadores de fechaduras de carros, controle de acesso.	Controle de acesso e *smartcards*.	Controle de estoque e inventário, cadeia de suprimentos.	Produtos de alto valor agregado.

As ondas de radiofreqüência variam de 30Khz a 300Ghz. Apenas algumas bandas de freqüências estão disponíveis para sistemas RFID. Certas faixas de freqüências adequam-se melhor a aplicações específicas. Sistemas RFID que operam nas faixas de 13.56Mhz e entre 860-930Mhz (terceira e quarta colunas) são os mais disseminados e utilizados pela industria. A freqüência de 13.56Mhz é bastante comum devido a disseminação dos cartões inteligentes (*smartcards*) baseados na tecnologia RFID. Já a freqüência entre 860-930Mhz são muito práticas para utilização em controle de estoque e cadeia de suprimentos pois oferece um bom alcance (alguns metros) como também possui o recurso da leitura de várias tags em alta velocidade. A presente dissertação utiliza como base para a proposta do Capítulo 3 tags que operam na faixa de 860 a 930Mhz.

O dado de identificação armazenado nas tags é um código eletrônico do produto, do tipo EPC (*Eletronic Product Code*) [Brock 2001] que é um número padronizado único para a identificação de objetos na rede EPCGlobal [EPCGlobal 2004]. É derivado de um sistema de numeração também padronizado com a capacidade de identificação única de objetos em nível global. O tamanho de representação usualmente utiliza códigos variando de 64 bits até 256 bits [Ranasinghe et al 2004].

Com relação à segurança desse sistema de numeração global temos que o código EPC é simplesmente um número de identificação para um objeto específico, não possuindo nenhuma informação adicional armazenada no código. Todas as informações associadas às EPC's são encontradas na rede EPCGLobal e somente é acessível para usuários autorizados.

2.1.2. LEITORES

Os leitores são os dispositivos que interrogam as tags por seus dados, através de uma interface de radiofreqüência. A comunicação com as tags além de envolver a requisição dos dados das tags pode ainda incluir comandos de escrita de dados na tag, se esta permitir [Ranasinghe et al 2004]. Leitores possuem sua própria fonte de energia, capacidade de processamento e uma antena de comunicação. Podem ser dispositivos a parte, ou seja, com funcionalidade apenas de leitor RFID, podem aparecer acoplados a outros dispositivos como *handhelds* e *palmtops* ou mesmo aparecerem fixos em prateleiras inteligentes [Knospe 2004].

No contexto do sistema RFID, os leitores aparecem como a entidade central, pois se localizam no centro da comunicação, entre as tags e o banco de dados. A Figura 4 ilustra o posicionamento e fluxo de dados entre todas as entidades do sistema.

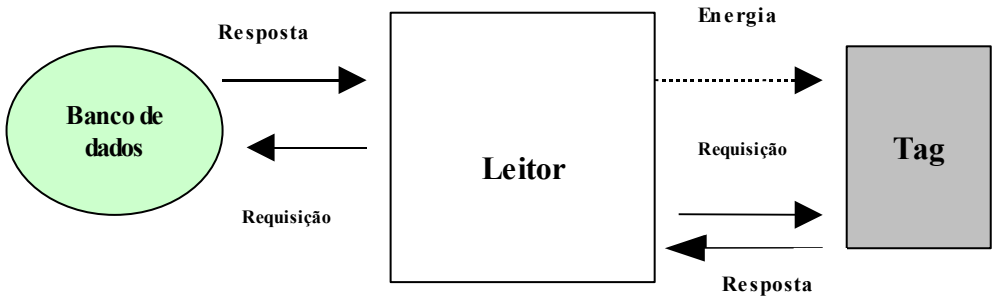


Figura 4. Posicionamento e fluxo de dados entre os componentes do sistema RFID.

O Leitor transfere energia para a Tag com a emissão de ondas eletromagnéticas através do espaço, no caso da tag ser do tipo passiva como ilustrado na Figura 4 pela seta tracejada. Em seguida, o leitor envia uma requisição para a tag, que responde. Recebendo a resposta da tag o leitor faz o processamento, ou seja, interage com uma aplicação externa que pode envolver um banco de dados ou uma aplicação específica [Moroz 2004][Finkenzeller 2003].

Segundo [Weis 2003], para prover funcionalidade adicional, os leitores podem também dispor de armazenamento de dados interno, computação de cálculos criptográficos entre outras funções de acordo com as necessidades da aplicação.

Chamamos de canal *forward* o canal leitor-tag e de *backward* o canal tag-leitor. O alcance de ambos os canais depende dos tipos de dispositivos utilizados, e podem variar de alguns centímetros a vários metros de distância [Knospe 2004]. Independente dos dispositivos utilizados no sistema, sempre existirá a chamada assimetria entre os canais *forward* e *backward*, ou seja, teremos que o canal *forward* terá um alcance bem maior que o canal *backward*, o que gera problemas de segurança que serão discutidos na seção 2.2.

2.1.3. BANCO DE DADOS

Segundo [Weis 2003], um sistema RFID só torna-se efetivamente útil na prática quando os leitores interagem de alguma maneira com um banco de dados externo, ou outro sistema computacional. O termo Banco de Dados é empregado no contexto do sistema RFID como uma forma de armazenamento externa de dados.

Devido à flexibilidade desta entidade, temos que base de dados independentes podem ser construídas por qualquer pessoa para atender as necessidades da aplicação específica, como ilustraremos na seção a seguir.

Os dados armazenados geralmente incluem nome do produto, fabricante, validade, chave de criptografia entre outros detalhes escolhidos de acordo com os usuários do sistema.

2.1.4. APLICAÇÕES RFID

A tecnologia RFID pode ser aplicada em inúmeras situações, como na segurança e no controle de acesso a prédios e áreas restritas, sendo a liberação feita com a utilização de cartões RFID que destravam as portas, catracas ou cancelas.

No travamento e destravamento de veículos as chaves com tags são detectadas à distância, liberando as travas das portas e a partida do veículo. No rastreamento de livros e processos, a tecnologia é usada para localização de livros e processos arquivados fora de ordem e, por isso, perdidos.

A movimentação de bagagens em aeroportos pode ser beneficiada fazendo as bagagens serem etiquetadas com tags contendo o número do vôo, o nome do passageiro e um número seqüencial que as identificam. São rastreadas durante a sua colocação nas aeronaves, minimizando a ocorrência de malas perdidas.

Já o controle de estoques faz com que todos os itens sejam identificados, sendo possível detectar a saída de um item do estoque. O software de controle notifica a remoção do item, associada à identificação do funcionário e ao horário em que isso aconteceu. Promove, também, uma varredura eletromagnética para leitura de todos os itens remanescentes no estoque. Se for o caso, efetua um comando para reposição.

O rastreamento animal, muito utilizado na pecuária com a colocação de tags ou *transponders* em animais permite que eles sejam identificados e associados a dados individuais e históricos de movimentação, sanidade, administração de medicamentos, etc. É uma aplicação bastante utilizada no Brasil.

No controle da cadeia de suprimentos a identificação de embalagens pelos fornecedores permite a uma empresa distribuidora ou varejista um controle mais preciso e ágil de sua cadeia de fornecedores, melhorando a gestão dos estoques e reduzindo perdas. É um dos campos mais promissores para utilização da tecnologia.

2.1.5. CÓDIGO ELETRÔNICO DE PRODUTO E A REDE EPCGLOBAL

O EPC utiliza a tecnologia RFID e foi criado como alternativa ao código de barras. Contudo, para que seja adotado em massa na identificação de mercadorias, mais importante do que se conseguir a redução dos preços das tags é se estabelecerem padrões globais para codificação da informação neles armazenada e para a sua recuperação. Foi esta visão que, em 2003, levou à criação da rede EPC. Trata-se de uma rede global e aberta para rastreamento de bens. Sua infra-estrutura é constituída por três elementos principais: o código EPC, o Serviço de Nomeação de objetos (ONS - *Object Name Service*) e a Linguagem de Marcação Física (PML - *Physical Markup Language*) [EPCGlobal 2004].

O código EPC é um número único que, atribuído a um item qualquer da cadeia de suprimentos, por meio de uma etiqueta nele fixada, permite que esse item seja identificado de forma também única. Assim, cada etiqueta, na verdade uma tag RFID, contém um código EPC. É constituído por um cabeçalho e três grupos de dados. O cabeçalho indica a versão EPC que está sendo utilizada. O primeiro grupo de dados identifica o fabricante do item e o segundo grupo, o tipo exato do produto, seja ele item individual ou múltiplo. O terceiro grupo de dados corresponde ao número seqüencial que identifica cada exemplar do produto, como por exemplo cada garrafa de água, cada caixa de sabão em pó etc [EPCGlobal 2004].

O Serviço de Nomeação de Objetos (ONS – *Object Name Service*) é um serviço de rede automático e é baseado no Sistema de Nomes de Domínios (DNS - *Domain Name System*) da Internet, que associa a um nome (mnemônico) um endereço IP. Dessa maneira, ao ser consultado pelo *middleware* do sistema RFID sobre um determinado EPC, o

ONS indica o endereço IP do servidor Web onde a informação sobre o EPC está armazenada [EPCGlobal 2004].

Os padrões EPC mais conhecidos correspondem a tags passivos. Entretanto, outros tipos de tags, ativos, foram definidos pela EPCglobal com o desenvolvimento de suas especificações. Serão certamente mais caros, porém deverão incorporar funções como a detecção de movimento ou o rastreamento de dados em tempo real, sendo destinados a aplicações de alta segurança e alto valor. O padrão considerado para a especificação do Capítulo 3 baseia-se no padrão EPCGlobal para tags Classe 1 com detalhamento na próxima seção.

2.1.6. PADRÕES PARA RFID

A padronização objetiva definir a mais eficiente plataforma na qual a indústria pode operar. A ISO (*International Organization for Standardization*) e a EPCGlobal [EPCGlobal 2005] promovem a padronização de RFID [Moroz 2004]. A presente dissertação abordará o padrão EPCGlobal para tags Classe 1 [EPCGlobal 2005] por ser este o mais utilizado nas aplicações típicas.

O padrão ISO/IEC 18000 descreve a comunicação das tags RFID com um leitor, com detalhamento e especificação da modulação, enquadramento, métodos anti-colisão e parâmetros de protocolo [ISO 2004].

O protocolo de transmissão define como trocar instruções e dados entre o leitor e a tag em ambas as direções. É baseado no conceito de "o leitor fala primeiro", isto é, nenhuma tag inicia uma comunicação, a não ser que esta tenha recebido e decodificado adequadamente a instrução enviada pelo leitor. A Figura 5 ilustra o conceito de que a comunicação inicia-se a partir de uma requisição do Leitor.

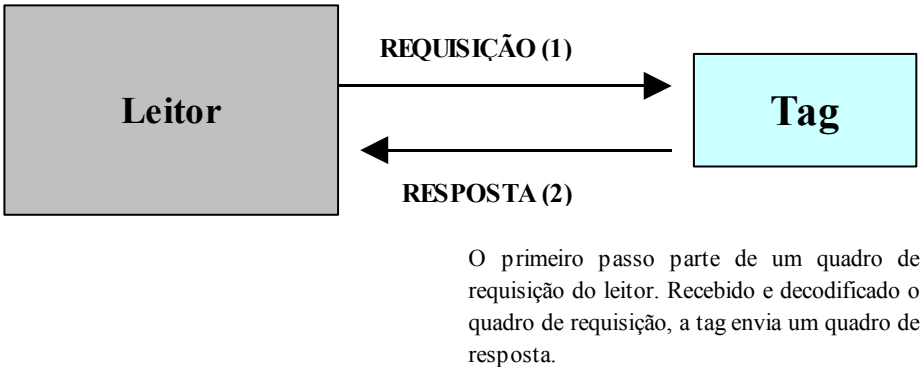


Figura 5. Seqüência de envio de quadros entre leitor e tag.

Todo comando consiste de uma requisição do leitor para a tag e uma resposta da tag para o leitor. Requisições e respostas são colocadas dentro de um quadro com os delimitadores de início de quadro (SOF - *Start of Frame*) e fim de quadro (EOF- *End of Frame*). O protocolo é orientado a bit e o número de bits transmitidos em um quadro é múltiplo de oito. A Figura 6 ilustra o formato dos quadros de requisição e resposta [ISO 2004]. Na Figura 6 (a) temos o formato do quadro de requisição, ou seja, o que parte do leitor para a tag. Todos os quadros iniciam com o delimitador de início de quadro (SOF - *Start of Frame*) e terminam com o delimitador de fim de quadro (EOF - *End of Frame*). O campo *flags* que é formado por oito bits, cada um deles indicando uma informação específica, como erro de escrita, erro de transmissão entre outros. Cada bit é responsável por fornecer uma informação específica, com descrição e detalhamento específico [ISO 2004]. O código de comando indica qual é o comando que está sendo enviado à tag. O protocolo especifica três comandos obrigatórios mostrados na Tabela 2. Estes comandos devem estar implementados na tag. Comandos opcionais podem ser implementados dependendo da aplicação e podem ser utilizados pelos fabricantes para incluir seus próprios comandos no protocolo. O campo de parâmetros é utilizado para processar informações relevantes ao tipo de comando que está sendo pedido, como por exemplo, escrever determinado dado no local especificado pelo parâmetro. A checagem de redundância cíclica (CRC – *Cyclic Redundancy Check*) é calculada em todos os bytes depois do SOF sem incluir o próprio CRC. É utilizado para a

detecção de erros durante a transmissão.

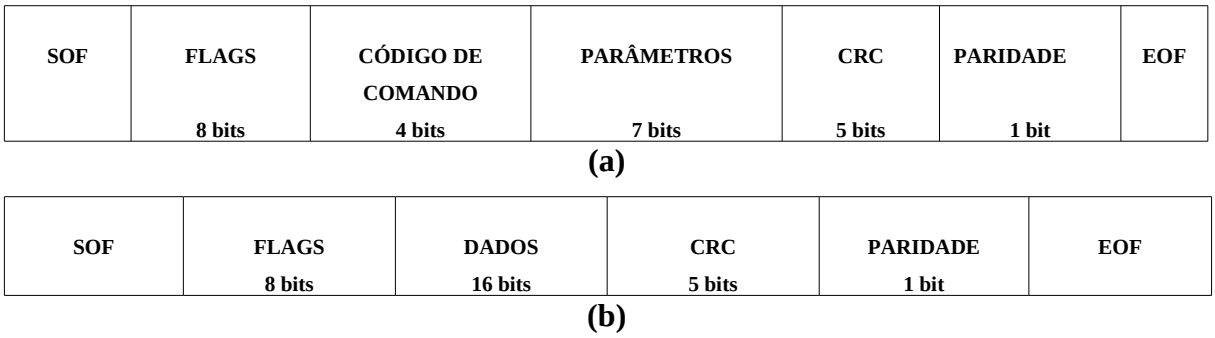


Figura 6. Formato de quadros de requisição e resposta.

Tabela 2. Comandos obrigatórios do protocolo de comunicação.

COMANDO	DESCRIÇÃO
LOAD	É utilizado para carregar os registradores internos com os dados de entrada enviados pelo leitor ou mesmo com o conteúdo da memória.
READ	É utilizado pelo leitor para consultar o conteúdo dos registradores da tag. Este comando gera uma resposta da tag.
WRITE	É utilizado para escrever o conteúdo enviado pelo leitor no endereço especificado também pelo leitor.

Na parte (b) da Figura 6 temos o formato do quadro de resposta enviado da tag ao leitor. Os campos *SOF*, *flags*, *CRC*, *Paridade* e *EOF* possuem mesma função que os dos quadros de requisição. O campo de dados com 16 bits carrega a informação solicitada pelo leitor, por exemplo, a ID[3] da tag. Nesse caso, vários quadros de resposta serão enviados para que o leitor receba a ID completa da tag.

A Tabela 3 mostra os principais parâmetros especificados para os canais de comunicação (tag-leitor e leitor-tag) que devem ser seguidos pelos fabricantes sendo que outros podem ainda ser especificados [ISO 2004].

3 Código EPC da tag, identificador da tag ou identificador único.

Tabela 3. Parâmetros dos canais de comunicação.

NOME DO PARÂMETRO	DESCRIÇÃO
Taxa de transmissão tag-leitor (*bit rate*)	45Kbits/s para recebimento e 20kbits/s para resposta.
Taxa de transmissão leitor-tag (*bit rate*)	45Kbits/s para transmissão e 20Kbits/s para recebimento.
Freqüência de operação	Variável
Freqüência padrão de operação	800-960Mhz
Quem fala primeiro	O leitor fala primeiro
Tamanho da memória	De 16 bits a 1Mbits
Mecanismo anti-colisão	Probabilístico ou determinístico

O padrão EPCGlobal [EPCGlobal 2005] detalha o protocolo em seu documento de forma similar ao padrão ISO, no entanto este possui algumas diferenças que serão explicitadas com detalhes adiante.

Uma tag Classe 1 contém um identificador único, código para correção de erros aplicado ao identificador e um pequeno "*password*" como únicos dados e informações. O identificador único será uma representação válida EPC. O código para detecção e correção de erros será uma Checagem de redundância cíclica (CRC - *Cyclic Redundancy Check*). Com relação ao "*password*", não existem restrições.

Os dados de uma tag Classe 1 são logicamente armazenados na Memória de Identificação da Tag (ITM - *Identifier Tag Memory*). A ITM é organizada logicamente como uma memória linear com o Bit Mais Significante (MSB – *Most Significant Bit*) do CRC localizado na posição zero (0). O Bit Menos Significante (LSB – *Least Significant Bit*) do CRC é seguido pelo MSB do código EPC. O LSB do código EPC é seguido pelo MSB do "*password*". O LSB do "*password*" é o último bit do ITM.

O identificador único são representações válidas de códigos EPC definidas pela EPCGlobal. Todos EPC's válidos possuem quatro partes: versão, gerente de domínio, classe de objeto e número serial, ordenados nesta ordem a partir do MSB até o LSB. Dessa forma, o MSB do EPC é o MSB da versão.

