Centro de Ensino Universitário de Brasília - UniCEUB
Faculdade de Ciências Exatas e Tecnologia - FAET
Departamento de Engenharia da Computação

Projeto Final

Segurança de dados aplicada à urna eletrônica com foco na autenticação

Sérgio Tsugumiti Kobayashi
RA 2031804-5

Orientador: Prof. Marco Antônio de O. Araújo

Brasília/DF - Julho/2006

SÉRGIO TSUGUMITI KOBAYASHI

SEGURANÇA DE DADOS APLICADA À URNA ELETRÔNICA COM FOCO NA AUTENTICAÇÃO

Trabalho apresentado à banca examinadora da Faculdade de Ciências Exatas e Tecnológicas, para a conclusão do curso de Engenharia de Computação. Prof. Orientador: Marco Antônio de O. araújo

Brasília - DF

Julho/2006

SÉRGIO TSUGUMITI KOBAYAHSI

SEGURANÇA DE DADOS APLICADA À URNA ELETRÔNICA COM FOCO NA AUTENTICAÇÃO

COMISSÃO EXAMINADORA

_____

Prof. Orientador

_____

[Nome do Examinador]

_____

[Nome do Examinador]

Brasília, 04 de julho de 2006.

AGRADECIMENTOS

A minha mãe, Maria de Lourdes Silva, por me ensinar a sonhar, meu pai, Tsugumito Kobayashi e todos os meus irmãos, Marcelo Hissakiti Kobayashi, Ricardo Takao Kobayashi, Cecília Akemi Kobayashi, Paulo Sadao Kobayashi, Ronaldo Massaaki Kobayashi, Karla Sanae Kobayashi, Flávia Satiko Kobayashi e Felipe Yukio Kobayashi, por sempre me apontar o norte.

Ao meu primo-irmão Sérgio Alberto de Figueiredo (Sérgio Lorran), por nunca me deixar desistir.

Ao meu filho Thiago Kobayashi, por ser a realização de um sonho e ter me ensinado a ser pai.

Aos grandes amigos que sempre acreditaram e em especial ao Sérgio Malta Massuda, por ter me trazido de volta e mostrado o caminho.

Ao meu professor orientador Marco Antônio de O. Araújo, pelo direcionamento, clareza e comprometimento.

Ao coordenador do curso de Engenharia de Computação, prof. Abiezer Amarília Fernandes, por ter compreendido e dado suporte nos meus momentos de turbulência.

E a todo o corpo docente da Faculdade de Ciências Exatas e Tecnologia – FAET, pelo alimento intelectual, meu muito obrigado.

“A vida é muito curta para ser pequena.”
(desconhecido)

Resumo

Este é um trabalho de natureza acadêmica que tem por objetivo oferecer outra possibilidade de segurança da informação no uso de urnas eletrônicas para votação. Baseando-se nos 3 (três) módulos distintos no procedimento de votação eletrônica, conforme fluxo descritivo do Tribunal Superior Eleitoral – TSE [1], é possível dividir em três momentos a votação eletrônica:

1. **Autenticação e/ou identificação do eleitor;**
2. **Votação;**
3. **Apuração / Totalização.**

Este trabalho ficará restrito ao item número *1. Autenticação e/ou identificação do eleitor,* sendo que o mesmo poderá vir a servir como referência a trabalhos posteriores que tenham interesse em desenvolver as demais relações do fluxo descritivo do TSE [1].

Através do estudo de viabilidade da inserção de autenticação do eleitor através de um cartão de identificação e que contenha dados codificados de validação do eleitor em sua zona eleitoral, cria-se uma nova formatação para um dos pilares da segurança da informação, a confidencialidade, o eleitor não será mais identificado no momento da votação, mas autenticado e através do cartão, o eleitor passa a ter acesso ao módulo de votação, propriamente dito, sem que seus dados tenham sido gravados ou identificados, diminuindo-se a possibilidade de quebra de sigilo ou confidencialidade da informação, disponível na eleição eletrônica e da possibilidade de cruzamento de informações para se descobrir a escolha do eleitor na eleição.

Palavras chave:

Autenticação, segurança, votação eletrônica, urna eletrônica.

Abstract

This is a research of academic bias that seeks to offer another possibility in information security at the use of electronic ballots. Using the three distinct modules as a reference according to the description by Tribunal Superior Eleitoral (Supreme Election Court) [1], it is possible to separeta in three phases the electronic voting process:

1. **Authentication and/or identification of the voter;**
2. **Voting;**
3. **Summing.**

This research will be restrict to point 1. “Authentication and/or identification of the voter”, and this could serve as a reference to subsequent research that have interest in exploring others relations in the description.

Through the study of viablity of inserting the voter authentication using a identification card and that it contains encrypted data the voter validation on his voting zone, a new structure changes one of the information security pillars, confidentiality, the voter will not be identified anymore at the voting moment, but he will be authenticated and through the identification card the voter will have access to the voting module without recording or identifying your personnal data, reducing the possibility of openning secret data or confidentility of the information available at the electronic ballot and the possiblity of cross reference of information to find out which was the voter choice in the election.

Keywords:

Authentication, security, electronic voting, electronic ballot.

Sumário

ÍNDICE DE TABELAS

ÍNDICE DE FIGURAS

1. Introdução

Ao se falar sobre eleições, tem-se em mente o demasiado tempo despendido pelo antigo sistema de votação manual onde o eleitor era identificado, recebia a cédula de votação, se encaminhava para a área restrita de voto, marcava a escolha (candidato, branco ou nulo), inseria o voto na urna e recebia o comprovante conforme o seguinte fluxo (figura 1):

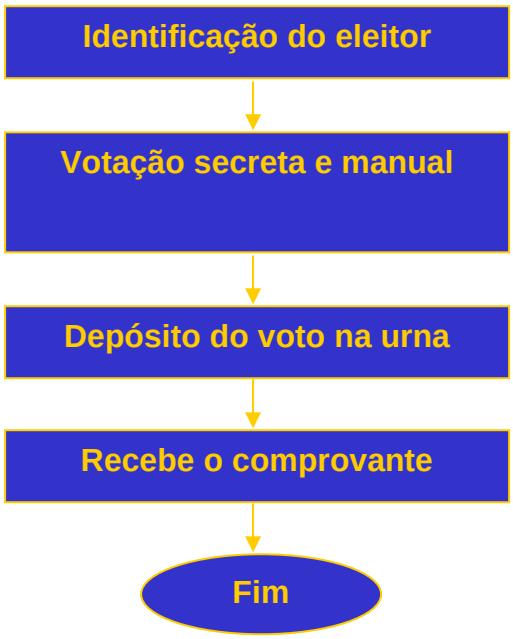


Figura 1 – Fluxo de votação manual

Além do tempo gasto em todo o processo da votação propriamente dita, diversas eram as vulnerabilidades e ameaças, com por exemplo a falsificação da cédula de votação, substituição da urna, alteração de voto nulo em voto válido durante a apuração/totalização, dentre outras possibilidades. Com a mudança de votação manual para votação eletrônica, houve um ganho de tempo extremamente considerável, simplificando-se o fluxo como segue (figura 2):

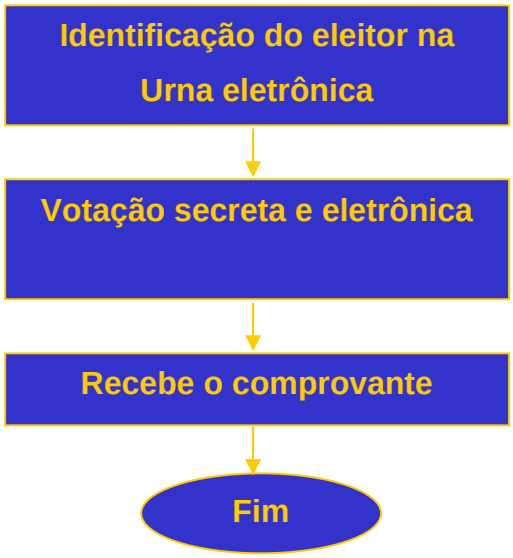


Figura 2 – Fluxo de votação eletrônica

O ganho mais expressivo pode ser constatado na apuração/totalização dos votos, que é basicamente a consolidação dos dados registrados nas unidades regionais (TRE – Tribunais Regionais Eleitorais), pela unidade central ou nacional (TSE – Tribunal Superior Eleitoral), tendo sido o ganho operacional da votação regional também substancial e expressivo, mas como todo sistema que necessite segurança e principalmente, segurança da informação, a urna eletrônica também está sujeita a vulnerabilidades e ameaças, no seu processo como um todo.

A partir do momento que o eleitor se apresenta para votar, o processo de uma eleição pode ser dividido nas seguintes etapas:

- Identificação do Eleitor
- Votação Secreta
- Apuração de cada urna e totalização dos votos

A proposta deste trabalho é aumentar a segurança das informações do eleitor, substituindo-se o título de eleitor em papel impresso, por um cartão do tipo smart card, com as informações referentes ao eleitor codificadas, diminuindo-se a possibilidade de uso indevido do documento por terceiros.

Baseado nos três pilares de segurança, definidos nas literaturas sobre segurança da informação em TI [2][3] – Integridade, confidencialidade e disponibilidade, este trabalho se propõe a minimizar as vulnerabilidades e ameaças da urna eletrônica com o foco específico no item confidencialidade do voto do eleitor, dividindo a urna em módulos, sendo o primeiro módulo referente à autenticação do eleitor para a votação. A interdependência dos módulos visa a garantir critérios de segurança individualizados e definidos de acordo com a identificação e definição das vulnerabilidades e ameaças em cada momento da votação.

Com a singularização do eleitor através do cartão do eleitor, modularização da urna eletrônica e autenticação do eleitor antes da votação, as vulnerabilidades e ameaças poderão ser restringidas e tratadas em instantes determinados e assistidos/auditados.

Nos capítulos seguintes deste trabalho, iremos demonstrar as formas de tratamento da informação para que se possa fazer o controle e a manutenção da segurança da informação, partindo-se de uma introdução sobre o sistema de votação no **primeiro capítulo**, de um breve histórico da atual urna eletrônica, histórico e normas referentes a segurança da informação em tecnologia da informação – TI, breve discurso sobre criptografia e os modelos criptográficos adotados na solução no **segundo capítulo**, as questões referentes ao hardware e software propostos no **terceiro capítulo**, a implementação no **quarto capítulo** e a conclusão no **quinto capítulo**.

2. A urna eletrônica

Este capítulo é a base histórica e conceitual para um melhor entendimento, desenvolvimento lógico e subsídio para a implementação do protótipo a que se propõe este trabalho.

2.1 Urna eletrônica

Como base histórica e evolutiva, esse capítulo irá discorrer sobre o sistema de votação eletrônica utilizado no Brasil, considerando-se os aspectos numéricos dos volumes considerados em uma votação eletrônica, bem como sobre a urna eletrônica propriamente dita e suas posteriores atualizações. Esse capítulo visa também, trazer informações sobre a legislação atual brasileira com relação ao processo eleitoral eletrônico, avaliações do sistema com relação aos prós e contras e a perspectiva mundial sobre a votação eletrônica, divergências e sugestões de modelos mais seguros.

2.1.1 Histórico / Evolução

O sistema de votação eletrônica no Brasil teve seu início no ano de 1996, quando por questões de segurança e viabilização do sistema, não foi coberto todo o território nacional, tendo atingido aproximadamente 1,04% do número total de municípios que na época era de 5.507 municípios (tabela 1), o que representou aproximadamente 32,07% do eleitorado nacional ou 32.478.153 de eleitores efetivos, no ano de 1998 ocorreu um salto do montante atingido pela votação eletrônica, chegando-se aos 9,74% do número total de municípios ou 537 municípios, que representou 57,62% do eleitorado nacional que na época já era de 61.111.992 de eleitores efetivos, nas eleições do ano 2000 esses números saltaram para 100% do número total de municípios com votação eletrônica, sendo de 5.559 municípios e 100% do eleitorado nacional ou 109.780.071 de eleitores efetivos, tendo esses valores sido seguidos nas eleições subseqüentes, 2002 e 2004, com as variações na quantidade de municípios e eleitorado efetivo nos seguintes valores, 5.561 para 5.563 e 115.254.113 para 119.782.000, respectivamente.

Tabela 1 – Histórico e evolução da votação eletrônica no Brasil

VOTAÇÃO ELETRÔNICA NO BRASIL					
Eleições	**1996**	**1998**	**2000**	**2002**	**2004**
Nº Total de Municípios	5.507	5.513	5.559	5.561	5.563
Nº de Municípios com Votação Eletrônica	57	537	5.559	5.561	5.563
% de Municípios com Votação Eletrônica	1,04%	9,74%	100%	100%	100%
Eleitorado com Votação Eletrônica	32.478.153	61.111.922	109.780.071	115.254.113	119.782.000
% do eleitorado atingido	32,07%	57,62%	100%	100%	100%
Total de Votantes	101.284.121	106.101.067	109.780.071	115.254.113	119.782.000
População Total	157.070.163	158.232.252	169.872.856	173.900.000	182.000.000
Seções eleitorais com urnas eletrônicas	72.311	145.213	322.500	367.300	408.810

Fonte: TSE

Pode-se perceber também que a quantidade de urnas eletrônicas nos anos eleitorais acompanhou a evolução numérica dos demais itens, tendo sido de 72.311 urnas eletrônicas no ano de 1996, 145.213 no ano de 1998, 322.500 no ano 2000, 367.300 no ano 2002 e 408.810 no ano 2004.

A própria urna eletrônica sofreu, também, alterações ao longo do período descrito acima, tanto em *hardware* quanto em *software,* com o intuito de se melhorar o desempenho utilizando-se de novas tecnologias disponíveis e dificultar quaisquer intervenções ou quebra de segurança da urna.

“Há três versões de *hardware* para a urna eletrônica, os modelos UE 96, UE 98 e UE 2000, que foram adquiridos nos anos de 1996, 1998 e 2000, respectivamente. Todos os modelos apresentam a mesma arquitetura básica, embora diferenças, decorrentes da evolução tecnológica, possam ser observadas no seu *hardware*” [1].

As características tecnológicas da urna em questão, bem como as modificações, estão descritas no anexo A – Hardware da urna eletrônica.

A UE 2000 permite a ligação de até dois terminais funcionando paralelamente, estando ligados nessa configuração, um terminal irá atuar como mestre e os outros dois como escravos, ou seja, todos os dados da votação ficarão armazenados no terminal

mestre, que também será o responsável pela totalização da seção eleitoral, cabe aos terminais escravos somente a função de entrada e saída de dados (figura 3).

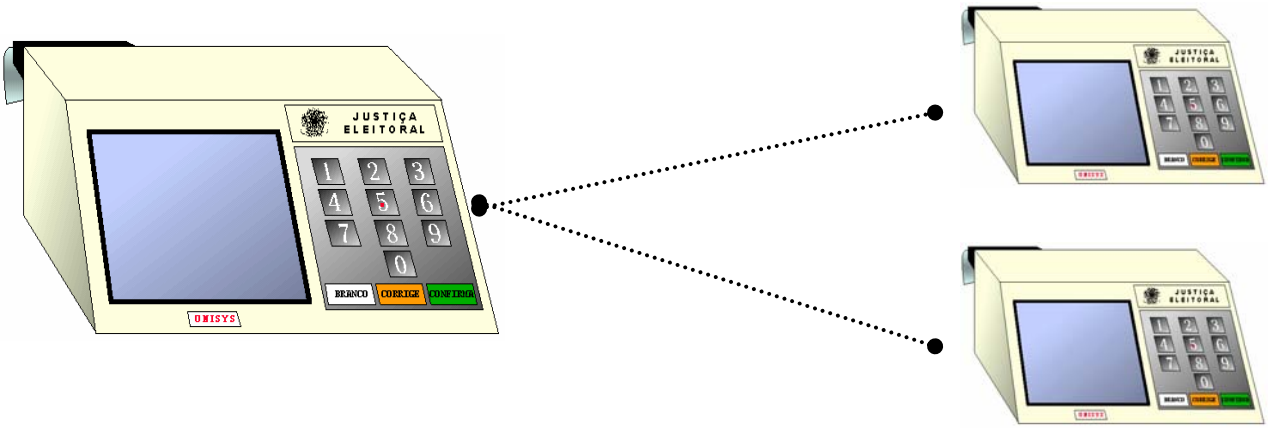


Figura 3 – Ligação da UE2000

A urna eletrônica possui ainda um microterminal, composto de um teclado numérico e um display de cristal líquido com 2/4 linhas por 20 caracteres (figura 2), ligado ao terminal do eleitor (UE), para uso exclusivo do mesário e tem as seguintes atribuições:

- identificação do eleitor e se está apto a votar na Seção;
- localização do comprovante na folha de votação;
- liberar a UE para votação;
- suspensão do voto;
- finalização da votação

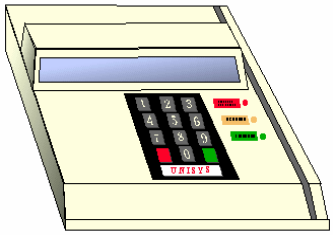


Figura 4 – Microterminal do eleitor

Os dispositivos de entrada e saída encontrados na urna eletrônica e “não-visíveis” pelo eleitor, tem funções específicas, sendo estes:

- Impressora utilizada para impressão dos boletins de urna, dos relatórios de teste e carga do software;
- Acionador de disquete 3 ½" para gravação dos boletins de urna;
- Dois *slots* para a inserção de cartões de memória removível tipo *flash*, denominados *flash* interno e *flash* externo, utilizados como *backup*;
- Um conector para teclado, utilizado para manutenção e testes;
- Dois conectores USB (*universal serial bus*);
- Um conector para fone de ouvido, utilizado para eleitores com deficiência visual;
- Um conector para conexão com outros terminais do eleitor, somente no microterminal;
- Uma saída para impressora, no microterminal.

Todos os dispositivos que tem acesso externo são lacrados após a carga do software e dos dados da eleição e assim ficam até a conclusão da eleição. O microterminal é conectado ao terminal do eleitor através de cabo serial diretamente na placa do terminal. A urna possui uma arquitetura similar a de um computador pessoal IBM/PC, com algumas alterações para uma melhoria do controle e da segurança, mudanças no *firmware* também são encontradas na urna, seja para autenticação e criptografia como para segurança.

A segurança da urna, segundo o TSE, consiste na integração de Hardware, Software básico, Softwares da eleição, Sistema de Criptografia e procedimentos, tornando a urna singular, como segue:

- Preparação do Hardware: Procomp
- Software básico específico: Microbase
- Sistema Operacional comum de mercado - VirtuOS
- Gerenciadores específico da Urna
- Sistema de Criptografia: CEPESC
- Validação: TSE
- Integração de hardware, software e procedimentos: TSE

Geração das informações para a UE.

Os aplicativos das eleições são apresentados aos Partidos Políticos antes das eleições, conforme determina o art. 66 da Lei nº 9.504. A convocação dos partidos para essa apresentação é realizada segundo Resolução do TSE.

Preparação no TSE

- Geração dos códigos executáveis dos aplicativos da eleição;
- Integração com o sistema de segurança;
- Autenticação dos códigos executáveis, empacotamento e envio aos TRE's;
- Envio de cadastro de eleitores, organizados por Zonas Eleitorais, aos TRE´s.

Preparação nos TREs

- Arquivos de candidatos, fotografias e coligações.

Preparação das matrizes para carga das Urnas

- Carrega o programa gerador de mídia (GM) a matriz para inseminar as urnas com: todos os softwares da urna, tabela de eleitores, candidatos, coligações, Zonas, Seções;
- O programa gera o FC (*flash* de carga) com softwares oficiais das Eleições, tabela de candidatos e eleitores, controlados e autenticados;
- O programa pára de funcionar se detectar aplicativos não oficiais;
- Ao inserir o FC na urna ligada, o sistema faz a limpeza e a inseminação nas Urnas.

