TRABALHO DE GRADUAÇÃO

Sistema Supervisor de Violação Física para Módulo de Segurança Criptográfica

Danielle Almeida Lima

Brasília, dezembro de 2019

ENGENHARIA
MECATRÔNICA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

Sistema Supervisor de Violação Física para Módulo de Segurança Criptográfica

Danielle Almeida Lima

Relatório submetido como requisito parcial de obtenção de grau de Engenheiro de Controle e Automação

Banca Examinadora

Prof. Adolfo Bauchspiess, ENE/UnB _____
Orientador

Eng. Enilton Antonio do Nascimento Júnior, Dinamo Networks _____
Co-orientador

Prof. José Camargo da Costa, ENE/UnB _____
Examinador interno

Brasília, dezembro de 2019

FICHA CATALOGRÁFICA

Danielle, Lima
Sistema Supervisor de Violação Física para Módulo de Segurança Criptográfica

[Distrito Federal] 2019.

x, 101p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2019). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1. Criptografia
2. Módulos de Segurança Criptográficas
3. Segurança Física
4. Ataques Físicos
4. Sistema Supervisor de Violação Física

I. Mecatrônica/FT/UnB
II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

LIMA, DANIELLE, (2019). Sistema Supervisor de Violação Física para Módulo de Segurança Criptográfica. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-$n°$05, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 101p.

CESSÃO DE DIREITOS

AUTOR: Danielle Almeida Lima

TÍTULO DO TRABALHO DE GRADUAÇÃO: Sistema Supervisor de Violação Física para Módulo de Segurança Criptográfica.

GRAU: Engenheiro ANO: 2019

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Danielle Almeida Lima

QNL 03 Bloco J casa 08, Taguatinga Norte.

7150-320 Brasília – DF – Brasil.

Dedicatória

Dedico este trabalho à minha família, ao meu namorado e aos meus amigos que foram fonte de amor e apoio durante toda essa trajetória.

Danielle Almeida Lima

Agradecimentos

Agradeço aos meus chefes Enilton Nascimento e Jorge Santos pela oportunidade e confiança de dar continuidade a um dos projetos da empresa, por todo apoio e suporte sem medir esforços ao me auxiliar e dar dicas.

Agradeço ao professor Adolfo por ter aceitado ser meu orientador, por toda atenção, correções, paciência e conhecimento. Agradeço ao professor José Camargo por todas as consultorias, ajuda e disponibilidade para realizar as medidas no laboratório.

Agradeço aos meus pais e avós por proporcionarem que eu realize todos os meus sonhos. Agradeço ao meu namorado por sempre acreditar em mim, por apoiar os meus sonhos e por me inspirar na busca de ser melhor a cada dia. Agradeço aos meus amigos que sempre estiveram ao meu lado me apoiando e motivando.

Agradeço também a faculdade por ter me ensinado a não desistir perante os fracassos. Sou muito grata à todos os ensinamentos e conhecimentos adquiridos durante esses cinco anos e meio e por todas as pessoas que fizeram parte dessa caminhada.

Danielle Almeida Lima

RESUMO

Muitas empresas possuem informações que necessitam de um nível de proteção elevado. O Módulo de Segurança Criptográfica (MSC) ou *Hardware Security Module* (HSM) é um dispositivo de criptografia, fisicamente seguro, projetado e utilizado para realizar autenticação forte, garantir o sigilo, integridade e autenticidade das informações armazenadas sob a responsabilidade da empresa, como número de cartões de crédito, dados sigilosos de transações, informações pessoais de usuários e históricos de consumo. É proposta deste projeto implementar um sistema supervisor de violação física para um MSC de rede. Este circuito tem como função detectar a tentativa de violação desse equipamento a partir da remoção da tampa superior. Desse modo, para cumprir com o seu objetivo são utilizados sensores com o intuito de enviar sinais caso ocorra uma invasão, além disso é necessário manter uma área de memória volátil segura, emitir alertas ao identificar uma violação e possuir a capacidade de gerenciamento de energia, uma vez que caso o sistema esteja desenergizado, as informações presentes na memória volátil devem ser mantidas.

A detecção foi feita com o auxílio do chip supervisor DS3645 que é responsável por detectar a violação e tomar a decisão de cortar a memória volátil do circuito. Na primeira etapa do projeto foi utilizado o Arduino Due que é uma placa baseada no microcontrolador arm de 32 bits, o *Atmel SAM3X8E ARM Cortex-M3*, para realizar todos os testes do chip e armazenar o programa desenvolvido para a função de detecção e de comunicação com a placa mãe do HSM. Na segunda etapa foi desenvolvida uma placa proprietária no software EAGLE para ser inserida no Arduino Due com o chip supervisor e mais alguns componentes auxiliares, como cristal, diodo, relé, resistores e transistores. Além de utilizar os sensores do tipo *micro switch*, também foi usado o sensor de temperatura presente no chip para detectar ataques do tipo impressão de temperatura. Esse sistema será submetido a um processo de homologação das normas MCT-7 e FIPS 140-2 para o nível de segurança física 3.

Palavras Chave: Criptografia, Segurança Física, Sistema Supervisor e Violação Física.

ABSTRACT

Most companies have information that require protection on an extremely high level. The Hardware Security Module (HSM) is a encryption device, physically secure, designed and used to perform strong authentication, ensure the confidentiality, integrity and authenticity of information stored under the company's responsibility, such as credit card numbers, sensitive transaction data, personal user information, and consumption histories. This project proposes the implementation of a tamper monitoring system for a network HSM. This circuit is intended to detect attempted tampering of this equipment by removing the top cover. Thus, in order to comply with its objective, sensors such as micro switches are used in order to send signals in case of violation, besides performing signal monitoring it is necessary to maintain a safe non-volatile memory area, and have the ability to manage power, since in case the system is de-energized, the information present in the volatile memory must be maintained.

The detection was done with the help of the DS3645 supervisory chip which is responsible for detecting the breach and making the decision to cut the circuit's volatile memory. In the first stage of the project was used Arduino Due which is a board based on a 32-bit ARM core microcontroller Atmel SAM3X8E ARM Cortex-M3 CPU, to perform all tests of the chip and store the program developed for the function of detection and communication with the HSM motherboard. In the second stage, a proprietary board in the EAGLE software was developed to be inserted into the Arduino Due with the supervisor chip and some auxiliary components, such as crystal, diode, relay, resistors and transistors. In addition to using micro switch sensors, the temperature sensor on the chip was also used to detect temperature impression attacks. This system will be subjected to an approval process of the MCT-7 and FIPS 140-2 standards for the level of physical security 3.

Keywords: Cryptography, Physical Security, Supervisor System, Physical Violation.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

Siglas e Acrônimos

AC	Autoridade Certificadora
HSM	*Hardware Security Module*
ICP	Infraestrutura de Chaves Públicas
ITI	Instituto Nacional de Tecnologia da Informação
MSC	Módulo de Segurança Criptográfico
NF-e	Nota Fiscal Eletrônica
CI	Circuito Integrado
IR	Infravermelho
PCB	Placa de Circuito Impresso
SVMK	*Server Master Key*
RTC	Relógio de Tempo Real
EDA	*Electronic Design Automation*
BGA	*Ball Grid Array*
SMD	*Surface Mounting Device*
CSBGA	*Chip-scale Ball Grid Array*
I2C	*Inter-Integrated Circuit*
IDE	*Integrated Development Environment*

Capítulo 1

Introdução

1.1 Contextualização

A revolução digital viabilizou o armazenamento e o processamento de uma vasta quantidade de informações, bem como a comunicação desses dados a uma alta velocidade por todo o mundo. Este avanço impacta diretamente a vida profissional e privada dos seres humanos, devido às suas interações com o mundo online. É importante que as informações compartilhadas sejam seguras [40], ou seja, a confidencialidade, integridade e autenticidade dos dados sejam garantidas.

Nas últimas décadas a ciência da criptologia desenvolveu um conjunto de ferramentas poderosas para proteger digitalmente os dados armazenados e em trânsito, criar interações seguras, garantir os direitos digitais e a privacidade [40]. Os processos criptográficos são responsáveis por prover a segurança lógica. A criptografia é baseada em algoritmos, protocolos, geração de números randômicos e dados secretos [41]. Sua implementação não é suficiente para garantir a completa segurança de uma unidade de processamento, mecanismos físicos de proteção também são necessários.

Tradicionalmente, o termo segurança física tem sido usado para descrever a proteção ativa de materiais contra o fogo, danos causados pela água, roubo, ou perigos similares. Entretanto, preocupações recentes com a segurança computacional fizeram com que a segurança física assumisse um novo significado, no qual tecnologias são usadas para proteger informações contra ataques físicos. Neste novo sentido, é uma barreira colocada em torno de um sistema computacional para deter o acesso físico não autorizado ao próprio sistema [43]. Este conceito é complementar ao de segurança lógica que está relacionada aos mecanismos pelos quais o sistema operacional e outros softwares impedem o acesso não autorizado aos dados [42].

A segurança física é frequentemente vista em segundo plano, sendo muitas vezes negligenciada [20]. A maioria das organizações se concentram em contramedidas de segurança orientadas à tecnologias para evitar ataques de hackers. As invasões de sistemas de rede não são o único modo pelo qual as informações confidenciais podem ser roubadas ou usadas contra uma empresa. Todos os firewalls, criptografia e outras medidas de segurança seriam inúteis caso não haja uma barreira física de proteção [21].

Um equipamento que agrega em si os conceitos de segurança física e lógica é o HSM (*Hardware Security Module*), também conhecidos como MSC (Módulo de Segurança Criptográfica). Segundo [22], O HSM é um servidor ou placa auxiliar criptográfica fisicamente segura resistente à violação que fornece funcionalidades criptográficas com capacidade de geração e armazenamento de chaves criptográficas simétricas e assimétricas voltados para utilização em uma Infraestrutura de Chaves Públicas (ICP). O MSC pode ser utilizado em bancos para transações financeiras, em tribunais para processos e dados sigilosos, em clínicas e hospitais para prontuários, em empresas que usam NF-e (Nota Fiscal Eletrônica), em escritórios de advocacia, telecomunicações, governo e comunidades internet.

Em Módulos de Segurança Criptográficas, a fronteira criptográfica define um perímetro no qual estão contidos componentes tais como processador(es), memórias e outros dispositivos de hardware, software e firmware [22]. Desse modo, para garantir a integridade, confidencialidade, autenticidade e disponibilidade dos dados armazenados no HSM, além da segurança lógica, a fronteira criptográfica é delimitada por mecanismos de segurança física que protegem contra sondagem, observação e manipulação direta. Com o crescente avanço nas tecnologias de ataques físicos e a necessidade de manter os dados seguros, os MSCs necessitam de aparatos eficientes para resistir, evidenciar, detectar e responder às violações físicas.

1.2 Objetivos

Os objetivos deste trabalho foram divididos em duas categorias: objetivos gerais e objetivos específicos. Ambos estão descritos a seguir.