O CRC é calculado sobre o código EPC completo com o bit mais significativo do EPC sendo o primeiro bit a ser checado pelo algoritmo de CRC. O resultado é um valor de

16 bits para códigos EPC com até 256 bits.

O "*password*" é uma "*string*" de 8 bits utilizada pelo comando KILL definido posteriormente.

A Figura 7 ilustra a organização interna da memória da tag. Temos inicialmente o valor do CRC seguidos pelo código EPC que pode ser até de 256 bits e terminando com um *password* de 8 bits.

CRC	**EPC**	**PASSWORD**
(16 bits)	**(até 256 bits)**	**(8 bits)**

Figura 7. Memória da tag.

O protocolo para tags Classe 1-Gen 2 especifica também que a tag possui uma função geradora de números aleatórios [EPCGlobal 2005b].

Outro aspecto importante definido no padrão EPCGlobal é a implementação de estados da tag. Tags EPCGlobal devem ter a capacidade de guardar informação de estado. O protocolo define sete estados: Pronto (*ready state*), Arbitrário (*arbitrate state*), Resposta (*reply state*), *Acknowledge state*, Aberto (open state), Seguro (*secure state*), Morto (*killed stated*). A tag encontra-se em estado "Pronto" quando ela está esperando por uma energização, ou seja, não está participando de nenhuma requisição no momento. O estado "arbitrário" é similar ao "Pronto", exceto que a tag deve estar participando de uma consulta. Após responder a uma consulta a tag entra no estado de "Resposta". Os demais estados são utilizados quando o recurso do password é usado [EPCGlobal 2005b].

A comunicação entre leitor e tag ocorre na forma de pacotes ou quadros onde um único pacote possui um comando completo do leitor e uma resposta completa da tag. O comando e a resposta permitem uma comunicação "*half-duplex*" entre leitor e tag. A Tabela 4 mostra o formato do quadro de dados enviado pelo leitor com a descrição dos respectivos campos.

Tabela 4. Formato e campos dos quadros de dados.

Campo	Nro. Bits	Descrição
PREAMBL	-	Preâmbulo.
CLOCKSYNC	20	Todo comando tem como prefixo uma série de 20 zero's binários para ser utilizado como sincronização de *clock*. O circuito de sincronização da tag utiliza este valor para estabelecer seu "*timer*" para leitura/decodificação.
SOF	1	Início de quadro. Um binário de valor um (1).
CMD	8	Especifica o comando que está sendo enviado à tag.
P1	1	Bit de paridade do campo [CMD].
PTR	8	Ponteiro para uma posição no identificador da tag. Este campo é o ponto de início para a tentativa de comparar dados especificados no campo [VALUE] definido posteriormente.
P2	1	Bit de paridade do campo [PTR].
LEN	8	Comprimento dos dados que estão sendo enviados no campo [VALUE]. Este valor deve ser maior do que zero (0).
P3	1	Bit de paridade do campo [LEN].
VALUE	variável	Este é o dado que a tag irá comparar com seu identificador após algum dos comandos ScrollID, PingID, Quiet, Talk e Kill. (A partir do [PTR] até o LSB).
P4	1	Bit de paridade do campo [VALUE].
P5	1	Bit de paridade de todos os campos de paridade.
EOF	1	Indicador de final de quadro. Um binário de valor um (1).

A Tabela 5 ilustra os comandos obrigatórios que devem ser implementados pelo leitor, ou seja, os comandos que qualquer sistema RFID deve disponibilizar. A coluna 1 mostra o nome do comando. A coluna 2 mostra o código do comando, ou seja, valor que aparecerá dentro do campo "*CMD*" do quadro de comando do leitor. A coluna 3 mostra o comando que a tag enviará como resposta e a coluna 4 apresenta a descrição do comando.

Tabela 5. Comandos obrigatórios.

Nome do comando	Código do comando	Resposta da tag	Descrição
ScrollAlID	0011 0100	ScrollID Reply	Todas as tags responderão enviando um Preâmbulo de 8 bits seguidos pelo CRC (bit MSB primeiro) e pelo código EPC completo (MSB do identificador primeiro).
ScrolID	0000 0001	ScrollID Reply	Tags dentro do valor definido em [VALUE] iniciando na posição [PTR] irão responder com um quadro contendo um Preâmbulo de 8 bits seguidos do CRC (MSB enviado primeiro) e do código EPC completo (MSB do identificado primeiro).
PingID	0000 1000	PingID Reply	Tags dentro do valor definido em [VALUE] iniciando na posição [PTR] irão responder enviando 8 bits do identificador iniciando no bit [PTR] + [LEN].
Quiet	0000 0010	Nenhuma	Tags dentro do valor definido em [VALUE] iniciando na posição [PTR] entrarão no modo desativado onde eles não responderão ou executarão comandos do leitor. Este modo é mantido até que um comando Talk seja recebido e corretamente interpretado.
Talk	0001 0000	Nenhuma	Tags dentro do valor definido em [VALUE] iniciando na posição [PTR] entrarão no modo ativo onde eles poderão responder aos comandos do leitor. Este modo de operação é mantido até que um comando Quiet seja recebido e corretamente interpretado.
Kill	0000 0100	Nenhum	Tags dentro do valor definido em [VALUE] iniciando na posição [PTR] entrarão no modo desativado definitivamente e não mais responderão a comandos do leitor. Funciona como um auto-destrutor.

Na comunicação tag-leitor uma tag nunca envia comandos ao leitor. A tag apenas executa comandos enviados pelo leitor. Apenas quatro comandos são obrigatórios para a tag interagir com o leitor: VerifyID, ScrollAllID, ScrollID e PingID.

As tags respondem aos comandos ScrollAllID e ScrollID com uma resposta do tipo ScrollID Reply. O formato é ilustrado na Tabela 6 adiante.

Tabela 6. Formato do quadro ScrollID Reply.

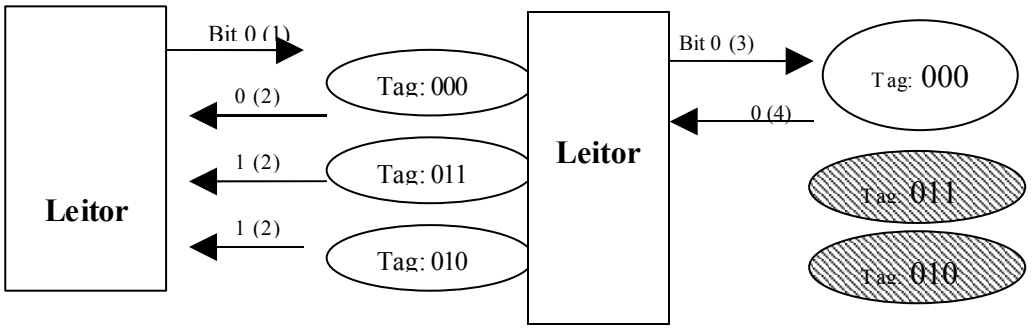
Campo	Número de bits	Descrição
PREAMBL	8	Preâmbulo.
TAGID	Variável	Código EPC da tag.
CRC	16	Cálculo de CRC em cima da TAGID.

As tags respondem ao comando PingID com um PingID Reply. O formato está ilustrado na Tabela 7 a seguir.

Tabela 7. Formato do quadro PingID Reply.

Campo	Número de bits	Descrição
8BITID	8	A tag modula um valor de 8 bits de sua ITM iniciando na posição [PTR]+[LEN].

Se existem mais de uma tag dentro do alcance do leitor, um mecanismo anti-colisão é necessário**.** Um exemplo de algoritmo determinístico simples é o percurso em árvore binária (*binary tree walking*) onde o leitor questionará todas as tags na vizinhança para o próximo bit de seus IDs. Se dois valores diferentes de bits forem transmitidos de um conjunto de tags, o leitor será capaz de detectar a colisão. O leitor então fará uma difusão de um bit indicando se tags que enviaram um 0 ou um 1 devem continuar. Essencialmente, o leitor escolhe uma ramificação da árvore binária de valores de IDs. Tags que não se enquadram na escolha do leitor não continuarão participando do protocolo. À medida que o leitor continua a percorrer as ramificações da árvore binária, poucas tags continuarão operando. Assim no fim do algoritmo apenas uma tag estará respondendo. Este processo de endereçar e isolar uma única tag pode ser chamado de singularização. A Figura 8 a seguir ilustra e explica o algoritmo.



Passo 1. O Leitor transmite o primeiro bit da ID.

Passo 2. As tags enviam o primeiro bit de cada uma.

Passo 3. O Leitor transmite o próximo bit da ID.

Passo 4. As tags enviam o primeiro bit de cada uma. Nesse caso apenas a primeira Tag respondeu, as demais foram eliminadas do protocolo. Nesse momento a única tag que respondeu é singularizada e o algoritmo termina.

Figura 8. Algoritmo para singularização das tags "Percurso em árvore binária".

É importante lembrar que o padrão não obriga a utilização deste ou daquele algoritmo. O algoritmo ilustrado na Figura 8 é apenas um exemplo de um algoritmo simples, eficiente, e de baixo custo de implementação. Outros algoritmos podem ser utilizados de acordo com a vontade do fabricante. Exemplos dele são algoritmos utilizados em redes como o *Aloha* e *Slotted Aloha* [Weis 2003].

2.2. SEGURANÇA

"Segundo o dicionário AURÉLIO, informação é um dado acerca de alguém ou algo; o conhecimento; segundo a teoria da informação, a medida da redução da incerteza. A segurança da informação pode ser definida como a proteção contra um grande número de ameaças às informações, de forma a assegurar a continuidade do negócio, minimizando danos comerciais e maximizando o retorno de possibilidades e investimentos. Tanenbaum [Tanenbaum 2003] caracteriza segurança pela preservação dos três atributos básicos da informação: confidencialidade, integridade e disponibilidade."

Um sistema seguro deve garantir os seguintes benefícios: privacidade,

autenticação, integridade, não repúdio, controle de acesso, e disponibilidade [Stallings 2002].

A privacidade nos garante que ninguém estará "escutando" o que está sendo transmitindo na rede (comunicação leitor–tag e tag-leitor) sem estar autorizado. A autenticação garante que a origem da mensagem foi corretamente identificada, com a certeza de que a identificação não é falsa, ou seja, a origem é realmente quem diz ser. A integridade garante que o que foi transmitido não foi alterado, de forma nenhuma, durante a transmissão. Garante que o que o destinatário recebeu foi exatamente o que o remetente enviou. O não-repúdio consiste no fato de requerer que nem o remetente nem o destinatário de uma mensagem sejam capazes de negar a mensagem, nem de negar que tenha sido enviada, nem negar que tenha sido recebida, se realmente isso tenha acontecido. O Controle de Acesso requer que o acesso à informação possa ser controlado pela rede que contenha a informação. Algumas vezes, se quer dar acesso somente de leitura a um arquivo, ou no caso dos sistemas RFID, pode ser desejável que apenas leitores autorizados consultem o conteúdo das tags. E finalmente, a disponibilidade requer que o sistema computacional esteja disponível para qualquer pessoa autorizada em qualquer momento que ela deseje.

Um ataque consiste na violação da transmissão de uma mensagem de uma origem para um destino. O intruso, ou atacante, é o responsável por interferir nessa transmissão.

O intruso pode ter quatro comportamentos principais em relação às posições da origem e do destino da mensagem. A Figura 9 a seguir [Stallings 2002] a seguir ilustra o comportamento normal, interrupção, interceptação, modificação e fabricação.

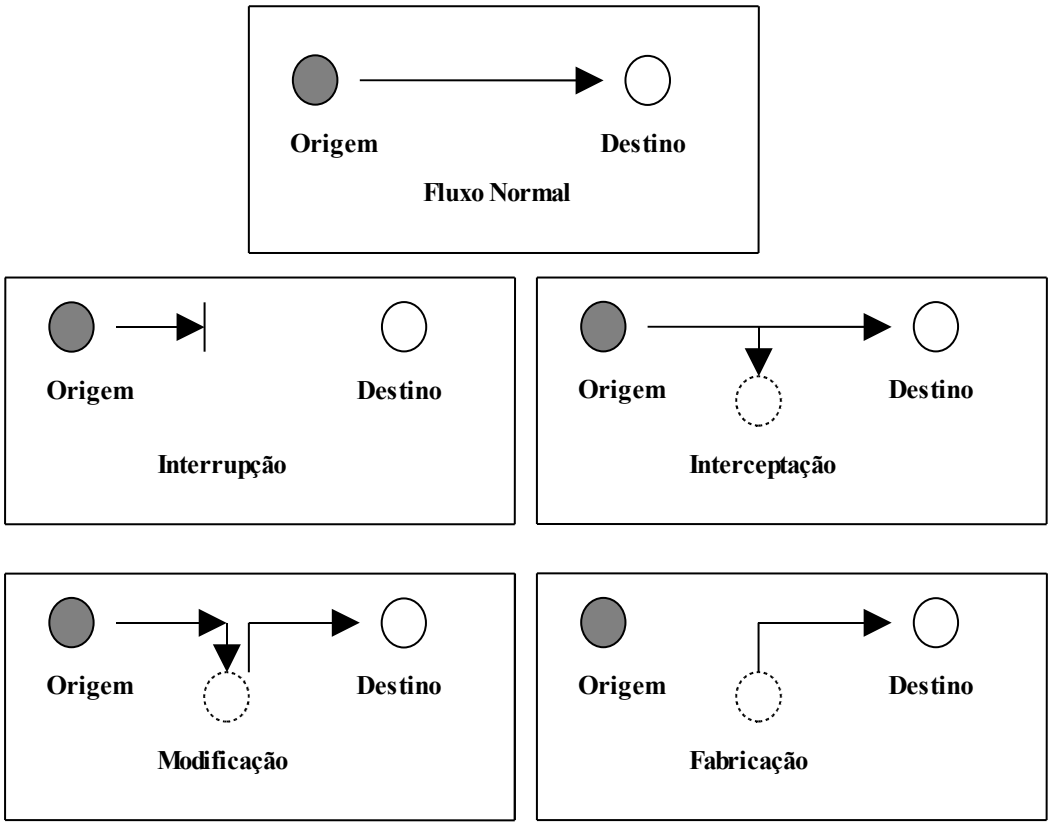


Figura 9. Comportamentos de um intruso em relação às posições de origem e destino.

Na interrupção o intruso objetiva interromper o fluxo de dados que parte da origem, deixando o dispositivo destino sem receber pacotes. Os ataques denominados Negação de Serviço (DoS – *Denial of Service*) são um exemplo de interrupção. Na interceptação o mesmo objetiva apenas tomar conhecimento de todo o fluxo de dados que trafega por essa conexão, também chamado de espionagem. Na modificação, o intruso, além de escutar o tráfego, intercepta os dados e os modifica, enviando-os para o destino. Este ataque pode ser chamado também de *man-in-the-middle*. E na fabricação ele cria dados para enviar para o destino. O dispositivo destino não tem como saber quem está enviando esses dados. A fabricação é muitas vezes chamada de ataque tipo *replay*.

Outro tipo de ataque não listado na Figura 9 é o chamado de reflexão, onde o intruso engana o mecanismo de autenticação aproveitando-se de sessões de autenticação já

abertas [Tanenbaum 2003].

A seguir trataremos dos tipos de ataques e aspectos de segurança em sistemas RFID.

2.2.1. SEGURANÇA EM RFID

Os grandes ganhos de eficiência oferecidos pelos sistemas RFID podem vir com um custo em segurança e privacidade. Vulnerabilidades como ataques físicos, plágio ou falsificação (*spoofing*[4]), captura de tráfego, análise de tráfego e negação de serviço podem todas estar presentes nesses sistemas. E cada um destes riscos pode afetar ambos os indivíduos e organizações [Sarma et al 2003] [Weis et al 2003b] [Henrici 2004]. Até mesmo as mais modernas tags especificadas pela EPCGlobal, as chamadas Gen 2 [EPCGlobal 2005b], não oferecem segurança adequada aos usuários [Duc et al 2006].

Os ataques físicos são os mais perigosos, porém os mais fáceis de serem evitados, pois o intruso precisa ter acesso direto a tag, sendo assim mais fácil de ser detectado. Esses tipos de ataques incluem ataques na fonte de energia, remoção de material, inundação com água, uso de radiação, destruição de circuitos entre outros.

Outro tipo de poder que um intruso pode possuir é o de participar nos protocolos, ou construir suas próprias tags para uso indevido. Ou seja, o atacante pode iniciar consultas a outras tags ou responder consultas de um leitor legítimo. Em [Weis 2003], são propostos alguns esquemas de controle de acesso e serão discutidos adiante na seção 2.4.

Já os ataques do tipo interceptação ou captura e análise de tráfego são chamados de ataques passivos. Analogamente funcionam como "*sniffers*[5]", ou seja, capturam os dados da comunicação e fazem uso indevido destas informações. Assim, o atacante pode "ouvir" as mensagens transmitidas nos protocolos através das capturas das ondas de radiofreqüência, o que muitas vezes não é desejável. A utilização de criptografia torna-se fundamental para se evitar este tipo de problema. A seção 2.3 abordará criptografia e justificará sua utilização em sistemas RFID. A Figura 10 [Weis 2003] ilustra como um intruso

4 Spoofing é o ato de falsificar o remetente de um pacote de transmissão de dados, para que o receptor o trate como se fosse de um outro utilizador. No caso de sistemas RFID é o ato de falsificar a origem ou o destino da comunicação [Stallings 2002].

5 [Ing. Ver. To Sniff (Farejar)] (Farejador). Programa que monitora o fluxo de dados numa rede [Stallings 2002].

pode capturar os dados negociados entre tag e leitor.