Após a Carga (inseminação) executa-se o aplicativo da eleição para:

- Carregar a tabela de candidatos e eleitores no aplicativo oficial;
- Associar os candidatos aos registradores (contadores) na seqüência de Cargo, Partido e Candidato (número e nome).

2.1.2 Legislação brasileira

A lei que regulamenta a votação eletrônica no Brasil é a Lei 9.504 de 30 de setembro de 1997, com as devidas alterações, sendo específico para o conteúdo desse trabalho o artigo número 59 que segue em seu teor original ou parte relacionada ao trabalho em questão no anexo B – Extrato da Lei no. 9.504 do TSE.

"Do Sistema Eletrônico de Votação e da Totalização dos Votos

Art. 59. A votação e a totalização dos votos serão feitas por sistema eletrônico, podendo o Tribunal Superior Eleitoral autorizar, em caráter excepcional, a aplicação das regras fixadas nos arts. 83 a 89.

§ 1º A votação eletrônica será feita no número do candidato ou da legenda partidária, devendo o nome e fotografia do candidato e o nome do partido ou a legenda partidária aparecer no painel da urna eletrônica, com a expressão designadora do cargo disputado no masculino ou feminino, conforme o caso.

§ 2º Na votação para as eleições proporcionais, serão computados para a legenda partidária os votos em que não seja possível a identificação do candidato, desde que o número identificador do partido seja digitado de forma correta.

§ 3º A urna eletrônica exibirá para o eleitor, primeiramente, os painéis referentes às eleições proporcionais e, em seguida, os referentes às eleições majoritárias.

§ 4º A urna eletrônica disporá de mecanismo que permita a impressão do voto, sua conferência visual e depósito automático, sem contato manual, em local previamente lacrado, após conferência pelo eleitor. (Parágrafo incluído pela Lei nº 10.408, de 10.1.2002).

§ 5º Se, ao conferir o voto impresso, o eleitor não concordar com os dados nele registrados, poderá cancelá-lo e repetir a votação pelo sistema eletrônico. Caso reitere a discordância entre os dados da tela da urna eletrônica e o voto impresso, seu voto será colhido em separado e apurado na forma que for regulamentada pelo Tribunal Superior Eleitoral, observado, no que couber, o disposto no art. 82 desta Lei.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 6º Na véspera do dia da votação, o juiz eleitoral, em audiência pública, sorteará três por cento das urnas de cada zona eleitoral, respeitado o limite mínimo de três urnas por

Município, que deverão ter seus votos impressos contados e conferidos com os resultados apresentados pelo respectivo boletim de urna.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 7º A diferença entre o resultado apresentado no boletim de urna e o da contagem dos votos impressos será resolvida pelo juiz eleitoral, que também decidirá sobre a conferência de outras urnas.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 4º A urna eletrônica disporá de recursos que, mediante assinatura digital, permitam o registro digital de cada voto e a identificação da urna em que foi registrado, resguardado o anonimato do eleitor. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 5º Caberá à Justiça Eleitoral definir a chave de segurança e a identificação da urna eletrônica de que trata o § 4º. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 6º Ao final da eleição, a urna eletrônica procederá à assinatura digital do arquivo de votos, com aplicação do registro de horário e do arquivo do boletim de urna, de maneira a impedir a substituição de votos e a alteração dos registros dos termos de início e término da votação. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 7º O Tribunal Superior Eleitoral colocará à disposição dos eleitores urnas eletrônicas destinadas a treinamento. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 8º O Tribunal Superior Eleitoral colocará à disposição dos eleitores urnas eletrônicas destinadas a treinamento.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)". [2]

2.1.3 Prós e contras

A implementação da votação eletrônica no Brasil trouxe uma agilidade ao processo de eleição, considerando-se os vários dias de contagem e apuração despendidos nas antigas eleições manuais, a quantidade de pessoal envolvido e os gastos com todo o modelo antigo, a consolidação dos dados pelos diversos tribunais regionais eleitorais espalhados na grandiosidade territorial brasileira, sem se falar das vulnerabilidades e ameaças a que o sistema se acometia. Quando da implantação da votação eletrônica no Brasil os critérios relevantes na época foram segurança e a velocidade da eleição, mesmo lembrando-se das enormes filas que se formaram nas zonas eleitorais, devido ao desconhecimento ou mesmo da "fobia" eletrônica que se fazia e faz presente em todo processo de automatização, o processo como um todo pode ser considerado satisfatório

uma vez que nas eleições seguintes houve a consolidação definitiva desse processo eletrônico de votação atingindo-se a totalidade ou 100% do eleitorado nacional.

Como todo sistema informatizado, a votação eletrônica e mais especificamente a urna eletrônica, tem lados opostos com relação à confiabilidade, de um lado pode-se considerar o próprio Tribunal Superior Eleitoral – TSE e do outro um grupo de parlamentares e intelectuais de diversas universidades, com se pode perceber na citação que segue:

"No dia 29 de maio de 2002 ocorreram dois fatos simultâneos no Congresso em Brasília relativos à questão da confiabilidade do voto eletrônico no Brasil. No Centro Cultural da Câmara dos Deputados, aconteceu o Seminário do Voto Eletrônico (SVE), promovido pelo Partido Democrático Trabalhista (PDT). E já quase no final dos trabalhos do SVE, o Ministro Nelson Jobim, presidente do Tribunal Superior Eleitoral (TSE) compareceu ao Congresso Nacional para apresentar ao presidente da Câmara, Aécio Neves e ao presidente do Senado, Rames Tebet, o Relatório Unicamp produzido pela Fundação de Desenvolvimento da Universidade Estadual de Campinas (FUNCAMP) sobre o Sistema Informatizado de Eleições (SIE)" [5].

O relatório apresentado pela FUNCAMP é referente ao contrato TSE no. 54/2001 de prestação de serviços técnicos especializados, tendo sido contratado para efetuar uma avaliação sobre o Sistema Informatizado de Eleições (SIE), que não se restringe à votação eletrônica, mas trata do processo inteiro, desde a urna eletrônica até os sistemas desenvolvidos para a contagem e apuração dos votos, e teve como resultado final um parecer positivo com relação à confiabilidade do sistema como um todo, mesmo trazendo algumas ressalvas e algumas sugestões para a melhoria da segurança e da própria confiabilidade do sistema. Por outro lado, tem-se uma mobilização com diversos questionamentos sobre a segurança do sistema como um todo e mais especificamente, sobre a urna eletrônica e a confiabilidade de todos os sistemas que a compõe (*hardware e software*), são feitas duras críticas com relação à confidencialidade, à integridade e a disponibilidade das informações constantes da eleição e ao mesmo tempo apresentam-se vulnerabilidades e possíveis ameaças a essas informações [6], como estes dois fatos ocorreram quase que simultaneamente, serviram como suporte para as análises a serem feitas no capítulo de segurança, tanto pelo lado positivo quanto pelo lado negativo.

2.1.4 Perspectiva mundial

O Sistema Eletrônico de Votação adotado no Brasil já foi oferecido a diversos países, tendo sido testado e mesmo utilizado em alguns países da América do Sul (Venezuela e Paraguai), mas em países considerados do primeiro mundo a votação eletrônica é um tópico extremamente discutido nos meios acadêmicos, Massachusetts Institute of technology – MIT, Harvard, Oxford, University of New Castle, dentre outros, e nessas discussões diversos são os questionamentos sobre a segurança e a garantia das informações, por exemplo, um estudo da MIT [7] questiona a utilização de impressora para a conferência do voto na urna, e o questionamento é referente à divergência, se houver, entre o voto impresso e o voto eletrônico, destes qual será o verdadeiro e porque? Como terá sido a adulteração, que meios deverão existir para garantir que tal situação não venha a ocorrer ou que seja minimizada a possibilidade de que venha a acontecer? Diante desses questionamentos e diversos outros, a implementação de um sistema eletrônico de votação é ainda hoje vetada.

Algumas propostas para a conferência do voto eletrônico com o voto impresso já estão bastante difundidas e sedimentadas, dentre essas pode-se citar o algoritmo de criptografia impressa de David Chaum [8][9], que por estar fora do escopo desse trabalho não será aqui discutido.

2.2 Segurança da informação

“O único sistema verdadeiramente seguro é aquele que está desligado, desplugado, trancado num cofre de titanium, lacrado, enterrado em um *bunker* de concreto, envolto por gás nervoso e vigiado por guardas armados muito bem pagos. Mesmo assim, eu não apostaria minha vida nisso” [10]

2.2.1 Conceito

A segurança da informação tem deixado de ser tratada como um assunto puramente técnico da área de Tecnologia de Informação – TI e vem sendo utilizada como uma ferramenta fundamental em todas as áreas que tenham a necessidade de

confidencialidade (sigilo), integridade e disponibilidade da informação. Sabe-se que a garantia total de segurança é um fato praticamente utópico, o que se procura é a utilização de ferramentas ou métodos que possibilitem reduzir os impactos sobre os ativos, ou seja, tudo o que tenha valor e necessite de proteção, sejam ativos físicos ou ativos lógicos.

2.2.2 Objetivo - Confidencialidade

Sendo objetivo principal desse trabalho aumentar a segurança na urna eletrônica, tem-se como objetivo de suporte a confidencialidade, especificamente o sigilo do voto eletrônico. Entenda-se como sigilo no voto eletrônico o momento em que o eleitor se dirige à urna e lá efetua o seu voto, sendo que nesse momento não pode ocorrer o relacionamento entre o eleitor e a escolha do mesmo.

"Do Voto Secreto

- *Legislação Complementar*: Lei nº 9.504/97, arts. 59 a 62: sistema eletrônico de votação e totalização dos votos. Arts. 82 a 89: aplicáveis, juntamente com as regras dos arts. 103 e 104 deste Código, ao sistema convencional.

Art. 103. O sigilo do voto é assegurado mediante as seguintes providências:

I - uso de cédulas oficiais em todas as eleições, de acordo com modelo aprovado pelo Tribunal Superior;

II - isolamento do eleitor em cabina indevassável para o só efeito de assinalar na cédula o candidato de sua escolha e, em seguida, fechá-la;

III - verificação da autenticidade da cédula oficial à vista das rubricas;

IV - emprego de urna que assegure a inviolabilidade do sufrágio e seja suficientemente ampla para que não se acumulem as cédulas na ordem em que forem introduzidas" [4].

2.2.3 Norma ISO/IEC 17799/2000

Em 1987 o departamento de comércio e indústria do Reino Unido (DTI) criou um centro de segurança de informações, o CCSC (Commercial Computer Security Centre)

que dentre suas atribuições tinha a tarefa de criar uma norma de segurança das informações para o Reino Unido.

Desde 1989 vários documentos preliminares foram publicados por esse centro, até que, em 1995, surgiu a BS7799 (British Standard 7799). Esse documento foi disponibilizado em duas partes para consulta pública, a primeira em 1995 e a segunda em 1998. Em 1o de dezembro de 2000, após incorporar diversas sugestões e alterações, a BS7799 ganhou status internacional com sua publicação na forma da ISO/IEC 17799:2000.

Por fim, em setembro de 2001, a ABNT homologou a versão brasileira da norma, denominada NBR ISO/IEC 17799.

A norma nacional de segurança de informação é dividida em 10 (dez) macro controles:

- Política de Segurança;
- Segurança Organizacional;
- Classificação e Controle dos Ativos da Informação;
- Segurança em Pessoas;
- Segurança Física e do Ambiente;
- Gerenciamento de Operações e Comunicações;
- Controle de Acesso;
- Desenvolvimento da Segurança de Sistemas;
- Gestão da Continuidade do Negócio;
- Conformidade.

Não é intuito desse trabalho, criar uma política de segurança, mas tão somente identificar as necessidades de segurança em um sistema de votação eletrônica. A norma esta sendo utilizada como norteadora, e é subsídio o macro controle "*Classificação e Controle dos ativos da informação"*, que é utilizado para delinear os ativos e possibilitar assim que se faça uma avaliação dos riscos aplicada à urna eletrônica.

Ativos são todos os possíveis "alvos" que venham a prejudicar ou derrubar um dos pilares da segurança [3].

2.2.4 Análise de riscos aplicada à urna eletrônica

Como dito anteriormente, os ativos são elementos fundamentais quando pensamos em Segurança da Informação e também são as peças principais quando iniciamos uma análise para verificarmos os possíveis riscos que o sistema possa apresentar. A abrangência na identificação dos elementos importantes que devam ser protegidos contra as ameaças, melhora a visualização das áreas de vulnerabilidades, e aumenta cada vez mais a probabilidade de sucesso e conseqüentemente a segurança, numa avaliação de riscos, a identificação das vulnerabilidades e ameaças a que está sujeita a informação e a probabilidade de que um ataque identificado possa ocorrer é a referência para uma boa análise de riscos.

O macro controle da norma ISO/IEC 17799:2000 que trata da *Classificação e Controle dos ativos da informação* tem por objetivo prover diretrizes para a gestão de segurança dos ativos, classificação, registro e controle, sendo o controle efetivado através de um inventário dos ativos da empresa e pode ser assim definido:

- Contabilização dos ativos

  Inventário – cada urna eletrônica tem um número de série e este número é registrado no TSE e consta do banco de dados das urnas, para o caso em que seja necessário algum tipo de rastreabilidade, além disso, o modelo sugerido nesse trabalho propõe que seja incluído no banco de dados os dados referentes aos módulos do sistema e uma vez que alguns desses dados serão utilizados para a comunicação entre os módulos e também para o processo de criptografia, existe uma necessidade de cuidados especiais para a guarda dessas informações;

- Classificação da informação

  A classificação da informação em um sistema eleitoral pode ser dividida em três tipos:

  1) Informações Confidenciais: informações de caráter exclusivo e sigiloso;
  2) Informações Corporativas: informações pertinentes somente ao TSE;
  3) Informações Públicas: informações a serem distribuídas e veiculadas.

A informação a ser tratada nesse trabalho se restringirá à informação confidencial, ou seja, extremamente sigilosa e inviolável referente à escolha do eleitor no voto secreto;

- Rótulos e tratamento da informação.

  Cada informação de voto deve ter o rótulo de confidencialidade, integridade e disponibilidade e deve ser tratado como definido por esses rótulos, sendo que esse trabalho se restringe à confidencialidade e à urna eletrônica, tratando portanto das informações que circularão nesse dispositivo e no critério de confidencialidade.

2.2.4.1 Ativos físicos e ativos lógicos

Dentro da definição dada aos ativos anteriormente, podemos dividi-los em duas categorias distintas:

Ativos físicos – todo elemento físico que possa ser manipulado, movido, remanejado, por exemplo, a urna eletrônica;

Ativos lógicos – todo elemento lógico que possa ser manipulado, alterado, apagado, por exemplo, programas de computador.

Dentro do foco a que se destina esse trabalho pode-se identificar os seguintes ativos classificados conforme a definição anterior.

Ativos físicos:

- Urna eletrônica;
- Módulo de autenticação;
- Módulo de TRE zonal;
- Módulo de votação.

Ativos lógicos:

- Fluxograma de programação;
- Programa assembler dos módulos;
- Esquema eletrônico dos módulos;
- Número de série dos microcontroladores;

- Número de série dos *transcoders*;
- Número "*seed*" do TSE;
- Número de série do TSE (ou fabricante da urna).

2.2.4.2 Vulnerabilidades e ameaças

A vulnerabilidade pode ser considerada como o ponto fraco de um ativo ou grupo de ativos, onde uma ameaça direcionada e que já conheça essa vulnerabilidade possa se aproveitar deste ponto fraco para causar danos ao ativo. Cabe ressaltar que não é a vulnerabilidade a responsável pelos danos causados, sendo apenas uma condição que permite a ação da ameaça [3].

As ameaças são mecanismos ou ferramentas que atuam nas vulnerabilidades do ativo causando perdas, danos ou alterações ao mesmo. Estes ataques realizados pelas ameaças sobre as informações podem ser diretos ou indiretos e podem ser acidentais ou propositais, no trabalho em questão considerar-se-á somente ataques propositais.

Para esse trabalho consideram-se ameaças no momento de transporte da urna eletrônica para a distribuição e do local de votação para o local de contagem e apuração, e esse tipo de ameaça seria aos ativos físicos descritos acima, para tanto o modelo sugere controles para esses tipos de ameaças e/ou ataques, além disso, considera-se a possibilidade de ataques "internos" sobre os ativos lógicos, programas e dados contidos na urna eletrônica disponibilizada para cada seção de votação, ou os Tribunais Regionais Eleitorais – TRE´s.

Para se evitar a troca ou o manuseio da urna eletrônica quando no percurso entre o TRE e a seção de votação, bem como, da seção de votação para a área de contagem e apuração, a mesma deverá ser autenticada tanto pelo TSE quanto pelo TRE, com uma chave de identificação única, utilizando-se para isso dos números de série dos componentes e de informações de criptografia que se façam necessárias para esse procedimento, como essa fase de autenticação foge ao escopo deste trabalho, serve como uma introdução para a autenticação do eleitor que é o foco a ser questionado e implementado. Partindo-se da premissa de que a urna possui mecanismos de controle para o envio e recebimento na seção, quais seriam as vulnerabilidades e ameaças a serem tratadas? A troca do sistema que foi gravado na urna pode ser considerada como

uma ameaça lógica ao sistema, mas a vulnerabilidade para esse tipo de ataque pode ser controlada com algumas ferramentas disponíveis no microcontrolador como por exemplo o *checksum*, as urnas eletrônicas ao serem programadas recebem um *checksum*, um cálculo de checagem utilizando-se de informações contidas no microcontrolador, como por exemplo o número de série, etc, e esse número pode ser verificado a qualquer momento, mesmo porque ao se instalar o programa da urna eletrônica, automaticamente, o microcontrolador irá se comunicar com os demais módulos informando seus dados de configuração, ID, no. de série, etc., e estes passam a trabalhar em conjunto e qualquer alteração no código do microcontrolador irá gerar uma não autenticação do mesmo com os demais módulos o que desabilitará a urna, o que funcionaria inclusive para a segurança contra ameaças ao ativo físico pela troca de algum componente dos módulos, como por exemplo a memória que contenha os dados dos candidatos ou dos eleitores. Como o conjunto não possui um sistema operacional que esteja sendo executado em um nível mais baixo, a seqüência do programa segue à mesma seqüência do fluxograma, sem que hajam chamadas para um sistema controlador que possa conter funções “maliciosas” com alterações de funcionalidade, ou que se possa identificar determinado dado direcionando-se a escolha do eleitor para determinado candidato, uma vez que o programa utiliza de ponteiros com endereço de memória, a função “maliciosa” deveria conhecer esse endereço para poder se utilizar de tal artifício, e esses endereços podem ser “misturados” para cada seção,o que dificultaria ainda mais qualquer intervenção externa ao sistema.

2.2.5 Criptografia

A palavra ”criptografia” é composta dos termos gregos ”*kryptós*” (secreto, oculto) e ”*gráphein*” (escrita, escrever). E é considerada a ciência de comunicar-se secretamente. O mais antigo problema na criptografia se refere à comunicação segura em um canal inseguro, o objetivo principal é tornar a mensagem indecifrável para terceiros, que possam vir a interceptar a comunicação. A criptografia é tão antiga quanto a escrita, porém, somente nas últimas décadas tornou-se alvo de estudos científicos mais extensos e aprofundados, com o surgimento de novas áreas como a criptoanálise e a criptologia.

A segurança da informação tem sentido desde a época em que não existiam meios eletrônicos de envio e armazenamento da informação, onde na maioria dos casos o conteúdo da informação era armazenado em meio físico como o papel, por exemplo, e

para que houvesse a garantia da segurança dessa informação, foram criados protocolos e mecanismos de segurança fundamentados na aplicação de leis, quando enviamos uma carta pelo correio, a segurança é o lacre de fechamento do envelope e o mecanismo de proteção é a lei de sigilo da correspondência, que me garante a pena ao fraudador por ter acesso ao conteúdo da correspondência de outra pessoa, ou em outro caso o papel moeda que tem um tipo de tinta e uma gramatura diferente de outros papéis encontrados no mercado dificultando a reprodução do mesmo.