1.2.1 Objetivos Gerais

A proposta central deste projeto é desenvolver um subsistema supervisor para garantir a segurança física de equipamentos criptográficos, de modo a evitar acessos não permitidos, bem como para deter o uso, modificações ou até mesmo substituição não autorizada dos componentes. O supervisor de segurança tem como função o monitoramento do perímetro e acionamento caso seja detectada uma violação física.

1.2.2 Objetivos Específicos

A proposta específica deste projeto é realizar a proteção da fronteira criptográfica de um HSM de rede. Desse modo, serão necessários os seguintes tópicos:

- Desenvolver um sistema com auxílio de um chip supervisor dedicado para detectar e responder a violação;
- Selecionar os sensores para detectar a violação, monitorar a temperatura e a tensão de alimentação;

- Desenvolver um sistema de gerenciamento de energia que permita manter dados críticos na memória volátil em caso de falha de energia e apagá-los em caso de violação;
- Desenvolver um firmware que implementa uma máquina de estados e o protocolo de comunicação com o HSM;

O cenário para teste e validação do circuito supervisor de violação de fronteira é o HSM de rede da empresa Dinamo Networks.

1.3 Apresentação do manuscrito

O trabalho tem seis capítulos. Para dar uma visão geral do texto, apresenta-se aqui o formato do trabalho. O capítulo atual, capítulo 1, contém a introdução, a contextualização, os objetivos gerais e específicos desse projeto.

O segundo capítulo descreve a fundamentação teórica e a literatura correlata utilizada como base para o trabalho, explicando os mais diversos conceitos englobando criptografia, segurança física, seus tipos, métodos, mecanismos e ataques e relacionando os conceitos de segurança física a um Módulo de Segurança Criptográfica.

O terceiro capítulo cita em detalhes os materiais e as ferramentas utilizadas no decorrer do projeto, assim como as suas especificações e os seus papéis no projeto. O capítulo é finalizado com um visão geral do sistema desenvolvido.

O quarto capítulo descreve os procedimentos e métodos seguidos no decorrer do projeto. Neste capítulo é mostrado como o hardware e o software foram desenvolvidos. A máquina de estados do funcionamento e o protocolo de comunicação são descritos, bem como o modelo operacional do sistema.

O capítulo cinco apresenta os resultados obtidos que validam todo o processo descrito pela metodologia. Já o capítulo seis, por sua vez, conta com a conclusão do trabalho. Além das sugestões para trabalhos futuros.

Capítulo 2

Fundamentos Teóricos

2.1 Criptografia

A criptografia é a arte e a ciência de manter mensagens seguras. Uma mensagem em texto claro ao ser criptografada torna-se um texto cifrado, assim ao ser decriptografado retorna ao estado inicial, ou seja a mensagem em texto claro [35], essa descrição é apresentada na figura 2.1. Além de garantir a confidencialidade, a criptografia também é utilizada para executar outras funcionalidades, como autenticidade, integridade e disponibilidade.

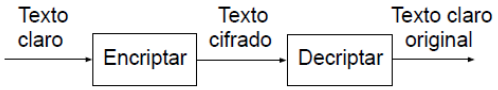


Figura 2.1: Processo de encriptar e decriptar uma mensagem em texto claro.

A autenticidade é a propriedade de ser genuíno e capaz de ser verificado e confiável seja na validação de uma transmissão, em uma mensagem ou na sua origem. A integridade está relacionada a prevenção contra a modificação ou destruição imprópria de informações. A disponibilidade assegura o acesso e uso rápido e confiável de informação. A confidencialidade tem como objetivo preservar restrições autorizadas sobre acesso e divulgação de informação, incluindo meios para proteger a privacidade de indivíduos e informações privadas [38].

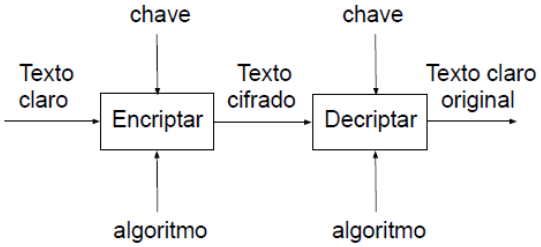


Figura 2.2: Modelo simplificado do processo de encriptar e decriptar uma mensagem.

Tanto para encriptar quanto para decriptar é necessário a utilização de uma chave secreta e de um algoritmo, assim como é apresentado na figura 2.2. A chave é uma entrada para o algoritmo e um valor independente do texto claro e do algoritmo. O algoritmo produzirá uma saída diferente, dependendo da chave usada no momento. As substituições e transformações exatas realizadas pelo algoritmo dependem da chave [38].

O processo de encriptação pode ser simétrico ou assimétrico. Na criptografia simétrica, a mesma chave é utilizada para encriptar e decriptar uma mensagem, já na criptografia assimétrica a chave usada para encriptar não é a mesma usada para decriptar. As funções de hash e a assinatura digital são algoritmos criptográficos. Uma boa função de hash garante que os resultados da aplicação da função a um grande conjunto de entradas produzirá saídas que são distribuídas por igual e de modo aleatório [38].

A assinatura digital é o conjunto de dados sob forma eletrônica, ligados ou logicamente associados a outros dados eletrônicos, utilizado como método de comprovação da autoria [23]. Utiliza o método de autenticação dos algoritmos de criptografia com chaves assimétricas operando em conjunto com uma função de hash com o intuito de verificar se o emissor da mensagem é aquele quem se diz ser.

A figura 2.3 mostra o processo criptográfico de criação de uma assinatura digital de modo simplificado. O signatário gera um hash de um documento eletrônico, em seguida, o cifra com sua chave privada gerando a assinatura digital. Para realizar esse procedimento é necessário utilizar uma chave privada para o processo de geração de assinatura e uma chave pública contida no certificado digital para a verificação da assinatura.

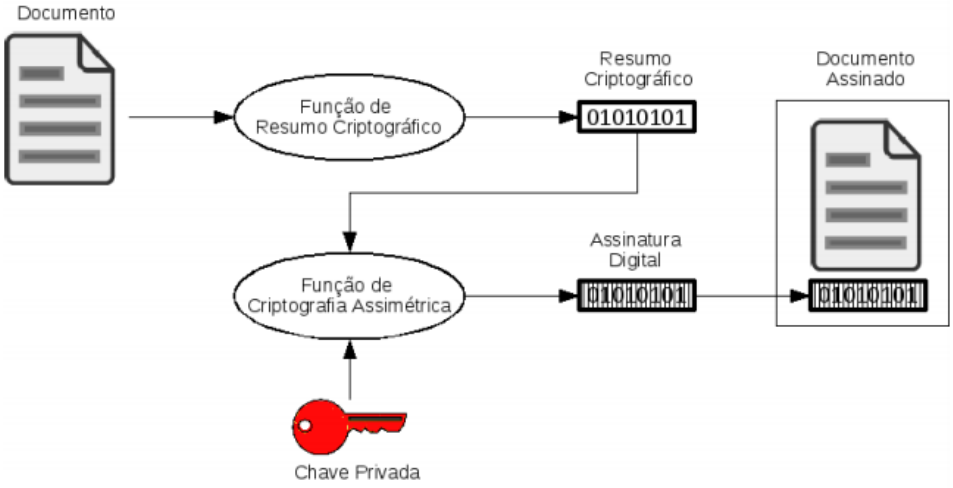


Figura 2.3: Modelo simplificado da criação de uma assinatura digital [23].

Considere que um banco quer iniciar uma comunicação confiável e criptografada com outro banco. Desse modo, para que essa situação ocorra é necessário que eles troquem a chave pública entre eles, além disso é preciso que se garanta que aquela chave pública pertença ao parceiro de comunicação. O certificado digital e a autoridade certificadora permitem a viabilidade e segurança dessa situação.

O certificado digital é um documento eletrônico que identifica pessoas e empresas no mundo digital, comprovando sua identidade. Permite acessar serviços on-line e assinar documentos eletrônicos com possibilidade de certificação da autenticidade e da integridade [36]. Os certificados utilizam criptografia para cifrar e decifrar as assinaturas utilizando chaves públicas e privadas.

A chave pública pode ser acessada e conhecida por todos, são dados do proprietário do próprio certificado. As principais informações que constam em um certificado digital são a chave pública, nome e e-mail do titular, período de validade do certificado, nome da Autoridade Certificadora (AC) que emitiu o certificado, número de série do certificado digital, assinatura digital da AC. Em relação à chave privada, somente o proprietário do certificado deve conhecer a chave privada, ou seja, é a senha de acesso ao certificado digital [44].

No Brasil quem controla o processo de certificação digital é o Instituto Nacional de Tecnologia da Informação (ITI) que cuida das inovações em tecnologia de ampliação da cidadania digital. A ICP define as diretrizes e normas que devem ser seguidas. Todas as certificadoras e empresas que fornecem esse serviço devem seguir os padrões estabelecidos por ela.

Uma Autoridade Certificadora é uma entidade, pública ou privada, subordinada à hierarquia da ICP-Brasil, responsável por emitir, distribuir, renovar, revogar e gerenciar certificados digitais. Tem a responsabilidade de verificar se o titular do certificado possui a chave privada que corresponde à chave pública que faz parte do certificado. Cria e assina digitalmente o certificado do assinante, onde o certificado emitido pela AC representa a declaração da identidade do titular, que possui um par único de chaves (pública/privada) [24].

2.1.1 Módulo de Segurança Criptográfica

O Módulo de Segurança Criptográfica (MSC), também conhecido como *Hardware Security Module* (HSM) é um dispositivo de armazenamento das chaves que são utilizadas pelas aplicações que necessitam fazer criptografia e/ou assinatura. Seu objetivo é prover um armazenamento seguro das chaves criptográficas para que terceiros não a utilizem para forjar assinaturas, nem a visualizem ou acessem os dados que foram encriptados pelo par público relacionado a chave.

Ademais, faz o gerenciamento de chaves digitais acelerando o processo criptográfico fornecendo autenticação forte para acesso às chaves críticas para aplicativos de servidor. Esses dispositivos podem ser fornecidos na forma de placas de expansão ou de um hardware de rede. A função de criptografia é realizada por outra máquina na rede, apresenta maior segurança, uma vez que a chave privada nasce dentro do dispositivo, o uso das chaves em um ambiente distribuído é simples. Seu software exige alto nível de segurança das chaves, redundância, alta disponibilidade e elevado desempenho em criptografia simétrica e assimétrica.

Desse modo, a função do MSC é ser um repositório seguro para chaves criptográficas e uma plataforma de criptografia para aplicações, além disso é um gerenciador de chaves, tendo em vista que pode gerar, usar, acessar, importar, exportar e destruir chaves. O HSM pode ser compartilhado por múltiplos usuários e aplicações com separação em partições, as chaves são separadas e não podem ser acessadas por usuários não autorizados, nem mesmo por administradores, isso fornece

a garantia do nível de segurança.

2.2 Segurança Física

A maioria dos engenheiros de segurança na atualidade estão preocupados com a proteção lógica, entretanto há várias razões pelas quais a proteção física não pode ser negligenciada. Possuir paredes e bloqueios são um fator importante em relação a estratégia geral de gerenciamento de risco de uma empresa [6], além disso independentemente de possuir a melhor proteção lógica, caso sua barreira física não seja eficaz, seus dados estarão vulneráveis.