Na Figura 10 são ilustrados os alcances dos canais tag-leitor ou "*backward*" e leitor-tag ou "*forward*", assim como o posicionamento de um intruso dentro do campo de alcance do canal leitor-tag. Como o canal *forward* possui alcance bem maior que o *backward*, invasores podem ter acesso a essas informações. A captura de tráfego dentro do canal *backward* também é possível, no entanto é mais fácil de ser detectada, pois o invasor precisa estar bem próximo a tag.

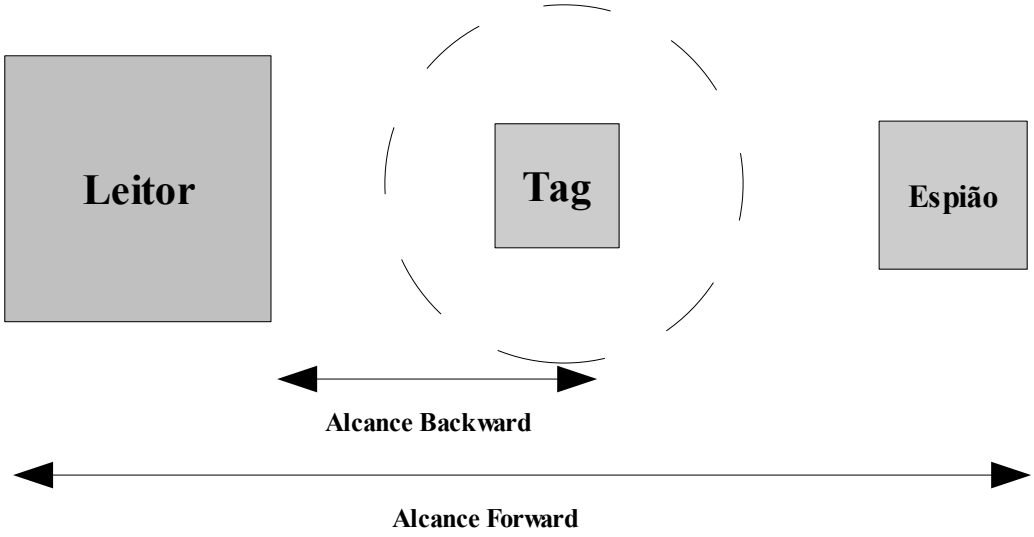


Figura 10. Ataque do tipo interceptação em sistemas RFID.

A negação de serviço[6], tão presente nos sistemas computacionais convencionais também ocorre em sistemas RFID. O atacante pode tentar interferir no sinal de radiofreqüência, corromper difusões ou mesmo bloquear mensagens ou tentar desabilitar a tag por outras formas. São ataques bastante danosos e muito difíceis de serem evitados [Sarma et al 2003].

O plágio ou falsificação da tag também se apresenta como mais um risco de segurança para o sistema RFID. Uma pessoa mal intencionada pode criar sua própria tag para plagiar uma tag válida e enganar um caixa de uma loja, por exemplo. Pode também tentar

6 Um ataque deste tipo tem por objetivo não permitir que um determinado serviço ou servidor responda a requisições e/ou fique bloqueado [Tanenbaum 2003].

reescrever dados na tag para baixar o preço de um produto. A utilização de esquemas de autenticação deve ser implementada para se evitar este tipo de problema. O Capítulo 3 tratará da especificação de mecanismos de autenticação com criptografia para sistemas RFID através de extensões de segurança ao protocolo de comunicação [Weis 2004].

A Tabela 8 a seguir, resume os principais tipos de problemas de segurança relacionados aos sistemas RFID abordados anteriormente e qual abordagem seguir para contornar o problema.

Tabela 8. Principais tipos de problemas de segurança em sistemas RFID.

TIPO	COMO FUNCIONA	ABORDAGEM
FÍSICO	O intruso pode realizar ataques físicos às tags.	Esquemas de segurança física de objetos, como contratação de seguranças, sistemas de câmeras, etc.
PLÁGIO ou FALSIFICAÇÃO (fabricação)	O intruso pode realizar consultas às tags e responder consultas a leitores legítimos.	Esquemas de autenticação e controle de acesso.
INTERCEPTAÇÃO (e modificação)	O intruso captura os quadros que trafegam no ar, também chamado de espionagem.	Esquemas de criptografia dos dados que são enviados entre as entidades leitor e tag.
NEGAÇÃO DE SERVIÇO (interrupção)	O intruso atrapalha as difusões interferindo nos sinais de radiofreqüência.	Muito difíceis de serem evitados, e por isso são bastante danosos.
PRIVACIDADE DE LOCALIZAÇÃO (RASTREAMENTO)	O intruso faz leituras nas tags que o usuário carrega podendo rastrear a localização do usuário.	Esquemas de autenticação e controle de acesso.

Conforme já citado anteriormente, os ataques físicos são bastante perigosos, porém fáceis de serem evitados. A contratação de serviços convencionais de segurança e sistemas de monitoramento eletrônico ajuda a evitar o problema.

Os ataques do tipo falsificação e interceptação são os mais comuns e são alvos do protocolo proposto no Capítulo 3. Quando fala-se em falsificação pode-se imaginar tanto a falsificação de tags como de leitores. O controle de acesso garante que apenas leitores legítimos possam fazer consultas as tags. Na seção 2.3 serão mostrados esquemas de controle

de acesso propostos em [Weis 2003]. Já mecanismos de autenticação permitem que as tags se autentiquem aos leitores e vice-versa, evitando assim que falsas tags respondam aos leitores legítimos. Os diversos tipos de mecanismos de autenticação assim como a justificativa de qual mecanismo utilizar para sistemas RFID serão abordados posteriormente na seção 2.4.

Finalmente, os ataques do tipo negação de serviço que aparecem como ataques bastante danosos por serem extremamente difíceis de serem evitados, pois o intruso interfere no sinal de radiofreqüência interrompendo a comunicação.

2.2.2. SEGURANÇA E ASSIMÉTRIA DOS CANAIS

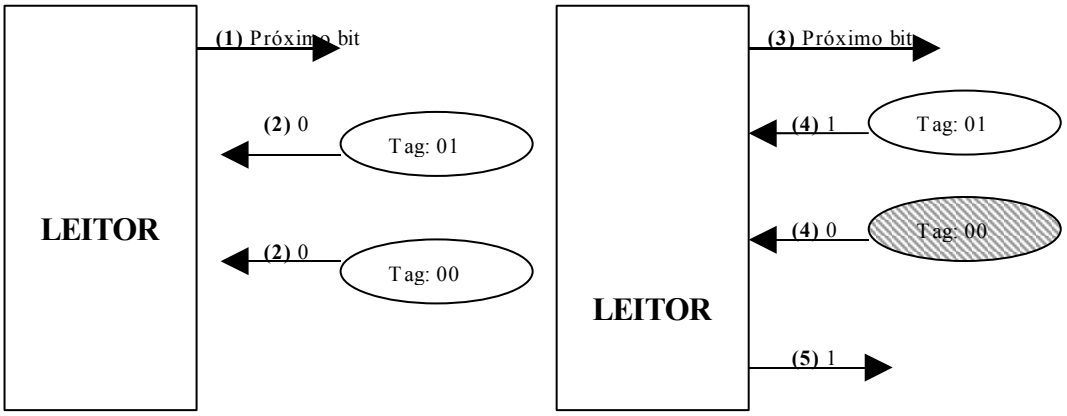
Um aspecto importante de segurança em RFID que deve ser levado em consideração é a potência do sinal leitor-tag (*forward channel*). Pessoas não autorizadas podem ter acesso aos dados que circulam pelo ar (*wireless*), mesmo estando a vários metros de distância. Esse aspecto torna vulnerável o algoritmo que individualiza uma tag, o percurso em árvore binária (*Binary Tree Walking*), pois conforme explicado na seção 2.1.6. à medida que o leitor vai individualizando a tag, parte da ID da mesma é enviada através do canal leitor-tag até que esta esteja totalmente individualizada.

Devido a esse problema [Weis et al 2003b] propôs uma variação do algoritmo de percurso em árvore binária chamado de Percurso Silencioso em Árvore Binária" (*Silent Binary Tree Walking*). Como o próprio nome sugere, essa variação propõe a eliminação da transmissão bit a bit através do canal leitor-tag conforme é ilustrado a seguir:

PERCURSO SILENCIOSO EM ÁRVORE BINÁRIA
1. O leitor solicita o próximo bit à população de tags. 2. As tags enviam o bit solicitado pelo leitor. 3. O leitor informa às tags qual bit deve continuar seguindo no protocolo. O envio deste bit através do canal leitor-tag não é feito de forma direta. O leitor envia o valor XOR do último bit do prefixo com o bit que continua na pesquisa. 4. Volta-se ao passo 1.

A Figura 11 [Weis et al 2003b] ilustra um exemplo de como funciona o algoritmo. O algoritmo utiliza o recurso da assimetria dos canais leitor-tag e tag-leitor (o canal leitor-tag possui alcance bem maior) para enviar os valores sensíveis, no caso os bits do ID da tag. Como o alcance do canal tag-leitor é de poucos metros torna-se fácil detectar a presença

de um intruso atuando nesse canal.



(1) O Leitor solicita o próximo bit.
(2) As tags enviam o primeiro bit (o primeiro 0 e a segundo 0 também).
(3) O Leitor solicita mais um bit.
(4) As tags enviam seus respectivos bits (o primeiro 1 e o segundo 0).
(5) O Leitor envia qual bit continua no protocolo que no caso é o bit 1, eliminando assim a segunda tag e consequentemente individualizando a primeira. Este bit é enviado através de um XOR com o bit anterior do prefixo, no caso o 0 evitando assim que este seja capturado por um intruso.

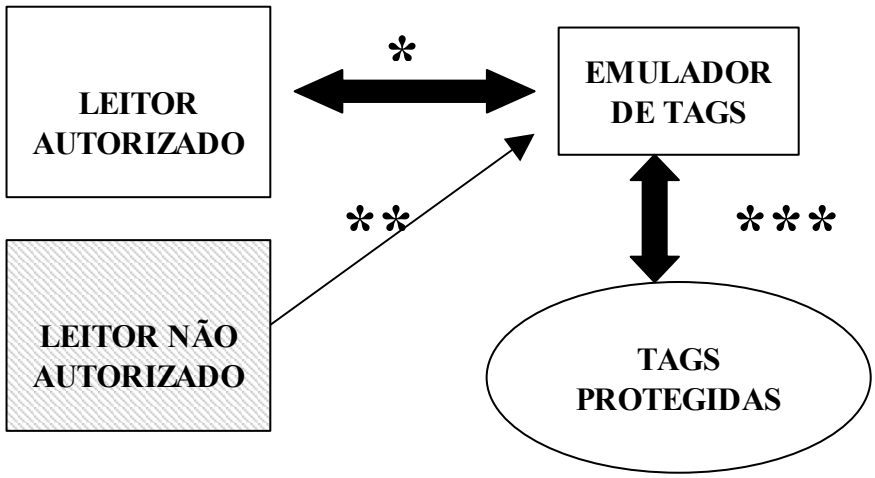
Figura 11. Percurso silencioso em árvore binária (*Silent Binary Tree Walking*).

[Chammas et al 2004], propõe uma abordagem com bom custo beneficio para a implementação de controle de acesso na identificação por radiofreqüência. A proposta chamada de "*Tag Emulator*" ou emulador de tag não utiliza nenhum tipo de algoritmo de criptografia ou função *hash*[7], o que evita uma maior complexidade no desenvolvimento dos circuitos das tags. O princípio básico do emulador de tag é o encapsulamento de um conjunto de tags que necessitam de segurança. Esse mecanismo permite que a informação (o código EPC das tags) somente seja acessível para leitores autorizados. O encapsulamento é obtido forçando com que o emulador aparente ser uma tag comum a todos os leitores. Apenas os leitores autorizados devem saber o código EPC do emulador.

O emulador funcionará nesse esquema como um intermediário entre os leitores

7 Uma função hash no contexto deste trabalho é uma equação matemática que utiliza texto para criar um código chamado Resumo de Mensagem (M*essage Digest)*. Alguns exemplos conhecidos de funções hash: MD4 (MD - *message digest*), MD5 e SHA1 [Tanenbaum 2003].

e as tags, mas para os leitores o emulador é visto como uma tag qualquer. Os leitores autorizados saberão a EPC do emulador que consulta a tag desejada e envia a resposta ao leitor autorizado. A Figura 12 [Chammas et al 2004] ilustra como funciona o sistema utilizando o emulador de tags.



(*) O Leitor autorizado para consultar a tag envia o EPC do emulador que consulta a tag verdadeira e devolve o resultado ao Leitor.

(**) O Leitor não autorizado, por não ter conhecimento da EPC do emulador não recebe resposta alguma, ficando assim sem acesso à tag.

(***) A tag é recebe a requisição do emulador e envia a resposta.

Figura 12. Funcionamento do emulador de tag.

A utilização desse esquema possibilita a disponibilização de controle de acesso ao sistema RFID, impedindo que leitores não autorizados tenham acesso ao conteúdo das tags. Por não utilizar nenhum tipo de algoritmo criptográfico ou função *hash*, temos que o mecanismo de emulador de tags é vulnerável a espionagem, pois o código EPC é transmitido ao emulador pelo leitor autorizado através do canal leitor-tag, que conforme foi dito anteriormente tem grande alcance sendo difícil a detecção de intrusos.

Uma solução recente para a disponibilização de segurança em RFID é o chamado Bloqueador de Tags (*Blocker Tag)*[Juels et al 2003]. Uma tag bloqueadora opera interferindo no protocolo de individualização das tags, ou seja, interferindo no algoritmo de

percurso em árvore binária. Quando o leitor faz a consulta em determinada sub árvore do protocolo, a tag bloqueadora envia ambos os bits 0 e 1 ao leitor, forçando assim uma colisão e fazendo com que o leitor explore a árvore inteira. Na prática a maioria dos leitores desiste da pesquisa após tentar cerca de uma centena de vezes na árvore. Dessa forma temos o bloqueio das tags, ou seja, o leitor não consegue individualizar tag alguma. Esse recurso de bloqueio protege a privacidade dos usuários das tags, evitando consultas não autorizadas. É uma abordagem atrativa devido ao baixo custo de implementação. As tags existentes nos mais diversos objetos não precisam ser modificadas. As próprias tags bloqueadoras podem ter custo bem baixo, pois elas consistem essencialmente de uma ou duas tags normais com pequenas modificações nos circuitos. Assim, temos um custo aproximado de uma tag bloqueadora equivalente ao custo de duas tags normais. Dessa forma a proposta ataca o problema da privacidade gerado com o uso das etiquetas RFID.

Outras propostas também com o objetivo de proteger as tags contra leituras indevidas aparecem nos trabalhos de [Juels 2004] e [Juels 2004b]. Na primeira, o autor sugere uma variação do bloqueador de tags e chama a proposta de bloqueador via software (*soft-blocking)*. Essa proposta envolve a construção de módulos de software no dispositivo bloqueador, oferecendo maior flexibilidade que o simples bloqueador de tags. Os leitores também precisam de programação especial para funcionar em conjunto com o bloqueador via software. Trata-se também de uma proposta que visa respeitar a privacidade dos usuários. Possui um custo um pouco maior devido à necessidade de um dispositivo especial que funcionará como o bloqueador via software. Mesmo assim, tem se mostrado uma solução barata, flexível e alternativa à primeira proposta do bloqueador de tags. No entanto, é importante lembrar que a solução apenas oferece privacidade, deixando de lado objetivos como controles de acesso e autenticação.

Já a segunda proposta, segundo o trabalho de [Juels 2004b], é uma proposta chamada de "*minimalist cryptography*" que é uma solução baseada em pseudônimos pré-programados carregados pelas tags. Através da utilização de diferentes pseudônimos durante sessões diferentes de leituras, a tag evita o rastreamento por entidades não autorizadas. Já uma entidade autorizada tem acesso aos pseudônimos e pode realizar o rastreamento se for o caso. A proposta descreve um protocolo que possui propriedades de autenticação e privacidade,

preocupando-se com as limitações do poder de computação e da capacidade de armazenamento. Não envolve cálculos criptográficos intensivos e nem envolve a necessidade de aumento de recursos das tags. A proposta deixa um pouco a desejar por não basear-se nos padrões de comunicação existentes como o EPCGlobal, base desta dissertação ou mesmo o ISO/IEC 18000.

2.3. CRIPTOGRAFIA

A palavra Criptografia vem das palavras gregas que significam "escrita secreta" [Tanenbaum 2003].

[Menezes et al 2001] define criptografia como sendo o estudo de técnicas matemáticas relacionadas a aspectos de segurança da informação como confidencialidade, integridade de dados, autenticação e autenticação de origem de dados. Não é a única forma de oferecer segurança da informação, mas sim uma dentro de um conjunto de técnicas.

O uso da criptografia tem como objetivo garantir que uma mensagem ou informação só será lida e compreendida pelo destinatário autorizado para isso. Assim, pode-se dizer que os objetivos do uso da criptografia são quatro: confidencialidade, integridade, autenticação e não repúdio. A confidencialidade visa assegurar que só os receptores autorizados tenham acesso às informações. É obtida através da encriptação dos dados. A integridade assegura que a informação não foi alterada durante o processo de transporte da informação. A autenticação permite que o remetente e o receptor possam confirmar as identidades uns dos outros assim como a origem e o destino da informação. O não-repúdio é um serviço que previne que uma entidade negue uma determinada ação realizada por ela [Menezes et al 2001].

Existem dois tipos básicos de criptografia em relação ao uso de chaves: Criptografia Simétrica e Criptografia Assimétrica. A Criptografia Simétrica foi o primeiro tipo de criptografia criado e funciona transformando um texto em uma mensagem cifrada, através da definição de uma chave secreta, que será utilizada posteriormente para desencriptar a mensagem, tornando novamente um texto simples.

Exemplos de algoritmos de criptografia simétrica são o DES, IDEA, RC2,

RC4, *Blowfish* [Silva 2004], TEA (*Tiny Encryption Algorithm*)[Wheeler et al 1995] dentre outros. O algoritmo TEA é descrito a seguir por ser citado na literatura como o mais provável recurso para ser utilizado em sistemas RFID [Weis 2003] [Sarma et al 2003].