A forma de troca de informações mudou com certeza, mas a filosofia da troca de informações não sofreu tanta alteração, afinal o que se pretende é tão simplesmente “movimentar” a informação, seja em meio físico (papel) ou em meio lógico (bits e bytes), mas talvez a maior alteração se perceba na facilidade de se alterar ou de se copiar (multiplicar) a informação seja ela falsa ou verdadeira, portanto a dificuldade reside na criação de mecanismos para que a informação seja autêntica e que esta possa ser verificada, voltando-se para o meio físico, uma solução largamente difundida é o uso de assinaturas, a assinatura passa representar uma parte da própria identidade da pessoa e por ela ser reconhecido e “autenticado”. A assinatura digital já não é um fato tão simples, uma vez que criar uma informação e anexar uma assinatura a esse documento digital é uma tarefa até certo ponto fácil, o estudo da criptografia entra nesse mérito, criando mecanismos e ferramentas, para dificultar a alteração, cópia ou outra forma de intervenção na informação.

2.2.5.1 Conceito

A criptografia pode ser considerada como sendo o estudo de técnicas matemáticas relacionadas aos aspectos de segurança para que se escreva através de códigos permitindo que somente o remetente e o destinatário possam identificar e decifrar o conteúdo da informação e por isso vem subsidiar a segurança da informação nas questões de confidencialidade, integridade e autenticidade da informação.

2.2.5.2 Características

A criptografia tem como característica a troca de informação com a cifragem da mesma utilizando-se de “chaves” para a composição do par emissor/receptor, com relação à característica dessa chave, podemos definir dois modelos, modelo simétrico, que utiliza uma chave única e secreta e o modelo assimétrico, que utiliza uma chave pública.

- **Modelo simétrico**

Considere um sistema de criptografia composto pela transformação de encriptação e decriptação, $\{E_e: e \in C\}$ e $\{D_d: d \in C\}$, respectivamente, onde “C” é a chave. O sistema de criptografia é chamado de simétrico, se para cada par de chaves (*e,d*), é fácil se determinar ‘*d’* conhecendo-se somente ‘*e’* e pode-se determinar ‘*e*’ conhecendo-se ‘*d*’. Na maioria dos modelos de chave simétrica *e*=*d*. Tendo como exemplo:

Supondo-se que a chave de encriptação ‘*e*’ é definida pelo deslocamento das letras do alfabeto em três posições, conforme mostrado abaixo:

E	Original	ABCDEFGHIJKLMNOPQRSTUVWXYZ
	Cifrado	DEFGHIJKLMNOPQRSTUVWXYZABC

Uma mensagem dividida em pedaços de tamanho 2

m = ES TA ME NS AG EM NA OE SE GU RA

É encriptada como

$c = E_e(m)$= HV WD PH QV DJ HP QD RH VH JX UD

O modelo de criptografia simétrico pode ser descrito conforme o diagrama de blocos da figura 5.

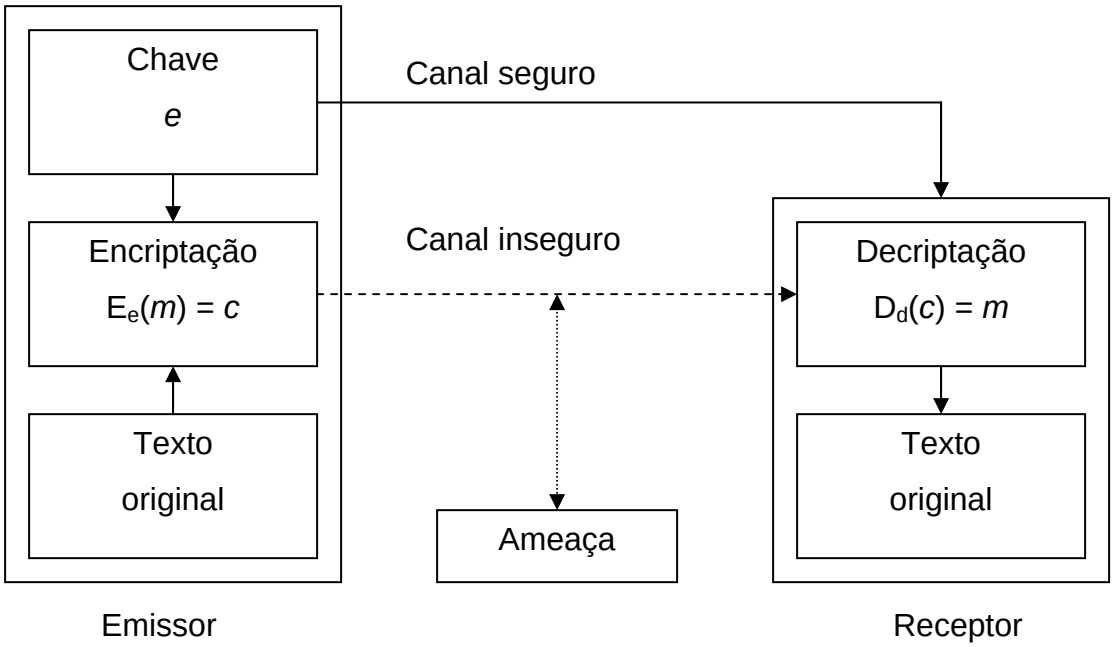


Figura 5 – Comunicação utilizando criptografia com o envio da chave por um canal seguro e a mensagem cifrada por um canal inseguro (simétrico). A chave de criptografia/decriptografia é única.

O grande problema enfrentado pelo método simétrico é a garantia de um canal seguro, ou então que os dois lados, emissor/receptor, conheçam o mecanismo de formação da chave simétrica. O método simétrico pode ser "quebrado" através de criptoanálise, utilizando-se de estatística da língua da mensagem, uma vez que o alfabeto possui características probabilísticas com relação à utilização de letras em percentuais, pode-se através de manipulações e projeções computacionais, prever a ocorrência de determinada letra, necessitando-se conhecer somente o tamanho de palavra escolhido para a cifragem, alguns mecanismos foram criados para tentar minimizar essa ocorrência e conseqüentemente, diminuir a probabilidade de ocorrência das letras e são conhecidos como: Substituição e transposição.

Substituição – cifragem por substituição é uma cifragem por blocos de tamanho previamente definidos e que sofrerão a substituição de símbolos ou grupo de símbolos, por outros símbolos ou grupo de símbolos, adiciona "*confusão*" à mensagem e pode ser subdividida em substituição simples, homofônica e polialfabética.

A cifragem por substituição simples pode ser entendida como no exemplo anterior, onde o bloco era de tamanho 2 e os conjuntos podem ser definidos da seguinte forma:

Encriptação

$E_e(m) = (e(m_1)\ e(m_2)...\ e(m_t)) = (c_1c_2...c_t) = c$

Decriptação

$D_d(c) = (d(c_1)\ d(c_2)...\ d(c_t)) = (m_1m_2...m_t) = m$

Onde t é o tamanho do bloco.

A cifragem por substituição homofônica é realizada pela troca de símbolos por valores escolhidos aleatoriamente dentro de um conjunto de combinações pré-definidas, por exemplo:

m = {a,b} e o conjunto de combinações é H(a) = {00,10} e H(b) = {01,11}, a mensagem "*m*" será transmitida por uma das seguintes formas,

m = {a,b} = {0001, 0011, 1001, 1011} = *c*

A cifragem por substituição polialfabética é a troca de símbolos dentro de blocos de tamanho definido, mas com a troca individual dos símbolos dentro do bloco, através de permutações definidas pela chave e os conjuntos são definidos da seguinte forma:

Encriptação

$E_e(m) = (p_1(m_1)\ p_2(m_2)...\ p_t(m_t))$

Decriptação

$d = (p_1^{-1}\ p_2^{-1}...\ p_t^{-1})$

e	Original	ABCDEFGHIJKLMNOPQRSTUVWXYZ
	p_1	DEFGHIJKLMNOPQRSTUVWXYZABC
	p_2	GHIJKLMNOPQRSTUVWXYZABCDEF

Do exemplo anterior temos:

m = ES TA ME NS AG EM NA OE SE GU RA

É encriptada como

$c = E_e(m)$= HY WG PK QY DM HS QG RK VK JA UG

A vantagem de cifragem polialfabética sobre a cifragem simples, é que com a mudança individual dos símbolos, a estatística da linguagem se perde uma vez que se usa de critérios de substituição diferentes para cada posição de símbolo dentro do bloco, mas uma criptoanálise pode determinar quais critérios foram utilizados, desde que se conheça o tamanho do bloco.

Transposição – cifragem por transposição é uma cifragem por blocos de tamanho previamente definidos e que sofrerão uma permutação de símbolos, mantendo-se os símbolos originais, a mensagem é dividida em blocos e esses blocos sofrem a permutação depois e são enviados para o receptor, adiciona “difusão” à mensagem e os conjuntos são definidos da seguinte forma:

Encriptação

$E_e(m) = (m_{e(1)}\ m_{e(2)}... \ m_{e(t)})$

Decriptação

$D_d(c) = (c_{d(1)}\ c_{d(2)}... \ c_{d(t)})$

A adição de *confusão* a uma mensagem está ligada à necessidade de tornar a relação entre a chave e o texto o mais complexo possível e a adição de *difusão* está ligada a necessidade de se diminuir as redundâncias dentro da mensagem, num sistema de criptografia pode-se utilizar a adição em separado de *confusão* e *difusão*, como fazer uma combinação com a adição dos dois.

- **Modelo assimétrico**

Considere um sistema de criptografia composto pela transformação de encriptação e decriptação, $\{E_e: e \in C\}$ e $\{D_d: d \in C\}$, respectivamente, onde “C” é a chave. O sistema de criptografia é chamado de assimétrico, se para cada par de chaves (*e,d*), uma chave “*e*” (chave pública) é tornada pública, ou seja, está disponível para qualquer pessoa,

enquanto a outra chave "*d*" (chave privada) é mantida secreta, de forma que tendo-se a chave pública, não existe como se descobrir a chave privada, é como se analogamente a mensagem fosse colocada dentro de uma maleta com um código de fechamento e que somente o receptor conhecesse esse código, qualquer pessoa que colocasse uma mensagem dentro da maleta, após fechá-la, seria incapaz de abri-la por desconhecer o segredo de abertura (chave privada).

O modelo de criptografia assimétrico pode ser descrito conforme o diagrama de blocos da figura 6.

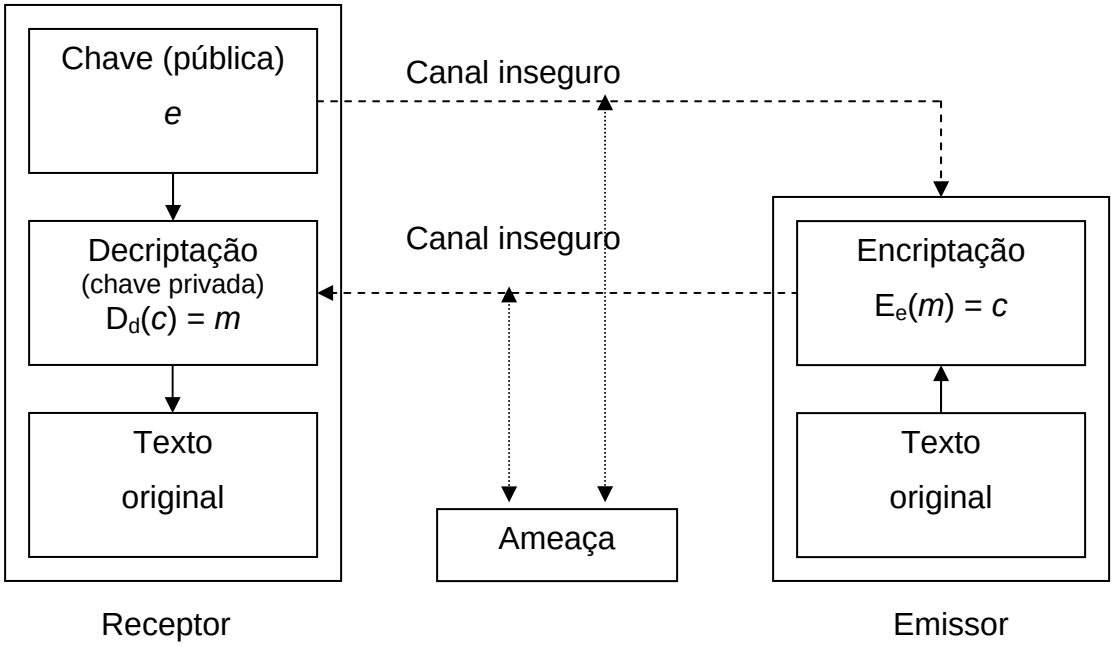


Figura 6 – Comunicação utilizando criptografia com o envio da chave pública por um canal inseguro e a mensagem cifrada por um canal inseguro, mantendo-se a chave privada somente no receptor (assimétrico). A chave pública *e* é utilizada para a criptografia do texto original pelo emissor, mantendo-se em segredo a chave privada *d* no receptor, formado o par (*d,e*) de criptografia, a mensagem é enviada e o receptor utiliza a chave privada para "abrir" a mensagem.

O modelo de criptografia assimétrica apesar de parecer sólido e seguro pode ser alvo de uma intervenção externa, por intermédio da interceptação do envio da chave pública e a modificação para uma chave falsa que é enviada ao emissor para que no momento de envio da mensagem cifrada, o interventor ou a ameaça, possa alterar o conteúdo da informação utilizando a chave falsa, e re-criptografar utilizando a chave pública interceptada e enviar a informação alterada para o receptor, como pode ser visto no diagrama de blocos da figura 7.

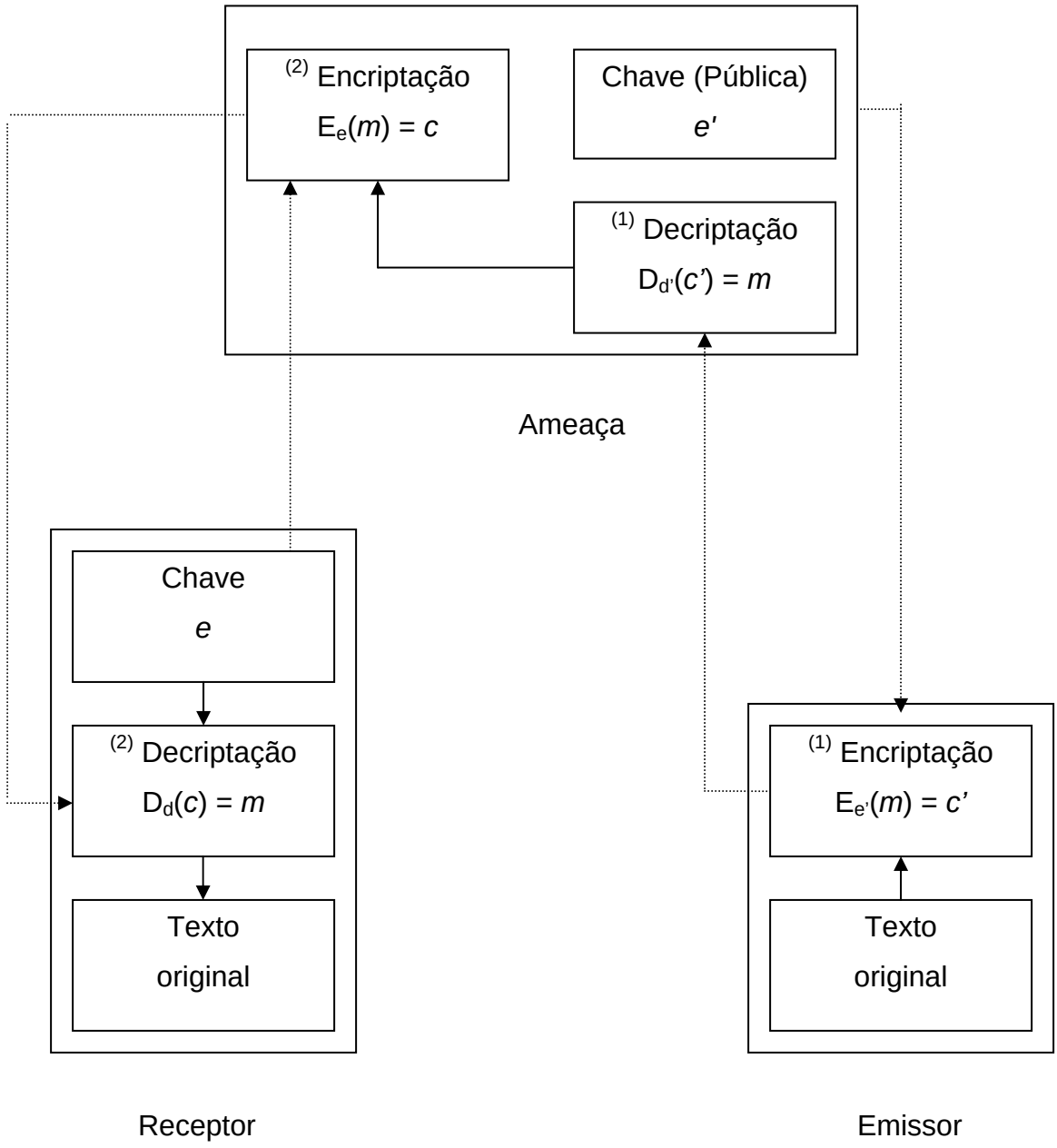


Figura 7 – Ataque utilizando-se de chave pública falsa. A ameaça envia uma chave pública (falsa) *e´* para o emissor, que interpreta como sendo a chave pública (verdadeira) do receptor e envia o texto cifrado para a ameaça [(1)] em seguida a ameaça criptografa a mensagem utilizando a chave pública *e* (verdadeira) enviada pelo receptor e envia para o receptor a mensagem alterada [(2)].

Resumidamente a comparação entre os dois modelos, simétrico e assimétrico, é feita com relação à forma de troca das chaves, enquanto no modelo simétrico a chave é única e mantida em segredo, no modelo assimétrico o par de chaves mantém uma chave privada e a outra pública, e o modelo simétrico requer a utilização de um canal seguro para a troca das chaves que bem poderia ser utilizado para a troca da informação uma vez que o canal é seguro, enquanto que no modelo assimétrico não há a necessidade de um

canal seguro, utilizando da citação popular “um segredo só é segredo enquanto somente uma pessoa o sabe!”, o mesmo se aplicaria à chave privada, mas fica vulnerável pela presença de uma chave falsa dentro do sistema, o que irá determinar que modelo utilizar é exatamente a aplicação.

2.2.5.3 Keeloq e AES

O sistema de criptografia utilizado no modelo proposto é definido em duas etapas, a primeira referente à autenticação propriamente dita e a segunda referente a comunicação entre os módulos (a ser implementada como uma continuação deste trabalho), para tanto segue um descritivo sobre cada um dos métodos criptográficos a serem utilizados.

- **KeeLoq - IFF**

Para entendermos o método de criptografia de propriedade da empresa Microchip, fabricante do microcontrolador e transcoder utilizados no projeto, é necessário um entendimento sobre o sistema chamado de IFF – Identify Friend or Foe (Identificação de amigo ou inimigo), que foi utilizado durante as guerras como um método de identificação dos aviões que se aproximavam como “amigos” ou “inimigos” e o conceito principal é baseado no uso de fórmulas ao invés de valores fixos no processo de autenticação. Nesse processo simples, o módulo central envia para o módulo de autenticação um número grande escolhido randomicamente, também conhecido como “desafio”. O módulo de autenticação utiliza uma fórmula convenientemente definida para calcular uma “resposta” que é então enviada de volta para o módulo central. O módulo central compara a “resposta” recebida com o valor calculado localmente, se a fórmula utilizada pelo módulo de autenticação for a correta, a comparação será exata sendo o módulo considerado autêntico ou compatível (figura 8).