A proteção física não é diferente no cerne da segurança computacional, ou seja, é realizada uma análise de ameaças, em seguida, projeta-se um sistema que envolve equipamentos e procedimentos, posteriormente, é testado. O próprio sistema apresenta vários elementos, entre eles, detectar, resistir, alarmar, responder e evidenciar. São utilizados mecanismos como sensores e barreiras para evitar e detectar intrusos, bem como medidas de controle operacionais.

A segurança física também está se tornando mais importante, uma vez que os sistemas computacionais estão se mudando para ambientes mais seguros. Ao mesmo tempo, o valor dos dados nesses sistemas está aumentando. Com o aprimoramento da segurança lógica, os ataques físicos passam a ser mais fáceis de realizar do que os ataques lógicos [7]. Pode-se notar que a motivação para realizar ataques de sistemas computacionais aumentou, devido a elevação das recompensas.

Para que a segurança física seja efetiva, caso um ataque ocorra, deve-se haver a baixa probabilidade de sucesso e alta probabilidade de detecção durante o ataque ou após a invasão [2]. Os sistemas que fornecem essa proteção devem tornar difícil o acesso não autorizado a dados, como um cofre de banco torna a tarefa de roubar dinheiro árdua, esse mecanismo resiste a violação, também podem utilizar sistemas de alarmes que impedem ataques e são caracterizados como um sistema que responde a violação, outro modo é fazer com que a tentativa de ataque se torne aparente para caso uma inspeção subsequente seja realizada, o ataque seja identificado, esse mecanismo é classificado como o que evidencia uma violação.

Sistemas de classificação que avaliam sistemas de computação de acordo com critérios que medem a dificuldade de montar um ataque bem-sucedido foram propostos. A exigência de documentação adicional, testes e qualidade garantem graus crescentes de segurança. O trabalho crescente levou ao avanço dos padrões [39], estas normas são responsáveis por avaliar de maneira rigorosa e detalhada, se os padrões estão sendo atendidos.

A tecnologia de segurança física adicionada ao design dos sistemas computacionais é relativamente nova. Um grande número de métodos para segurança física está atualmente em uso, por ser um novo campo no mercado comercial ainda está em desenvolvimento. Sua ampliação é contínua em decorrência dos avanços nas metodologias de ataque. A avaliação de adequação de um sistema de segurança física é dependente do tempo e deve ser repetida periodicamente.

2.2.1 Tipos de Segurança Física

2.2.1.1 Resiste a Violação

A resistência à violação consiste na utilização de materiais especializados com o intuito de dificultar a adulteração de um módulo ou dispositivo. É o mecanismo mais fácil de aplicar. Existem variadas maneiras de restringir o acesso físico a um dispositivo a partir da resistência. Abaixo segue uma lista com tais métodos:

- As tecnologias de cofre bancário utilizam as técnicas de tornar o dispositivo muito grande ou pesado para que diminua significativamente a probabilidade de um ataque. Verifica-se que isso não é conveniente para dispositivos portáteis, assim novas tecnologias foram desenvolvidas [3].
- Uma barreira física dura envolvendo o dispositivo. Materiais como aço, plásticos duros, cerâmica, cimento ou tijolo resistem à violação. A figura 2.4 apresenta um exemplo desse método.

Figura 2.4: Chips envolvidos por uma barreira resistente [27].

- Uma gaiola metálica ajuda a proteger o dispositivo contra interferências eletromagnéticas e a incorporação de camadas de metal à placa de circuito ajudam na detecção das causas do campo magnético [27].
- Materiais como SiMOX (silício/óxido de metal) e SOS (silício com safira) são utilizados para detectar a radiação infravermelha.
- Chip projetado de um certo modo que seja possível garantir que as camadas com os dados que precisam ser protegidos sejam envolvidas pelas camadas responsáveis pelas funcionalidades. Isso garante que as áreas secretas não sejam expostas sem remover ou danificar as camadas de funcionalidades necessárias para ler o dado protegido.
- Bits seguros destinados a dificultar o processo de abertura do dispositivo, além disso para ocultá-los são utilizados etiqueta e bumpons. A figura 2.5 mostra os padrões desses bits.

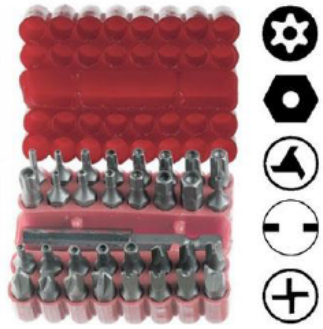


Figura 2.5: Bits seguros [13].

2.2.1.2 Evidencia a Violação

Os sistemas que evidenciam a violação são projetados para garantir que, se ocorrer uma invasão uma evidência será gerada. Esses sistemas não protegem do ataque em si, apenas provam a ocorrência de uma violação. Meios químicos e mecânicos são utilizados para gerar as evidências. Como esses sistemas não ativam alarmes ou notificam o proprietário de que um ataque ocorreu, é necessário uma supervisão frequente do equipamento, tendo em vista que esse método gera visualmente uma evidência do ataque. Além disso, seu uso é combinado com o sistema que responde a violação para alertar e provar que um ataque realmente ocorreu. Abaixo segue uma lista com os métodos:

- Pacotes frágeis é a maneira mais trivial de provar que um dispositivo foi invadido, ao tentar abrir ou penetrar o equipamento uma evidência é gerada, já que ao ser violado o pacote é difícil de reconstruir.
- Materiais polidos como o alumínio ou outro semelhante submetidos a um tratamento térmico provocam evidências aos serem violados como rachaduras e impressões digitais, desse modo ao supervisionar por intermédio de fotografias ou dispositivos de comparação óptica é possível detectar a violação.
- Lacres de segurança, como etiquetas void e fita casca de ovo geram evidências caso haja a tentativa de invasão, seja marcando o equipamento ou se quebrando quando há a tentativa de remoção.

Figura 2.6: Lacre para proteção do parafuso.

Figura 2.7: Etiquetas void [34].

2.2.1.3 Detecta a Violação

A detecção de uma violação é feita a partir da instalação de sensores no dispositivo. A forma e o tipo do sensor depende da finalidade para a qual ele é utilizado, entretanto independentemente do tipo, é o sensor que fornece uma saída quando o ataque é detectado. Uma característica importante de um circuito de detecção de violação é o seu funcionamento sem interrupções, ou seja, mesmo que o equipamento esteja ligado ou não à rede elétrica, o circuito deve estar operando. Para atender a esse requisito o circuito deve consumir baixa energia que é fornecida por uma bateria por um período maior do que 5 anos. Abaixo, seguem os mecanismos utilizados:

- Os switches são dispositivos que detectam o movimento mecânico, como o da tentativa de remoção da tampa ou de um componente ou de quando a barreira física é violada.

Figura 2.8: Microswitch [13].

- Os sensores de temperatura detectam mudanças ambientais na temperatura de funcionamento, os sensores de tensão verificam mudanças na tensão e os de raio-x identificam íons e feixes.
- Os componentes eletrônicos detectam e monitoram as mudanças nas frequências, pulsos de clock ou tensões que entram e saem do chip [33].

- Manta mesh e fibras ópticas são utilizadas para envolver a área crítica de hardware para detectar uma tentativa de violação, os sensores de monitoramento desses dispositivos reconhecem pequenas mudanças na capacitância ou resistência da malha.

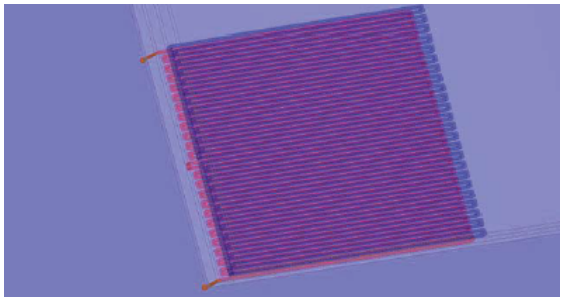


Figura 2.9: Estrutura da manta mesh [41].

Os circuitos de detecção de violação são de dois tipos: passivos e ativos. O tipo passivo mede as variações de corrente na estrutura da malha condutiva e dispara o evento de violação caso o circuito abra. Devido as pequenas correntes usadas, o método é suscetível a interferências eletromagnéticas, assim deve ser blindado para evitar o disparo falso. Já o circuito do tipo ativo realiza a sondagem periódica da manta mesh, por meio do envio de sinais em uma extremidade e a análise da resposta na outra. Caso determinados parâmetros mudem, o circuito detecta a invasão [41].

2.2.1.4 Responde a Violação

Após a detecção da violação, o sistema precisa responder ao ataque impedindo que o invasor acesse os dados secretos. Essas respostas podem ser o soar de um alarme, limpar a ROM ou destruir o próprio dispositivo físico. Além disso, é responsável por capturar a saída gerada pelo sistema de detecção.

Muitas vezes, um ataque é detectado antes que o invasor tenha obtido todos os dados necessários e, nesses casos, é essencial que o dispositivo tente impedir que o invasor obtenha o acesso ao restante dos dados. Na maioria dos dispositivos embarcados e *smart cards*, os segredos são armazenados na memória RAM ou ROM. Enquanto a memória RAM é relativamente fácil de limpar durante o ataque, a ROM é significativamente mais difícil.

O modo mais simples de apagar os dados da RAM é fazer a retirada da energia, o que efetivamente elimina o conteúdo. Outro modo, é sobrescrever os dados, esse método requer o uso de energia. A maneira mais eficaz de garantir que um invasor não tenha acesso aos dados é destruir completamente o dispositivo quando um ataque é detectado, isso pode ser feito a partir de um curto circuito em alguma parte do circuito. A destruição pode ser tão sutil que o responsável pelo ataque não percebe que o dispositivo parou de funcionar.

A figura 2.10 apresenta uma nova tecnologia capaz de fazer com que os chips de computador se autodestruam quando disparados remotamente. Esse método utiliza pastilhas de silício do computador ligados a um pedaço de vidro temperado que se rompe ao ser aquecido.

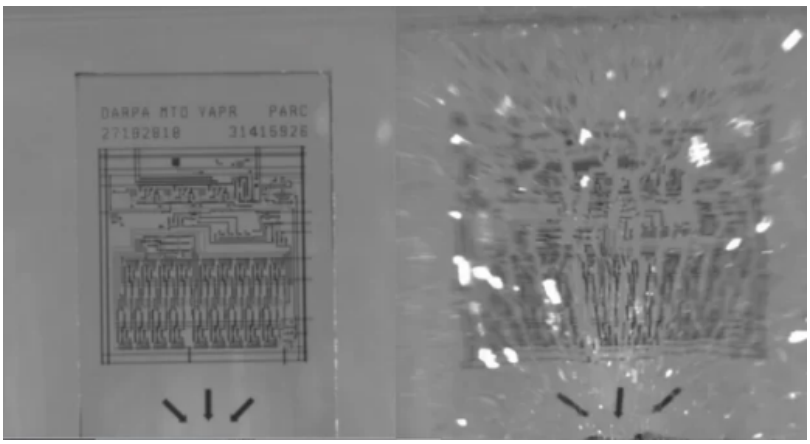


Figura 2.10: Chip autodestrutivo [13].