Como principal vantagem da criptografia simétrica temos a rapidez na criptografia e desencriptação das informações e um menor número de portas lógicas necessárias para implementação em hardware [Sarma et al 2003]. Como desvantagem temos o gerenciamento de chaves necessário pois a chave secreta deve ser transmitida ou comunicada para o receptor, tornando-a mais vulnerável a roubo.

A criptografia assimétrica é aquela baseada no uso de pares de chaves para cifrar/decifrar mensagens. As duas chaves são relacionadas através de um processo matemático, usando funções unidirecionais para a codificação da informação. Uma chave, chamada chave pública, é usada para cifrar, enquanto a outra, chamada chave secreta, é usada para decifrar.

Uma mensagem cifrada com uma chave pública só pode ser decifrada pela outra chave secreta com a qual esta relacionada. A chave usada para cifrar recebe o nome de chave pública porque ela deve ser publicada e amplamente divulgada pelo seu possuidor, fazendo com que qualquer pessoa possa lhe enviar mensagens cifradas. Já a chave usada para decifrar as mensagens, deve ser mantida em sigilo [Trinta 1998].

Como vantagem principal tem-se a não necessidade de comunicar ao receptor a chave para decifrar a mensagem criptografada. No entanto, a implementação em hardware exige mais recursos e a velocidade é mais lenta que a criptografia simétrica [Trinta 1998].

A implementação de criptografia para RFID necessita de atenção especial em alguns tópicos. Especialmente o poder de computação limitado de um lado e o baixo tamanho e consumo de energia de outro lado devem ser considerados. Isto nos leva a necessidade de implementações eficientes em software e hardware.

O TEA [Wheeler et al 1995], foi proposto inicialmente pelos pesquisadores David Wheeler e Roger Needham da Universidade de Cambridge, para ser usado em plataformas que não possuem grande poder de processamento. O princípio básico desse algoritmo, um dos mais simples da criptografia, é baseado apenas em um grande número de interações com XORs e somas na hora de cifrar e XORs e subtrações na hora de decifrar.

Com isso, a sua complexidade diminui e o seu desempenho melhora, ao contrário da grande maioria dos algoritmos criptográficos existentes na atualidade, que quase sempre são baseados em tabelas [Shepard 2005]. Estima-se que o TEA seja, pelo menos, três vezes mais rápido que o DES [Sarma et al 2003].

O TEA (*Tiny Encryption Algorithm)* é um algoritmo de chaves simétricas, trabalhando com blocos de 64 bits de mensagem e com uma chave de 128 bits, que se subdivide em quatro sub-chaves (K[0..3]) de 32 bits cada. A quantidade de interações mais comumente usada na execução do TEA é de trinta e dois ciclos (com sessenta e quatro operações). Com isso o processamento não se torna tão pesado e a segurança não fica em segundo plano. O processamento para cifrar uma mensagem no TEA é simples: primeiro a mensagem é dividida em dois blocos de 32 bits e cada bloco é processado de três maneiras diferentes; em seguida, é feito um XOR entre esses três processamentos e o resultado final é somado ao outro bloco [Wheeler et al 1995].

A implementação deste algoritmo em hardware requer um número menor de *gates* que a maioria dos algoritmos para sistemas computacionais convencionais, em torno de 2500 *gates* [Buth et al 2002].

O TEA aparece assim como um forte recurso a ser utilizado nos sistemas RFID, por ter se mostrado seguro, simples, e necessitar de menos *gates* em hardware para sua implementação [Weis 2003] [Sarma et al 2003]. Sua utilização em um protocolo de comunicação seguro para os sistemas RFID será mencionado no próximo Capítulo.

As funções *hash* também denominadas resumo de mensagens ("*message digest*") são funções de transformações que recebem uma entrada de tamanho variável e produzem uma saída de tamanho fixo, independente do tamanho da entrada. Quando usadas para fins criptográficos, possuem outras propriedades adicionais como a de ser chamada de "*one-way*", ou seja, é computacionalmente inviável encontrar a mensagem original a partir da saída *hash* gerada. Essas características fazem as funções *hash* funcionarem como "impressões digitais" de documentos eletrônicos. Exemplos muito comuns desses tipos de funções são a SHA-1 e MD5 [Menezes et al 2001].

Propostas de controle de acesso para sistemas RFID segundo trabalhos de [Weis 2003] utilizam funções *hash*, justificando assim a sua importância também nesses

sistemas.

O controle de acesso, ou seja, a forma como um leitor acessa a tag é um dos alvos dos estudos no trabalho de [Weis 2003]. A utilização de controle de acesso evita que leitores não legítimos consultem o conteúdo das tags. São descritas duas propostas de controle de acesso chamadas de travas por *hash* ou "*hash lock*". A utilização de criptografia para segurança de sistemas será mostrada posteriormente.

O sistema de controle de acesso através da trava *Hash* (*Hash Lock*), é um mecanismo simples baseado nas funções *hash*. As tags precisam basicamente de uma função *hash* otimizada em hardware. O funcionamento é descrito a seguir:

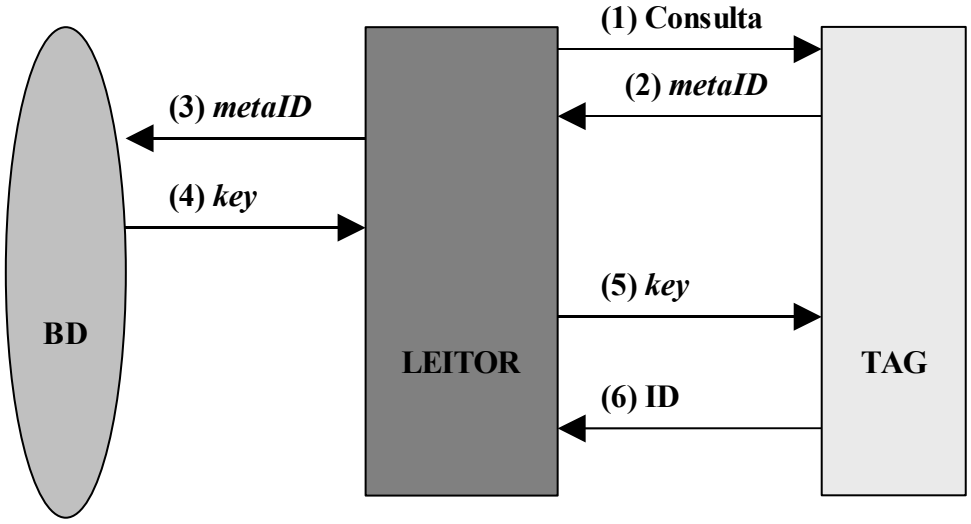
TRAVAMENTO DA TAG
1. O leitor R seleciona uma chave aleatória *key* e calcula *metaID = hash(key).*
2. O leitor R escreve a *metaID* na tag.
3. A tag entra no estado "travado".
4. O leitor R armazena no banco de dados o par (*metaID, key*) localmente.

A tag para ser consultada deve ser destravada. Apenas leitores legítimos devem ser capazes de realizar tal operação. O mecanismo para destravar é explicitado a seguir:

DESTRAVAMENTO DA TAG
1. O leitor R solicita a *metaID* da tag.
2. O leitor R procura o par (*metaID, key*) localmente.
3. O leitor R envia a chave *key* para a tag.
4. A tag confere se (*hash(key)=metaID*) e destrava caso positivo.

Dessa forma, somente leitores legítimos podem destravar as tags para realizar consultas, pois leitores ilegítimos não terão o par (*metaID, key*) evitando assim acessos não autorizados.

No entanto, como a *metaID* atua como um identificador, é possível o rastreamento da tag nesse esquema. A seguir é mostrada uma alternativa a trava por *hash* que evita o rastreamento das tags. A Figura 13 [Weis et al 2003b] ilustra o algoritmo.



(1) Consulta: O Leitor solicita através de um comando requisição a *metaID* da tag.
(2) *metaID*: A tag responde enviando a *metaID* ao Leitor.
(3) *metaID*: O Leitor envia a *metaID* ao BD e solicita a chave relacionada a esta *metaID*.
(4) *key*: O BD devolve a chave para o Leitor.
(5) *key*: O Leitor envia a chave para a tag.
(6) ID: E finalmente a tag devolve seu ID ao Leitor.

Figura 13. Trava por *Hash* (*Hash Lock*).

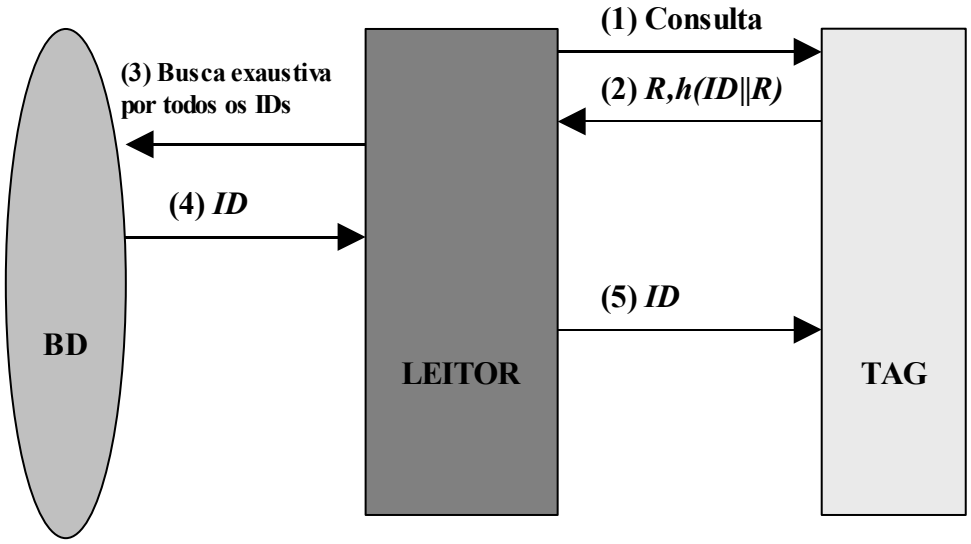
Para eliminar o problema do rastreamento da tag (devido à utilização do identificador *metaID*) [Weis 2003], também propôs uma variação do esquema de trava por *hash* chamada de trava por *hash* aleatório (*Randomized Hash Lock*). Trata-se também de um mecanismo simples que necessita de uma função *hash* otimizada em hardware e também de uma rotina para geração de números aleatórios. O funcionamento é descrito a seguir:

TRAVAMENTO DA TAG
1. O leitor envia uma instrução simples de travamento para a tag.

O travamento da tag é bastante simples, nenhum dado precisa ser enviado à tag. Apenas um comando de travamento deve ser enviado. Por sua vez, o destravamento da tag segue os passos mostrados a seguir:

DESTRAVAMENTO DA TAG
1. A tag responde a consulta do leitor enviando o par (R,h(ID||R)), onde R é um número aleatório, h(ID||R) é o hash gerado em cima do ID da tag concatenado com o número R. 2. O leitor faz uma busca exaustiva no BD aplicando h(ID||R) em todos os IDs conhecidos. 3. Encontrada a ID o leitor envia a ID para a tag que destrava-se.

Como não existe a presença de um identificador fixo, como no caso anterior, temos que o rastreamento da tag não pode ser realizado, o que é desejável em muitas aplicações. A Figura 14 ilustra o algoritmo da trava por *hash* aleatório.



(1) Consulta: O leitor envia uma consulta para a tag.
(2) *R,h(ID||R)*: A tag responde enviando o par R (número aleatório) mais o resultado do *hash* na *string* ID concatenada com R ao Leitor.
(3) *Busca*: O Leitor faz uma busca exaustiva pela ID da tag
(4) *ID*: O BD devolve o ID da tag para o Leitor.
(5) *ID*: O Leitor envia o ID para a tag que destrava-se.

Figura 14. Trava por *hash* aleatório.

2.4. AUTENTICAÇÃO

Autenticação é técnica através da qual um processo confirma que seu parceiro na comunicação é quem deve ser e não um impostor [Tanenbaum 2003].

Autenticação é a certificação da identidade de uma entidade para o outro lado do canal de comunicação, ou seja, é a garantia de que você é quem está dizendo ser. Vários métodos de autenticação forte[8] existem na atualidade. A principal diferença consiste em que tipo de criptografia será usada, se por chave pública ou por chave secreta. Na criptografia de chave secreta o problema da troca de chaves deve ser resolvido. Na criptografia por chave pública não temos este problema. Os dois principais tipos de protocolos de autenticação são chamados de protocolos com conhecimento zero (*zero-knowledge protocols)* e protocolos desafio-resposta (*challenge-responce protocols)* [Menezes et al 2001] [Tanenbaum 2003] [Stallings 2002]. O protocolo desafio-resposta será utilizado na proposta deste trabalho no capítulo 3 e será descrito a seguir.

Nos protocolos desafio-resposta, o verificador envia uma requisição desafio para o solicitante. Esse desafio pode ser um número escolhido aleatoriamente que varia de uma requisição para outra. O solicitante prova sua identidade através da manipulação do desafio, fazendo uso da chave secreta associada a ele. Assim, é muito importante que o segredo não seja mostrado ao verificador durante a comunicação. Após receber a resposta do solicitante, o verificador valida a resposta e pode ter certeza de que o solicitante conhece o segredo. Quando um atacante observa a comunicação entre o verificador e o solicitante, ele não pode ter acesso a nenhuma informação para uma identificação subseqüente, porque o próximo desafio será diferente.

Quando usamos criptografia por chave pública, o verificador envia um desafio para o solicitante. O número é encriptado pelo solicitante utilizando sua chave secreta e enviado de volta ao verificador que desencripta utilizando a chave pública do solicitante. Quando os resultados são iguais, o verificador reconhece a identidade do solicitante. A vantagem desse método é que o solicitante não precisa saber a chave secreta de ninguém mais. Esse mesmo protocolo pode ser implementado utilizando-se criptografia simétrica. Ao

8 Aquela que é computacionalmente inviável de ser quebrada com os recursos computacionais atuais [Menezes et al 2001].

invés de desencriptar a resposta do solicitante utilizando-se de sua chave pública, o verificador utiliza a chave secreta para desencriptar o resultado. A desvantagem deste método é o problema da troca de chaves. No entanto algoritmos de chave secreta funcionam bem mais rápido que métodos com chave pública.

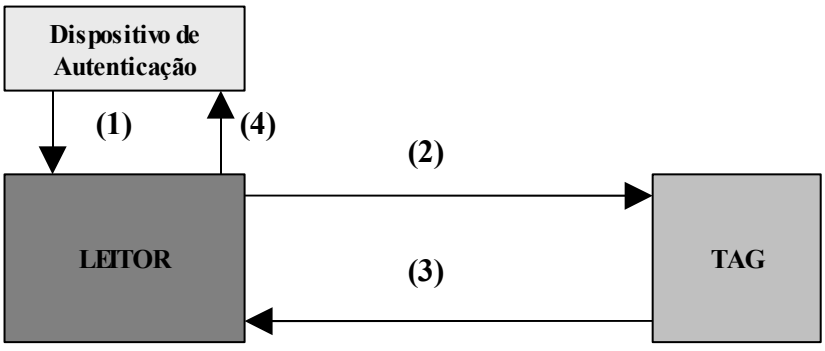
O protocolo onde uma entidade A é autenticada para uma entidade B é chamado de autenticação unilateral. Protocolos desafio-resposta "*one-way*" e "*two-way*" são utilizados neste tipo de autenticação. Quando utilizamos protocolos "*one-way*", um mecanismo de *timestamp* é necessário. O verificador A envia o *timestamp* encriptado t_a para o solicitante B que desencripta e verifica se o *timestamp* é aceitável. O protocolo "*two-way*" funciona com a utilização de números aleatórios. O solicitante B deve primeiro enviar um número aleatório r_b para o verificador A que encripta e envia de volta. A verificação funciona através da desencriptação da resposta e comparando com o número aleatório enviado.

Se ambas as entidades desejarem se autenticar uma para a outra (autenticação mútua), um protocolo "*two-way*" ou "*three-way*" é utilizado. A entidade A deve encriptar o *timestamp* t_a e aleatoriamente escolher um número r_a e enviar para a segunda parte B. Desencriptando e verificando o *timestamp* autentica a parte A para a entidade B. Depois disso, o número r_a é encriptado e enviado de volta a entidade A que desencripta e compara com o número gerado inicialmente. O mecanismo "*three-way*" trabalha similarmente, mas uma transmissão adicional tem que ser realizada. A entidade A escolhe um número aleatório r_a e envia para B. Este número r_a e outro número aleatório r_b são encriptados por B. O valor encriptado é devolvido para A que verifica seu próprio número aleatório e encripta o número de B. Esse resultado é enviado para B que finalmente encripta seu número e compara com o número escolhido r_b.

[Tanenbaum 2003] lista quatro regras que podem ajudar no projeto de protocolos de autenticação seguros e resistentes a diversos tipos de ataques. A primeira delas é fazer com que o transmissor prove quem é antes de o receptor responder. A utilização de duas chaves compartilhadas, ou seja, uma para cada autenticação também é aconselhada. A terceira regra é a utilização de desafios extraídos de conjuntos distintos, por exemplo, números pares e impares. Por último o autor sugere tornar o protocolo resistente a ataques que envolvam uma segunda seção. A sua aplicação previne a maioria dos problemas de segurança

de protocolos de autenticação.