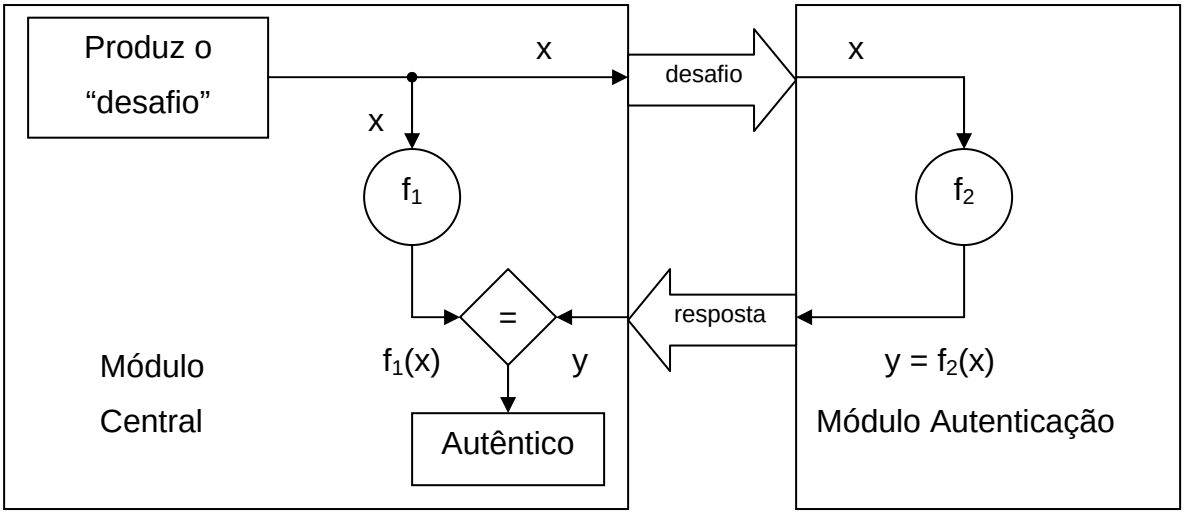


Figura 8 – Identify Friend or Foe

Pode-se perceber que nesse tipo de transmissão, não há o envio da fórmula ou de chaves criptográficas e o sistema pode ser repetido diversas vezes para que se possa ter uma maior consistência, uma vez que a cada nova transmissão um novo "desafio" é gerado randomicamente, qualquer intervenção externa resultaria em diferentes "desafios".

O sistema KeeLoq – IFF parte da base do IFF com o acréscimo de criptografia simétrica (patenteada) no módulo central e no módulo de autenticação. O KeeLoq é uma ferramenta de criptografia robusta, com chave criptográfica de 64 bits (cifragem por bloco) operando sobre blocos de dados com 32 bits (desafio/resposta). O método pode ser ilustrado através da seqüência:

1. Um desafio (x) de 32 bits é gerado pelo módulo central, randomicamente;
2. O desafio é enviado para o módulo de autenticação;
3. O módulo de autenticação, que também possui o algoritmo de criptografia KeeLoq, efetua a criptografia do desafio e retorna a resposta de 32bits;
4. O módulo central decriptografa a resposta e gera um novo valor de 32bits (x´);
5. O novo valor (x´) é comparado ao valor inicial do desafio (x);
6. Se os dois valores forem iguais, o módulo é então autenticado e pode dar seguimento aos processamentos.

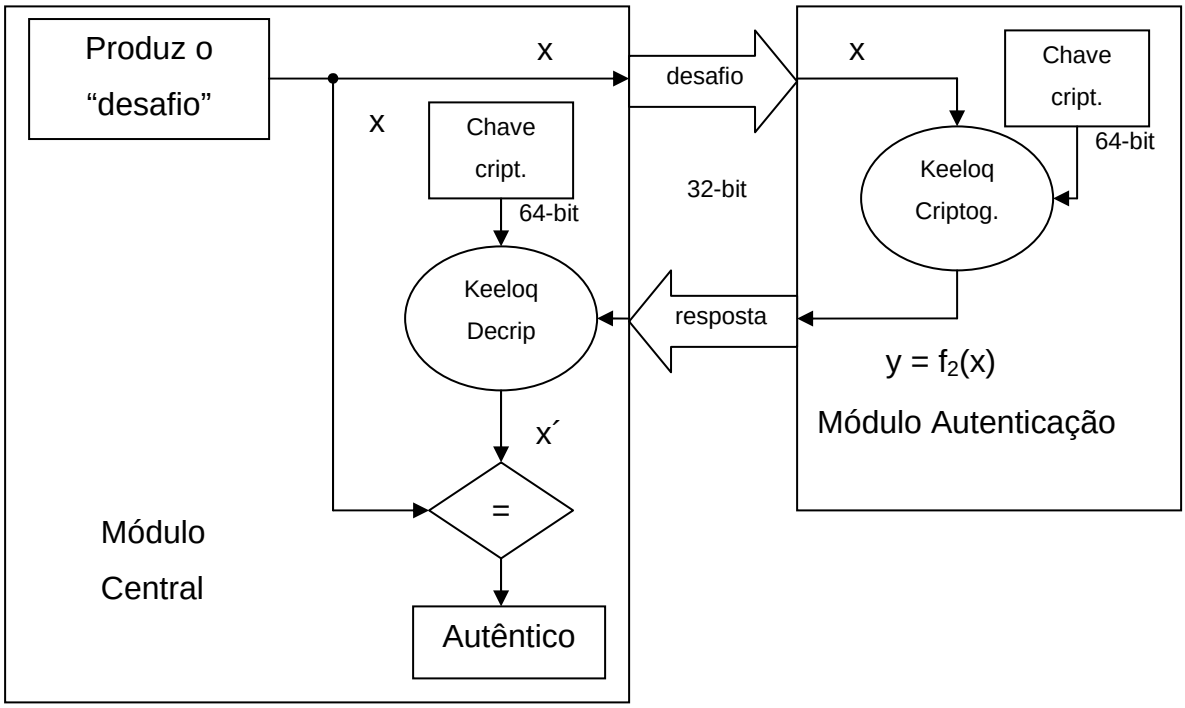


Figura 9 – KeeLoq - IFF

Se o módulo de autenticação não for compatível, a probabilidade de se descobrir a resposta (y) do desafio (x) é de $1/2^{32}$ ou em outras palavras uma em quatro bilhões. O processo pode ser repetido para que se tenha uma maior segurança e consistência.

- **Advanced Encryption Standard - AES**

Em 12 de setembro de 1997, o NIST – National Institute of Standards and Technology (Instituto Nacional de Padrões e Tecnologia), publicou um concurso com o intuito de selecionar um algoritmo para cifradores de bloco simétrico para as três primeiras décadas do século 21 e que seria o substituto do então DES – Data Encryption Standard. De todos os trabalho apresentados, restaram para a segunda fase somente 5: MARS, RC6, Rijndael, twofish e Serpent. [5]

Os algoritmos foram testados com relação à segurança e velocidade de execução, de abril a maio de 2000 o NIST fez uma ampla pesquisa com os conferencistas participantes das rodadas AES e dentre todos os candidatos, o preferido foi o Rijndael. Em outubro de 2000 o cifrador Rijndael passou a ser chamado de AES. Os criadores do Rijndael são Vincent Rijmen e Joan Daemen da Universidade Católica de Leuven na

Bélgica. O fato do Rijndael ser pequeno e rápido, mesmo com uma chave com o tamanho de 128 bits, permite o seu uso em aplicações aonde a segurança podia ser a comparada ao desejado pelo DES, mas aonde os recursos computacionais são muito limitados como celulares, smartcards e microcontroladores.

O AES é um cifrador de bloco com tamanho de bloco e chave variáveis entre 128, 192 e 256 bits. Isto significa que pode-se ter tamanho de blocos com tamanhos de chaves diferentes. A quantidade de rodadas necessárias para cifrar/decifrar é função do tamanho de bloco e chaves. O processo para cifrar e decifrar pode ser visualizado de uma maneira genérica na figura 10.

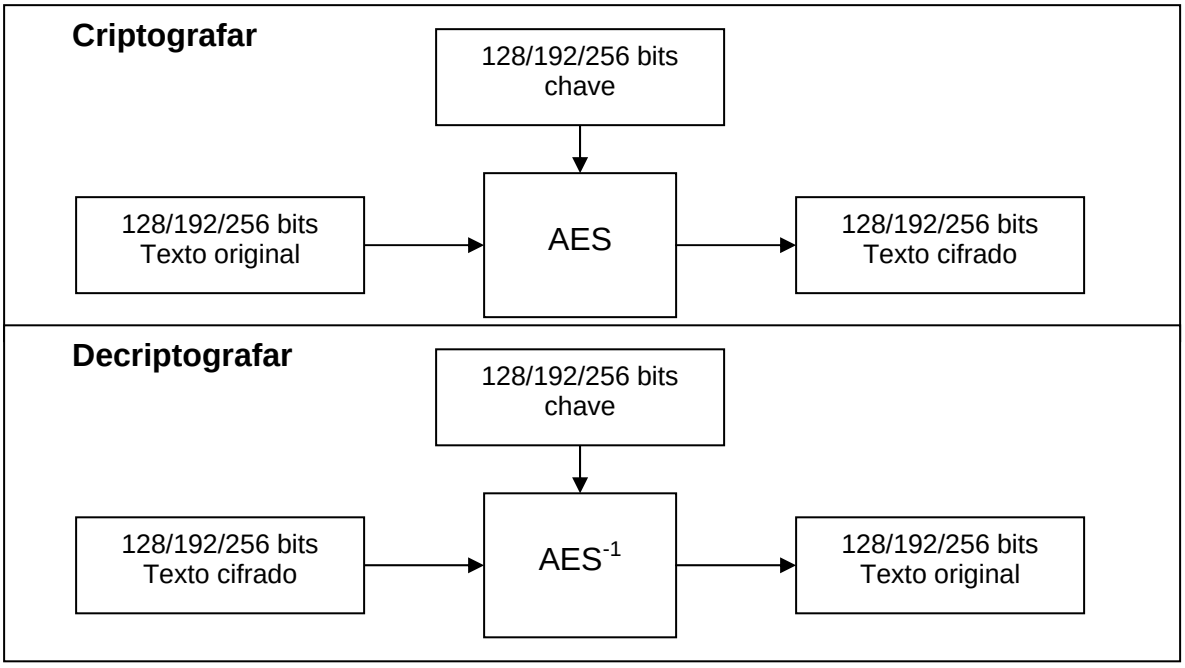


Figura 10 – AES simplificado

O AES opera com um determinado número de blocos de 32 bits que são ordenados em colunas de 4 bytes, as quais são chamadas de números de blocos *Nb*. Os valores de *Nb* possíveis são de 4, 6 e 8 equivalentes a blocos de 128, 192 e 256 bits (Nb x 32). A chave é agrupada da mesma forma em colunas que o bloco de dados, e é chamada de números de chaves, com a sigla *Nc*. Na tabela 2 tem-se um exemplo para *Nb*=4 e *Nc*=4.

Tabela 2 – Blocos de dados ($d_{x,x}$) e bloco de chaves ($c_{x,x}$).

Dado [0]	dado [4]	Dado [8]	dado [12]
Dado [1]	dado [5]	Dado [9]	dado [13]
Dado [2]	dado [6]	Dado [10]	dado [14]
Dado [3]	dado [7]	Dado [11]	dado [15]

chave[0]	chave[4]	chave[8]	chave[12]
chave[1]	chave[5]	chave[9]	chave[13]
chave[2]	chave[6]	chave[10]	chave[14]
chave[3]	chave[7]	chave[11]	chave[15]

Com base nos valores que *Nb* e *Nc* podem assumir é que será determinada a quantidade de rodadas a serem executadas, identificadas pela sigla *Nr* – número de rodadas (tabela 3).

Tabela 3 – Número de rodadas por tamanho de chave e de blocos.

No de rodadas (Nr)	Bloco de 16bytes	Bloco de 24bytes	Bloco de 32bytes
Chave de 16bytes	10	12	14
Chave de 24bytes	12	12	14
Chave de 32bytes	14	14	14

Em cada rodada o bloco de dados sofre as seguintes alterações:

1. **Byte Sub** – Os bytes de cada bloco de dados são substituídos por seu equivalente em uma tabela de substituição (tabela-s) que não é linear;
2. **Shift row (deslocamento de linha)** – Os bytes são rotacionados em grupos de quatro bytes;
3. **Mix Column** – O grupo de quatro bytes sofre uma multiplicação modular;
4. **Add round Key (Adição da chave da rodada)** – Aplica-se a subchave da rodada ao grupo de dados através de uma operação de XOR (OU exclusivo), a chave deve possuir o mesmo tamanho do bloco.

- **Transformação byte-sub**

A transformação byte-sub utiliza a substituição de bytes através de uma tabela-S (tabela de substituição) guardada na memória do microcontrolador, obtendo-se o novo valor através do endereço do valor a ser alterado, um exemplo de tabela-S pode ser visualizado na tabela 4, a volta pode ser obtida através da mesma tabela-S em sentido contrário, ou seja, para um determinado endereço inicial o seu equivalente seria igual a 255-D (onde D seria o valor em Hexadecimal do caracter).

Tabela 4 – Tabela-S em hexadecimal

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C6	37	87	47	DF	46	6	AC	F3	E0	86	42	1F	8D	4A	97
1	5C	D8	6C	27	5F	65	84	FF	2A	BD	DA	A	39	BA	D7	FC
2	8B	2F	C9	92	93	3	8F	3C	B3	AA	AE	EF	E7	7D	E3	A1
3	B0	8C	C2	CC	71	99	A0	59	80	D1	F8	DE	4E	82	DB	A7
4	60	C8	32	51	41	16	55	FA	D5	43	9D	CB	62	CE	2	B8
5	C5	ED	F0	2E	F2	3F	EB	45	56	4C	1B	63	54	34	75	C
6	FD	E	5A	4F	C4	24	C3	A8	A4	6F	D0	7	F5	33	9	7A
7	E5	CA	F4	8	D9	29	73	AF	3B	9B	5D	E2	F1	F	CF	DD
8	2C	30	C1	3E	5	89	B4	81	BC	8A	17	23	B6	25	61	C7
9	F6	E8	4	3D	D2	52	F9	78	94	1E	7B	B1	1D	15	40	4D
A	FE	D3	53	50	64	90	B2	35	DC	CD	3A	D6	E9	A9	BE	67
B	8E	7C	83	26	28	AD	14	6A	36	95	BF	5E	A6	57	1A	70
C	5B	77	A2	12	31	9A	BB	9C	7E	2D	B7	1	44	2B	48	58
D	F7	13	AB	96	74	C0	9F	10	E6	A3	85	6B	98	EC	21	19
E	EE	7F	79	E1	66	6D	18	B9	49	11	88	6E	1C	A5	72	D
F	38	EA	68	20	B	9E	D4	76	E4	69	22	0	FB	B5	4B	91

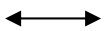
- **Transformação Deslocamento de linha (Shift row)**

A transformação shift row utiliza o rotacionamento do bloco de dados de acordo com uma tabela de rotação. A primeira linha do bloco de dados não sofre alteração e as linhas seguintes são rotacionadas à esquerda com a alteração dos bytes, um exemplo de tabela de rotação pode ser visto na tabela 5.

Tabela 5 – Shift row para uma configuração de 3 linhas (C1 a C3) e Nb=4

Nb	**C_1**	**C_2**	**C_3**
4	1	2	3
6	1	2	3
8	1	3	4

Dado [0]	dado [4]	dado [8]	dado [12]
Dado [1]	dado [5]	dado [9]	dado [13]
Dado [2]	dado [6]	dado [10]	dado [14]
Dado [3]	dado [7]	dado [11]	dado [15]



dado [0]	dado [4]	dado [8]	dado [12]
dado [13]	dado [1]	dado [5]	dado [9]
dado [10]	dado [14]	dado [2]	dado [6]
dado [7]	dado [11]	dado [15]	dado [3]

A operação inversa é efetuada da mesma forma com a utilização de rotação à direita.

- **Transformação MixColumn**

Na transformação de MixColumn as colunas dos bytes são consideradas como polinômios de grau 8 (1byte = 8bits) e é feita uma multiplicação modular (x4 + 1) com um polinômio fixo e reversível dado por:

p(x) = *03h*x^3 + *01h*x^2 + *01h*x + *02h;*

A multiplicação é equivalente a uma rotação cíclica dentro do vetor multiplicado (coluna) e pode ser definida como uma matriz de multiplicação do tipo:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02h & 03h & 01h & 01h \\ 01h & 02h & 03h & 01h \\ 01h & 01h & 02h & 03h \\ 03h & 01h & 01h & 02h \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

O polinômio escolhido é reversível o que torna a operação inversa simples, bastando para tanto multiplicar as colunas pelo polinômio:

p´(x) = *0Bh*x^3 + *0Dh*x^2 + *09h*x + *0Eh*

O que pode ser efetuado através da matriz de multiplicação inversa que segue:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0Eh & 0Bh & 0Dh & 09h \\ 09h & 0Eh & 0Bh & 0Dh \\ 0Dh & 09h & 0Eh & 0Bh \\ 0Bh & 0Dh & 09h & 0Eh \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

- **Adição da chave da rodada (Add round Key)**

A transformação de adição da chave da rodada, nada mais é do que a operação de OU exclusivo (XOR) byte a bytes com a sub-chave da rodada, para isso o tamanho da chave da rodada deve ser o mesmo do tamanho definido para o bloco de dados.

3. Modelo Proposto

Como o objetivo principal deste trabalho é aumentar a segurança na urna eletrônica com a utilização de cartão, é necessário integrar as disciplinas aprendidas no curso de engenharia de computação, sendo divididas entre a engenharia elétrica/eletrônica e a ciência da computação (figura 11), com ênfase em engenharia eletrônica e programação com a utilização de linguagem de baixo nível, utilizando-se de conteúdo adquirido em disciplinas como microcontroladores, linguagem técnica de programação e banco de dados.

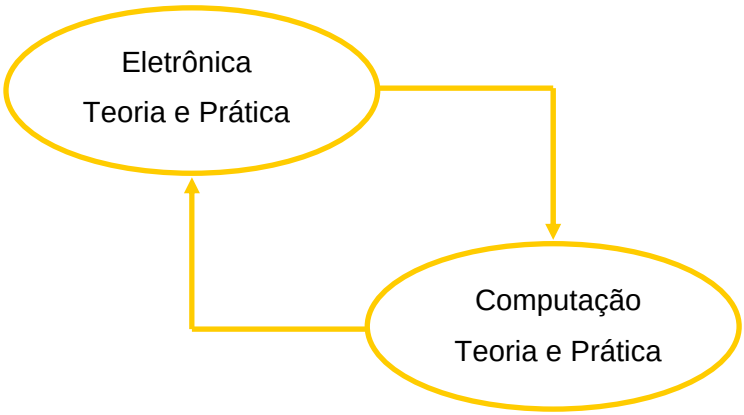


Figura 11 – Engenharia de computação

Como este trabalho se limita a confidencialidade da informação na urna eletrônica, ou seja, autenticação do eleitor sem a correspondência do voto, o mesmo ficou restrito à seção eleitoral, sendo determinado o fluxo da informação conforme consta da figura 12.

Figura 12 – Fluxo da votação eletrônica

O eleitor ao entrar na seção de votação, se identifica junto ao mesário utilizando o cartão do eleitor com foto, após a identificação visual pelo mesário, o eleitor se encaminha para a urna e insere o cartão do eleitor no leitor indicado na urna, o módulo de autenticação da urna confere os dados do cartão do eleitor com os dados contidos no módulo banco de dados da urna e valida o cartão para a votação, liberando o módulo de votação (figura 13).