2.2.2 Níveis de Segurança

A norma FIPS 140-2 [17] especifica os requisitos de segurança que serão atendidos por um módulo criptográfico utilizado em um sistema de segurança com proteção à informações sensíveis. O padrão fornece quatro níveis crescentes e qualitativos de segurança: Nível 1, Nível 2, Nível 3 e Nível 4. Estes níveis destinam-se a cobrir a ampla gama de potenciais aplicações e ambientes nos quais módulos criptográficos podem ser empregados.

- Nível 1: define os requisitos básicos de segurança para um módulo criptográfico, deve possuir pelo menos um algoritmo ou função de segurança aprovado, além disso não são necessários mecanismos de segurança física específicos, apenas deve atender aos requisitos básicos para componentes de nível de produção. Esse nível permite que os componentes de software e firmware de um módulo criptográfico sejam executados em um sistema de computação de propósito geral usando um sistema operacional não avaliado.

- Nível 2: aprimora os mecanismos de segurança física adicionando o requisito de evidência de violação, o qual inclui o uso de revestimentos ou vedantes invioláveis, fechaduras resistentes, desse modo o acesso físico só ocorre por meio da quebra dos mecanismos. Além disso, requer autenticação baseada em função, ou seja autentica a autorização de um operador para assumir uma função específica e executar um conjunto correspondente de serviços.

- Nível 3: exige a tentativa de impedir que o invasor tenha acesso aos dados mantidos dentro de um módulo criptográfico. Os mecanismos de segurança física devem ter alta probabilidade de detectar e responder a tentativas de acesso físico, uso ou modificação do módulo. Exemplos de métodos utilizados são gabinetes fortes, com detecção, resposta à violação e circuitos que apagam os parâmetros críticos de segurança quando as tampas ou portas removíveis são abertas. Os mecanismos de autenticação são baseados em identidade. Tanto a entrada, quanto a saída de parâmetros críticos de segurança de texto claro devem ser executadas usando portas fisicamente separadas de outras portas ou interfaces que são logicamente separadas usando um caminho confiável.

- Nível 4: os mecanismos de segurança física fornecem um envelope completo de proteção em torno do módulo criptográfico com a intenção de detectar e responder a todas as tentativas não autorizadas de acesso físico. A tentativa de violação em qualquer direção possui a probabilidade muito alta de ser detectada resultando na destruição de todos os parâmetros críticos de segurança em texto simples. Esse nível fornece a proteção contra o comprometimento da segurança devido a condições ambientais ou flutuações fora das faixas operacionais do módulo para tensão e temperatura. Os equipamentos que atendem a esse nível de segurança são submetidos a testes de falhas ambientais.

2.3 Ataques

Os ataques podem ser usados para finalidades diferente, dependendo do objetivo. Quem projeta um produto deve primeiramente pensar sobre os possíveis motivos que levam alguém a querer realizar o ataque do mesmo, e em seguida, concentrar-se na proteção com mecanismos. Há basicamente três classes de atacantes representadas na tabela 2.1. Essa classificação é baseada em [4].

Tabela 2.1: Comparação entre cada classe de atacante e os recursos disponíveis.

Recursos	Classe I	Classe II	Classe III	Classe III
Categoria	*Script-kiddie*	Acadêmico	Crime Organizado	Governo
Tempo	Limitado	Moderado	Grande	Grande
Despesas	<$1000	$10k - $100k	>$100k	Desconhecido
Criatividade	Varia	Alta	Varia	Varia
Detectabilidade	Alta	Alta	Baixa	Baixa
Objetivo	Desafio/prestígio	Publicidade	Dinheiro	Varia
Número	Muitos	Moderados	Poucos	Desconhecido
Organizado	Não	Não	Sim	Sim
Libera Informação	Sim	Sim	Varia	Não

Ademais, J. Grand [18] definiu quatro classes principais de ameaças à segurança, definidas da seguinte maneira:

- Interceptação ou espionagem: acesso a informações protegidas sem abrir o produto. Um interceptor silencioso pode não deixar rastros pelos quais a interceptação possa ser prontamente detectada.
- Interrupção ou geração de falhas: um ativo de um produto torna-se indisponível, inutilizável ou removido. Um exemplo, é a destruição maliciosa de um dispositivo de hardware, o apagamento intencional de conteúdo de programa ou de dados ou um ataque de rede de negação de serviço. A geração de falhas consiste em provocar intencionalmente o mau funcionamento, o que pode levar a ruptura de determinadas medidas de segurança.

- Estilo de modificação: violação de um ativo de um produto. A modificação é tipicamente uma técnica invasiva tanto para o hardware, quanto para o software/firmware, seja pela modificações de circuitos ou micropadrões, como alteração dos valores de dados ou a modificação de um programa para que ele execute uma computação diferente.

- Estilo de fabricação: criação de ativos falsificados em um produto ou sistema, como a adição de dados em um dispositivo, inserindo transações espúrias em um barramento ou interface, ou um ataque *Man-in-the-Middle* em uma rede. Às vezes, esses acréscimos podem ser detectados como falsificações, entretanto se feito de maneira hábil, eles podem ser indistinguíveis do produto real.

O tipo de ataque depende do objetivo da invasão, de qual classe o atacante faz parte, de qual é o tipo do dispositivo e quais métodos de segurança foram implementados durante a construção dos dispositivos [3].

2.3.1 Ataques Invasivos

Os ataques invasivos tendem a destruir a embalagem e em alguns casos o dispositivo inteiro, além disso exigem equipamentos muito caros. Esse tipo de ataque só é viável em situações em que a destruição do dispositivo não importa ou apesar do dano é possível reconstruir o dispositivo de modo que haja pouca ou nenhuma evidência do ataque. Abaixo, seguem algumas classificações dos tipos mais comuns de ataques invasivos.

- Ataques de sondagem: tem como objetivo conectar diretamente um condutor ao circuito que está sendo protegido, de modo que as informações possam ser obtidas ou que as alterações possam ser injetadas no sistema [43]. Esse tipo de ataque [25] também é comumente usado como o primeiro passo para ataques mais avançados, considerando que o atacante tenha equipamentos de sondagem no local, ele pode fazer diversos ataques, como de temporização [33], baseados em cache [30], de monitoramento de energia [28, 25, 33, 27], como análise de potência simples e análise de potência diferencial, bem como ataques de análise de falhas [27, 28, 5].

- Métodos de usinagem: outro ataque invasivo em *smart cards* ou dispositivos embarcados consiste em cortar partes do chip, peça por peça até que o invasor entenda como dispositivo foi construído. Normalmente, os circuitos integrados são embalados por uma cobertura ou outro revestimento resistente a violações, com o intuito de garantir que o ataque de sondagem não seja feito. Assim, ao usinar o chip e remover as tampas e os revestimentos é possível alcançar o circuito real e usar um ataque de sondagem. A usinagem pode ser feita manualmente, geralmente com o atacante usando uma faca de outra ferramenta para remover o material do dispositivo. Já a usinagem mecânica é o processo automatizado para remoção de material. O tipo mais preciso de usinagem é feito usando água desmineralizada, deionizada ou pura ou um laser. A água pura possui a vantagem de não ser condutora, desse modo é difícil detectar o seu uso, entretanto os equipamentos que utilizam essa técnica

são muito grandes e geralmente disponíveis apenas para alguns atacantes da classe III. A usinagem a laser possui as mesmas vantagens da água pura, todavia seu equipamento é menor e possui a desvantagem de gerar calor. O último tipo envolve o uso de produtos químicos, esse método é semelhante ao da água, entretanto produtos corrosivos são utilizados para dissolver o material de modo rápido e eficiente, como os agentes químicos são condutores, são mais fáceis de detectar e podem até causar curtos-circuitos não intencionais [43].

- Tecnologia de carga moldada: é uma carga explosiva moldada para focalizar o efeito da energia do explosivo. Usando explosivos minúsculos, é possível penetrar em um circuito integrado (CI) tão rapidamente que os circuitos que detectam intrusões podem ser desativados antes que possuam a chance de responder. Como as explosões fazem com que os cortes sejam feitos em velocidades hipersônicas de até mais de 7 km/s, quase não há tempo para o circuito sinalizar seus alarmes. Uma desvantagem deste método é o fato de ser puramente destrutivo e relativamente impreciso.

- Falhas: modificar as entradas de um microchip de um modo inesperado pode causar falhas no chip e fazer com que o mesmo comece a realizar operações erradas. A falha pode ser causada pela alteração da tensão de entrada, fazendo com que as instruções sejam interpretadas de maneira errada e os circuitos falhem. Fazê-lo no momento correto é vantajoso ao atacante, em decorrência da falta de memória e instruções confusas. Um efeito similar pode ser alcançado alongando e encurtando os pulsos de relógios indo para o CI. As temporizações do chip dessincronizam e resultam em comportamentos errôneos. Outro modo de introduzir falhas é por intermédio de interferência eletromagnética que podem causar rupturas nos diodos e nos circuitos dos transistores [25, 43]. As falhas também podem ser causadas por fatores ambientais e, portanto, não é estritamente um ataque invasivo.

- Microscópios eletrônicos de varredura: os atacantes da classe III que têm acesso a esse equipamento podem usá-lo para ler e possivelmente escrever bits para ROMs ou RAM em um nível molecular. Essa técnica exige que a superfície do chip esteja exposta, desse modo o microscópio pode acessar e ler qualquer parte do chip para obter e possivelmente modificar os segredos armazenados lá.

2.3.2 Ataques Não Invasivos

Os ataques não invasivos são geralmente mais sofisticados em seu design do que os ataques invasivos, e sua implementação geralmente depende de pequenas vulnerabilidades de projeto. Os ataques não invasivos exigem conhecimento detalhado do processador e do software usado, em contraste com um ataque invasivo em que o invasor pode simplesmente sondar a lógica e verificar o que faz o quê. Projetar um ataque invasivo requer conhecimento e habilidade, contudo uma vez que essa técnica esteja disponível para um dispositivo específico e uma versão de software, ela pode ser reproduzida com confiabilidade em segundos em outro dispositivo do mesmo tipo [25]. Abaixo, seguem algumas classificações dos tipos mais comuns de ataques não invasivos.

- Ataques de energia e radiação: podem ser usados para travar ou congelar certas partes de

um circuito em um determinado estado, além disso podem ser tanto invasivos quanto não invasivos e exigem acesso próximo ao dispositivo. Um exemplo, é o ataque de impressão por radiação que consiste em irradiar partes do CI, como o CMOS RAM, de modo que os valores dos bits são "queimados"da memória. Isso significa que uma operação normal de limpeza ou gravação não alterará o valor dos bits nessa ROM. Isso permite que um invasor leia a ROM em um momento posterior sem ter que se preocupar com a perda acidental de dados. Esse processo é similar a impressão por temperatura que consiste em um método que literalmente congela os bits na ROM, de modo que eles possam ser lidos minutos ou até horas após a energia ser removida de um chip. Um laser IR pode ser usado para ler e gravar nas células de uma ROM ou RAM. O silício é transparente para as frequências de infravermelho, portanto, é possível ler ou escrever um valor de bit ao focar um feixe de laser infravermelho em um determinado local no chip, sem exigir que ele seja usinado ou invadido [25, 43].