[Engberg et al 2004] propôs uma solução para o problema da autenticação em RFID, baseado em outro esquema citado anteriormente, ou seja, o "*Zero-Knowledge protocol*". A proposta foi baseada na utilização de outro componente no sistema, chamado de "*Authentication Device*", ou dispositivo de autenticação. Nessa proposta, a requisição de autenticação não é gerada pelo leitor e sim pelo usuário que está utilizando o dispositivo de autenticação. A requisição é então direcionada ao leitor que se comunica com a tag. Depois que o leitor é adequadamente autenticado, a tag envia a resposta ao leitor da forma usual. A partir desse ponto, o usuário pode realizar as operações desejadas com a tag. Na prática, o dispositivo de autenticação e o leitor podem estar unidos em uma única entidade, como um PDA, por exemplo. Outra característica da proposta é que o código EPC não precisa estar armazenado na tag, pois outro identificador intermediário é armazenado a partir do momento em que a tag passa a responsabilidade do usuário. Dessa forma, esse identificador intermediário é traduzido ao código EPC do produto apenas quando chega ao dispositivo de autenticação, evitando que o código EPC seja revelado de alguma outra maneira. O protocolo proposto por [Engberg et al 2004], está ilustrado na Figura 15.



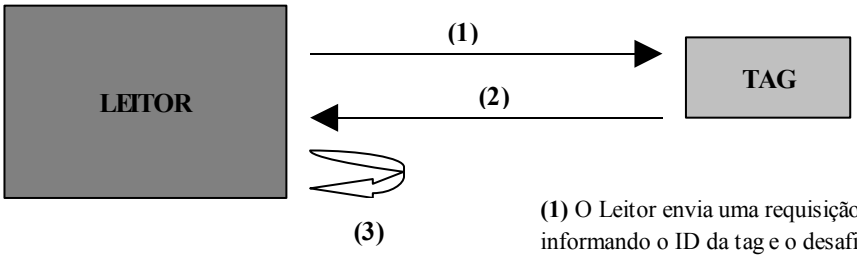
(1) O Dispositivo de autenticação que é controlado pelo usuário envia requisição de autenticação para o Leitor.
(2) O Leitor envia requisição para a tag.
(3) A tag processa a requisição e autentica o Leitor baseado no protocolo de autenticação "Zero-Knowledge" e informa ao Leitor.
(4) O Leitor informa ao dispositivo que a autenticação foi bem sucedida.

Figura 15. Autenticação para RFID utilizando "*Zero-Knowledge Protocol*".

Em todo o processo de autenticação envolvido, apenas a utilização de uma função *hash* aumenta o custo de fabricação da tag. O custo do sistema como um todo é aumentado também pela necessidade de utilização do dispositivo de autenticação. A proposta, assim, explora o problema da autenticação de leitores e da privacidade dos consumidores, impedindo que leitores não autorizados consultem e tenham acesso às informações das tags. Outro aspecto importante é que o autor deixa claro que a proposta não se aplica às tags somente-leitura, que são as de custo mais baixo, devido à necessidade da gravação de um identificador intermediário ou chave secreta utilizada no processo de autenticação do protocolo. Formatos de mensagens de autenticação e resposta também são mostrados no trabalho de [Engberg et al 2004]. O principal ponto fraco da proposta é a utilização de um dispositivo à parte fazendo assim com que as aplicações necessitem de uma reorganização de infra-estrutura. Outro ponto fraco é que o mecanismo de autenticação não menciona nenhum tipo de extensão ao protocolo de comunicação ISO/IEC 18000-3 ou mesmo o padrão EPCGlobal, ou seja, o padrão atual da indústria não é levado em consideração.

Em [Aigner e Feldhofer 2005], o autor explica como os protocolos de autenticação funcionam e como eles podem ser incluídos no padrão ISO/IEC 18000. A proposta baseia-se na utilização de criptografia simétrica para autenticação e em como a solução pode beneficiar o sistema RFID se incluído diretamente no padrão existente. O autor sugere 3 esquemas para autenticação: A autenticação da tag, a autenticação do leitor e a autenticação mútua.

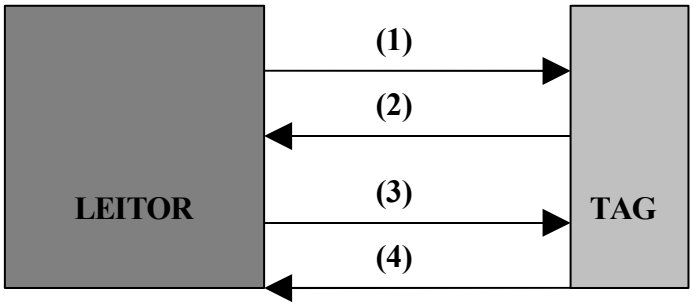
A autenticação da tag é realizada através de uma simples troca de dados, como ilustrado pela Figura 16.



(1) O Leitor envia uma requisição de autenticação, informando o ID da tag e o desafio.
(2) A tag encripta o desafio com a chave secreta e envia o resultado de volta ao Leitor.
(3) O Leitor confere o resultado e autentica a tag.

Figura 16. Autenticação da tag .

A autenticação do leitor é também proposta, com um mecanismo similar à autenticação da tag. Invasores podem ter acesso à ID da tag através da escuta do canal de comunicação, porém estes não podem iniciar uma comunicação. O autor cita também a importância da análise de ataques do tipo "seqüestro de uma comunicação autorizada" em um cenário de aplicação real, mas não indica nenhum tipo de solução. O processo é ilustrado na Figura 17.



(1)O Leitor faz a consulta à tag.
(2) A tag responde informando que o Leitor precisa autenticar-se com um desafio.

(3) O Leitor envia o desafio encriptado para a tag.
(4) A tag confere o resultado e envia sua ID para o Leitor.

Figura 17. Autenticação do leitor.

Ambas os esquemas de autenticação da tag e leitor são propostos com o objetivo de se propor extensões para o protocolo de comunicação ISO/IEC 18000. O autor

fala da necessidade de um módulo de criptografia utilizando uma implementação simplificada do algoritmo de criptografia simétrica AES. O projeto também requer a necessidade de implementação de uma função geradora de números aleatórios. Como objetivos finais do projeto o autor lista os seguintes, mostrados na Tabela 9 adiante.

Tabela 9. Objetivos da proposta de segurança.

OBJETIVO	DESCRIÇÃO
1	Estender o protocolo de comunicação da tecnologia RFID através de rotinas de segurança.
2	Desenvolvimento e implementação de novas tags com criptografia forte[9] assim como protótipos de tags e leitores.
3	Melhorar o alcance de leitores.
4	Investigar os campos de aplicações em potencial.
5	Pesquisar o campo de segurança na área de etiquetas inteligentes.

O trabalho apresenta também a criação de um emulador do sistema para simular os diversos contextos de aplicações que podem se beneficiar da proposta de segurança. Todos os aspectos relevantes como consumo de energia, quantidade de *gates*, implementação em hardware entre outros aspectos são detalhados na proposta.

Como conclusões do trabalho tem-se que a disponibilização de segurança para sistemas RFID através da utilização de autenticação com criptografia utilizando o algoritmo AES [Tanenbaum 2003] é aplicável para a atual tecnologia de RFID sem que os custos tenham aumento significativo.

[Yang et al 2005] propõe em sua pesquisa um mecanismo de autenticação mútua entre tag, leitor e banco de dados para sistemas RFID de baixo custo. O esquema oferece proteções contra violação de privacidade dos usuários do sistema. Inicialmente, o protocolo dispõe da autenticação do leitor prevenindo contra ataques passivos. Considera o canal de comunicação entre leitor e bancos de dados inseguro, ou seja, três entidades estão envolvidas no protocolo. Além do aspecto privacidade, problemas de falsificação de tags também são prevenidos. Para o funcionamento do protocolo proposto, as tags devem dispor de uma função *hash* e operações de ou-exclusivo assim como uma função geradora de

9 [Aigner e Feldhofer 2005] define criptografia forte como sendo aquela que é computacionalmente inviável de ser quebrada com os recursos computacionais atuais.

números aleatórios. A proposta é validada através da utilização de lógica GNY para provar que o protocolo atende aos objetivos propostos. Embora a avaliação geral da pesquisa seja satisfatória, o autor peca no quesito da função *hash*, ou seja, não é especificada qual função deve ser utilizada. Este fator é importante pois a função *hash* é a principal causa do aumento de custo do esquema proposto por [Yang et al 2005]. Unido a este ponto fraco tem-se também a não preocupação com os padrões existentes como o [ISO 2004] e o [EPCGlobal 2005] deixando o trabalho um tanto quanto impraticável, devido à grande importância da padronização em se tratando de sistemas RFID.

[Dimitrou 2005] apresenta um protocolo de autenticação para RFID focando a privacidade do usuário e a proteção contra a clonagem de tags. O mecanismo permite a autenticação mútua entre tag e leitor baseada no método desafio-resposta onde o envio da ID da tag é realizado simultaneamente com o processo de autenticação . É baseado na utilização de um segredo compartilhado entre a tag e o banco de dados, segredo este que é atualizado frequentemente para evitar o rastreamento das tags. Sua simplicidade é a grande vantagem, porém a utilização de uma função *hash* pode elevar o aspecto custo da solução. Assim como [Yang et al 2005] não houve preocupação com o aspecto custo, fator tão importante e tão discutido ao longo desta dissertação. A padronização também não esteve entre os objetivos do autor. O protocolo especificado altera um pouco a forma de comunicação atual das entidades tag e leitor, ou seja, muda o conceito de requisição-resposta devido o envio da ID da tag estar incorporado ao processo de autenticação. Apesar dos pontos negativos citados anteriormente, o protocolo proposto é eficiente, pois realmente oferece proteções importantes aos principais problemas que envolvem a utilização da tecnologia RFID como falsificação de leitores e tags, espionagem, privacidade do usuário e rastreamento indevido de tags.

[Tsudik 2006] descreve em seu trabalho uma técnica simples para proteção da privacidade do usuário, ou seja, tornar as tags não rastreáveis. O autor considera uma tag "não rastreável" como aquela em que é computacionalmente difícil de se obter informações sobre a identidade da tag. O protocolo proposto é chamado de YA-TRAP, envolve um número mínimo de interações entre tag e leitor e exige poucos recursos extras a serem disponibilizados pela tag, que é a implementação de uma função *hash* e uma função geradora de números aleatórios. O esquema também requer pouca carga computacional por parte da

entidade banco de dados. O autor assume que o canal entre leitor e banco de dados é seguro e que este apenas se comunica com leitores legítimos. Em resumo, a protocolo como um todo lida com o problema da privacidade do usuário, ou seja, exige que apenas leitores legítimos ao sistema tenham acesso ao conteúdo das tags, realizando assim a autenticação do leitor. Assim como [Yang et al 2005], [Tsudik 2006] não descreve qual função *hash* utilizar nem justifica o fato da solução ser considerada de baixo custo. Trata-se de uma solução de bom desempenho e que atinge o objetivo de proteger a privacidade, no entanto falha no aspecto comentado anteriormente e também nos aspectos já abordados, pois em nenhum momento o autor mostrou-se preocupado em padronizar o protocolo.

[Duc et al 2006] expõe em sua pesquisa a proposta de um protocolo de comunicação para dispositivos RFID baseado no padrão para tags Classe 1-Gen 2 [EPCGlobal 2005b], ou seja, para os tipos de tags que serão lançadas ainda no ano de 2006 segundo previsão da EPCGlobal [Gutierrez et al 2005]. O protocolo proposto é seguro no aspecto em que protege o sistema contra falsificação de leitores e tags como também oferece privacidade aos usuários. O trabalho de [Duc et al 2006] está um passo na frente dos demais [Yang et al 2005], [DIM 2005], [Tsudik 2006], [Lee et al 2006] pois preocupou-se com o fator padronização e baseou-se no protocolo largamente utilizado que é o da EPCGlobal. Embora ainda não existam as tags definidas no protocolo [EPCGlobal 2005b], os resultados da pesquisa são de grande importância. Apesar da preocupação em propor alterações para um protocolo existente, [Duc et al 2006] não especifica formalmente nenhuma extensão, ou seja, o trabalho restringe-se à exposição de um protocolo em alto nível onde não são especificados detalhes como formato de quadros e campos, aspectos também importantes para tornar mais completo o trabalho. O autor não se preocupa com o fator custo da solução, mas apresenta análise de complexidade do mecanismo como um todo, porém sem mencionar os recursos necessários.

[Lee et al 2006] ilustra em sua pesquisa mais uma proposta de mecanismo de autenticação mútua para sistemas RFID. Seu objetivo é oferecer privacidade aos usuários do sistema e consequentemente proteção contra clonagem de tags e leitores. A proposta assim como [Yang et al 2005], [Dimitrou 2005] e [Tsudik 2006] utiliza uma função *hash* como operação criptográfica. Trata-se também de um mecanismo eficiente pois oferece proteção

contra os principais problemas de segurança relacionados a RFID. Possui como ponto negativo os mesmos citados para [Yang et al 2005], [Dimitrou 2005] e [Tsudik 2006] que é a falta de preocupação com os aspectos padronização e custos, pois também não especifica que função *hash* utilizar nem menciona a solução ser de baixo custo.

O Capítulo 3 adiante detalhará a especificação de segurança para sistemas RFID utilizando metodologia similar a [Aigner e Feldhofer 2005], [Yang et al 2005], [Dimitrou 2005], [Tsudik 2006] e [Lee et al 2006]. No entanto trata-se de uma proposta mais ampla que também propõe extensões ao protocolo existente EPCGlobal sem que os custos sejam significativamente elevados. A utilização de outro algoritmo de criptografia simétrica é detalhada e a justificativa de sua utilização é citada. O contexto de aplicações que podem ser beneficiadas também é abordado.

3. EXTENSÕES DE SEGURANÇA PARA O PROTOCOLO EPCGLOBAL PARA TAGS CLASSE 1

A pesquisa realizada foi dividida em dois estágios, sendo que o primeiro deles objetivou mostrar que o algoritmo TEA pode ser implementado com um pequeno número de recursos adicionais às tags de baixo custo, justificando assim sua utilização posterior no protocolo de autenticação. Este estágio envolveu a avaliação das variáveis relevantes do algoritmo TEA utilizando microcontroladores PIC. Em seguida foi especificado um protocolo de autenticação mútua para ser adicionado no protocolo de comunicação EPCGlobal. Esta etapa envolveu a especificação e análise formal do protocolo utilizando lógica BAN resultando na padronização com adequação ao padrão existente para tags Classe 1. O segundo estágio finaliza comparando a proposta descrita com os trabalhos mais recentes publicados na literatura especializada, sinalizando a contribuição gerada pelo presente trabalho.

3.1. AVALIAÇÃO DO ALGORITMO TEA

Os recursos de software utilizados na avaliação do TEA foram um sistema Microsoft Windows XP, um simulador de PIC disponível gratuitamente, o MPLAB 7.21 da *Microchip*[tm] [Microchip 2005], uma implementação modificada em *Assembly* do algoritmo TEA [Warren 1998] e o compilador da Microchip[tm], o MPASM versão 1.30. Adicionalmente foi utilizado um ambiente de programação para PICs código aberto (*OpenSource*) chamado Pikdev [Pikdev 2005] para análise da quantidade de memória de programa necessária para implementação do algoritmo. Não foram utilizados recursos de hardware, sendo estes substituídos pelo ambiente e simulador acima citado. O modelo de hardware utilizado foi o PIC16F84A conforme metodologia de Chammas [Chammas et al 2004] onde ele utiliza este

tipo de PIC para simular uma tag RFID. As características deste tipo de componente assemelham-se bastante às características exigidas pelos microchips acoplados às tags RFID.

A escolha do ambiente de desenvolvimento MPLAB 7.21 levou em consideração os aspectos de custo (gratuito), facilidade de uso e confiabilidade, por ser fornecido pelo próprio fabricante dos microcontroladores. Além destas características, o MPLAB possui um simulador embutido de uma série de modelos de PICs disponíveis e também um compilador de linguagem *Assembly*. Todos esses recursos supriram as necessidades em termos de software para o desenvolvimento da pesquisa.

O experimento levou em consideração tags RFID com código EPC [EPCGlobal 2004] de 64 bits, velocidade de processamento de 20Mhz e pelo menos 40 bytes de memória. Para aumentar ainda mais a velocidade de encriptação do algoritmo, o número de ciclos foi reduzido a 16 (normalmente é 32), por ainda ser considerado seguro segundo os criadores do TEA [Wheeler et al 1995]. Este número pode ser modificado facilmente no código fonte caso desejável. Com pequenas modificações também podemos aumentar o tamanho do código EPC da tag.

O modelo de microcontrolador PIC utilizado possui as características mostradas na Tabela 10.

Tabela 10. Características do microcontrolador utilizado.

Tipo de Memória	*Flash*
Tamanho da memória de programa	1792 bytes
RAM	68 bytes
EEPROM	64 bytes
Microcontrolador	8 bits
Velocidade Máxima	20Mhz

[Aigner e Feldhofer 2005] construiu um protótipo de seu projeto, denominado de ART, utilizando uma placa FPGA assim como [Buth et al 2002], com o uso do algoritmo AES. Foram realizadas medidas de consumo de energia e quantidade de ciclos de *clock* necessários bem como medidas da área do *chip*. A metodologia utilizada no presente trabalho também seguiu idéias utilizadas por [Buth et al 2002] e [Aigner e Feldhofer 2005] para obtenção de resultados. Após a decisão sobre os materiais utilizados, uma breve comparação

entre os recursos necessários e recursos disponíveis foi realizada.

As principais variáveis analisadas foram memória utilizada, quantidade de registradores e velocidade.

O algoritmo TEA modificado foi avaliado seguindo a implementação em pseudo-linguagem, conforme Tabela 11. Foram considerados "informações" quaisquer dados que a tag possa enviar para o leitor. Por exemplo, a ID da tag ou mesmo uma requisição de autenticação ao leitor. A Tabela 12 ilustra em pseudo-linguagem como foi avaliado a utilização do algoritmo caso a tag precise desencriptar dados ao invés de encriptar dados. Como exemplo dessa situação temos que o leitor pode estar enviando uma requisição de autenticação para a tag e dessa forma a tag precisa processar a rotina de desencriptação.

Tabela 11. Algoritmo para envio de informação.