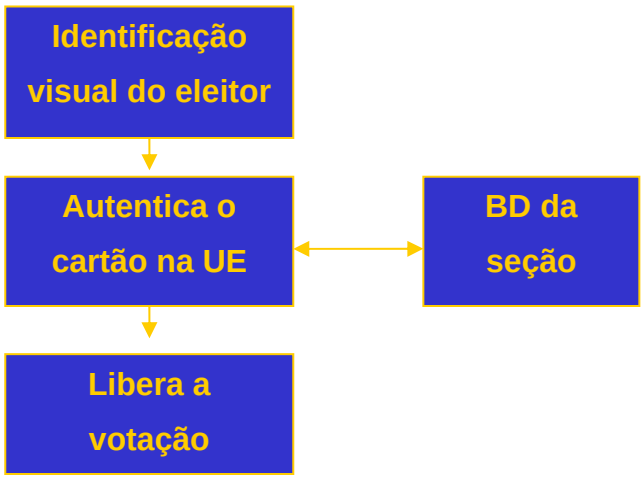


Figura 13 – Fluxo da votação eletrônica específico

A autenticação do cartão na urna é feita através do modelo criptográfico KeeLoq, patenteado pelo fabricante dos microcontroladores PIC - Microchip, a serem utilizados nesse projeto e após a autenticação, as informações constantes do cartão são acessadas e o número da inscrição, juntamente com a zona eleitoral, são criptografados com o modelo AES e a chave pré-definida e então pesquisados no banco de dados criptografado da seção. A escolha do número de inscrição com a zona eleitoral se deve ao fato de ser do formato numérico e de tamanhos 12bytes e 4 bytes, respectivamente, o que gera um campo de 16bytes, que não por coincidência é o tamanho de bloco a ser utilizado para a criptografia AES. A tabela-S e a chave criptográfica ficarão “escondidas” no código da aplicação, para uma maior segurança e dificuldade de acesso, uma vez que os microcontroladores da Microchip possuem uma segurança contra a leitura do código fonte, após ter sido gravado no microcontrolador.

O cartão do eleitor contém os dados do eleitor de forma aberta (pública), uma vez que o modelo criptográfico KeeLoq utiliza-se de dados como o número de série do transcoder e do microcontrolador, e outras informações que podem ser introduzidas em

tempo de programação, as informações referentes ao eleitor podem ficar “expostas”, desde que não se conheçam as demais informações.

3.1 Hardware e Software

A utilização do microcontrolador, além de atingir o objetivo desse trabalho (aumentar a segurança na urna eletrônica), também simplifica o modelo. Se considerarmos o processo de votação como um acumulador de informações pré-definidas, a função da urna fica restrita à acumulação dessas informações, sem referência a hora, eleitor ou outro dado além do número escolhido e que deve ser guardado para uma posterior totalização. Partindo-se deste princípio ocorre uma minimização das funcionalidades excedentes, aumentando a segurança, com a diminuição de portas de possíveis ataques e/ou vulnerabilidades, para tanto o sistema modular com dupla criptografia visa dificultar a ação de agentes externos ao processo e durante o processo. Utilizando-se de microcontroladores de baixo custo e segurança criptográfica o projeto pode ser facilmente ampliado ou remodelado, a programação segue a linha de execução com desvios devido a interrupções ou mesmo por desvios de espera, utilizando-se ferramentas de programação na linguagem de máquina “Assembler”, com a possibilidade de utilização de rotinas externas em C++.

O modelo proposto pode ser descrito pela figura 14, que segue:

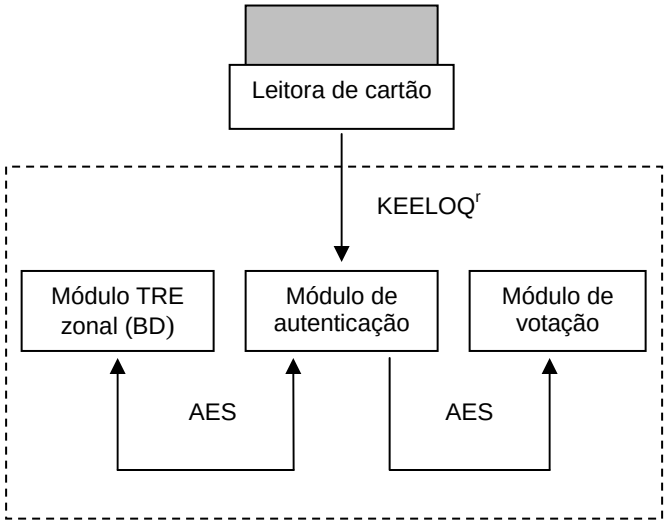


Figura 14 – diagrama do modelo proposto

4. Implementação

Dentro da filosofia adotada neste trabalho, a implementação tem como objetivo a integração das disciplinas engenharia elétrica/eletrônica e a ciência da computação, utilizando microcontroladores e elementos ativos e passivos de eletrônica, bem como a programação em linguagem de máquina “Assembler”

4.1 Hardware e Software

Para o hardware foi escolhido o microcontrolador PIC16F636 do fabricante Microchip, por ter uma relação custo/benefício muito boa, é um microcontrolador de 8 bits, com clock de 20Mhz, arquitetura RISC com um set de 35 instruções, memória “flash” de programa com 2048 palavras, memória de dados SRAM com 128 bytes e EEPROM com 256 bytes, 12 portas de entrada e saída e 2 comparadores, possui um módulo, em hardware, de criptografia, compatível com a criptografia KeeLoq utilizada no cartão do eleitor, transcoder HCS410.

Os microcontroladores PIC possuem uma estrutura de máquina interna do tipo Harvard, o que o diferencia da maioria dos microcontroladores de mercado, que utilizam uma arquitetura do tipo Von-Neumann. A diferença entre estas duas arquiteturas está na forma de processamento dos dados e do programa pelo microcontrolador. Na arquitetura tradicional, Von-Neumann, o barramento interno (bus) de dados e instruções é único e é geralmente de 8 bits. A arquitetura Harvard, possui dois barramentos internos, um para os dados e o outro para as instruções. Nos microcontroladores PIC, o barramento de dados é sempre de 8 bits, enquanto que o barramento de instruções varia de 12, 14 ou 16 bits dependendo do microcontrolador (neste projeto o PIC16F636 possui barramento de instrução de 14 bits). Este tipo de arquitetura permite que enquanto uma instrução esteja sendo executada, outra instrução possa ser “buscada” da memória, melhorando o desempenho do processamento, além disso como o barramento de instruções é maior que o barramento de dados, 14 e 8 bits, respectivamente, o operador da instrução já inclui o dado e o local onde a instrução irá operar (se for necessário), o que significa que apenas uma posição de memória é utilizada por instrução, economizando-se endereços de memória de programa.

Como o operador da instrução não reserva muito espaço para o código da instrução propriamente dito, os microcontroladores utilizam a tecnologia RISC (Reduced Instruction Set Computer – computador com set de instruções reduzido), que possui apenas 35 instruções, ao contrário dos microcontroladores com a tecnologia CISC (Complex Instruction Set Computer – computador com set de instruções complexo) que possuem mais de cem instruções.

Para o cartão do eleitor, este projeto utiliza o transcoder HCS410, que é um codificador/decodificador com a tecnologia Keeloq de criptografia, na configuração de Token com a conexão de 2 fios, dados e alimentação, conforme figura 15.

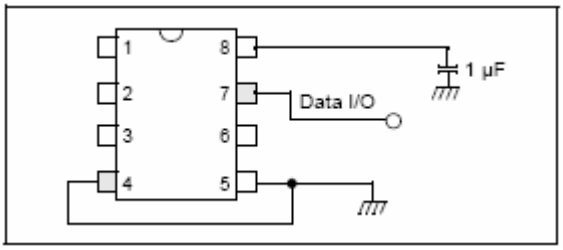


Figura 15 – modelo de conexão 2 fios (Token)

A função do HCS410 é de guardar os dados do eleitor e comunicar com o módulo de autenticação (PIC16F636) utilizando a autenticação com criptografia KeeLoq.

Para o desenvolvimento da implementação, é utilizado um protoboard para a montagem do Hardware e a programação é feita de forma serial com um programador compatível com o PICStart Plus. A conexão com o protoboard é feita através de um cabo serial conectado ao computador de desenvolvimento que executa o programa MPLAB IDE v6.60 do fabricante Microchip, disponibilizado no site do fabricante (www.microchip.com). Como o microcontrolador PIC16F636, permite a gravação “in-circuit”, ou seja, é permitido gravar o programa diretamente no microcontrolador, com este montado no circuito, bastando para isso tomar alguns cuidados com os elementos do circuito que estejam em contato com as portas utilizadas na gravação. A gravação é feita de forma serial com a utilização de 5 (cinco) pinos a seguir:

- Vdd – Alimentação de 5Vcc;
- Vss – Ground (GND);
- MCLR – Tensão de programação, o microcontrolador necessita de uma tensão de 13Vcc neste pino para entrar em modo de programação, por isso a necessidade

de se proteger os demais elementos do circuito que estejam em contato com este pino, contra possíveis sobre-tensões e queimas;

- RB6 – Clock da comunicação serial, controlado pelo gravador;
- RB7 – Dados na comunicação serial, que podem ser pelo gravador (escrita) ou pelo PIC (leitura).

Como o modelo proposto é referente a autenticação do eleitor, a seqüência lógica é descrita a seguir:

PIC16F636

- Recebe os dados do HCS410;
- Criptografa a informação;
- Busca na memória (criptografada) os dados do eleitor;
- Autentica ou não o eleitor na seção.

Baseado na seqüência lógica descrita acima se pode determinar o seguinte fluxo de programação:

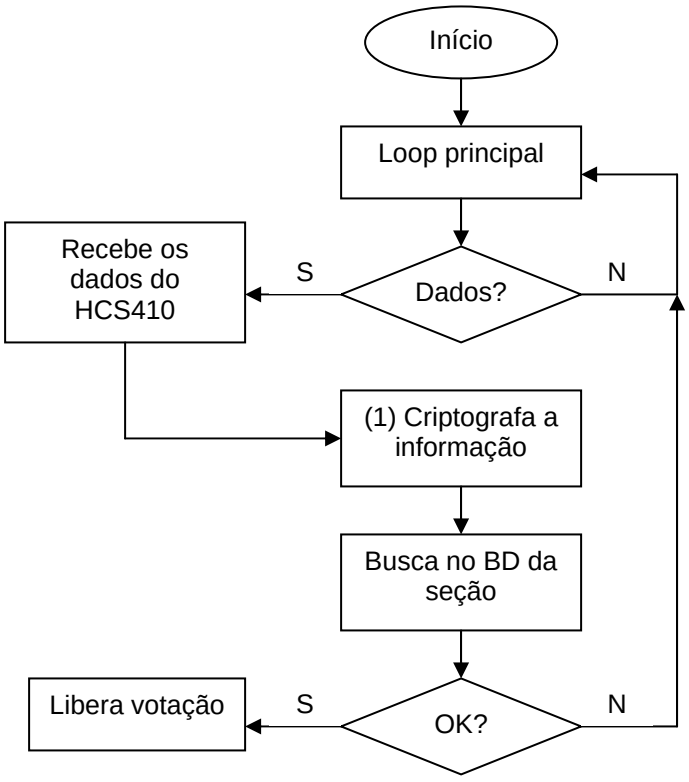


Figura 16 - Fluxo de programação principal

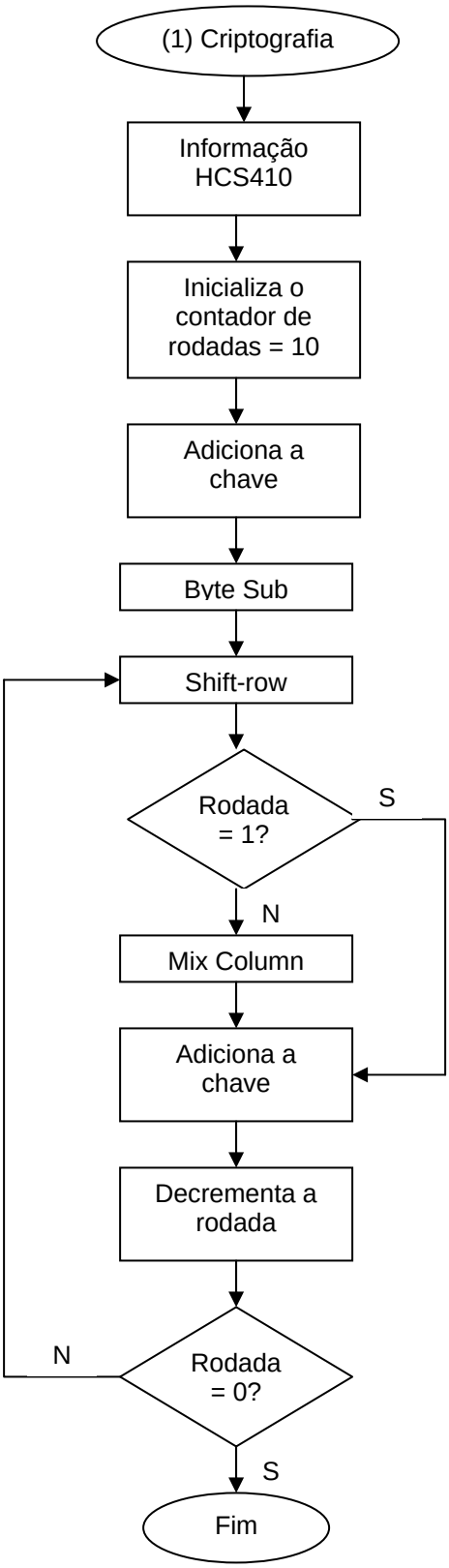


Figura 17 - Fluxo de criptografia AES

A programação é feita em linguagem de máquina (Assembler) e compilada pelo programa MPLAB IDE v6.60 do fabricante Microchip, sendo depois enviado para o microcontrolador através do gravador serial.

Para facilitar a visualização do resultado, foi incluído um display de cristal líquido para a comunicação entre o programa e o eleitor, sendo mostrado o dado coletado no HCS410 em formato original e o dado criptografado, seguido da informação de liberação ou não para a votação (figura 18).

1	0	0	8	4	7	2	3	5	5	0	6	1	4	2	9
G	D	M	H	A	E	J	Y	W	X	H	M	Q	L	K	B

Figura 18 – Display de saída

Algumas considerações são efetivadas com relação à programação do microcontrolador, ao detectar os sinais de ACK “de acordo” do módulo HCS410, o microcontrolador inicia a comunicação com o módulo utilizando-se dos registradores de criptografia KeeLoq CRDAT [1] e recebe as informações para serem guardadas em um vetor que será criptografado em modo AES e então buscado no BD contido na memória EEPROM de dados. Por ser uma linguagem de baixo nível o controle sobre essas informações tem de ser feito bit a bit, como por exemplo na rotina de adição da chave criptográfica como segue:

```
; ************************************ ADICAO_CHAVE ***********************************************
; ESSA ROTINA FAZ: BLOCO ^= CHAVE ( ^= SIGNIFICA BLOCO = BLOCO XOR CHAVE )
; ENTRADA:
; SAIDA..:
; GLOBAL.:        BLOCO, CHAVE
; OBS....:
; *************************************************************************************************************

ADICAO_CHAVE:

  MOVF  CHAVE+0x0,W                ; BLOCO[0] ^= CHAVE[0];
  XORWF         BLOCO+0x0,F

  MOVF  CHAVE+0x1,W                ; BLOCO[1] ^= CHAVE[1];
  XORWF         BLOCO+0x1,F
```

[1] É necessário a autorização expressa do fabricante Microchip para a descrição desses registradores e a sua utilização, através do “*KeeLoq Encoder License Agreement*”.

```
MOVF  CHAVE+0x2,W          ; BLOCO[2] ^= CHAVE[2];
XORWF       BLOCO+0x2,F


MOVF  CHAVE+0x3,W          ; BLOCO[3] ^= CHAVE[3];
XORWF       BLOCO+0x3,F

MOVF  CHAVE+0x4,W          ; BLOCO[4] ^= CHAVE[4];
XORWF       BLOCO+0x4,F

MOVF  CHAVE+0x5,W          ; BLOCO[5] ^= CHAVE[5];
XORWF       BLOCO+0x5,F

MOVF  CHAVE+0x6,W          ; BLOCO[6] ^= CHAVE[6];
XORWF       BLOCO+0x6,F

MOVF  CHAVE+0x7,W          ; BLOCO[7] ^= CHAVE[7];
XORWF       BLOCO+0x7,F

MOVF  CHAVE+0x8,W          ; BLOCO[8] ^= CHAVE[8];
XORWF       BLOCO+0x8,F

MOVF  CHAVE+0x9,W          ; BLOCO[9] ^= CHAVE[9];
XORWF       BLOCO+0x9,F

MOVF  CHAVE+0x0A,W         ; BLOCO[10] ^= CHAVE[10];
XORWF       BLOCO+0x0A,F

MOVF  CHAVE+0x0B,W         ; BLOCO[11] ^= CHAVE[11];
XORWF       BLOCO+0x0B,F

MOVF  CHAVE+0x0C,W         ; BLOCO[12] ^= CHAVE[12];
XORWF       BLOCO+0x0C,F

MOVF  CHAVE+0x0D,W         ; BLOCO[13] ^= CHAVE[13];
XORWF       BLOCO+0x0D,F

MOVF  CHAVE+0x0E,W         ; BLOCO[14] ^= CHAVE[14];
XORWF       BLOCO+0x0E,F

MOVF  CHAVE+0x0F,W         ; BLOCO[15] ^= CHAVE[15];
XORWF       BLOCO+0x0F,F

RETURN
```

Mas ao mesmo tempo esse controle permite que se conheça o estado atual de cada bit, o que pode ser útil em uma “procura” por problemas de execução.

Com relação à leitura dos dados gravados na EEPROM, uma facilidade implementada é a não necessidade de uma rotina de gravação de dados, uma vez que os mesmos estão previamente gravados. No caso da implementação os dados estão armazenados na própria memória EEPROM do microcontrolador, mas caso seja necessário, os dados podem ser armazenados em uma memória externa, sendo preciso criar modelos de segurança para o acesso a esses dados.

Outros controles importantes a serem considerados em toda implementação que utilize microcontroladores, são os controles de tempo ou *delay*, a comunicação dos microcontroladores com outros periféricos (memórias, Displays, etc) devem ser temporizados, por exemplo, na comunicação como display de cristal líquido (LCD), o microcontrolador interno do display, necessita de um delay de pelo menos 15ms para que a tensão de alimentação do display atinja o pico de 4,5V.

```
;*******************************************************************************
; Rotina de criação de delays para a comunicação como LCD
;
; Delay_tempo    = ((DELAY_valor * 3) + 4) * tempo_ciclo
; DELAY_valor    = (Delay_tempo - (4 * Cycle_tempo)) / (3 * tempo_ciclo)
;
; Delay_tempo    = ((32 * 3) + 4) * 1uSeg
;                = 100uSeg
; DELAY_valor    = (500uSeg - 4) / 3
;                = 165.33
;                = 165
;*******************************************************************************
DELAY500            MOVLW       D'165'          ; +1            1 cycle
                    MOVWF       DELAY           ; +2            1 cycle
DELAY500_LOOP       DECFSZ      DELAY, F        ; passo 1       1 cycle
                    GOTO  DELAY500_LOOP         ; passo 2       2 cycles
DELAY500_FIM        RETURN                      ; +3            2 cycles
;
;
X_DELAY500          MOVWF       X_DELAY         ; +1            1 cycle
X_DELAY500_LOOP     CALL   DELAY500             ; passo1        espera 500uSeg
                    DECFSZ      X_DELAY, F      ; passo2        1 cycle
                    GOTO  X_DELAY500_LOOP       ; passo3        2 cycles
X_DELAY500_FIM      RETURN                      ; +2            2 cycles
```

A programação é baseada no fluxograma descrito acima (figura 16) e tem a seqüência de execução linear, ficando “preso” em um loop infinito esperando o sinal de dados, na implementação, o acionamento do botão conforme figura 19.