Figura 2.11: Ataque por impressão de temperatura [13].

- Tecnologias de imagem: a maioria das tecnologias de imagem disponível podem ser usadas para fazer imagens de um chip. Microscópios com dispositivos de gravação, equipamentos de raio X e ultra-som são capazes de ajudar um atacante a visualizar os componentes internos de um chip sem precisar fisicamente abrir ou adulterar o dispositivo [3].
- Ataques de software: são feitos pela comunicação com o dispositivo embarcado pelos canais normais e pela tentativa de aprender mais sobre o dispostivo, explorando as vulnerabilidades de segurança no software [27].
- Técnicas de geração de falhas: geralmente utilizam fatores ambientais externos para causar falhas e outros problemas no dispositivo embarcado. Isso é basicamente uma combinação entre o ataque do tipo falha e energia/radiação, além disso pode ser usado juntamente com ataques baseados em software ou sondagem.

Ao contrário da crença convencional, as memórias voláteis (SRAM, DRAM) não perdem inteiramente seu conteúdo quando a energia é desligada [37]. Para SRAM, à temperatura ambiente, o tempo de retenção varia de 0,1 a 10 segundos, ao resfriá-la a -20^{o}C o tempo de retenção aumenta para 1 segundo até 17 minutos e a -50^{o}C o tempo de retenção é de 10 segundos até 10 horas. Desse modo, o congelamento de memória é um tipo de ataque não invasivo de impressão por temperatura para acesso a dados confidenciais. Já para o caso de retenção de dados em memórias não voláteis (EEPROM, flash, entre outras) pode levar vários ciclos para apagar os dados, assim para superar esse problema é recomendado que se escreva só zeros, só 1 e dados aleatórios na memória.

2.3.3 Tecnologias de Defesa

Os métodos de defesa a seguir se enquadram em três categorias: prevenção de intrusão, detecção de intrusão, detecção de ataques de energia não invasivos (frio, radiação, entre outros). Após a detecção, existem vários métodos de resposta. Cada método deve ser examinado de acordo com as especificidades do dispositivo. Por exemplo, um projeto que exige um sensor de baixa temperatura deve levar em conta as temperaturas às quais a unidade pode ser exposta durante o transporte [43].

- Revestimentos de chip único: essa técnica é usada para evitar ataques de sondagem. A superfície do chip não pode ser sondada, pois há um revestimento que protege de modo que a remoção danifique o chip e não seja possível recuperá-lo.
- Topografias de semicondutores: para evitar ataques de microscópio eletrônico de varredura ou sondagem, mesmo na presença de usinagem química ou outras técnicas que possam remover revestimentos, um chip pode ser projetado de forma a não expor estruturas críticas sem remover as camadas ativas do dispositivo.
- Sensores de tensão: são úteis em quase todos os projetos que exigem uma distribuição de energia adequada para uma operação correta. Com o intuito de garantir o funcionamento adequado dos circuitos, todas as fontes de alimentação devem ser monitoradas. Qualquer operação fora da faixa nominal operacional deve ser considerado um ataque e a resposta deve ser acionada. As referências para os equipamentos de monitoramento devem ser independentes das variações da fonte de alimentação.
- Sensores de sondagem: são barreiras ativas de violação que podem apresentar resistência ou evidências de violação, bem como a sua detecção para segurança adicional.
- Sensores de aceleração: são usados para detectar movimento ou vibração. Seus principais usos são para evitar roubos e detectar perfurações ou marteladas.
- Sensores de placa de circuito impresso (PCB): é similar a manta mesh, entretanto pode ser feito com um custo menor e com os fios sendo impressos em uma PCB. Contudo, o espaçamento regular das linhas e o material condutor usual feito de cobre fornece menos segurança, devido a facilidade com que os condutores podem ser isolados. Quando um condutor é localizado, é mais fácil conectar outro fio a ele com a finalidade de fornecer informações falsas ao circuito de detecção de violação.
- Sensores de movimento: são normalmente usados para detectar movimento em uma área ou caixa e em pares, pois cada tipo pode causar um falso positivo ou falhar em condições não usuais.
- Sensores de fluxo: são usados para detectar a intensidade da radiação em tempo real. Os fototransistores podem ser sensores de fluxo de radiação eficazes. Esses sensores tendem a se degradar com o tempo e a exposição à radiação, pois os fabricantes não especificam a sensibilidade na faixa de radiação de interesse.

2.4 Segurança Física da Fronteira Criptográfica de um HSM

O HSM é um equipamento que deve seguir os padrões internacionais de segurança com o intuito de garantir a segurança das chaves armazenadas nele contra ataques físicos e lógicos. Algumas normas reconhecidas são a FIPS 140-2 [9], 186-4 [17] e Manuais de Condutas Técnicas 7 (MCT-7) da ICP-Brasil [22].

Os objetivos de segurança do HSM é manter a confidencialidade, disponibilidade e integridade dos dados do sistema, bem como manter a capacidade de rastreamento de ações e seus responsáveis. O HSM é projetado para atender o nível de segurança 2 da norma FIPS 140-2 [9]. Esse equipamento possui mecanismos de segurança física que evidenciam, detectam, respondem e resistem à violação para impedir o acesso ao seu interior, e assim garantir a integridade evitando uso, modificação ou substituição não autorizada de componentes do HSM.

O primeiro mecanismo de proteção ocorre a partir do processo de cifragem das chaves mantidas em memória persistente pela *Server Master Key* (SVMK) carregada a partir dos *smart cards* para a memória volátil e não paginável do HSM. Caso o HSM seja desligado ou detecte uma violação, a SVMK é destruída.

A fronteira criptográfica é feita de aço, material rígido, resistente e opaco à luz visível. Qualquer tentativa de acesso ao interior não autorizada, como perfuração ou corte do material resultará em danos que evidenciam a violação. Além disso, o gabinete conta com uma proteção interna para todos os pontos de possíveis acesso. Essas proteções são formadas por chapas de aço conformadas que se encaixam aos pontos de acesso garantindo uma segunda blindagem. As entradas e saídas de ar contam com defletores que permitem a livre passagem de ar e ao mesmo tempo impedem a sondagem e visualização dos componentes internos da fronteira criptográfica.

O HSM possui mecanismos que evidenciam a violação, como lacres metálicos aplicados para proteger os parafusos, assim qualquer tentativa de acessar o parafuso gera uma evidência. Também possui uma etiqueta adesiva de lacre do tipo casca de ovo aplicada a junção da tampa superior com a estrutura do gabinete, desse modo se houver a tentativa de retirada dessa etiqueta, a mesma se desmanchará, não podendo ser reconstruída.

O único meio de acesso físico autorizado ao interior do HSM é através da remoção da tampa de manutenção superior. Esta tampa é protegida por *micro switches* que detectam tentativas de remoção. A ativação destes sensores iniciam o processo de resposta pelo circuito supervisor de violação da fronteira criptográfica. Caso o HSM esteja ligado quando o sensor é ativado, o circuito supervisor emite imediatamente um sinal para o firmware do HSM e inicia a destruição da SVMK na memória volátil e o equipamento é desligado, se o MSC estiver desligado, o circuito supervisor registra a violação, pois dispõe de um bateria interna para manter a memória volátil que armazena parâmetros críticos de segurança. Esse dispositivo conta com um sub-circuito de gerenciamento de energia capaz de fazer com que a carga da bateria só seja utilizada quando o equipamento esteja desligado da fonte de energia AC e caso uma violação seja detectada a bateria é desconectada.

Capítulo 3

Materiais e Ferramentas

3.1 *Arduino*

O *Arduino* é uma placa de prototipagem de hardware livre e de placa única. A placa trabalha com um microcontrolador do tipo *Atmel AVR*. O hardware dá suporte a diversas formas de entrada e saída de dados de um modo simples. A linguagem utilizada é programada através de um ambiente de desenvolvimento integrado (IDE) próprio do Arduino baseado no *Wiring* que consiste em uma variação de C/C++. Além disso, conta com um *bootloader*, o qual permite a gravação do programa no chip de memória.

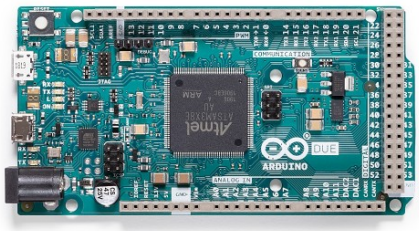


Figura 3.1: Arduino Due [8].

O Arduino Due apresentado na figura 3.1 é uma placa microcontroladora baseada no Atmel SAM3X8E ARM Cortex-M3 CPU. Ele contém toda a plataforma necessária para suportar o microcontrolador, para ligá-lo é preciso conectá-lo a um computador com um cabo USB ou a um adaptador de CA-CC ou a uma bateria. Esta placa foi utilizada para desenvolver o sistema de teste, sua escolha foi devido a disponibilidade e por possuir um barramento I2C capaz de transferir os dados da placa Arduino para os dispositivos de saída. Além disso, a seguir são explicitadas outras especificações que colaboraram para a escolha dessa placa.

- Processador: AT91SAM3X8E;
- Tensão de operação: 3.3 V;
- Tensão de entrada: 7-12 V;

- Corrente DC para o pino de 3.3 V e 5 V: 800 mA;
- Memória *flash*: 512 KB;
- Cristal: 84 MHz;
- Porta USB 2.0;

Na primeira etapa do projeto, o Arduino foi utilizado para testar o chip supervisor através de uma comunicação por I2C, bem como para entender o princípio de funcionamento do chip. Já na segunda etapa foi usado para realizar a comunicação serial com o HSM e armazenar o programa proprietário para o funcionamento do circuito supervisor.

3.2 Chip Supervisor - DS3645

A Maxim Integrated Products desenvolveu o DS3645, um chip supervisor de segurança com 4096 bytes de SRAM para aplicações que exigem o armazenamento seguro de dados confidenciais e as funcionalidades físicas de resposta à detecção de violações necessárias em processadores criptográficos e equipamentos de segurança de dados.

O DS3645 possui 32 bancos de 128 bytes e incorpora a funcionalidade de limpeza direta de alta velocidade. A memória de chave 4KB é constantemente complementada em segundo plano para evitar o ataque do tipo impressão de memória. No caso da detecção de uma violação, a memória da chave é rapidamente limpa e um viés negativo é aplicado para limpar a SRAM externa ao DS3645.

Além disso, possui um contador de segundos em tempo real, um temporizador de *watchdog*, supervisor de CPU, controlador de SRAM não volátil e sensor de temperatura embutido no chip. No caso de uma falha de energia primária, uma fonte de bateria externa é automaticamente ligada para manter ativa a memória volátil, o contador de segundos e o circuito de detecção de violação.