Passo 1. Buscar ID da tag na EEPROM.	Considera-se como uma instrução trivial.
Passo 2. Encriptar ID através de 16 *rounds* do algoritmo TEA.	Instruções de encriptação, responsável pelo maior consumo de energia e utilização de memória.
Passo 3. Enviar informação encriptada.	Instrução que não faz parte da unidade controladora da tag, e sim da unidade *front-end* e está fora do escopo deste trabalho.

Tabela 12. Algoritmo para recepção de informação.

Passo 1. Receber dados encriptados	Instrução que não faz parte da unidade controladora da tag e sim da unidade *front-end*.
Passo 2. Desencriptar dados através de 16 *rounds* do algoritmo TEA.	Instruções de desencriptação, responsável pelo maior consumo de energia e utilização de memória.
Passo 3. Processar informação recebida.	Pode variar conforme o cenário de aplicação onde o sistema RFID está sendo utilizado.

3.2. RESULTADO DA AVALIAÇÃO

Os recursos oferecidos pelos PICs vão além dos necessários para tags de baixo

custo. A Tabela 13 ilustra os recursos necessários pelas tags para implementação do algoritmo TEA.

Tabela 13. Recursos necessários para implementação do algoritmo TEA.

EEPROM	A chave utilizada possui 128 bits, sendo necessário 16 bytes para armazenamento da chave. Adicionalmente é necessário gravar a ID da tag, de 64 bits, ou oito bytes. A área de armazenamento é de **24 bytes** (192 bits).
Memória de programa	**1442** bytes
Quantidade de instruções	**3** tipos de instruções simples e **11** instruções complexas. Em 16 *rounds* do algoritmo são executadas 3696 instruções na rotina de encriptação e 3746 instruções na rotina de desencriptação.
Quantidade de Registradores utilizados	**16** registradores para variáveis temporárias. Os demais dados são armazenados na EEPROM.
RAM	**37 bytes** segundo os próprios resultados de encriptação do MPLAB gerados em um arquivo com extensão "lst" e também segundo o próprio autor da implementação do algoritmo [Warren 1998].
Velocidade	Operando a 20Mhz, o tempo para encriptação e desencriptação dos dados foi incrivelmente rápido da ordem de dezenas de milisegundos, assim não produziu nenhum atraso na comunicação global do sistema.
Quantidade de gates para implementação em hardware [Buth et al 2002]	2560 portas lógicas

Como a quantidade de dados a serem encriptados é um número muito pequeno de bytes (oito bytes) observamos que a velocidade de processamento foi extremamente rápida, da ordem de milisegundos, ou seja, um aumento não significativo de atraso. A memória de EEPROM utilizada também apresenta bons resultados, 24 bytes, valor aceitável para tags RFID segundo [Finkenzeller 2003]. Este valor utiliza 44 bytes a menos que os 68 bytes

oferecidos pelos PIC's.

A quantidade de memória de programa utilizada foi de 1442 bytes conforme resultados da compilação do programa em linguagem *Assembly*, a partir do MPLAB. A quantidade de instruções de baixo nível necessária são apenas 14 instruções. As rotinas de encriptação e desencriptação executam cerca de 3700 instruções para realizar suas respectivas tarefas. É uma quantidade relativamente baixa quando comparada aos principais algoritmos para uso geral.

A memória de programa utilizada é indicada pelo tamanho do arquivo compilado *hex* e também pelo resultado da análise do visualizador de memória ocupada, mostrado a partir do ambiente PikDev.

A contagem da utilização de registradores e de memória RAM também mostrou resultados aceitáveis, 16 registradores e 37 bytes de RAM. A análise levou em consideração os parâmetros definidos no protocolo de descrição e comunicação das tags RFID [EPCGlobal 2005].

Resultados de [Buth et al 2002] foram analisados e uma variável adicionada na última linha da Tabela 13 por ser importante para as conclusões posteriores. A quantidade de *gates* necessários para implementação em hardware e o consumo de energia mostram que a utilização do TEA para dispositivos limitados como é o caso das tags RFID pode ser realizada com um aumento aceitável de custos. Sendo esses custos direcionados para a variável memória, principal responsável pelo aumento de recursos requeridos. Outra variável importante a ser considerada é o consumo de energia. [Buth et al 2002] mostra que o consumo de energia da implementação em software do algoritmo é cerca de 1090mW. Os equipamentos e a implementação utilizados são bem mais robustos que os microcontroladores PICs consumindo assim uma quantidade bem maior de energia. A partir deste número obtido em [Buth et al 2002], comparativamente obtivemos um consumo energético de cerca de 5mW (considerando as cerca de 3700 instruções de encriptação/desencriptação descritas na Tabela 13) utilizando PICs. Esta quantidade pode ser disponibilizada até mesmo por tags que são energizadas a partir do leitor, ou seja, tags que não possuem bateria própria [Finkenzeller 2003].

3.3. ESPECIFICAÇÃO DO PROTOCOLO DE AUTENTICAÇÃO E SEGURANÇA PROPOSTO

A metodologia utilizada para elaboração do protocolo de autenticação segue idéias dos projetos descritos por [Aigner e Feldhofer 2005], [Tsudik 2006], [Yang et al 2005] e [Dimitrou 2005] onde os autores propõem e especificam um protocolo de autenticação para sistemas RFID e validam os mesmos.

No primeiro projeto, o protocolo é validado através de simulações feitas por um software desenvolvido pelos próprios autores. No projeto de Tsudik são utilizados métodos formais para validação. Já Yang e Dimitrou validam seus protocolos através da análise minuciosa dos mesmos. Dessa forma a metodologia utilizada nesta dissertação procura unir as idéias dos projetos citados.

O protocolo proposto neste trabalho é validado através de um método formal [Santos et al 2002] utilizando a lógica BAN[10] amplamente utilizada para validação de protocolos de autenticação. A lógica BAN será descrita posteriormente, bem como a análise detalhada do protocolo de comunicação e dos problemas de segurança que afetam o sistema. O algoritmo TEA [Wheeler et al 1995] foi o escolhido para realizar as operações criptográficas envolvidas no processo de autenticação. Na Tabela 14 são detalhados todos os passos executados para elaboração e avaliação da pesquisa.

O primeiro passo para a elaboração e especificação das extensões de autenticação para o protocolo EPCGlobal foi a escolha de um mecanismo de autenticação. O mecanismo escolhido foi o conhecido como desafio-resposta ou "*challenge-response*" por melhor se encaixar na forma de comunicação entre as entidades do sistema, ou seja, a requisição-resposta.

A especificação formal do protocolo segue a chamada "notação padrão" descrita em [Carlsen 1994][Junior 1999]. A descrição do mesmo por meio de uma figura, assim como a forma de discussão dos resultados seguem as notações de [Tsudik 2006], [Yang et al 2005], [Dimitrou 2005], [Duc et al 2006] e [Lee et al 2006].

10 A lógica BAN [Burrows et al 1990], ou lógica Burrows-Abadi-Needham é um método formal utilizado na validação de protocolos de autenticação.

Tabela 14. Metodologia utilizada na pesquisa.

Passo	Descrição
Passo 1 – Análise do protocolo e especificação de um novo protocolo com extensões de segurança.	Desenvolvimento, especificação e descrição formal de um protocolo que disponibilize autenticação e criptografia para o sistema.
Passo 2 – Padronização do protocolo proposto adequando-se ao padrão EPCGlobal para tags Classe 1.	Através da análise dos comandos já existentes e do formato dos quadros utilizados na comunicação, foram especificados novos comandos baseados nos comandos existentes para a disponibilização de segurança no protocolo.
Passo 3 - Avaliação do protocolo com as extensões de segurança e prova formal que seus objetivos são alcançados.	Foram avaliadas todas as extensões criadas no aspecto segurança, detalhando quais tipos de problemas podem ser resolvidos e/ou minimizados com a utilização no mundo real. A cada passo do protocolo, foi justificado como determinado ataque é protegido. Utilizando um método formal para validação de protocolos de autenticação (lógica BAN) a proposta é validada, ou seja, é provado que os objetivos propostos foram alcançados.

3.3.1. CONSIDERAÇÕES INICIAIS

Inicialmente, a Tabela 15 apresenta as notações utilizadas na descrição do protocolo, que serão utilizadas até o final da dissertação.

As tags são do tipo EPCGlobal Classe 1 ou seja, possuem memória gravada uma única vez. Esta memória deve vir de fábrica com seu código EPC e uma chave de criptografia de 128bits do algoritmo TEA. Devem implementar funções de encriptação e desencriptação assim como uma função geradora de números aleatórios. Seguindo orientações descritas por [Tanenbaum 2003 pp. 838] estabelece-se que leitores apenas utilizarão números aleatórios impares e tags números aleatórios pares, para evitar ataques do tipo reflexão. Considera-se também que as tags possuirão energia suficiente para realizar as operações

necessárias e que as tags possuem capacidade de guardar informação de estado conforme especificado em protocolo [EPCGlobal 2005][EPCGlobal 2005b].

Tabela 15. Notação utilizada.

Símbolo/Notação	Descrição
T	Tag RFID.
L	Leitor RFID.
k	Chave de criptografia compartilhada por T e L.
S	Solicitação de autenticação.
RT	Número aleatório gerado pela tag.
RL	Número aleatório gerado pelo leitor.
ID	Código EPC da tag.
$Enc_k()$	Função de encriptação utilizando o algoritmo TEA com chave *k*.
$Denc_k()$	Função de desencriptação utilizando o algoritmo TEA com chave *k*.

Conforme mostrado na avaliação do TEA, este algoritmo pode ser disponibilizado nas tags de baixo custo com segurança adequada.

Os demais dados relacionados às tags (características particulares por exemplo) são armazenados em um banco de dados externo.

Como o canal de comunicação entre leitor e banco de dados provavelmente é inseguro deve-se também utilizar criptografia entre ambos. Outro aspecto a ser considerado é que leitor e banco de dados devem possuir capacidade energética e computacional adequada para executar operações criptográficas.

No protocolo proposto não são considerados ataques do tipo físico pois estes devem ser evitados através de sistemas de segurança tradicionais como o monitoramento de vídeo por exemplo. Os ataques *man-in-the-middle*, ataques do tipo *replay*, espionagem, falsificação de tags e leitores e ataques de reflexão foram considerados e previnidos pelo

protocolo proposto.

3.3.2. CONFIGURAÇÕES INICIAIS

Considera-se que cada tag possui armazenada em sua memória interna o seu código EPC e a sua chave de criptografia, bem como possui implementadas as funções de encriptar/desencriptar e de geração de números aleatórios. Vale observar que em tags Gen-2, a função geradora de números aleatórios já é padronizada. O leitor deve possuir estas mesmas funções. O sistema RFID que utiliza o protocolo proposto pode realizar quatro tipos diferentes de comunicações mostradas na Tabela 16.

Tabela 16. Tipos possíveis de comunicações.

Tipo de comunicação	Número de mensagens	Descrição e exemplo
Autenticação da tag	4 mensagens: - 2 de autenticação - 2 de requisição/resposta.	Consiste na comunicação onde apenas existe a necessidade de autenticação da tag, ou seja, considera-se que todos os leitores são legítimos. É um típico caso de aplicação em grandes redes de supermercados, ou seja, os leitores estão localizados nos caixas e estes são garantidos serem legítimos.
Autenticação do leitor	5 mensagens: - 3 de autenticação - 2 de requisição/resposta.	Consiste na comunicação onde apenas existe a necessidade de autenticação do leitor, ou seja, considera-se que todas as tags são legítimas. É um típico caso de aplicação em transporte de cargas, ou seja, evita que terceiros tenham acesso ao valor carregado por um caminhão por exemplo.
Autenticação mútua	7 mensagens: - 5 de autenticação - 2 de requisição/resposta.	Comunicação onde ambos são autenticados. É o tipo mais seguro de comunicação. Pode e deve ser aplicado na maioria das aplicações onde os custos gerados não sejam significativos.
Sem autenticação	2 mensagens: - 2 de requisição/resposta.	Comunicação definida pelo atual padrão EPCGlobal, ou seja, nenhuma entidade é autenticada.

A Figura 18 ilustra o protocolo com utilização de autenticação mútua, já que este envolve as comunicações com autenticação da tag e do leitor, não sendo necessário assim ilustrar os três diferentes tipos de comunicação.

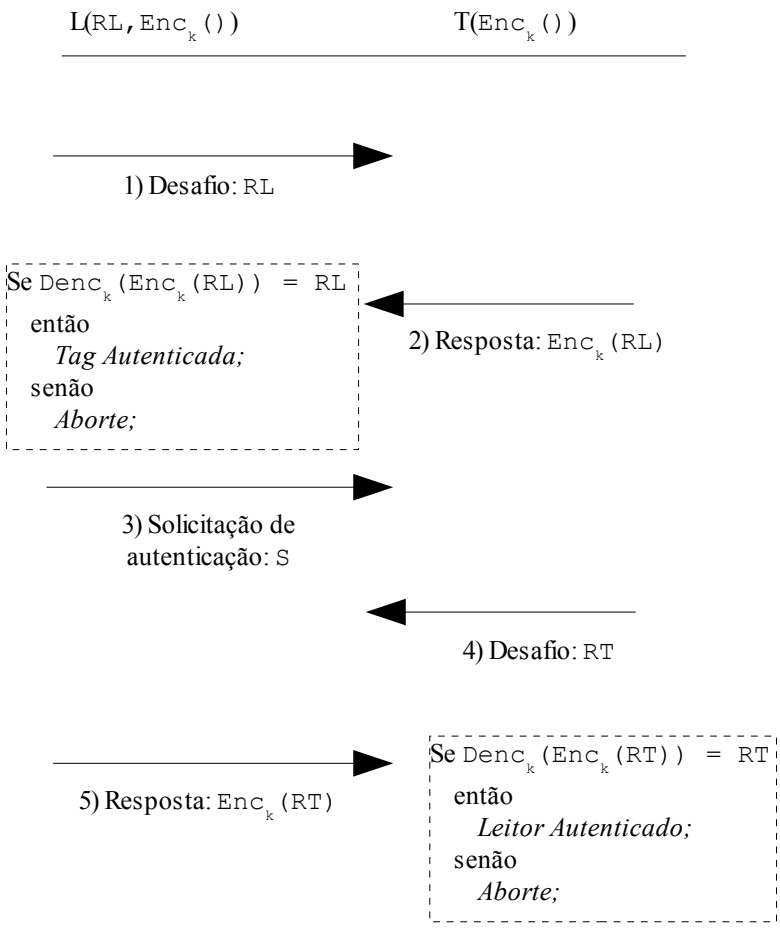


Figura 18. Comunicação com autenticação mútua.

PASSO 1 (DESAFIO de L): Neste passo, o leitor aplica um protocolo seguro para singularização da tag. O leitor gera um número aleatório RL (impar) e envia para a tag com o qual está se comunicando, sendo que este número é diferente a cada leitura, evitando assim ataques do tipo *replay*. Ataques de reflexão são prevenidos através da utilização de números aleatórios de conjuntos diferentes, neste caso do leitor, um número ímpar.

PASSO 2 (RESPOSTA de T): A tag recebe RL de L, aplica a função $Enc_k(RL)$ e devolve este valor para o leitor como resposta. Como apenas L e R conhecem k então apenas o leitor estará apto a desencriptar o valor enviado por T. Assim, temos que a tag está devidamente autenticada a partir deste ponto, evitando assim o problema da falsificação de tags e também prevenindo a violação de privacidade ou ataque de rastreamento/localização.

PASSO 3 (SOLICITAÇÃO DE AUTENTICAÇÃO de L): Caso a comparação $Denc_k(Enc_k(RL)) = RL$ seja verdadeira, L considera T autêntica. Nesse ponto L solicita sua própria autenticação.

PASSO 4 (DESAFIO de T): A tag gera um número aleatório RT (par) e envia para L. A cada solicitação T gera um RT diferente para evitar ataques do tipo *replay*. Ataques de reflexão são prevenidos através da utilização de números aleatórios de conjuntos diferentes, neste caso do leitor, um número par.

PASSO 5 (RESPOSTA de L): O leitor recebe RT de T e aplica a função $Enc_k(RT)$. O resultado é enviado para T, finalizando o protocolo de autenticação mostrado na Figura 18. T e L estão devidamente autenticados, o que previne qualquer ataque de falsificação, ou seja, apenas leitores e tags autorizadas estarão se comunicando.

Os passos 6 e 7 a seguir não fazem parte do protocolo de autenticação, e sim da comunicação normal do sistema, ou seja, requisição-resposta. A diferença é que ambas as partes a partir desse ponto já encontram-se autenticadas. Para evitar ataques do tipo *man-in-the-middle* e/ou espionagem L deve solicitar a ID encriptada de T.

PASSO 6 (REQUISIÇÃO de L): Caso o L tenha sido devidamente autenticado, este já pode se comunicar normalmente com T. A solicitação pode ser a requisição da ID ou a requisição da ID encriptada, ou seja, $Enc_k(ID)$.

PASSO 7 (RESPOSTA de T): T responde a solicitação de L.

3.4. ESPECIFICAÇÃO FORMAL DO PROTOCOLO

Os métodos formais comprometem-se a auxiliar o desenvolvedor no planejamento e na análise dos protocolos criptográficos, principalmente os de autenticação e distribuição de chaves [Santos et al 2002]. O planejamento informal dos protocolos criptográficos está sujeito a erros, evidenciando a importância do uso de métodos formais para o planejamento de tais protocolos.

Os métodos formais permitem fazer uma análise completa, considerando os diferentes modos de ataque, para verificar se os objetivos propostos pelo protocolo foram alcançados [Santos et al 2002]. Os métodos formais podem ser baseados em linguagem de verificação, sistemas especialistas, lógicas modais e sistemas algébricos.