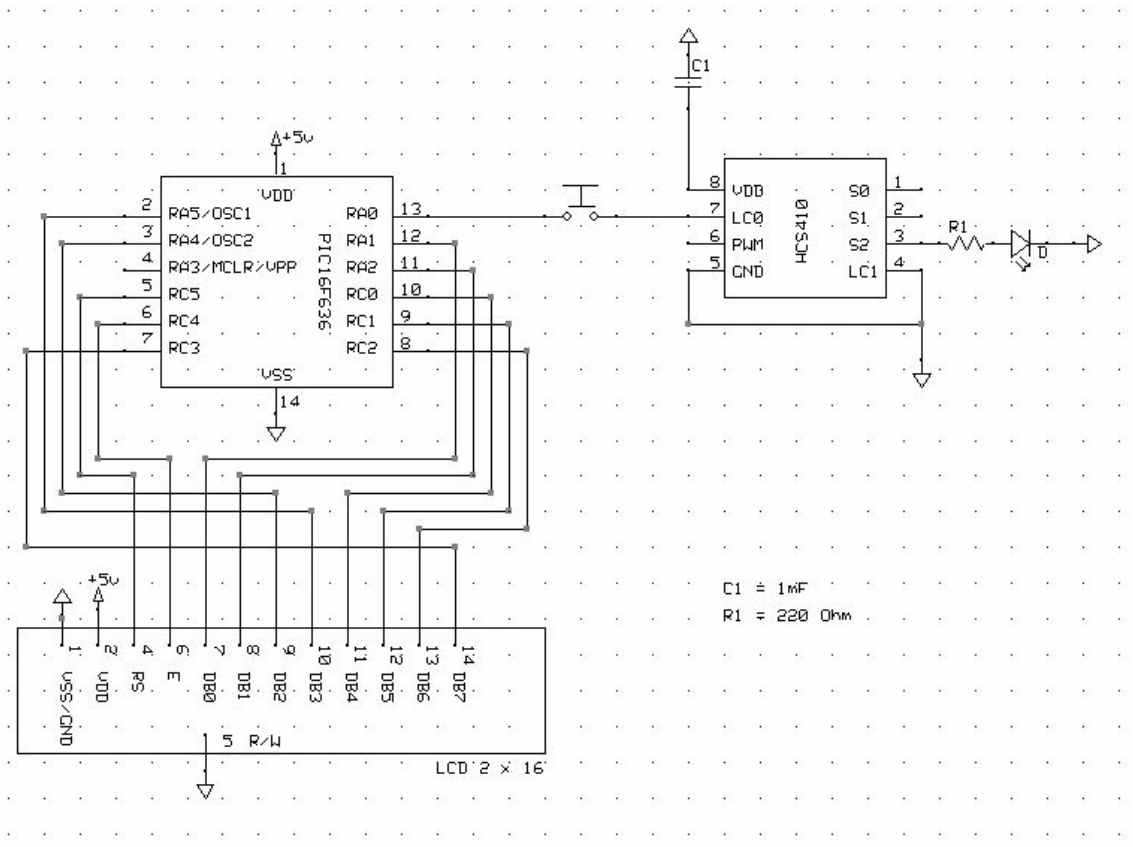


Figura 19 – Esquema eletrônico do modelo proposto.

5. Conclusão

A segurança da informação nos tempos atuais é uma disciplina que está sendo detalhada à exaustão, com diversas pesquisas e trabalhos sendo desenvolvidos. A necessidade de se desenvolver mecanismos que venham a dificultar “ataques” ou o uso indevido de informação por terceiros, tem gerado soluções das mais complexas às mais simples, tendo sido foco deste trabalho o aumento da segurança na urna eletrônica.

Com a utilização de microcontroladores o modelo proposto permitiu a utilização de conceitos de engenharia eletrônica (hardware) e de conceitos da ciência da computação (software). A simplicidade da arquitetura do microcontrolador aliada a sua disponibilidade de implementação de segurança através de modelos criptográficos, o torna uma ferramenta bastante potente e ao mesmo tempo eficiente com relação ao modelo proposto, aumentando-se o grau de dificuldade de intervenção externa e diminuindo-se as prováveis vulnerabilidades do sistema como um todo.

As dificuldades encontradas no desenvolvimento deste projeto estiveram em grande parte relacionadas à programação em linguagem de máquina (912 linhas de código), por ser essa uma linguagem de baixo nível e por ter o controle bit a bit do microcontrolador, envolvendo uma parte substancial de tempo para a implementação. Com a utilização das ferramentas disponíveis no software de desenvolvimento MPLAB, foi possível efetuar “investigações” em determinados momentos da programação para se perceber o estado de determinados bits e de algumas variáveis, o que despendeu muito tempo mas permitiu um controle maior do fluxo da programação.

Como sugestão de trabalhos futuros, pode-se considerar a implementação do módulo de votação e/ou o desenvolvimento de aplicativo de banco de dados para a apuração e totalização dos votos, bem como o aplicativo de configuração dos microcontroladores para a distribuição nos TRE´s.

Concluindo, a solução apresentada tem como característica principal o aumento da segurança da urna eletrônica agregando também a simplificação de hardware, o que permite um controle maior sobre as possibilidades de intervenção com um número menor de portas de acesso e uma vez gravada a informação no microcontrolador, este pode operar em modo de proteção de código (code-protection) que não permite o acesso às informações contidas no microcontrolador, sejam estas, dados gravados na EEPROM ou o programa propriamente dito.

6. Referências Bibliográficas

[1]http://www.tse.gov.br – último acesso em 12/05/2006.

[2]MARTINS, José Carlos Cordeiro. "Gestão de projetos de segurança da informação", São Paulo-SP, 2003;

[3]COLTRO, Renata. "Segurança: prioridade coorporativa" Computerworld, São Paulo, p. 26, 13 mar 2002.

[4]ARTIGOS, extraído da Lei 9.504 do Tribunal Superior Eleitoral;

[5]RELATÓRIO UNICAMP – Avaliação do sistema informatizado de eleições / maio 2002;

[6]JAKOBSKIND, Mário Augusto e MANESCHY, Osvaldo, "Burla eletrônica", Rio de Janeiro-RJ, 2002;

[7]ADIDA, Ben, Special Topics in Cryptography, Instrutor: Ran Canetti, "*Lecture 17: Introduction to Electronic Voting*", Abril 8, 2004, MIT/OCW.

[8]CHAUM, D., "*Security without Identification, Card Computers to make Big Brother Obsolete",* Comunications of the ACM, october 1985, pp 1030-1044.

[9]CHAUM, D., "*Achieving Electronic Privacy*," Scientific American, August 1992, pp. 96-101.

[10]SPAFFORD, G., Diretor de Operações de Computador, Auditoria e Tecnologia da Segurança, Purdue University/France;

[11]ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - NBR ISO/IEC17799: Tecnologia da informação - Código de prática para a gestão da segurança da informação. Rio de Janeiro, 2001.

[12]GOLDREICH, O. "*Foundations of Cryptography: Basic Tools*". New York, NY: Cambridge University Press, 2001.

[13]DIFFIE, W. and HELLMAN, W. E., "*New directions in cryptography*", IEEE trans. on information theory, nov. 1976.

[14]MENEZES, A. J., OORSCHOT, P. C. V. and VANSTONE, S. A., "*Handbook of applied cryptography*", California, CRC press, August, 2001.

[15]FERGUSON, N., SCHNEIER, B., "*Practical cryptography*", Wiley Publishing, Indianapolis, Indiana, 2003.

[16]http://www.esat.kuleuven.ac.be/~rijmen/rijndael - acessado em 12/05/2006.

[17]http://csrc.nist.gov/CryptoToolkit/aes/ - acessado em 12/05/2006.

[18]RIJMEN, V. & DAEMEN, J., "*AES proposal: Rijndael*", NIST, october,1999.

Anexo A

Hardware da Urna Eletrônica (UE).

Urna eletrônica de 1996 (UE 96):

- Processador 386 SX 40 MHz
- Código Braille
- 2 Mb de memória
- vídeo de cristal líquido monocromático de 9” Sharp
- teclado tipo membrana
- Impressora de impacto Epson
- Dois drives de disquetes Sony
- Sistema Operacional: VirtuOS
- Peso: 10 Kg
- Fonte de alimentação: AC 90 – 240V / DC 12V
- Autonomia de 1:30 horas com bateria interna
- Autonomia de 12 horas com bateria externa automotiva

Urna eletrônica de 1996 (UE 96) - Upgrade:

- Processador 386 SX 40 MHz
- Código Braille
- 2 Mb de memória
- vídeo de cristal líquido monocromático de 9”
- teclado de borracha condutiva
- Impressora impacto Epson
- 2 flash card
- Peso: 10 Kg
- Fonte de alimentação: AC 90 – 240V / DC 12V
- Autonomia de 1:30 horas com bateria interna
- Autonomia de 12 horas com bateria externa automotiva

Urna eletrônica de 1998 (UE 98):

- Cyrix media GX-133 Mhz
- Código Braille
- 8 Mb de memória
- vídeo de cristal líquido monocromático de 9” sharp
- teclado tipo borracha condutiva
- 2 flash card
- 1 Drive de disquete Sony
- Fonte alimentação: 35 W
- Impressora: Axiohm - Térmica 35 colunas
- Sistema Operacional: VirtuOs
- Peso: 8 Kg
- Fonte de alimentação: AC 90 – 240V / DC 12V
- Autonomia de 4 horas com bateria interna
- Autonomia de 15 horas com bateria externa automotiva

Urna eletrônica de 2000 (UE 2000):

- Processador Cyrix media GX-166 Mhz
- 16 Mb de memória
- vídeo de cristal líquido monocromático de 9” Nanya
- teclado mecânico resistente e especial para eleições
- Duas Flash Card (FI e FV)
- 1 Drive de disquete
- Impressora: MECAF/ROHM
- fone auricular para cegos
- Peso: 8 Kg
- Fonte de alimentação: AC 90 – 240V / DC 12V
- Autonomia de 12 horas com bateria interna

Anexo B

Extrato da Lei no. 9.504 do TSE

...

Do Sistema Eletrônico de Votação e da Totalização dos Votos

Art. 59. A votação e a totalização dos votos serão feitas por sistema eletrônico, podendo o Tribunal Superior Eleitoral autorizar, em caráter excepcional, a aplicação das regras fixadas nos arts. 83 a 89.

§ 1º A votação eletrônica será feita no número do candidato ou da legenda partidária, devendo o nome e fotografia do candidato e o nome do partido ou a legenda partidária aparecer no painel da urna eletrônica, com a expressão designadora do cargo disputado no masculino ou feminino, conforme o caso.

§ 2º Na votação para as eleições proporcionais, serão computados para a legenda partidária os votos em que não seja possível a identificação do candidato, desde que o número identificador do partido seja digitado de forma correta.

§ 3º A urna eletrônica exibirá para o eleitor, primeiramente, os painéis referentes às eleições proporcionais e, em seguida, os referentes às eleições majoritárias.

§ 4º A urna eletrônica disporá de mecanismo que permita a impressão do voto, sua conferência visual e depósito automático, sem contato manual, em local previamente lacrado, após conferência pelo eleitor. (Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 5º Se, ao conferir o voto impresso, o eleitor não concordar com os dados nele registrados, poderá cancelá-lo e repetir a votação pelo sistema eletrônico. Caso reitere a discordância entre os dados da tela da urna eletrônica e o voto impresso, seu voto será colhido em separado e apurado na forma que for regulamentada pelo Tribunal Superior Eleitoral, observado, no que couber, o disposto no art. 82 desta Lei.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 6º Na véspera do dia da votação, o juiz eleitoral, em audiência pública, sorteará três por cento das urnas de cada zona eleitoral, respeitado o limite mínimo de três urnas por Município, que deverão ter seus votos impressos contados e conferidos com os resultados apresentados pelo respectivo boletim de urna.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 7º A diferença entre o resultado apresentado no boletim de urna e o da contagem dos votos impressos será resolvida pelo juiz eleitoral, que também decidirá sobre a conferência de outras urnas.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 4º A urna eletrônica disporá de recursos que, mediante assinatura digital, permitam o registro digital de cada voto e a identificação da urna em que foi registrado, resguardado o anonimato do eleitor. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 5º Caberá à Justiça Eleitoral definir a chave de segurança e a identificação da urna eletrônica de que trata o § 4º. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 6º Ao final da eleição, a urna eletrônica procederá à assinatura digital do arquivo de votos, com aplicação do registro de horário e do arquivo do boletim de urna, de maneira a impedir a substituição de votos e a alteração dos registros dos termos de início e término da votação. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 7º O Tribunal Superior Eleitoral colocará à disposição dos eleitores urnas eletrônicas destinadas a treinamento. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 8º O Tribunal Superior Eleitoral colocará à disposição dos eleitores urnas eletrônicas destinadas a treinamento.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

Art. 60. No sistema eletrônico de votação considerar-se-á voto de legenda quando o eleitor assinalar o número do partido no momento de votar para determinado cargo e somente para este será computado.

Art 61. A urna eletrônica contabilizará cada voto, assegurando-lhe o sigilo e inviolabilidade, garantida aos partidos políticos, coligações e candidatos ampla fiscalização.

Art. 61A. Os tribunais eleitorais somente proclamarão o resultado das eleições depois de procedida a conferência a que se referem os §§ 6º e 7º do art. 59.(Artigo incluído pela Lei nº 10.408, de 10.1.2002) (Revogada pela Lei nº 10.740, de 1º.10.2003)

Art. 62. Nas Seções em que for adotada a urna eletrônica, somente poderão votar eleitores cujos nomes estiverem nas respectivas folhas de votação, não se aplicando a ressalva a que se refere o art. 148, § 1º, da Lei nº 4.737, de 15 de julho de 1965 - Código Eleitoral.

Parágrafo único. O Tribunal Superior Eleitoral disciplinará a hipótese de falha na urna eletrônica que prejudique o regular processo de votação.

Das Mesas Receptoras

Art. 63. Qualquer partido pode reclamar ao Juiz Eleitoral, no prazo de cinco dias, da nomeação da Mesa Receptora, devendo a decisão ser proferida em 48 horas.

§ 1º Da decisão do Juiz Eleitoral caberá recurso para o Tribunal Regional, interposto dentro de três dias, devendo ser resolvido em igual prazo.

§ 2º Não podem ser nomeados presidentes e mesários os menores de dezoito anos.

Art. 64. É vedada a participação de parentes em qualquer grau ou de servidores da mesma repartição pública ou empresa privada na mesma Mesa, Turma ou Junta Eleitoral.

Da Fiscalização das Eleições

Art. 65. A escolha de fiscais e delegados, pelos partidos ou coligações, não poderá recair em menor de dezoito anos ou em quem, por nomeação do Juiz Eleitoral, já faça parte de Mesa Receptora.

§ 1º O fiscal poderá ser nomeado para fiscalizar mais de uma Seção Eleitoral, no mesmo local de votação.

§ 2º As credenciais de fiscais e delegados serão expedidas, exclusivamente, pelos partidos ou coligações.

§ 3º Para efeito do disposto no parágrafo anterior, o presidente do partido ou o representante da coligação deverá registrar na Justiça Eleitoral o nome das pessoas autorizadas a expedir as credenciais dos fiscais e delegados.

Art. 66. Os partidos e coligações poderão fiscalizar todas as fases do processo de votação e apuração das eleições, inclusive o preenchimento dos boletins de urna e o processamento eletrônico da totalização dos resultados, sendo-lhes garantido o conhecimento antecipado dos programas de computador a serem usados.

§ 1º No prazo de cinco dias, a contar do conhecimento dos programas de computador a que se refere este artigo, o partido ou coligação poderá apresentar impugnação fundamentada à Justiça Eleitoral.

§ 2º Os partidos concorrentes ao pleito poderão constituir sistema próprio de fiscalização, apuração e totalização dos resultados, contratando, inclusive, empresas de auditoria de sistemas, que, credenciadas junto à Justiça Eleitoral, receberão, previamente, os programas de computador e, simultaneamente, os mesmos dados alimentadores do sistema oficial de apuração e totalização.

Art. 66. Os partidos e coligações poderão fiscalizar todas as fases do processo de votação e apuração das eleições e o processamento eletrônico da totalização dos resultados.(Redação dada pela Lei nº 10.408, de 10.1.2002)

§ 1º Todos os programas de computador de propriedade do Tribunal Superior Eleitoral, desenvolvidos por si ou sob encomenda, utilizados nas urnas eletrônicas para o processo de votação e apuração, serão apresentados para análise dos partidos e coligações, na forma de programas-fonte e programas-executáveis, inclusive os sistemas aplicativo e de segurança e as bibliotecas especiais, sendo que as chaves eletrônicas privadas e senhas eletrônicas de acesso se manterão no sigilo da Justiça Eleitoral.(Redação dada pela Lei nº 10.408, de 10.1.2002)

§ 2º A compilação dos programas das urnas eletrônicas, referidos no § 1º, será feita em sessão pública, com prévia convocação dos fiscais dos partidos e coligações, após o que serão lacradas cópias dos programas-fonte e dos programas compilados.(Redação dada pela Lei nº 10.408, de 10.1.2002)

§ 3º No prazo de cinco dias, a contar da sessão referida no § 2º, o partido ou coligação poderá apresentar impugnação fundamentada à Justiça Eleitoral.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 4º Havendo necessidade de modificação dos programas, a sessão referida no § 3º realizar-se-á, novamente, para este efeito.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 1º Todos os programas de computador de propriedade do Tribunal Superior Eleitoral, desenvolvidos por ele ou sob sua encomenda, utilizados nas urnas eletrônicas para os processos de votação, apuração e totalização, poderão ter suas fases de especificação e de desenvolvimento acompanhadas por técnicos indicados pelos partidos políticos, Ordem dos Advogados do Brasil e Ministério Público, até seis meses antes das eleições. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 2º Uma vez concluídos os programas a que se refere o § 1º, serão eles apresentados, para análise, aos representantes credenciados dos partidos políticos e coligações, até vinte dias antes das eleições, nas dependências do Tribunal Superior Eleitoral, na forma de programas-fonte e de programas executáveis, inclusive os sistemas aplicativo e de segurança e as bibliotecas especiais, sendo que as chaves eletrônicas privadas e senhas eletrônicas de acesso manter-se-ão no sigilo da Justiça Eleitoral. Após

a apresentação e conferência, serão lacradas cópias dos programas-fonte e dos programas compilados. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 3º No prazo de cinco dias a contar da data da apresentação referida no § 2º, o partido político e a coligação poderão apresentar impugnação fundamentada à Justiça Eleitoral. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 4º Havendo a necessidade de qualquer alteração nos programas, após a apresentação de que trata o § 3º, dar-se-á conhecimento do fato aos representantes dos partidos políticos e das coligações, para que sejam novamente analisados e lacrados. (Redação dada pela Lei nº 10.740, de 1º.10.2003)

§ 5º A carga ou preparação das urnas eletrônicas será feita em sessão pública, com prévia convocação dos fiscais dos partidos e coligações para a assistirem e procederem aos atos de fiscalização, inclusive para verificarem se os programas carregados nas urnas são idênticos aos que foram lacrados na sessão referida no § 2º deste artigo, após o que as urnas serão lacradas.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 6º No dia da eleição, será realizada, por amostragem, auditoria de verificação do funcionamento das urnas eletrônicas, através de votação paralela, na presença dos fiscais dos partidos e coligações, nos moldes fixados em resolução do Tribunal Superior Eleitoral. (Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

§ 7º Os partidos concorrentes ao pleito poderão constituir sistema próprio de fiscalização, apuração e totalização dos resultados contratando, inclusive, empresas de auditoria de sistemas, que, credenciadas junto à Justiça Eleitoral, receberão, previamente, os programas de computador e os mesmos dados alimentadores do sistema oficial de apuração e totalização.(Parágrafo incluído pela Lei nº 10.408, de 10.1.2002)

Art. 67. Os órgãos encarregados do processamento eletrônico de dados são obrigados a fornecer aos partidos ou coligações, no momento da entrega ao Juiz Encarregado, cópias dos dados do processamento parcial de cada dia, contidos em meio magnético.