O DS3645 fornece entradas de detecção de violação para sensores externos e para manta *mesh*. Caso o nível da bateria ou de temperatura caia abaixo de um limite especificado ou se a frequência do oscilador do cristal estiver fora da faixa determinada ou se a taxa de variação de temperatura exceder os limites programados um evento de violação é acionado.

O acesso ao contador de segundos, à memória e o monitoramento de violação e configuração do dispositivo é realizado por meio de uma interface compatível com I2C. O DS3645 utiliza o método de encapsulamento CSBGA, que aprimora a segurança, pois os terminais não são expostos às bordas externas do pacote [12]. Este dispositivo suporta os níveis de segurança 3 e 4 da norma FIPS 140-2. As principais características que atendem ao escopo do projeto são especificadas a seguir:

- Memória de chave de 4096 bytes com uma limpeza de alta velocidade sem impressão;
- RAM de 64 bytes;

- Contador de segundos de 32 bits;
- *Watchdog timer*;
- Supervisor da CPU;
- Quatro comparadores de detecção de violação;
- Quatro comparadores de janela com tensão de referência;
- Duas entradas digitais para detecção de violação;
- Detecção de temperatura e suas variações;
- Registro e travamento dos eventos de violação;
- Monitoramento da oscilação do cristal;
- Baixo consumo de energia;
- Ampla faixa de temperatura: -55 °C a +95 °C;
- Encapsulamento CSBGA;
- Interface compatível com I2C.

O DS3645 foi escolhido por ser o chip supervisor mais completo do mercado e que apresenta suporte para o desenvolvimento das funcionalidades necessárias do circuito supervisor de violação de fronteira criptográfica. A tabela 3.1 apresenta o estudo comparativo entre variados chips supervisores. Além disso, a partir da procura por esses componentes verificou-se que o mercado oferece uma gama limitada de soluções.

Tabela 3.1: Comparação entre os chips supervisores.

Componente	DS3645	STM1404CSNIQ6F	M41ST87W	MSFIPS
Empresa	Maxim	STMicroelectronics	STMicroelectronics	MSI
Circuito de gerenciamento de energia	Sim	Sim	Sim	Sim
Watchdog timer	Sim	Não	Sim	Não
RTC ou contador de segundos	Sim	Sim	Sim	Não
Entradas para detectar violação	Sim	Sim	Sim	Sim
Sensor de temperatura	Sim	Sim	Não	Sim
Baixo consumo de energia	Sim	Sim	Sim	Sim
Limpeza rápida da memória volátil	Sim	Sim	Sim	Sim
Detecção do nível de bateria	Sim	Sim	Não	Não
Memória interna segura	Sim	Não	Sim	Não
Compatibilidade para SRAM externa	Sim	Sim	Sim	Sim
Obsoleto	Não	Sim	Não	Sim

3.2.1 *Breakout Board*

A *breakout board* é uma placa que converte componentes SMD (*Surface Mounting Device*) e BGA em *through hole* e permitem que eles sejam soldados em placas padrão ou encaixados em *protoboards*. Desse modo, é possível inserir o componente SMD ou BGA em um projeto sem a necessidade de fabricar uma placa de circuito impresso ou soldar os fios diretamente nos terminais dos componentes.

O chip DS3645 possui o encapsulamento CSBGA, isso significa que é um tipo de BGA termicamente aprimorado com cavidade reduzida, fatores de forma e meio de interconexões menores. O BGA é um dos métodos de encapsulamento utilizados para a conexão de componentes em circuitos integrados. Com uma grande quantidade de pinos, o BGA realiza a conexão por meio da soldagem de pequenas esferas com a fixação direta entre componente e placa. É um dos encapsulamentos mais críticos do processo e exige um cuidado extra durante a montagem, pois sem auxílio de equipamento radiológico, é possível inspecionar apenas os *balls* externos do componente.

A soldagem é realizada por uma estação de solda BGA que realiza o aquecimento do componente, desse modo ao atingir a temperatura de fusão, uma leve movimentação realiza a junção das esferas dos terminais de contato do componente à placa. Utiliza-se um estêncil para facilitar o processo de soldagem. Nesse projeto para a fase de testes, o chip DS3645 foi soldado à *breakout board* especificada na figura 3.2 com o auxílio do estêncil apresentado nessa mesma imagem.

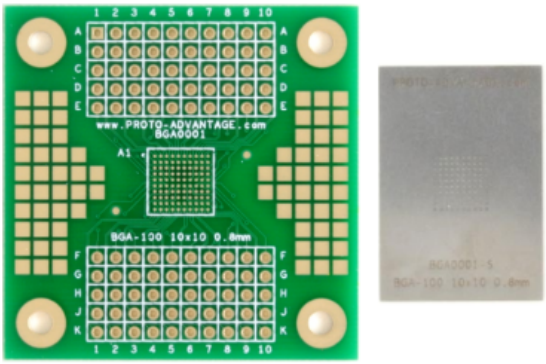


Figura 3.2: *Breakout board* e estêncil [32].

Esta placa possui 100 pinos com um espaçamento de 0,1 inch, um *pitch* de 0,8 mm e diâmetro dos *balls* de 0,45, desse modo apesar do diãmetro dos *balls* do DS3645 ser de 0,36 mm, as duas placas possuem compatibilidade. Suas dimensões são 48,26 mm x 48,26 mm x 1,6 mm, a faixa de temperatura de operação é de -40°C a +130°C e a máxima temperatura para o processo de soldagem é 260°C. Já o estêncil compatível com essa placa é feito de aço inoxidável. Além do uso dessa *breakout board* também foi desenvolvida uma placa no software *EAGLE* para a fase de testes do DS3645.

3.3 Relé

O relé é uma chave eletromecânica composto por uma bobina e contatos elétricos que podem estar normalmente abertos (NA) ou normalmente fechados (NF). Quando uma corrente elétrica circula na bobina ocorre a produção de um campo magnético que gera uma força eletromagnética capaz de atrair os contatos elétricos de modo a abrir ou fechar os circuitos conectados a ele. Assim que a corrente na bobina é interrompida o contato elétrico retorna ao seu estado normal com o auxílio de uma mola interna [16]. No projeto foi usado um relé para controlar e interceptar o sinal de teclado na porta USB durante o *start-up* até que o firmware entre em execução. Como apenas o sinal de teclado é controlado foi escolhido um relé de 5 V, 2 A e tensão de operação menor ou igual a 3,75 V mostrado na figura 3.3.

Figura 3.3: Relé METALTEX ML2RC-5V [15].

3.4 Chave *Micro switch*

A chave *micro switch* é um tipo de interruptor usado para acionar ou interromper uma saída, possui três terminais, sendo um normalmente aberto, um normalmente fechado e um comum. Nesse projeto esta chave é utilizada como sensor de violação, posicionada em pontos estratégicos do HSM de modo a detectar a tentativa de remoção da tampa superior, a partir da retirada dos parafusos que a fixam a tampa ao gabinete e pressionam os sensores.

Figura 3.4: Detector de pressão [11].

Os sensores são do modelo MS0850503F055P1A apresentado na figura 3.4 e com as seguintes características técnicas:

- Contatos: 5 A 125/250 V AC;
- Expectativa de vida: 50.000 ciclo no mínimo;
- Vida mecânica: 1.000.000 ciclos;
- Resistência mínima de isolamento: 100 MΩ;
- Temperatura de operação: $-25\,^{\circ}\text{C}$ a $85\,^{\circ}\text{C}$;

3.5 Bateria

Com o intuito de sustentar a alimentação da memória volátil do DS3645 quando o HSM está desligado foi escolhida uma bateria de lítio do modelo CR2477 de 3 V e 1000 mAh de carga nominal, sua tensão final é de cerca de 2,5 V e o seu peso padrão de 10,5 g. A figura 3.5 apresenta a bateria escolhida.

Figura 3.5: Bateria de 3V [29].

3.6 EAGLE

O Autodesk EAGLE é um software de automação de projetos eletrônicos (EDA - *Eletronic Design Automation*) utilizado pelos projetistas de placas de circuito impresso, com ele é possível conectar com perfeição diagramas esquemáticos, posicionamento de componentes, roteamento de PCBs e um amplo conteúdo de bibliotecas [14]. Esta ferramenta foi utilizada para projetar a *breakout board* do chip supervisor de segurança e a *shield* juntamente com os componentes complementares, como resistores, cristal, bateria e relé para ser utilizada com o Arduino Due.

Primeiramente desenvolveu-se os esquemáticos do chip DS3645 para incorporar o componente a biblioteca do software, no formato *Ball Grid Array* (BGA) com 49 *balls*, tendo como base as dimensões e especificações dos documentos apresentados no anexo desse trabalho, em seguida a *breakout board* foi projetada. Devido a dificuldade para realizar o roteamento dos 49 *balls* do chip com a barra de pinos foi desenvolvida uma placa com duas camadas. Durante o projeto foi tomado o cuidado para que a disposição de cada pino da barra estivesse em conformidade com os espaçamentos dos furos da *protoboard*, uma vez que inicialmente todos os testes foram realizados com o auxílio dessa matriz de contato.

3.7 Protocolo I2C

O protocolo I2C, ou *Inter-Integrated Circuit* é um protocolo de comunicação que funciona por meio de um barramento onde apenas um dispositivo, denominado mestre, é responsável por requisitar informações dos dispositivos conectados. A conexão I2C é feita através de dois fios, o SDA responsável por transmitir dados entre receptor e transmissor via barramento e o SCL que tem como objetivo temporizar e sincronizar unidades conectadas ao sistema.

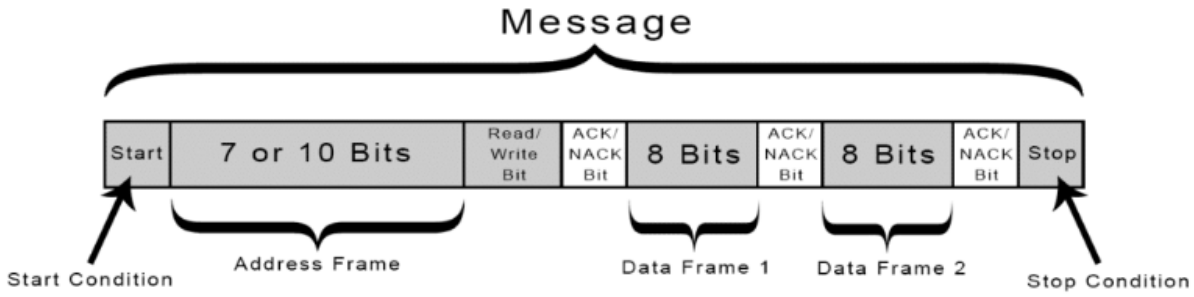


Figura 3.6: Formato da mensagem enviada na comunicação I2C [10].