Os métodos de análise formal baseados em lógicas modais são os que apresentam os melhores resultados em relação aos processos de autenticação, destacando-se os métodos baseados em lógica BAN pelo baixo nível de complexidade. Ela foi publicada em 1989 por Michael Burrows, Martin Abadi e Roger Needham [Burrows et al 1990], e foi a primeira sugestão para formalizar a descrição e análise de protocolos de autenticação.

A lógica BAN tem como objetivo principal a descrição dos objetivos do protocolo, possibilitando determinar se a quantidade de suposições é suficiente em relação a outros protocolos e se existem passos desnecessários na execução do protocolo.

A execução da lógica implica na segmentação da análise em 3 etapas distintas. A primeira etapa consiste em idealizar o protocolo (*Protocol Idealization*). A segunda etapa é o levantamento das suposições de crença (*assertions*) e os objetivos com afirmações numa notação simbólica. A terceira e última etapa transforma os passos do protocolo numa notação simbólica e aplica as regras dos postulados (*inference rules*) para atingir os objetivos do protocolo, denominada esta etapa como a de análise do protocolo (*Protocol Analysis*). A lógica divide o tempo em duas épocas: passado e presente. Considera-se presente o tempo durante a atual execução do protocolo, sendo que qualquer mensagem enviada antes desse tempo é considerada passado, devendo ser rejeitada pelo protocolo [Santos et al 2002]. A

Tabela 17 descreve a notação utilizada pela lógica BAN.

Tabela 17. Notação básica da lógica BAN.

Representação	Significado
$P \mid\equiv X$	P *acredita em* X, ou seja, para P a mensagem contendo X é verdadeira.
$P \lhd X$	P *recebe* X, ou seja, P recebeu uma mensagem contendo X.
$P \mid\sim X$	P *disse* X, ou seja, P enviou uma mensagem contendo X.
$P \Rightarrow X$	P *tem jurisdição sobre* X, ou seja, P é responsável por X.
$\#(X)$	*Novo* X, ou seja, X é novo e não foi utilizado antes em nenhuma sessão por nenhum participante.
$P \leftrightarrow^{k} Q$	P e Q *compartilham* a chave criptográfica k, ou seja, P e Q podem utilizar k para se comunicarem com privacidade.
$\{X\}_k$	Fórmula X *cifrada* com a chave k.

Na lógica são distinguidos os seguintes objetos representados na Tabela 17: Principal ou Participantes (representados pela letra P e Q), Chaves de Ciframento (representada por k) e Fórmulas Lógicas (representadas por X). Os símbolos P e Q utilizados na Tabela 17 representam os participantes genéricos; X mensagens e declarações; e K a chave de ciframento / deciframento.

Os postulados da lógica BAN são descritos por um conjunto de regras as quais determinam a diversas operações lógicas [Burrows et al 1990]. Existem várias regras, porém as principais utilizadas na análise dos protocolos criptográficos são oito, sendo utilizadas no caso específico do protocolo proposto nesta dissertação apenas as três apresentadas a seguir.

Regra 1 (R1) – Regra de Significado da Mensagem (*Message Meaning Rule*) - Esta regra verifica o significado da mensagem.

$$\frac{P|\equiv (Q \leftrightarrow^{k} P),\ P \lhd \{X\}_{k}}{P|\equiv Q|\sim X}$$

"*Se P acredita que Q e P compartilham a chave k e P recebe uma mensagem cifrada com k, então P acredita que Q disse X em algum momento*".

Regra 2 (R2) – Regra de Verificação de Identificador (*Nonce Verification Rule*) - Esta regra verifica se a mensagem é recente.

$$\frac{P|\equiv \#(X),\ P|\equiv Q\ |\sim X}{P|\equiv Q\ \ |\equiv X}$$

"*Se P acredita que X é novo e P acredita que Q disse X em algum momento, então P acredita que Q acredita em X*".

Regra 3 (R3) – Regra de Jurisdição (*Jurisdiction Rule*) - Esta regra verifica a confiança sobre alguma declaração.

$$\frac{P|\equiv Q \Rightarrow X,\ P|\equiv Q\ |\equiv\ X}{P|\equiv X}$$

"*Se P acredita que Q tem autoridade sobre a declaração de X e acredita que Q acredita em X, então P acredita em X*".

A partir da base teórica descrita anteriormente, o protocolo proposto nesta foi analisado para que possamos provar que seus objetivos são alcançados.

A simplicidade do protocolo de autenticação apresentado na seção 3.3 e ilustrado na Figura 18 nos mostra que a definição formal é idêntica a definição idealizada pela lógica BAN. A idealização do protocolo é mostrada a seguir na Tabela 18.

Tabela 18. Protocolo formalizado.

MSG 1 $L \rightarrow T$: #(RL)
MSG 2 $T \rightarrow L$: $\{RL\}_k$
MSG 3 $L \rightarrow T$: #(S)
MSG 4 $T \rightarrow L$: #(RT)
MSG 5 $L \rightarrow T$: $\{RT\}_k$

Seguindo o primeiro passo indicado pela lógica BAN obteve-se o protocolo idealizado mostrado na Tabela 18. Na mensagem 1, o Leitor envia para a Tag um novo desafio. A tag por sua vez encripta esse desafio e envia o resultado para o Leitor na mensagem 2 (MSG 2). Na mensagem 3 (MSG 3), o Leitor solicita sua própria autenticação. As mensagens 4 e 5 (MSG 4 e MSG 5) completam a autenticação do Leitor da mesma forma como foi realizada a autenticação da Tag. O protocolo idealizado na Tabela 18 segue exatamente os passos ilustrados na Figura 18.

A segunda etapa da lógica BAN resulta na especificação das suposições e objetivos do protocolo apresentadas na Tabela 19 e na Tabela 20.

Tabela 19. Suposições para análise do protocolo.

1) $L \mid\equiv T \leftrightarrow^k L$	4) $T \lhd \{RT\}_k$
2) $T \mid\equiv L \leftrightarrow^k T$	5) $L \mid\equiv$ #(RL)
3) $L \lhd \{RL\}_k$	6) $T \mid\equiv$ #(RT)

Tabela 20. Objetivos do protocolo.

1) Provar que T é autêntico, ou seja, T precisa provar para L que é uma tag válida no sistema.
2) Provar que L é autêntico, ou seja, L precisa provar para T que é um leitor autorizado no sistema.

A simplicidade da prova mostrada na Tabela 21 está no envolvimento de apenas duas entidades, no caso, tag e leitor como também no fato de que os objetivos do protocolo envolvem apenas a autenticação mútua de ambas as partes. Aliado a estes dois aspectos, concluímos também que a lógica BAN é um poderoso recurso para formalização de provas sem deixar de lado o aspecto simplicidade.

As mensagens 1, 3 e 4 (MSG 1, MSG 3 e MSG 4) da Tabela 21 são apresentadas mas não são consideradas pois tratam de informações enviadas em texto claro (*plaintext*[11]) e na lógica BAN estes tipos de mensagens são insignificantes [Burrows et al 1990]. Ao final de **MSG 2** prova-se que a Tag obteve sua autenticação. A autenticação mútua termina ao final da autenticidade de L mostrado em **MSG 5**.

Tabela 21. Prova do protocolo.

MSG 1	-	-
MSG 2	$L \mid\equiv L \leftrightarrow^{k} T,\ L \triangleleft \{RL\}_k$ 7) $L \mid\equiv T \mid\sim (RL)$ **R1(1,3)** $L \mid\equiv \#(RL), L \mid\equiv T \mid\sim (RL)$ 8) $L \mid\equiv T \mid\equiv RL$ **R2(5,7)** **Resultado:** L *autentica* T.	A segunda mensagem é utilizada para autenticar T. Se a resposta ao desafio for igual ao valor recebido na 1ª mensagem então T é uma tag legítima. Este resultado é obtido aplicando-se a Regra 1 às suposições 1 e 3 e em seguida aplicando-se a Regra 2 às suposições 5 e 7.
MSG 3	-	-
MSG 4	-	-
MSG 5	$T \mid\equiv T \leftrightarrow^{k} L,\ T \triangleleft \{RT\}_k$ 9) $T \mid\equiv L \mid\sim (RT)$ **R1(2,4)** $T \mid\equiv \#(RT), T \mid\equiv L \mid\sim (RT)$ 10) $T \mid\equiv L \mid\equiv RT$ **R2(6,9)** **Resultado:** T *autentica* L.	A quinta mensagem é utilizada para autenticar L. Se a resposta ao desafio for igual ao valor recebido na 4ª mensagem então L é um leitor válido. Este resultado é obtido aplicando-se a Regra 1 às suposições 2 e 4 e em seguida aplicando-se a Regra 2 às suposições 6 e 9.

11 Uma informação não-encriptada que é enviada de uma entidade para outra é chamada de "texto em claro, ou texto claro" (*plaintext*) [Tanenbaum 2003].

Formalizada a prova de que o protocolo atinge seus objetivos de autenticação das entidades do sistema RFID, o próximo passo é a padronização do protocolo especificado na seção 3.3.

3.5. PADRONIZAÇÃO DO PROTOCOLO DE AUTENTICAÇÃO

A padronização do protocolo consistiu em passar os comandos necessários às extensões de autenticação para uma linguagem compatível com o protocolo existente da EPCGlobal para tags Classe 1.

O protocolo não define faixas para criação de comandos personalizados [EPCGlobal 2005]. Dessa forma os códigos dos comandos que serão mostrados a seguir foram escolhidos aleatóriamente tomando-se o cuidado para não causar conflitos com os códigos dos comandos já existentes.

A solicitação de autenticação da tag, ou seja, o passo onde o leitor pede que a tag se identifique chamamos de comando *AuthTagRequest*. É um comando de solicitação, ou seja, originado do leitor. A Tabela 22 descreve este comando.

Tabela 22. O comando *AuthTagRequest*.

CMD (Código de comando)	0001 0001
LEN (Tamanho do conteúdo do campo Value)	-
Value (Valor enviado)	RL

Ao receber um comando AuthTagRequest, a tag responde com um comando AuthTagReply, que consiste na resposta para a solicitação do leitor. O formato do quadro deste comando é mostrado na Tabela 23.

Tabela 23. O comando de resposta *AuthTagReply*.

PREAMBL (Preâmbulo)	-
CRC	-
Value (Valor)	$Enc_k(RL)$

O protocolo original EPCGlobal dispõe de um comando chamado ScrollID denominado e descrito no Capítulo 2. Uma das extensões proposta nesta dissertação é a possibilidade de envio da ID encriptada. Dessa forma um novo comando foi especificado, o ScrollEncID, ou seja, um comando originado do leitor solicitando que a tag envie seu ID encriptado. A Tabela 24 ilustra o formato do quadro deste comando.

Tabela 24. O comando *ScrollEncID*.

CMD (Código de comando)	0001 0010
LEN (Tamanho do conteúdo do campo *Value*)	-
Value (Valor enviado)	-

Como resposta a solicitação ScrollEncID a tag responde com um comando ScrollEncIDReply cujo formato de quadro é mostrado na Tabela 25.

Tabela 25. O comando de resposta *ScrollEncIDReply*.

PREAMBL (Preâmbulo)	-
CRC	-
Value (Valor)	$Enc_k(ID)$

O leitor para autenticar-se necessita solicitar que a tag envie um desafio. Esta solicitação foi padronizada como *AuthRequest*, ou seja, o leitor solicita sua própria autenticação. O formato do quadro é mostrado na Tabela 26 abaixo.

Tabela 26. O comando *AuthRequest*.

CMD (Código de comando)	0001 1000
LEN (Tamanho do conteúdo do campo *Value*)	-
Value (Valor enviado)	-

Para responder a solicitação do comando AuthRequest, a tag envia o quadro AuthReply, ou seja, envia o desafio para o leitor. Este quadro é explicitado na Tabela 27 a seguir.

Tabela 27. O comando de resposta *AuthReply*.

PREAMBL (Preâmbulo)	-
CRC	-
Value (Valor)	RT

E finalmente para que o processo de autenticação do leitor seja concluído, este envia um comando Auth contendo o desafio encriptado, ou seja, valor que a tag vai conferir para decidir se autentica ou não o leitor. A Tabela 28 mostra o formato deste tipo de quadro.

Tabela 28. O comando *Auth*.

CMD (Código de comando)	0001 0100
LEN (Tamanho do conteúdo do campo *Value*)	-
Value (Valor enviado)	$Enc_k(RT)$

A Tabela 29 resume os comandos envolvidos no processo de autenticação proposto pelas extensões ao protocolo EPCGlobal.

Tabela 29. Comandos personalizados para disponibilização de segurança.

Comando	*Código*	*Comando de resposta*	*Descrição*
AuthRequest	00001000	AuthReply	Authrequest é o comando enviado pelo leitor solicitando sua própria autenticação junto à tag.
AuthTagRequest	00010001	AuthTagReply	AuthTagRequest é o comando enviado pelo leitor solicitando que a tag autentique-se, ou seja, que ela prove que é uma tag legítima do sistema.
ScrollEncID	00010010	ScrollEncIDReply	ScrollEncID é o pedido pela ID encriptada da tag.
Auth	00010100	-	Auth é o comando que o leitor envia para autenticar-se junto a tag. É sempre enviado após um AuthRequest.

3.6. DISCUSSÃO

O protocolo proposto garante três tipos de autenticação (da tag, do leitor, e de ambos) utilizando apenas uma função de criptografia e outra geradora de números aleatórios. A Tabela 30 compara o trabalho realizado nesta dissertação com trabalhos desenvolvidos e publicados por outros pesquisadores da área no quesito segurança e privacidade.

Quando fala-se na utilização em massa de etiquetas RFID, o primeiro problema que vem à mente da maioria das pessoas é justamente o aspecto privacidade. Dessa forma, a maioria das propostas publicadas na literatura atacam este problema, conforme pode-se ver na primeira linha da Tabela 30. O que varia entre as diversas propostas é exatamente como o problema é abordado. Dos trabalhos comparados na Tabela 30 apenas [Aigner e Feldhofer 2005] e [Duc et al 2006] preocuparam-se em não utilizar funções *hash*, pois sabe-se que estas possuem custo alto para disponibilização nas tags. Isto justifica a não utilização no protocolo proposto nesta dissertação. [Yang et al 2005], [Tsudik 2006] e [Dimitrou 2005] afirmam que suas propostas são possíveis de implementar com baixo custo, mas não justificam suas afirmações, pois nem mesmo indicam uma função *hash* a ser utilizada. [Duc et al 2006] e

[Lee et al 2006] não mencionam o aspecto custo em suas pesquisas.

Tabela 30. Comparação entre protocolos no aspecto segurança.

Trabalho/Pesquisa	A	B	C	D	E	F	G
Privacidade do usuário	O	O	O	O	O	O	O
Autenticação Mútua	O	O	X	O	Δ	O	O
Proteção contra espionagem	X	X	X	X	X	X	O
Não utiliza função hash	O	X	X	X	O	X	O
Proteção contra *man-in-the-middle*	Δ	Δ	X	X	X	X	O
Ataque do tipo *Replay*	X	O	O	O	X	O	O
Falsificação de tags	O	O	O	O	O	O	O
Falsificação de leitores	O	O	X	O	O	O	O
Autenticação apenas da tag	X	X	X	O	O	X	O
Autenticação apenas do leitor	X	X	X	O	O	X	O
Segue algum padrão existente	Δ (ISO[12])	X	X	X	O	X	O (EG[13])

Notação
Δ – Parcialmente satisfeita O- Totalmente satisfeita X – Não satisfeita
A – [Aigner e Feldhofer 2005] B- [Yang et al 2005] C- [Tsudik 2006] D- [Dimitrou 2005] E- [Duc et al 2006] F- [Lee et al 2006]
G- Este trabalho

Outro fator bastante relevante na análise realizada trata-se da utilização de autenticação mútua. Sabe-se que esse mecanismo de autenticação por si só já resolve os principais problemas relacionados com sistemas RFID. Apenas [Tsudik 2006] não propõe tal mecanismo. A justificativa para a não utilização de autenticação mútua é a simplificação do protocolo e para o objetivo específico de oferecer proteção relacionada à privacidade do usuário. [Aigner e Feldhofer 2005], [Yang et al 2005], [Dimitrou 2005] e [Lee et al 2006] propõem autenticação mútua sendo que os três últimos incorporam funções *hash* não especificadas em tais mecanismos, tornando as propostas incompletas e de custo mais alto.

12 Padrão ISO/IEC 18000.
13 Padrão EPCGlobal para tags Classe 1.

Este trabalho e as propostas de [Aigner e Feldhofer 2005] e [Duc et al 2006] utilizam criptografia de baixo custo com especificação dos algoritmos utilizados no caso do AES em [Aigner e Feldhofer 2005], o TEA nesta dissertação e funções XOR's em [Duc et al 2006]. [Duc et al 2006] apenas especifica a autenticação da tag. A especificação de que algoritmo utilizar representa um grande ponto positivo na proposta pois sua justificativa é de fundamental importância quando falamos no aspecto "custo de implementação" da solução.

Os ataques de falsificação (de leitores e tags) são facilmente resolvidos com a autenticação mútua falada anteriormente. [Duc et al 2006] não especifica autenticação mútua mas sua proposta também protege contra ambos os tipos de ataques.

Finalmente, os aspectos padronização e proteção contra análise de tráfego fazem o grande diferencial da proposta desta dissertação. Apenas [Aigner e Feldhofer 2005] e [Duc et al 2006] mencionam algum tipo de protocolo, no caso, o ISO/IEC 18000 em [Aigner e Feldhofer 2005] e o EPCGlobal em [Duc et al 2006]. A escolha do EPCGlobal é devido a este ser o padrão mundial que vem sendo adotado pela indústria, justificando assim sua maior importância. Aliado ao fato de ser o mais adotado pela indústria, temos também a ausência de propostas de segurança para este protocolo da EPCGlobal, fazendo assim que este trabalho seja um dos primeiros juntamente com [Duc et al 2006] a atacar problemas de segurança em RFID para o protocolo da EPCGlobal.