Art. 68. O boletim de urna, segundo modelo aprovado pelo Tribunal Superior Eleitoral, conterá os nomes e os números dos candidatos nela votados.

§ 1º O Presidente da Mesa Receptora é obrigado a entregar cópia do boletim de urna aos partidos e coligações concorrentes ao pleito cujos representantes o requeiram até uma hora após a expedição.

§ 2º O descumprimento do disposto no parágrafo anterior constitui crime, punível com detenção, de um a três meses, com a alternativa de prestação de serviço à comunidade pelo mesmo período, e multa no valor de um mil a cinco mil UFIR.

Art. 69. A impugnação não recebida pela Junta Eleitoral pode ser apresentada diretamente ao Tribunal Regional Eleitoral, em quarenta e oito horas, acompanhada de declaração de duas testemunhas.

Parágrafo único. O Tribunal decidirá sobre o recebimento em quarenta e oito horas, publicando o acórdão na própria sessão de julgamento e transmitindo imediatamente à Junta, via telex, fax ou qualquer outro meio eletrônico, o inteiro teor da decisão e da impugnação.

Anexo C

...

2.6 – CONTROLE E CLASSIFICAÇÃO DE ATIVOS DE INFORMAÇÃO

Os objetivos desta seção são definir a classificação, o registro e o controle das informações da organização.

2.6.1 – Contabilização dos ativos de informação

A contabilização tem como objetivo manter a proteção adequada dos ativos de informação da organização. Por isso, faz-se necessário inventariar todos os ativos de informação, para garantir que a proteção seja mantida de forma correta.

A informação deve possuir um proprietário responsável e identificado. A ele é atribuída a responsabilidade pelo controle da implementação e manutenção.

- Inventário dos ativos de informação

O objetivo do inventário dos ativos da informação é garantir a implementação efetiva e correta das proteções. Assim, é necessário que a organização seja capaz de identificar seus ativos de informação, com seus respectivos valores e importância, e cujos níveis de proteção implementados estejam relacionados diretamente com eles. Para tanto, se faz necessário estruturar e manter um inventário dos principais ativos associados com seus respectivos sistemas de informação.

Cada ativo de informação e seu respectivo proprietário devem estar claramente identificado, assim como a classificação de segurança desse ativo deve estar acordada e documentada, juntamente com sua localização atual (dado importante para o plano de contingência).

2.6.2 – Classificação da informação

O objetivo da classificação da informação é garantir que os ativos de informação recebam um nível adequado de proteção, pois a informação possui vários níveis de sensibilidade e criticidade. A informação deve ser classificada para indicar a importância, a prioridade e o nível de proteção. Pode ser que informações mais sensíveis recebam um

nível adicional de proteção ou um tratamento especial. Um sistema de classificação da informação deve ser usado com intuito de definir níveis mais adequados de proteção.

- Recomendações para classificação

É recomendado que a classificação da informação e seus respectivos controles de proteção levem em consideração as necessidades de compartilhamento ou restrição da informação com seus respectivos impactos nos negócios. A informação deve ser classificada e rotulada de acordo com seu valor,

sensibilidade e criticidade, e esta rotulação deve ser feita de forma bem criteriosa, para evitar classificações que não condizem com a realidade da informação.

As regras de classificação devem, ainda, levar em consideração que algumas informações não devem possuir uma classificação fixa, pois sua sensibilidade varia com o tempo, e que sua rotulação pode ser modificada de acordo com uma política predeterminada.

Atenção especial deve ser dada à interpretação dos rótulos nas informações de terceiros, para averiguar se as definições destes são as mesmas utilizadas pela organização.

A responsabilidade pela classificação da informação e sua revisão periódica devem ficar a cargo de seu autor ou do proprietário responsável por ela.

- Rótulos e tratamento da informação

É importante definir os procedimentos para rotular a informação em conformidade com a política de classificação da informação adotada pela organização. Cada classificação deve abranger as atividades de cópia,armazenamento e tipos de transmissão a que a informação está sujeita, e ainda, como e quando deverá ser executada a destruição de sua mídia.