Cada dispositivo conectado ao sistema possui um endereço, tendo em vista que a comunicação I2C é feita via um barramento. Este código é composto por um valor de 7 bits, disponibilizando um total de 127 endereços onde apenas o intervalo 0x8 até o intervalo 0x77 está disponível para utilização. O sinal de dados é transferido em sequências de 8 bits. Assim que uma condição de partida ocorre, a primeira sequência de 8 bits que indica o endereço do escravo para o qual os dados estão sendo enviados é recebida. Após cada sequência de 8 bits um ACK é recebido. Após esse primeiro ACK, são recebidos os registros internos do dispositivo escravo. Após receber as sequências de endereçamento, são enviadas as sequências de dados, assim que são completamente enviados uma condição de parada define o término do processo.

3.8 Máquina Virtual

Uma máquina virtual (*Virtual Machine* - VM) é definida como uma duplicata eficiente e isolada de uma máquina real. Em uma máquina real, uma camada de software de baixo nível fornece acesso aos vários recursos do hardware para o sistema operacional, que os disponibiliza de forma abstrata às aplicações [26].

Nesse projeto foi utilizado o Oracle VM VirtualBox para executar o sistema operacional CentOs 7, o qual é uma distribuição Linux. Nesse ambiente foi adicionado um programa de teste com a finalidade de simular o comportamento do HSM de modo que seja possível realizar as trocas de mensagens via comunicação serial entre o HSM e o sistema supervisor de segurança.

3.9 Visão Geral

A figura 3.7 mostra uma visão geral do circuito supervisor de violação de fronteira com os seus componentes após a implementação de todas as suas funcionalidades e a relação entre esse circuito e o HSM.

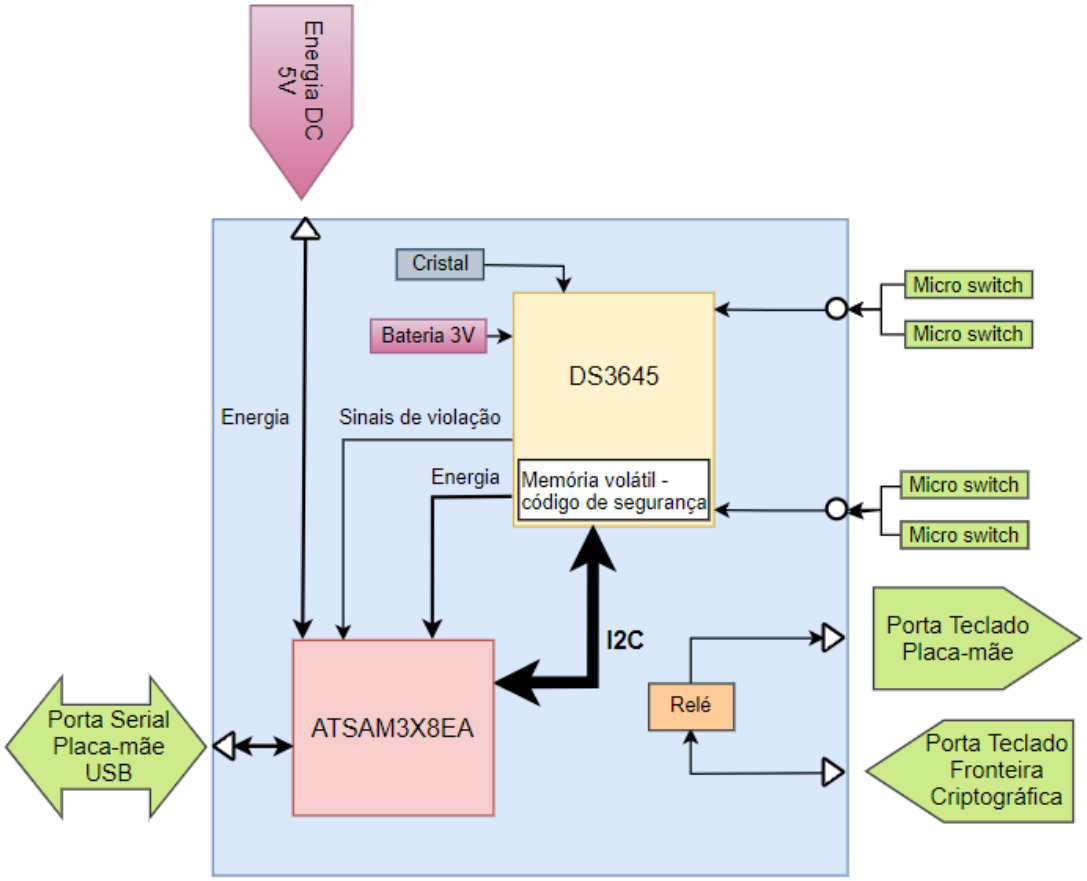


Figura 3.7: Diagrama de blocos do circuito supervisor de violação.

Em resumo, o sistema contará com um chip supervisor conectado diretamente aos *micro switches* para detectar a violação, desse modo quando algum dos sensores é ativado o circuito emite um sinal via porta serial para o firmware do HSM que ao tratar esse sinal tomará uma série de medidas, entre elas a de desligar o equipamento. O Arduino Due é utilizado para armazenar e executar o programa desenvolvido que realiza a comunicação serial com a placa mãe. Na primeira etapa foi utilizado um programa de teste na máquina virtual para simular o comportamento do HSM durante a troca de mensagens.

Do ponto de vista de desenvolvimento de firmware desse circuito será necessário implementar o protocolo de comunicação com o HSM. Além disso, no *setup* será realizada a configuração inicial responsável por setar todos os parâmetros para fazer a retirada do circuito do modo de fábrica e deixá-lo funcional. Já na parte principal do programa estará a lógica da máquina de estados implementada a partir de funções como detecta_violação, não_detecta_violação, grava_código_segurança, monitora, entre outros.

Além disso, serão implementadas rotinas de leitura de tensão e de temperatura. Um ponto importante que deve ser considerado durante o desenvolvimento é que há dois modos de funcionamento, quando está conectado a fonte e quando está conectado a bateria. No segundo caso não há comunicação I2C, nesse caso sempre que for verificado que o circuito retornou para o modo de funcionamento alimentado pela fonte deverá haver uma verificação se o código de segurança está na memória RAM do circuito, se estiver não houve violação, caso contrário houve invasão.

Capítulo 4

Métodos

4.1 Sistema de Teste

Primeiramente realizou-se todas as conexões entre o arduino e a *breakout board* com o chip DS3645. A partir da leitura do *datasheet* do chip verificou-se a necessidade de conectar ao circuito uma bateria de 3V e um cristal de 32.768kHz e 6pF. O diagrama de conexão desses materiais é apresentado na figura 4.1.

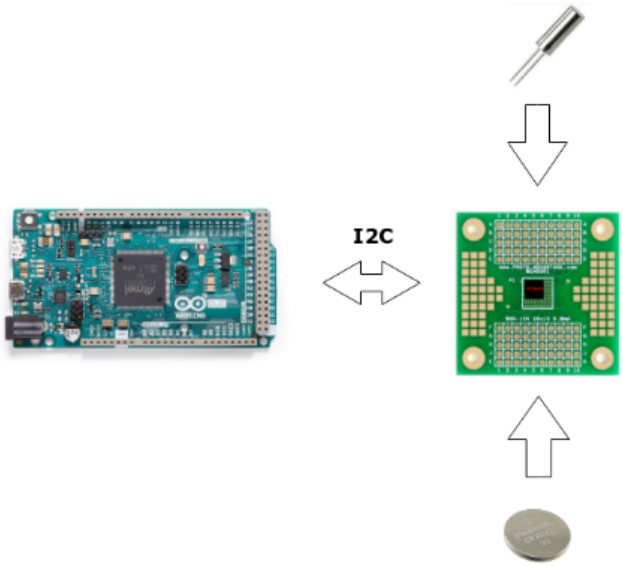


Figura 4.1: Diagrama de conexão dos materiais para a etapa de teste do chip.

A segunda etapa consistiu em verificar se o chip foi soldado de maneira correta na placa e se estava funcionando, desse modo com o auxílio da comunicação I2C foi desenvolvido um programa que realiza o escaneamento de todos os dispositivos I2C no ambiente e retorna o endereço, o qual é único para cada dispositivo. O *datasheet* do chip informa qual é o endereço do escravo. Assim, ao executar o programa e como há apenas um dispositivo I2C conectado ao sistema houve o retorno do endereço esperado de acordo com o *datasheet*. A partir disso, pode-se concluir que o processo de soldagem foi efetivo e o chip estava funcionando.

Em seguida, foi desenvolvido um programa com o intuito de ler a memória para obter os sinais monitorados pelo chip. Desse modo, o primeiro endereço de memória lido foi o do contador de segundos, a partir do monitoramento desse endereço verificou-se que o valor incrementava de tempos em tempos como esperado. Posteriormente, todos os sinais que indicam violações, medidas de temperatura e tensão, número de série e o gerador de número aleatório foram lidos.

A partir da análise dos resultados obtidos da leitura, concluiu-se que o circuito estava em um estado de violação e nessa condição não é possível ler os valores de temperatura, tensão e entradas analógicas, uma vez que em um estado de violação as atualizações do conversor analógico-digital são suspendidas. Portanto, foram desabilitadas todas as entradas do chip e para modificar os sinais foi desenvolvido um programa para escrever bits específicos em determinadas posições de memória no chip. Além disso, todas as entradas e comparadores foram aterrados para não haver flutuações das mesmas. Desse modo, o sinal de violação foi retirado, as medidas foram atualizadas e puderam ser monitoradas.

A próxima etapa do projeto consistiu em colocar os *micro switch*. A figura 4.2 apresenta o circuito desenvolvido. O pino Vcco do chip é a tensão de alimentação de saída caracterizada pelo chaveamento interno entre o pino de tensão de alimentação (Vcci=3,3 V) e o pino de tensão da bateria (Vbat). Quando a tensão é fornecida pela fonte primária Vcco=Vcci, já quando o equipamento é desligado a energia é fornecida pela bateria e Vcco=Vbat. O pino de entrada digital escolhido foi o de lógica inversa, de tal modo que se IN=LOW detecta a violação, caso contrário para IN=HIGH a violação não é detectada. Além disso, no *micro switch* foram utilizados os terminais comum e normalmente aberto.

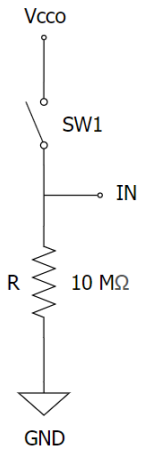


Figura 4.2: Esquemático de conexão dos sensores.

Como mostrado na figura 4.2, a configuração de ligação escolhida foi de *Pull Down*. Assim, quando o equipamento está fechado com a tampa e pressionado pelo parafuso, a chave está fechada e IN=HIGH, logo não há violação, entretanto quando o parafuso é retirado a chave é aberta, IN=LOW e ocorre a detecção da violação. Desse modo, para verificar se o chip detectou o comportamento descrito foram lidos alguns bits específicos da memória, a partir disso ao abrir a chave, os registradores indicavam que havia ocorrido uma violação e ela tinha sido causada pela

entrada digital escolhida. O teste descrito foi realizado com o circuito ligado e desligado. Além disso, foi escolhido um valor de resistor grande para que a corrente seja a menor possível de modo que não ocorra fuga de corrente pela bateria quando o equipamento estiver desligado.