Nenhuma das propostas da literatura tem solução para a análise de tráfego inibindo que pessoas mal-intencionadas tenham acesso às informações trocadas entre tags e leitores. [Dimitrou 2005] enumera que isto é solucionado com sua proposta, no entanto detectamos que em seu trabalho a análise de tráfego é protegida apenas no mecanismo de autenticação, ou seja, comunicações posteriores não são protegidas. Dessa forma, o protocolo proposto nesta dissertação especificou um comando para comunicação segura entre tag e leitor, ou seja, a capacidade do envio da ID encriptada, equacionando dessa maneira o problema da análise e captura de tráfego. Outra solução muito questionada pela comunidade foi relacionada à pergunta: "Por que não guardar a ID encriptada na tag?". A princípio parece ser uma solução simples e sem custo adicional algum. Porém uma rápida análise da solução nos leva às seguintes conclusões: O gerenciamento de chaves seria um sério problema, assim como o problema da privacidade do usuário não seria resolvida pois a ID encriptada seria

sempre a mesma, permitindo que a tag fosse indevidamente rastreada.

A Tabela 31 compara os resultados com outros trabalhos publicados na literatura no quesito desempenho e requerimentos extras.

Tabela 31. Comparação entre protocolos no aspecto desempenho.

Trabalho/Pesquisa	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Número de Mensagens	7	7	2	7	4	5	7
Número de operações de encriptação	2	1	0	0	2	0	2
Número de operações de desencriptação	2	1	0	0	2	0	2
Número de operações hash	0	3	1 pela tag n pelo BD	3	0	3	0
Geração de números aleatórios	2	2	1	2	2	2	2

Notação

A – [Aigner e Feldhofer 2005] B- [Yang et al 2005] C- [Tsudik 2006] D- [Dimitrou 2005] E- [Duc et al 2006] F- [Lee et al 2006]

G- Este trabalho

A análise e comparação do aspecto desempenho e requisitos extras nos levou à conclusão de que as propostas possuem resultados similares. Nenhum trabalho propõe solução com significativa diferença de desempenho. A diferença básica entre as propostas é a utilização ou não de criptografia e funções *hash*. O número de mensagens reduzidas envolvidas do trabalho de [Tsudik 2006] é justificada pelos objetivos do autor, ou seja, atacar o problema da privacidade do usuário enquanto as demais preocupam-se com uma gama maior de problemas. O trabalho de [Duc et al 2006] envolve poucas mensagens pelo fato de não oferecer autenticação mútua. Dessa forma, no aspecto desempenho, este trabalho leva vantagem com relação aos demais por utilizar criptografia de baixo custo com o algoritmo TEA enquanto [Aigner e Feldhofer 2005] utiliza o AES e os demais utilizam funções *hash* não especificadas.

A Tabela 32 resume os resultados obtidos com a utilização das extensões de segurança propostos, ou seja, resume quais problemas são resolvidos e como eles são

resolvidos.

Tabela 32. Proteções oferecidas pelas extensões de segurança.

Resultado	Descrição
Proteção contra falsificação de tags (*tags spoofing or cloning*).	A utilização de autenticação forte evita que tags falsas, criadas para fraudar um sistema, sejam inseridas no contexto da aplicação para realização de atos mal-intencionados.
Proteção contra falsificação de leitores (*readers spoofing or cloning*).	Funciona da mesma forma da proteção contra falsificação de tags. Nenhum leitor autorizado pode consultar tags até que este esteja devidamente autenticado.
Proteção contra captura e análise de tráfego e *man-in-the-middle*.	As extensões criadas oferecem a possibilidade de enviar dados entre tag e leitor utilizando criptografia.
Proteção contra ataques do tipo *replay*.	A autenticação mútua previne este tipo de problema, pois um número aleatório criado em determinada sessão de autenticação apenas tem validade naquela sessão.
Proteção contra ataques do tipo reflexão.	Ataques de reflexão são prevenidos através da utilização de números aleatórios de conjuntos diferentes pelas entidades envolvidas.
Proteção contra rastreamento indevido do usuário (*tracking*)	A autenticação mútua permite apenas que leitores autorizados consultem às tags, impedindo assim que usuários não autorizados façam consultas às tags.

3.7. CONTRIBUIÇÕES

O trabalho apresentado tem contribuição na padronização completa do protocolo de autenticação proposto. [Duc et al 2006] foi quem mais se aproximou de uma padronização adequada, no entanto este não especificou formalmente seus acréscimos ao protocolo. Dessa forma, pode-se afirmar que este trabalho contribui em se tratando de segurança em RFID para o padrão EPCGlobal. O trabalho de [Aigner e Feldhofer 2005] apesar de mencionar o padrão ISO/IEC 18000 também não especifica formalmente nenhum acréscimo ou extensão ao protocolo e nem mesmo menciona a entidade EPCGlobal.

Pesquisadores como [Weis 2003] e [Sarma et al 2003] sinalizam a importância

da utilização do algoritmo TEA em pesquisas de segurança em sistemas RFID. Este caminho foi seguido e os resultados evidenciaram sucesso em sua utilização [Mota e Zorzo 2005].

Os resultados mostrados ao longo do trabalho nos aspectos padronização e utilização do algoritmo TEA certificam que a união de idéias já publicadas e aceitas pela comunidade científica possibilitaram a elaboração de uma nova proposta com resultados promissores.

4. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foram descritos os principais problemas de segurança envolvidos com a utilização da identificação por radiofreqüência assim como as mais diversas soluções propostas na literatura. A partir desse ponto, chegou-se à conclusão de que mecanismos de autenticação forte e criptografia são necessários para a disponibilização de segurança aos usuários da tecnologia. Esta conclusão levou à especificação de extensões padronizadas de segurança para o protocolo da EPCGlobal para tags Classe 1. Um protocolo de autenticação foi descrito e formalmente provado e detalhado passo a passo. Formato de quadros e campos foram formalmente especificados tendo como base o protocolo existente. E finalmente, foram analisados quais problemas podem ser solucionados com a adoção das modificações no protocolo.

No decorrer dos estudos outras conclusões surgiram como a do aspecto aplicação e algoritmo de criptografia. Com relação à aplicação temos que esta é de fundamental importância para a decisão de se usar ou não extensões de segurança pois em todo cenário de utilização da tecnologia pode ser necessário o uso de segurança. Embora, como foi dito ao longo do trabalho, a grande maioria da gama de aplicações pode ser beneficiada com disponibilização de segurança e privacidade sem prejuízo da eficiência e aumento de custos. Já o algoritmo de criptografia também é outro elemento que mereceu atenção neste trabalho, pois é o principal gerador de custos do protocolo de autenticação proposto. Após os testes realizados chegou-se a conclusão de que o TEA pode sim ser utilizado de forma barata e segura.

E, finalmente, temos o aspecto da padronização, que é a principal contribuição do trabalho descrito. Ou seja, chegou-se à conclusão de que tão importante quanto propor um protocolo seguro de comunicação é padronizar este protocolo com o padrão que está sendo utilizado mundialmente utilizado, no caso, o protocolo EPCGlobal.

Os trabalhos futuros incluem a utilização, implementação e testes das novas

extensões desenvolvidas nesta pesquisa em dispositivos reais, quando estes se tornarem facilmente disponíveis no Brasil, com o objetivo de mostrar que o protocolo proposto tratou de mecanismos de segurança prontos para serem disponibilizados no mundo real. Incluem também estudos de casos para delimitação detalhada de quais aplicações podem e devem realmente utilizar e dispor de segurança e privacidade. E, finalmente, o projeto, especificação e implementação de um software emulador do sistema RFID deve ser realizado com o objetivo de disponibilizar para a literatura especializada uma ferramenta de testes e avaliações de aspectos de segurança em identificação por radiofreqüência.

REFERÊNCIAS BIBLIOGRÁFICAS

[Aigner e Feldhofer 2005] AIGNER, M., Feldhofer M., "*Secure Symmetric Authentication for RFID Tags*", Telecommunication and Mobile Computing - TCMC 2005, Graz, Austria, **2005**.

[Brock 2001] BROCK, D., "The *Eletronic Product Code (EPC) – A naming Scheme for Physical Objects*", Auto-ID Center, White Paper, Cambridge, EUA, **2001**.

[Burrows et al 1990] BURROWS, M., Abadi, M., Needham, R., "*A logic of authentication*". ACM transactions on Computer Systems, vol 8, pp. 18-36, **1990**.

[Buth et al 2002] BUTH M. D., Geesing, V., Otto, M., Smits, J.P., "*The tradeoffs between TEA in hardware and Software*", in Hardware/Software Co-Design Proceedings, University of Twente, Computer Science Departament, Holanda. Disponível em: http://wwwhome.cs.utwente.nl/~smit/HWSWcodesign/index-eng.html, **2002**.

[Carlsen 1994] CARLSEN, U. "*Generating formal cryptographic protocol specifications*". In: IEEE SYMPOSIUM ON RESEARCH IN SECURITY AND PRIVACY, 1994, Oakland. Proceedings, New York: IEEE Computer Society Press, 1994. p. 137-146.

[Chammas et al 2004] CHAMMAS M.E., El-Khoury B., Halaby A., "*Implementing Security in RFID systems: The "tag emulator"*", American University of Beirut, http://itpapers.zdnet.com/abstract.aspx?scid=1120&sortby=comp&docid=113378 , Beirut, Líbano, **2004**.

[Duc et al 2006] DUC, D. N., Park, J., Lee, H., e Kim, K., "*Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning*". In Symposium on Cryptography and Information Security (SCIS) , Hiroshima, Japan, **2006**.

[Dimitrou 2005] DIMITROU T., "*A Lightweight RFID Protocol to protect against Traceability and Cloning attacks",* Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005, Atenas, Grécia, IEEE, **2005**.

[Engberg et al 2004] ENGBERG S.J., Harning M.B., Jensen C. D., "*Zero-knowledge Device Authentication: Privacy & Security Enhanced RFID preserving Business Value and Consumer Convenience*", Second Annual Conference on Privacy, Security and Trust, Session 2: Privacy and Identity, http://dev.hil.unb.ca/Texts/PST/, páginas 89-102, New Brunswick**,** Canada, **2004**.

[EPCGlobal 2004] EPCGlobal Inc, "*The EPCGlobal Network™ : Overview of Design , Benefits and Security*", http://www.epcglobalinc.org/ , **2004**.

[EPCGlobal 2005] EPCGlobal Inc, "*860MHz-930MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical Communication Interface Specification Candidate Recommendation, Version 1.0.1***"**, Technical Report**,** EPCGlobalInc, **2005.**

[EPCGlobal 2005b] EPCglobal Inc., "*Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.09*", Disponível em at http://www.epcglobalinc.org/ standards_technology/specifications.html. **2005**.

[Finkenzeller 2003] FINKENZELLER, K. , "*RFID Handbook*", Second Edition, Jonh Wiley & Sons, Ltd, 434p , Munique, Alemanha, **2003**.

[Gutierrez et al 2005] GUTIERREZ, R. M.V., Filha D.C.M., Neves, M.E.T.M.S, "*Complexo Eletrônico: Identificação por radiofreqüência*", BNDES Setorial, n.22, páginas 29-70, Rio De Janeiro, Brasil, **2005**.

[ISO 2004] International Organization for Standardization, "*Automatic Identification - Radio Frequency Identification for Item Management — Communications and Interfaces — Part 6: Physical Layer, Anti collision System and Protocol Values at 800-960 Mhz MODE* 4", **2004**.

[Juels et al 2003] JUELS A., Rivest R. L., Szydlo M. "*The blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy*", V. Atluri, ed. 8th ACM Conference on Computer and Communications Security, páginas. 103-111. ACM Press, Washington DC, EUA, **2003**.

[Juels 2004] JUELS A., Brainard J., "*Soft Blocking: Flexible Blocker Tags on the cheap*", S. De Capitani di Vimercati and P. Syverson, eds., Workshop on Privacy in the Electronic Society (WPES), páginas 1-7, ACM Press, Washington,DC, EUA, **2004**.

[Juels 2004b] JUELS A., "*Minimalist Cryptography for Low-Cost RFID Tags*", C. Blundo, ed., Security of Communication Networks (SCN), páginas 149-164, Springer-Verlag, Amalfi, Itália, **2004**.

[Henrici 2004] HENRICI D., Muller P., "*Tackling Security and Privacy Issues in Radio Frequency Identification Devices*", Pervasive Computing, Lecture Notes in Computer Science, páginas 219-224, Vienna, Austria, **2004**.

[Junior 1999] JUNIOR, R.B., "*Uma teoria de primeira ordem para especificação e análise de protocolos de criptografia*", Tese de Mestrado em Ciência da Computação, Universidade Federal do Ceará, 198pp, Fortaleza, Ceará, **1999**.

[Knospe 2004] KNOSPE, H. , "*RFID Security*", Information Security Technical Report, Vol 9, No 4, **2004**.

[Landt 2001] LANDT, J., AIM , "*Shrouds of Time – The history of RFID*", Global – The Association for automatic Identification and mobility, http://www.aimglobal.org, **2001**.

[Lee et al 2006] LEE, S., Asano, T., e Kim, K., "*RFID mutual authentication scheme based on synchronized secret information*". In Symposium on Cryptography and Information Security (SCIS), Hiroshima, Japan, **2006**.

[Silva 2004] SILVA, L.S., "*Public Key Infrastructure – PKI*", Novatec, 352pp, **2004**.

[Menezes et al 2001] MENEZES A.J., van Oorschot P.C., Vanstone S.A., "*Handbook of applied cryptography*", CRC Press , http://www.cacr.math.uwaterloo.ca/hac/ , 816pp, **2001**.

[Microchip 2005] MICROCHIP TECHNOLOGY INC, http://www.microchip.com. Acessado em Setembro de **2005**.

[Moroz 2004] MOROZ, R., "*Understanding Radio Frequency Identification (RFID) (Passive RFID)*", R Moroz Ltd, http://www.rmoroz.com/rfid.html, Markram, Canada, **2004**.

[Mota e Zorzo 2005] Mota, R.P.B., Zorzo, S.D., "*Análise de Recursos do algoritmo de criptografia simétrica TEA em microcontroladores PICs para utilização em tags RFID*" , 4th International Information and Telecomunication Technologies Symposium - I2TS 2005, Proceedings, Florianópolis, Santa Catarina, Brasil, **2005**.

[Nebojsa 2005] Nebojsa, Matic., "*PIC Microcontrollers for beginners too*". Disponível em http://www.mikroelektronika.co.yu/english/product/books/PICbook/picbook.htm, 2003. Acessado em Setembro de **2005**.

[Pikdev 2005] PIKDEV, "*Pikdev, and IDE for the development of PIC based applications under KDE*". Acessado em Setembro de **2005**.

[Ranasinghe et al 2004] RANASINGHE, D.C., Engels, D.W., Cole, P.H., "*Low-Cost RFID Systems: Confronting Security and Privacy*", Auto-ID Labs Research Workshop, Zurich, Suiça, **2004**.

[Santos et al 2002] SANTOS, M. C. M., Xexéo J. A.M., Rezende, J. F. "*Análise Formal de Protocolos de Segurança para Redes Celulares"*. In: IV Workshop de Comunicação sem Fio e Computação Móvel - WCSF2002 – São Paulo, **2002**.

[Sarma et al 2003] SARMA S.E., Weis, S.A., Engels, D.W., "*Radio Frequency Identification: Security Risks and Challenges*", Cryptobytes, RSA Laboratories, Volume 6, No 1, páginas 2-9, **2003**.

[Shepard 2005] SHEPHERD S., "*Tiny Encryption Algorithm*", http://www.simonshepherd.supanet.com/tea.htm , Bradford University, Inglaterra, **2005**.

[Stallings 2002] STALLINGS, W., "*Network Security Essentials*", Prentice Hall 2a Edição, 432pp, **2002**.

[Tanenbaum 2003] TANENBAUM, A. S., "*Redes de Computadores*", Tradução da 4a edição americana, Editora Campus, pp. 950, **2003**.

[Trinta 1998] TRINTA F. A. M., Macêdo R. C., "*Um estudo sobre criptografia e assinatura digital*", http://www.di.ufpe.br/~flash/ais98/cripto/criptografia.htm , Universidade Federal de Pernambuco, Refice-PE, **1998**. Acessado em Fevereiro de **2005**.

[Tsudik 2006] TSUDIK Gene., "*YA-TRAP: Yet Another Trivial RFID Authentication Protocol*", International Conference on Pervasive Computing and Communications-PerCom 2006, Pisa, Italy, IEEE Computer Society Press, **2006**.

[Warren 1998] Warren Andrew, "*Tiny Encryption Algorithm*". Disponível em http://www.geocities.com/SiliconValley/2499/code.html. Acessado em Setembro de 2005, **1998**.

[Weis 2003] WEIS, S.A. "*Security and Privacy in Radio-Frequency Identification Devices*", Master Thesis in Computer Science, MIT Massachusetts Institure of Technology, 79pp, Massachusetts, EUA, **2003**.

[Weis et al 2003b] WEIS, S.A., Sarma, E.S., Rivest, R.L., and Engels, D. W., "*Security and privacy aspects of low-cost radio frequency identification systems*", Security in Pervasive Computing, Dallas, Texas, EUA, **2003**.

[Weis 2004] WEIS S.A.., "RFID*: Concerns, Consensus, and Questions*", IEEE Security & Privacy, http://www.computer.org/security, páginas 48-50,publicado em março de **2004**.

[Wheeler et al 1995] Wheeler David J. and Needham Robert M., "*TEA, a Tiny Encryption Algorithm*", http://www.ftp.cl.cam.ac.uk/ftp/papers/djw-rmn/djw-rmn-tea.html , Technical report, páginas 363-366, Computer Laboratory, University of Cambridge, **1994**.

[Yang et al 2005] Yang J., Park J.,Lee H., Ren K., Kim K., "*Mutual authentication protocol for low-cost RFID",* Ecrypt Workshop on RFID and Lightweight Crypto, Graz, Austria, **2005**.