Anexo D

```
; *****************************************************************
; DESCRIÇÃO: CODIGO FONTE DA AUTENTICACAO NA URNA ELETRONICA
; AUTOR....: SERGIO KOBAYASHI
; TRABALHO DE PROJETO FINAL DE GRADUACAO
; PROGRAMA URNA.ASM
; VERSÃO...: 1.0      DATA: 05/05/06       ATUALIZACAO: 15/06/06
; *****************************************************************
;                  DESCRIÇÃO DO ARQUIVO
; *****************************************************************

; *****************************************************************

  LIST      P=PIC16F636
  #INCLUDE P16F636.inc
  #INCLUDE S_TABELA.inc              ; GUARDA DA TABELA S E INICIALIZA NO
                                     ; ENDERECO 0x0700

; *****************************************************************
; CONFIGURAÇÕES DO MICROCONTROLADOR
; *****************************************************************
;
  __CONFIG   _CP_OFF & _WDT_ON & _BODEN_OFF & _PWRTE_ON & _RC_OSC

; ************************ CONSTANTES ******************************

INIT_RAM0       EQU   0x20          ; PONTO INICIAL DA RAM NO SEGMENTO 0
ROUNDS          EQU   0x0A    ; NUMERO DE RODADAS
POS_MEM         EQU   0x00    ; ENDERECO ONDE ESTA ARMAZENADA A INSCRICAO

LCD_DATA_TEMP   EQU   PORTA; PORTA DE DADOS PARA O LCD
LCD_CTRL        EQU   PORTC; PORTAS DE CONTROLE DO LCD

LCD_E           EQU   4             ; LCD ENABLE
LCD_RS          EQU   5             ; LCD RS

RC3             EQU   7             ; LCD dataline 7 (MSB)
RC2             EQU   6             ; LCD dataline 6
RC1             EQU   5             ; LCD dataline 5
RC0             EQU   4             ; LCD dataline 4
RA5             EQU   3             ; LCD dataline 3
RA4             EQU   2             ; LCD dataline 2
RA2             EQU   1             ; LCD dataline 1
RA1             EQU   0             ; LCD dataline 0 (LSB)

DELAY           EQU   0x23          ; USADO NA ROTINA DE DELAY
X_DELAY         EQU   0x24          ; USADO NA ROTINA DE X_DELAY

; *************************** VARIAVEIS ****************************

BLOCO           EQU   0x20          ; VETOR DE BLOCO ( 16 bytes*8 => 128 bits)
CHAVE           EQU   0x30          ; VETOR DA CHAVE ( 16 bytes*8 => 128 bits)

AUX             EQU   0x40          ; VARIAVEIS
AUX1            EQU   0x41          ;
```

```
AUX2            EQU   0x42
AUX3            EQU   0x43

ROUND_CNT       EQU   0x44          ; CONTADOR DE RODADAS PARA CRIPTOGRAFIA
RCON            EQU   0x45          ;

; ************************ EEPROM **********************************
; DEFINICAO DO ENDERECO INICIAL DA EEPROM
; ******************************************************************

          ORG           H'2100'+POS_MEM          ; INICIO DA EEPROM

;****************** INICIO DA AREA DE PROGRAMA **********************
; DEFINICAO DO ENDERECO INICIAL DO PROGRAMA E ROTINAS DE INTERRUPCAO
; ********************************************************************************

          ORG   0x00            ; ENDERECO INICIAL DO PROGRAMA

INICIO:
  GOTO  MAIN                    ; VAI A ROTINA PRINCIPAL

;********** INICIO DA AREA DE TRATAMENTO DE INTERRUPCAO *************
; COMO NAO HA INTERRUPCOES A ROTINA E RESTRITA AO ENDERECO
; E AO RETORNO. ESSA ROTINA NAO E NECESSARIA, MAS A INCLUSAO
; PODE EVITAR QUALQUER "BUG" CASO SEJA INTERPRETADA ALGUMA INT.
; ********************************************************************

          ORG   0x04            ; ENDERECO DE CONTROLE DE INTERRUPCAO

INT_CTRL:
  RETFIE                        ; RETORNO DA INTERRUPCAO

;********************** ROTINA DE CRIPTOGRAFIA **********************
; ESSA ROTINA CRIPTOGRAFA O CODIGO DO ELEITOR PARA A BUSCA
; ********************************************************************

MAIN:
  BTFSC DADOS                   ; CHECA SE TEM DADOS
  GOTO  MAIN                    ; NAO, LOOP INFINITO
  CALL   INIT_CPU               ; SIM, INICIALIZA OS REGISTRADORES DA CPU
  CALL   INIT_DISPLAY           ; INICIALIZA DISPLAY LCD
  CALL   LE_BLOCO               ; LE A INSCRICAO A SER CRIPTOGRAFADO
  CALL   LE_CHAVE               ; LE A CHAVE USADA NA CRIPTOGRAFIA
  CALL   CRIPT                  ; CRIPTOGRAFA
  CALL   BUSCA_MEM              ; BUSCA NA EEPROM
  GOTO  $

; ********************* INIT_CPU ***********************************
; ESSA ROTINA LIMPA A RAM NO SEGMENTO 0
; ENTRADA:
; SAIDA..:
; GLOBAL.:
; OBS....:
; ******************************************************************

INIT_CPU:
  MOVLW   INIT_RAM0             ; APONTA O INICIO DA RAM
  MOVWF   FSR
LOOP_CRAM0
  CLRF    INDF                  ; APAGA O PONTEIRO DA RAM
```

```
  INCF    FSR,F           ; INCREMENTA O PONTEIRO
  BTFSS   FSR,7           ; ENQUANTO FSR < 0x80 (D128) EXECUTA O LOOP
  GOTO    LOOP_CRAM0
  RETURN

; ********************* INIT_DISPLAY ***********************************
; ESSA ROTINA INICIALIZA O DISPLAY LCD
; ENTRADA:
; SAIDA..:
; GLOBAL.:
; OBS....:
; **********************************************************************

INIT_DISPLAY:
  MOVLW         0x1E
  CALL          X_DELAY500          ; DELAY DE 15ms (30 * 0,5ms = 15ms)

  MOVLW         B'00001101'         ;DISPLAY ON, CURSOR OFF, BLINKING OFF
  CALL          LCD_INST

  MOVLW         B'00000001'         ;CLEAR DISPLAY
  CALL          LCD_INST

  MOVLW         B'00000110'         ;CURSOR_RIGHT ON, AUTO_SHIFT OFF
  CALL          LCD_INST
  RETURN

TTT:

  MOVWF         LCD_DATA_TEMP       ;COPIA W PARA LCD
  BCF           RS                  ;MODO INSTRUÇÃO DO DISPLAY
  BSF           E                   ;SET BIT E (ENABLE)
  BCF           E                   ;CLEAR BIT E (DISABLE)
  RETURN

; ************************** LE_BLOCO ******************************
; ESSA ROTINA LE O BLOCO DE DADOS DO CARTAO DO ELEITOR
; ENTRADA:      INSC[12] + ZN_EL[4]
; SAIDA..:      BLOCO[16]={B0,B1,...,B15}
; GLOBAL.:      BLOCO,AUX,FSR
; OBS....:
; **********************************************************************

LE_BLOCO:
  MOVLW         0x0
  MOVWF         BLOCO+0x0
  MOVLW         0x0
  MOVWF         BLOCO+0x1
  MOVLW         0x0
  MOVWF         BLOCO+0x2
  MOVLW         0x0
  MOVWF         BLOCO+0x3
  MOVLW         0x0
  MOVWF         BLOCO+0x4
  MOVLW         0x0
  MOVWF         BLOCO+0x5
  MOVLW         0x0
  MOVWF         BLOCO+0x6
  MOVLW         0x0
  MOVWF         BLOCO+0x7
```

```
  MOVLW       0x0
  MOVWF       BLOCO+0x8
  MOVLW       0x0
  MOVWF       BLOCO+0x9
  MOVLW       0x40
  MOVWF       BLOCO+0xA
  MOVLW       0x0
  MOVWF       BLOCO+0xB
  MOVLW       0x0
  MOVWF       BLOCO+0xC
  MOVLW       0x0
  MOVWF       BLOCO+0xD
  MOVLW       0x0
  MOVWF       BLOCO+0xE
  MOVLW       0x0
  MOVWF       BLOCO+0xF
  RETURN

; ************************** LE_CHAVE *******************************
; ESSA ROTINA LE A CHAVE CRIPTOGRAFICA PARA A SER UTILIZADA
; "PROJETOFINALURNA"
; ENTRADA:
; SAIDA..:        CHAVE[16]={CH0,CH1,...,CH15}
; GLOBAL.:        CHAVE[16],FSR,AUX
; *******************************************************************

LE_CHAVE:
  MOVLW       0x50
  MOVWF       CHAVE+0x0                   ; P
  MOVLW       0x52
  MOVWF       CHAVE+0x1                   ; R
  MOVLW       0x4F
  MOVWF       CHAVE+0x2                   ; O
  MOVLW       0x4A
  MOVWF       CHAVE+0x3                   ; J
  MOVLW       0x45
  MOVWF       CHAVE+0x4                   ; E
  MOVLW       0x54
  MOVWF       CHAVE+0x5                   ; T
  MOVLW       0x4F
  MOVWF       CHAVE+0x6                   ; O
  MOVLW       0x46
  MOVWF       CHAVE+0x7                   ; F
  MOVLW       0x49
  MOVWF       CHAVE+0x8                   ; I
  MOVLW       0x4E
  MOVWF       CHAVE+0x9                   ; N
  MOVLW       0x41
  MOVWF       CHAVE+0xA                   ; A
  MOVLW       0x4C
  MOVWF       CHAVE+0xB                   ; L
  MOVLW       0x55
  MOVWF       CHAVE+0xC                   ; U
  MOVLW       0x52
  MOVWF       CHAVE+0xD                   ; R
  MOVLW       0x4E
  MOVWF       CHAVE+0xE                   ; N
  MOVLW       0x41
  MOVWF       CHAVE+0xF                   ; A
  RETURN
```

```
; ************************ ADICAO_CHAVE ****************************
; ESSA ROTINA FAZ: BLOCO ^= CHAVE ( ^= SIGNIFICA BLOCO = BLOCO XOR CHAVE )
; ENTRADA:
; SAIDA..:
; GLOBAL.:      BLOCO, CHAVE
; OBS....:
; ********************************************************************

ADICAO_CHAVE:
  MOVF      CHAVE+0x0,W     ; BLOCO[0] ^= CHAVE[0];
  XORWF     BLOCO+0x0,F

  MOVF      CHAVE+0x1,W     ; BLOCO[1] ^= CHAVE[1];
  XORWF     BLOCO+0x1,F

  MOVF      CHAVE+0x2,W     ; BLOCO[2] ^= CHAVE[2];
  XORWF     BLOCO+0x2,F

  MOVF      CHAVE+0x3,W     ; BLOCO[3] ^= CHAVE[3];
  XORWF     BLOCO+0x3,F

  MOVF      CHAVE+0x4,W     ; BLOCO[4] ^= CHAVE[4];
  XORWF     BLOCO+0x4,F

  MOVF      CHAVE+0x5,W     ; BLOCO[5] ^= CHAVE[5];
  XORWF     BLOCO+0x5,F

  MOVF      CHAVE+0x6,W     ; BLOCO[6] ^= CHAVE[6];
  XORWF     BLOCO+0x6,F

  MOVF      CHAVE+0x7,W     ; BLOCO[7] ^= CHAVE[7];
  XORWF     BLOCO+0x7,F

  MOVF      CHAVE+0x8,W     ; BLOCO[8] ^= CHAVE[8];
  XORWF     BLOCO+0x8,F

  MOVF      CHAVE+0x9,W     ; BLOCO[9] ^= CHAVE[9];
  XORWF     BLOCO+0x9,F

  MOVF      CHAVE+0x0A,W    ; BLOCO[10] ^= CHAVE[10];
  XORWF     BLOCO+0x0A,F

  MOVF      CHAVE+0x0B,W    ; BLOCO[11] ^= CHAVE[11];
  XORWF     BLOCO+0x0B,F

  MOVF      CHAVE+0x0C,W    ; BLOCO[12] ^= CHAVE[12];
  XORWF     BLOCO+0x0C,F

  MOVF      CHAVE+0x0D,W    ; BLOCO[13] ^= CHAVE[13];
  XORWF     BLOCO+0x0D,F

  MOVF      CHAVE+0x0E,W    ; BLOCO[14] ^= CHAVE[14];
  XORWF     BLOCO+0x0E,F

  MOVF      CHAVE+0x0F,W    ; BLOCO[15] ^= CHAVE[15];
  XORWF     BLOCO+0x0F,F
  RETURN
```

```
; ********************** SUBSTITUICAO_S ****************************
; ESSA ROTINA FAZ: BLOCO[i] = S_BOX[BLOCO[i]]
; ENTRADA:
; SAIDA..:
; GLOBAL.: AUX,BLOCO,FSR,PCLATH
; OBS....:
; ******************************************************************

SUBSTITUICAO_S:                         ; IMPLEMENTA A SUBSTITUICAO DIRETA
  MOVLW     HIGH S_TABELA               ; SELECIONA UM SEGMENTO
  MOVWF     PCLATH                      ;
                                        ; FUNCIONA COMO UM FOR: for i(0....15)
PUT_S_TABELA_B0:
  MOVLW     D'99'                       ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x0,F
  BTFSC STATUS,Z
  GOTO  PUT_B0
  DECF  BLOCO+0x0,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B0:
  MOVWF     BLOCO+0x0

PUT_S_TABELA_B1:
  MOVLW     D'99'                       ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x1,F
  BTFSC STATUS,Z
  GOTO  PUT_B1
  DECF  BLOCO+0x1,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B1:
  MOVWF     BLOCO+0x1

PUT_S_TABELA_B2:
  MOVLW     D'99'                       ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x2,F
  BTFSC STATUS,Z
  GOTO  PUT_B2
  DECF  BLOCO+0x2,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B2:
  MOVWF     BLOCO+0x2

PUT_S_TABELA_B3:
  MOVLW     D'99'                       ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x3,F
  BTFSC STATUS,Z
  GOTO  PUT_B3
  DECF  BLOCO+0x3,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B3:
  MOVWF     BLOCO+0x3

PUT_S_TABELA_B4:
  MOVLW     D'99'                       ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x4,F
  BTFSC STATUS,Z
  GOTO  PUT_B4
  DECF  BLOCO+0x4,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B4:
```

```
  MOVWF         BLOCO+0x4

PUT_S_TABELA_B5:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x5,F
  BTFSC STATUS,Z
  GOTO  PUT_B5
  DECF  BLOCO+0x5,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B5:
  MOVWF         BLOCO+0x5

PUT_S_TABELA_B6:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x6,F
  BTFSC STATUS,Z
  GOTO  PUT_B6
  DECF  BLOCO+0x6,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B6:
  MOVWF         BLOCO+0x6

PUT_S_TABELA_B7:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x7,F
  BTFSC STATUS,Z
  GOTO  PUT_B7
  DECF  BLOCO+0x7,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B7:
  MOVWF         BLOCO+0x7

PUT_S_TABELA_B8:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x8,F
  BTFSC STATUS,Z
  GOTO  PUT_B8
  DECF  BLOCO+0x8,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B8:
  MOVWF         BLOCO+0x8

PUT_S_TABELA_B9:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0x9,F
  BTFSC STATUS,Z
  GOTO  PUT_B9
  DECF  BLOCO+0x9,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_B9:
  MOVWF         BLOCO+0x9

PUT_S_TABELA_BA:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0xA,F
  BTFSC STATUS,Z
  GOTO  PUT_BA
  DECF  BLOCO+0xA,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_BA:
```

```
  MOVWF         BLOCO+0xA

PUT_S_TABELA_BB:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0xB,F
  BTFSC STATUS,Z
  GOTO  PUT_BB
  DECF  BLOCO+0xB,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_BB:
  MOVWF         BLOCO+0xB

PUT_S_TABELA_BC:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0xC,F
  BTFSC STATUS,Z
  GOTO  PUT_BC
  DECF  BLOCO+0xC,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_BC:
  MOVWF         BLOCO+0xC

PUT_S_TABELA_BD:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0xD,F
  BTFSC STATUS,Z
  GOTO  PUT_BD
  DECF  BLOCO+0xD,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_BD:
  MOVWF         BLOCO+0xD

PUT_S_TABELA_BE:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0xE,F
  BTFSC STATUS,Z
  GOTO  PUT_BE
  DECF  BLOCO+0xE,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_BE:
  MOVWF         BLOCO+0xE

PUT_S_TABELA_BF:
  MOVLW         D'99'                   ; SE (BLOCO[i]==0) --> BLOCO[i]=99D)
  MOVF  BLOCO+0xF,F
  BTFSC STATUS,Z
  GOTO  PUT_BF
  DECF  BLOCO+0xF,W
  CALL  S_TABELA                        ; SENAO BLOCO[i] = S_TABELA[BLOCO[i]-1]
PUT_BF:
  MOVWF         BLOCO+0xF
  RETURN


; ********************** SHIFT_ROW ******************************
; ESSA ROTINA MUDA AS LINHAS (1,2 e 3 MAS NAO A 0)
; ENTRADA:
; SAIDA..:      BLOCO MODIFICADO
; GLOBAL.:      AUX, BLOCO, VETOR_COL
; OBS:
```

```
; *********************************************************************

SHIFT_ROW:

; ROTACIONA LINHA ESQUERDA 1 UMA POSICAO
  MOVF          BLOCO+0x1,W
  MOVWF         AUX
  MOVF          BLOCO+0x5,W         ;
  MOVWF         BLOCO+0x1
  MOVF          BLOCO+0x9,W         ;
  MOVWF         BLOCO+0x5
  MOVF          BLOCO+0xD,W         ;
  MOVWF         BLOCO+0x9
  MOVF          AUX,W
  MOVWF         BLOCO+0xD           ;


; ROTACIONA LINHA ESQUERDA 2 DUAS POSICOES A PARTIR DO BLOCO+0x0A
  MOVF          BLOCO+0x2,W
  MOVWF         AUX
  MOVF          BLOCO+0x6,W
  MOVWF         AUX1

  MOVF          BLOCO+0xA,W
  MOVWF         BLOCO+0x2
  MOVF          BLOCO+0xE,W
  MOVWF         BLOCO+0x6

  MOVF          AUX,W
  MOVWF         BLOCO+0xA
  MOVF          AUX1,W
  MOVWF         BLOCO+0xE

; ROTACIONA LINHA ESQUERDA 3 TRES POSICOES A PARTIR DO BLOCO+0xB
  MOVF          BLOCO+0xF,W
  MOVWF         AUX
  MOVF          BLOCO+0xB,W
  MOVWF         BLOCO+0xF
  MOVF          BLOCO+0x7,W
  MOVWF         BLOCO+0xB
  MOVF          BLOCO+0x3,W
  MOVWF         BLOCO+0x7
  MOVF          AUX,W
  MOVWF         BLOCO+0x3
  RETURN

; ************************ MIX_COLUMN ************************
; ESSA ROTINA E EQUIVALENTE A MULTIPLICACAO DO BLOCO EM COLUNAS
; DE UMA MATRIZ 4x4 PELA SEGUINTE MATRIZ
;
;  0x2 0x3 0x1 0x1
;  0x3 0x1 0x1 0x2
;  0x1 0x1 0x2 0x3
;  0x1 0x2 0x3 0x1
;
; ENTRADA:
; SAIDA..:
; GLOBAL.:
; OBS....:
; *********************************************************************
```

```
MIX_COLUMN:

; INICIA COM A MISTURA DA COLUMN 0
  MOVF      BLOCO+0x0,W
  XORWF     BLOCO+0x1,W
  MOVWF     AUX1                    ; AUX1 = BLOCO+0x0 ^ BLOCO+0x1
  XORWF     BLOCO+0x2,W
  XORWF     BLOCO+0x3,W
  MOVWF     AUX                     ; AUX = BLOCO+0x0 ^ BLOCO+0x1 ^ BLOCO+0x2
^ BLOCO+0x3

  MOVF      BLOCO+0x1,W             ; AUX2 = BLOCO+0x1 ^ BLOCO+0x2
  XORWF     BLOCO+0x2,W
  MOVWF     AUX2

  MOVF      BLOCO+0x2,W             ; AUX3 = BLOCO+0x2 ^ BLOCO+0x3
  XORWF     BLOCO+0x3,W
  MOVWF     AUX3

  CALL   CALC_XVEZES                ; AUX = XVEZES(AUX), AUX1 = XVEZES(AUX1),
AUX2 = XVEZES(AUX2), AUX3 = XVEZES(AUX3)

  MOVF      AUX,W                   ; BLOCO+0x0 ^= AUX ^ AUX1
  XORWF     AUX1,W
  XORWF     BLOCO+0x0,f

  MOVF      AUX,W                   ; BLOCO+0x1 ^= AUX ^ AUX2
  XORWF     AUX2,W
  XORWF     BLOCO+0x1,f

  MOVF      AUX,W                   ; BLOCO+0x2 ^= AUX ^ AUX3
  XORWF     AUX3,W
  XORWF     BLOCO+0x2,f

  MOVF      BLOCO+0x0,W       ; BLOCO+0x3 = BLOCO+0x1 ^ BLOCO+0x2 ^ BLOCO+0x2
^ AUX
  XORWF     BLOCO+0x1,W
  XORWF     BLOCO+0x2,W
  XORWF     AUX,W
  MOVWF     BLOCO+0x3


; MISTURA COLUNA 1
  MOVF      BLOCO+0x4,W
  XORWF     BLOCO+0x5,W
  MOVWF     AUX1              ;
  XORWF     BLOCO+0x6,W
  XORWF     BLOCO+0x7,W
  MOVWF     AUX

  MOVF      BLOCO+0x5,W
  XORWF     BLOCO+0x6,W
  MOVWF     AUX2

  MOVF      BLOCO+0x6,W
  XORWF     BLOCO+0x7,W
  MOVWF     AUX3

  CALL   CALC_XVEZES
```

```
  MOVF      AUX,W
  XORWF     AUX1,W
  XORWF     BLOCO+0x4,f

  MOVF      AUX,W
  XORWF     AUX2,W
  XORWF     BLOCO+0x5,f

  MOVF      AUX,W
  XORWF     AUX3,W
  XORWF     BLOCO+0x6,f

  MOVF      BLOCO+0x4,W
  XORWF     BLOCO+0x5,W
  XORWF     BLOCO+0x6,W
  XORWF     AUX,W
  MOVWF     BLOCO+0x7

; MISTURA COLUNA 2
  MOVF      BLOCO+0x8,W
  XORWF     BLOCO+0x9,W
  MOVWF     AUX1            ;
  XORWF     BLOCO+0xA,W
  XORWF     BLOCO+0xB,W
  MOVWF     AUX

  MOVF      BLOCO+0x9,W
  XORWF     BLOCO+0xA,W
  MOVWF     AUX2

  MOVF      BLOCO+0xA,W
  XORWF     BLOCO+0xB,W
  MOVWF     AUX3

  CALL   CALC_XVEZES

  MOVF      AUX,W
  XORWF     AUX1,W
  XORWF     BLOCO+0x8,f

  MOVF      AUX,W
  XORWF     AUX2,W
  XORWF     BLOCO+0x9,f

  MOVF      AUX,W
  XORWF     AUX3,W
  XORWF     BLOCO+0xA,f

  MOVF      BLOCO+0x8,W
  XORWF     BLOCO+0x9,W
  XORWF     BLOCO+0xA,W
  XORWF     AUX,W
  MOVWF     BLOCO+0xB


; MISTURA COLUNA 3
  MOVF      BLOCO+0xC,W
  XORWF     BLOCO+0xD,W
  MOVWF     AUX1            ;
```

```
  XORWF         BLOCO+0xE,W
  XORWF         BLOCO+0xF,W
  MOVWF         AUX

  MOVF          BLOCO+0xD,W
  XORWF         BLOCO+0xE,W
  MOVWF         AUX2

  MOVF          BLOCO+0xE,W
  XORWF         BLOCO+0xF,W
  MOVWF         AUX3

  CALL    CALC_XVEZES

  MOVF          AUX,W
  XORWF         AUX1,W
  XORWF         BLOCO+0xC,f

  MOVF          AUX,W
  XORWF         AUX2,W
  XORWF         BLOCO+0xD,f

  MOVF          AUX,W
  XORWF         AUX3,W
  XORWF         BLOCO+0xE,f

  MOVF          BLOCO+0xC,W
  XORWF         BLOCO+0xD,W
  XORWF         BLOCO+0xE,W
  XORWF         AUX,W
  MOVWF         BLOCO+0xF

  RETURN

CALC_XVEZES:
  MOVLW         0x1B
  BCF           STATUS,C
  RLF           AUX1,F
  BTFSC         STATUS,C
  XORWF         AUX1,F
  BCF           STATUS,C
  RLF           AUX2,F
  BTFSC         STATUS,C
  XORWF         AUX2,F
  BCF           STATUS,C
  RLF           AUX3,F
  BTFSC         STATUS,C
  XORWF         AUX3,f
  RETURN

; ********************* ENC_CHAVE ******************************
; ESSA ROTINA ATUALIZA A CHAVE CRIPTOGRAFICA GERANDO UMA NOVA
; ENTRADA:      CHAVE
; SAIDA..:
; GLOBAL.:      AUX,RCON
; *************************************************************

ENC_CHAVE:
  CALL    CALC_S_TABELA_VALORES
  MOVF  RCON,W                              ; CHAVE[0] ^= RCON
```

```
  XORWF     CHAVE,F
  BCF       STATUS,C
  RLF       RCON,F          ; RCON = XVEZES(RCON)
  BTFSS     STATUS,C
  GOTO      COMP_ROUND
  MOVLW     0x1B
  MOVWF     RCON

COMP_ROUND:
  MOVF      CHAVE+0x0,W     ; EQUIVALENTE A UM
  XORWF     CHAVE+0x4,F     ; XOR DE CADA COLUNA
                            ; COM A ANTERIOR
  MOVF      CHAVE+0x1,W
  XORWF     CHAVE+0x5,F
                            ; PRIMEIRO COLUNA[1] ^= COLUNA[0]
  MOVF      CHAVE+0x2,W
  XORWF     CHAVE+0x6,F

  MOVF      CHAVE+0x3,W
  XORWF     CHAVE+0x7,F
                            ; COLUNA[2] ^= COLUNA[1]
  MOVF      CHAVE+0x4,W
  XORWF     CHAVE+0x8,F

  MOVF      CHAVE+0x5,W
  XORWF     CHAVE+0x9,F

  MOVF      CHAVE+0x6,W
  XORWF     CHAVE+0xA,F

  MOVF      CHAVE+0x7,W
  XORWF     CHAVE+0xB,F
                            ; COLUNA[3] ^= COLUNA[2]
  MOVF      CHAVE+0x8,W
  XORWF     CHAVE+0xC,F

  MOVF      CHAVE+0x9,W
  XORWF     CHAVE+0xD,F

  MOVF      CHAVE+0xA,W
  XORWF     CHAVE+0xE,F

  MOVF      CHAVE+0xB,W
  XORWF     CHAVE+0xF,F
  RETURN

CALC_S_TABELA:
  MOVLW     HIGH S_TABELA
  MOVWF     PCLATH

  MOVF      CHAVE+0xD,W
  BTFSS     STATUS,Z
  GOTO  S_TABELA_0xD
  MOVLW     D'99'
  XORWF     CHAVE+0x0,F
  GOTO  LE_CHAVE_0xE

S_TABELA_0xD:
  DECF      CHAVE+0xD,W     ; CHAVE[0x0] ^= s_box[CHAVE[0xD]]
  CALL   S_TABELA
```

```
  XORWF         CHAVE+0x0,F

LE_CHAVE_0xE:
  MOVF          CHAVE+0xE,W
  BTFSS         STATUS,Z
  GOTO  S_TABELA_0xE
  MOVLW         D'99'
  XORWF         CHAVE+0x1,F
  GOTO  LE_CHAVE_0xF

S_TABELA_0xE:
  DECF          CHAVE+0xE,W         ; CHAVE[0x1] ^= s_box[CHAVE[0xE]]
  CALL    S_TABELA
  XORWF         CHAVE+0x1,F

LE_CHAVE_0xF:
  MOVF          CHAVE+0xF,W
  BTFSS         STATUS,Z
  GOTO  S_TABELA_0xF
  MOVLW         D'99'
  XORWF         CHAVE+0x2,F
  GOTO  LE_CHAVE_0xC

S_TABELA_0xF:
  DECF          CHAVE+0xF,W         ; CHAVE[0x2] ^= s_box[CHAVE[0xF]]
  CALL    S_TABELA
  XORWF         CHAVE+0x2,F

LE_CHAVE_0xC:
  MOVF          CHAVE+0xC,W
  BTFSS         STATUS,Z
  GOTO  S_TABELA_0xC
  MOVLW         D'99'
  XORWF         CHAVE+0x3,F

  RETURN

S_TABELA_0xC:
  DECF          CHAVE+0xC,W         ; CHAVE[0x3] ^= s_box[CHAVE[0xC]]
  CALL    S_TABELA
  XORWF         CHAVE+0x3,F

  RETURN

; ************************* CRIPT *******************************
; ESSA ROTINA CRIPTOGRAFA O TEXTO UTILIZANDO A CHAVE
; ENTRADA: INSCRICAO DO ELEITOR + ZONA ELEITORAL
;              A CHAVE DE CRIPTOGRAFIA
; SAIDA..: TEXTO CIFRADO
; OBS....:
; ***************************************************************

CRIPT:
  MOVLW         0x01
  MOVWF         RCON                ; INICIALIZA O RCON
  CALL    ADICAO_CHAVE              ; ADICAO_CHAVE INICIAL
  MOVLW         ROUNDS              ; INICIALIZA AS RODADAS
  MOVWF         ROUND_CNT

LOOP_CRIPT:
```

```
  CALL   SUBSTITUICAO_S          ;
  CALL   SHIFT_ROW
  DECF   ROUND_CNT,W
  BTFSC STATUS,Z
  GOTO  ULTIMA_RODADA
  CALL   MIX_COLUMN

ULTIMA_RODADA:
  CALL   ENC_CHAVE
  CALL   ADICAO_CHAVE
  DECFSZ         ROUND_CNT,F
  GOTO  LOOP_CRIPT
  RETURN

; ******************** BUSCA_MEM ******************************
; ESSA ROTINA BUSCA OS DADOS NA MEMORIA DE DADOS DA EEPROM
; ENTRADA:       BLOCO
; SAIDA..:       LCD
; GLOBAL.:
; *****************************************************************

BUSCA_MEM:
  BSF            STATUS,RP0          ; BANK1
  MOVLW          POS_MEM
  MOVWF          EEADR               ; ENDERECO DA MEMORIA PARA LEITURA
  BSF            EECON1,RD           ; LE O CONTEUDO DO ENDERECO (BAIXO - 1o. BYTE)
  MOVF           EEDAT,W             ; MOVE O CONTEUDO DO ENDERECO PARA W
  BCF            STATUS,RP0          ; BANK0
  XORWF          BYTE_LOW            ; COMPARA O VALOR COM BYTE_LOW
  BTFSC          STATUS,Z            ; VALOR IGUAL?
  RETURN                             ; NAO, ENTAO VOLTA

  MOVLW          .0                                  ; COLUNA 0 DA LINHA 2
  CALL   LINHA2                      ;
  CALL   LCD_DATA                    ; MOSTRA O PRIMEIRO BYTE NO DISPLAY
                                             ; SIM, LE O VALOR SEGUINTE (ALTO)
  BSF    STATUS,RP0                  ; BANK1
  BCF    STATUS,RP1
  MOVLW          POS_MEM+H'10'       ; ENDERECO + 16BYTES (EM HEXA)
  MOVWF          EEADR               ; ENDERECO DA MEMORIA PARA LEITURA
  BSF    EECON1,RD                   ; LE O CONTEUDO DO ENDERECO (ALTO - 2o. BYTE)
  MOVF   EEDAT,W                     ; MOVE O CONTEUDO DO ENDERECO PARA W
  BCF    STATUS,RP0                  ; BANK0
  XORWF          BYTE_HIGH           ; COMPARA O VALOR COM BYTE_HIGH
  BTFSC STATUS,Z                     ; VALOR IGUAL?
  RETURN                             ; NAO, ENTAO VOLTA
                                      SIM, MOSTRA O VALOR NO DISPLAY CRIPTOGRAFADO
  MOVLW          .8                  ; COLUNA 9 DA LINHA 2
  CALL   LINHA2
  CALL   LCD_DATA                    ; MOSTRA O SEGUNDO BYTE NO DISPLAY

; ******************** ESCRITA NO DISPLAY ************************
;  ROTINA DE ESCRITA NO DISPLAY (LCD)
; *****************************************************************

LCD_DATA:

  MOVWF          LCD_DATA_TEMP              ;MOVE W PARA LCD_DATA_TEMP
  MOVLW          0x1E
  CALL   X_DELAY500                         ;DELAY 15ms (30 * 0,5ms = 15ms)
```

```
  BSF             RS                          ;MODO DADOS DO DISPLAY
  BSF             E                           ;SETA BIT E (ENABLE)
  BCF             E                           ;CLEAR BIT E (DISABLE)
  RETURN

; ********************************************************************
; *                          ROTINA DE INSTRUÇÃO DO LCD
; ********************************************************************

LCD_INST

  MOVWF         LCD_DATA_TEMP       ;MOVE W PARA LCD_DATA_TEMP
  CALL    X_DELAY500                ;DELAY 15ms
  BCF             RS                ;MODO INSTRUÇÃO DO DISPLAY
  BSF             E                 ;SETA BIT E (ENABLE)
  BCF             E                 ;CLEAR BIT E (DISABLE)

  RETURN

; ********************************************************************
; *                    ROTINA PARA ENDEREÇOS DA LINHA 1
; ********************************************************************

LINHA1:

  ADDLW         B'10000000'               ;LINHA 1 + POSIÇÃO DO CURSOR
  CALL    LCD_INST
  RETURN

; ********************************************************************
; *              ROTINA PARA ENDEREÇOS DA LINHA 2
; ********************************************************************

LINHA2:

  ADDLW         B'11000000'               ;LINHA 2 + POSIÇÃO DO CURSOR
  CALL    LCD_INST
  RETURN

;*****************************************************************************
; ROTINA DE CRIAÇÃO DE DELAYS PARA A COMUNICACAO COM O LCD
;
; DELAY_TEMPO         = ((DELAY_VALOR * 3) + 4) * TEMPO_CICLO
; DELAY_VALOR= (DELAY_TEMPO - (4 * CYCLE_TEMPO)) / (3 * TEMPO_CICLO)
;
; DELAY_TEMPO         = ((32 * 3) + 4) * 1uSeg
;                     = 100uSeg
; DELAY_VALOR= (500uSeg - 4) / 3
;                     = 165.33
;                     = 165
;*****************************************************************************

DELAY500            MOVLW       D'165'          ; +1        1 cycle
                    MOVWF       DELAY           ; +2        1 cycle
DELAY500_LOOP       DECFSZ      DELAY, F        ; passo 1   1 cycle
                    GOTO  DELAY500_LOOP         ; passo 2   2 cycles
DELAY500_FIM  RETURN                            ; +3        2 cycles
                                                ;
                                                ;
X_DELAY500          MOVWF       X_DELAY         ; +1        1 cycle
```

```
X_DELAY500_LOOP     CALL    DELAY500                ; passo1        espera 500uSeg
                    DECFSZ          X_DELAY, F      ; passo2        1 cycle
                    GOTO  X_DELAY500_LOOP           ; passo3        2 cycles
X_DELAY500_FIM      RETURN                          ; +2            2 cycles


  END                                               ; PONTO FINAL DO PROGRAMA
```

Anexo E

Extrato do Manual do microcontrolador PIC16F636

Anexo F

Manual do transcoder HCS410