Para concluir a etapa de teste do chip foi desenvolvido um código de escrita e leitura na memória. Essa etapa foi fundamental, pois durante a fabricação do HSM é gravado um código de segurança de 17 bytes na memória protegida do chip usado para verificação de integridade do equipamento, uma vez que quando uma violação é detectada o código de segurança é destruído imediatamente da memória, assim só é possível recuperar esse valor no laboratório do fabricante. Ao executar o código de escrita e leitura na memória verificou-se que os registradores específicos da memória de chave do DS3645 armazenavam o código e ao gerar uma violação esse valor era apagado.

Por fim, foi acrescentado o circuito do relé apresentado na figura 4.3 com o intuito de interceptar e controlar os sinais de teclado que entram e saem da fronteira criptográfica. A partir dos valores de tensão nominal e consumo nominal apresentados no anexo obteve-se a corrente Ic=40 mA necessária para o relé fechar os seus contatos, assim optou-se pelo uso do transistor BC547A que atende a especificação de corrente de condução necessária para o acionamento. A tabela 4.1 mostra os valores utilizados para calcular o valor do resistor que deve ser colocado no terminal de base do transistor a partir da lei de *Kirchhoff*. A tensão de 3,3 V corresponde ao estado lógico 1 fornecido pelo Arduino Due.

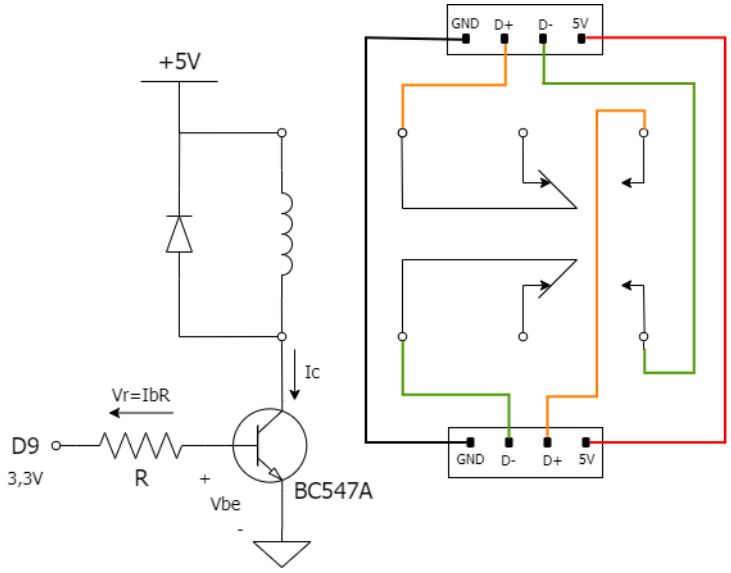


Figura 4.3: Esquemático de conexão do relé.

Tabela 4.1: Característica do transistor BC547A.

Ganho beta mínimo	110
Tensão coletor-emissor	45 V
Tensão base-emissor	0,7 V
Corrente de coletor máxima	100 mA

$$3,3 - I_b R - V_{be} = 0 \tag{4.1}$$

$$R = \frac{(3,3 - V_{be})\beta_{min}}{Ic} \therefore R < 7,15k\Omega \tag{4.2}$$

Como beta varia em função da tensão coletor-emissor e da temperatura é conveniente escolher um valor de resistência menor que o calculado, assim foi feita uma combinação de resistores em série para resultar em 7 kΩ. O diodo foi utilizado nesse circuito por possuir uma ação rápida e suportar picos instantâneos de corrente, além disso ao ser usado em paralelo com a bobina do relé reduz a corrente induzida quando o relé é desligado, desse modo com o caminho fornecido para essa corrente é evitado queima de componentes e interferência no circuito de controle. A figura 4.4 mostra o protótipo em *protoboard* do sistema de teste descrito.

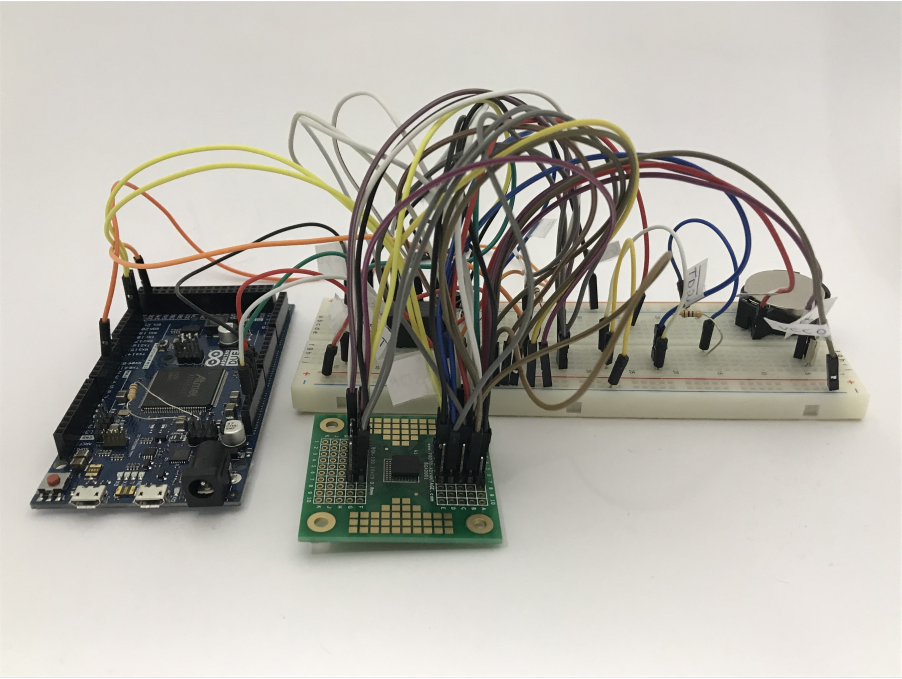


Figura 4.4: Sistema de teste do supervisor na *protoboard*.

4.2 Máquina de Estados

A construção da máquina de estados foi baseada na versão anterior desse produto, como os componentes do circuito supervisor estão obsoletos foi necessário projetar um novo com o auxílio de materiais que ainda não possuem *end of life*. Desse modo, para garantir a compatibilidade com o *firmware* foi utilizada a mesma máquina de estados. Assim, para identificar as transições e os estados foi usado o programa de teste que simula o comportamento do HSM na máquina virtual juntamente com o circuito supervisor desenvolvido anteriormente e obteve-se como resultado a figura 4.5.

Os estados são descritos por:

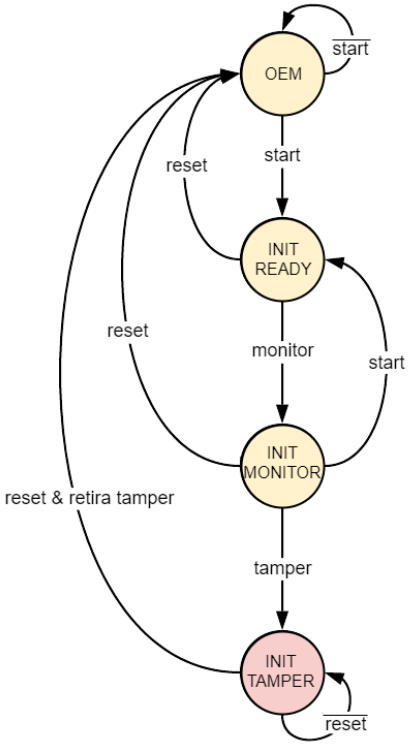


Figura 4.5: Máquina de estados do funcionamento do sistema.

- OEM: estado inicial e de retorno após o comando de reset;
- INIT READY: estado de parada que não detecta violação;
- INIT MONITOR: estado que monitora e é capaz de detectar a violação;
- INIT TAMPERED: estado após ocorrer um *tamper* (violação);

4.3 Protocolo de Comunicação

Com base na premissa de manter a compatibilidade com o firmware o protocolo de comunicação utilizado foi baseado no desenvolvido para a primeira versão do circuito supervisor que foi implementado para um microcontrolador intel 8051 de 8 bits.

As próximas seções serão destinadas para a especificação do protocolo de baixo nível implementado pelo controlador e pelo cliente de seus serviços, ou seja, o hospedeiro. A comunicação entre controlador e hospedeiro é feita pela porta serial com uma troca de mensagens bem definidas, na qual o hospedeiro inicia o diálogo e gera uma saída, assim o controlador espera essas mensagens, responde e gera uma entrada que será esperada pelo hospedeiro. No sistema o hospedeiro é o HSM e o controlador é o circuito supervisor.

4.3.1 Formato da Mensagem

Em relação as mensagens trocadas entre hospedeiro e controlador, a taxa de transmissão utilizada foi de 115200 e o formato usado é mostrado na figura 4.6.

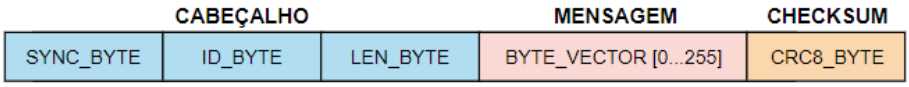


Figura 4.6: Formato da mensagem.

Toda mensagem utiliza o formato simples apresentado, cada campo possui 1 byte e tem como significado:

- SYNC_BYTE: é definido como 0x07 e indica o início de uma mensagem;
- ID_BYTE: identifica qual é a mensagem;
- LEN_BYTE: informa qual é tamanho do conteúdo da mensagem;
- BYTE_VECTOR [0...255]: informa o conteúdo da mensagem, o seu tamanho pode ser de até 32 bytes;
- CRC8_BYTE: esse byte é utilizado para conferir a integridade da mensagem, é utilizado o método pseudo-CRC sobre um campo finito de 9 bits para a detecção de erros;

A tabela da figura 4.7 apresenta todas as mensagens do protocolo juntamente com o seu formato.

MENSAGEM	SYNC_BYTE	ID_BYTE	LEN_BYTE	BYTE_VECTOR [0...255]			CRC8_BYTE
start	0x07	0x01	0x00	-	-	-	0xXX
reset	0x07	0x02	0x00	-	-	-	0xXX
monitor	0x07	0x03	0x00	-	-	-	0xXX
tampering_detected	0x07	0x04	0x00	-	-	-	0xXX
ack_ok	0x07	0x05	0x00	-	-	-	0xXX
ack_unknown	0x07	0x06	0x00	-	-	-	0xXX
ack_invalid	0x07	0x07	0x00	-	-	-	0xXX
ack_need_start	0x07	0x08	0x00	-	-	-	0xXX
read_mem	0x07	0x09	0x02	endereço1	endereço2	-	0xXX
write_mem	0x07	0x0A	0x03	endereço1	endereço2	valor	0xXX
ack_info	0x07	0x0B	0x01	estado	-	-	0xXX
get_state	0x07	0x0C	0x00	-	-	-	0xXX
turn_relay	0x07	0x0D	0x02	numero	status	-	0xXX
get_battery_status	0x07	0x0E	0x00	-	-	-	0xXX

Figura 4.7: Definição do formato das mensagens do protocolo.

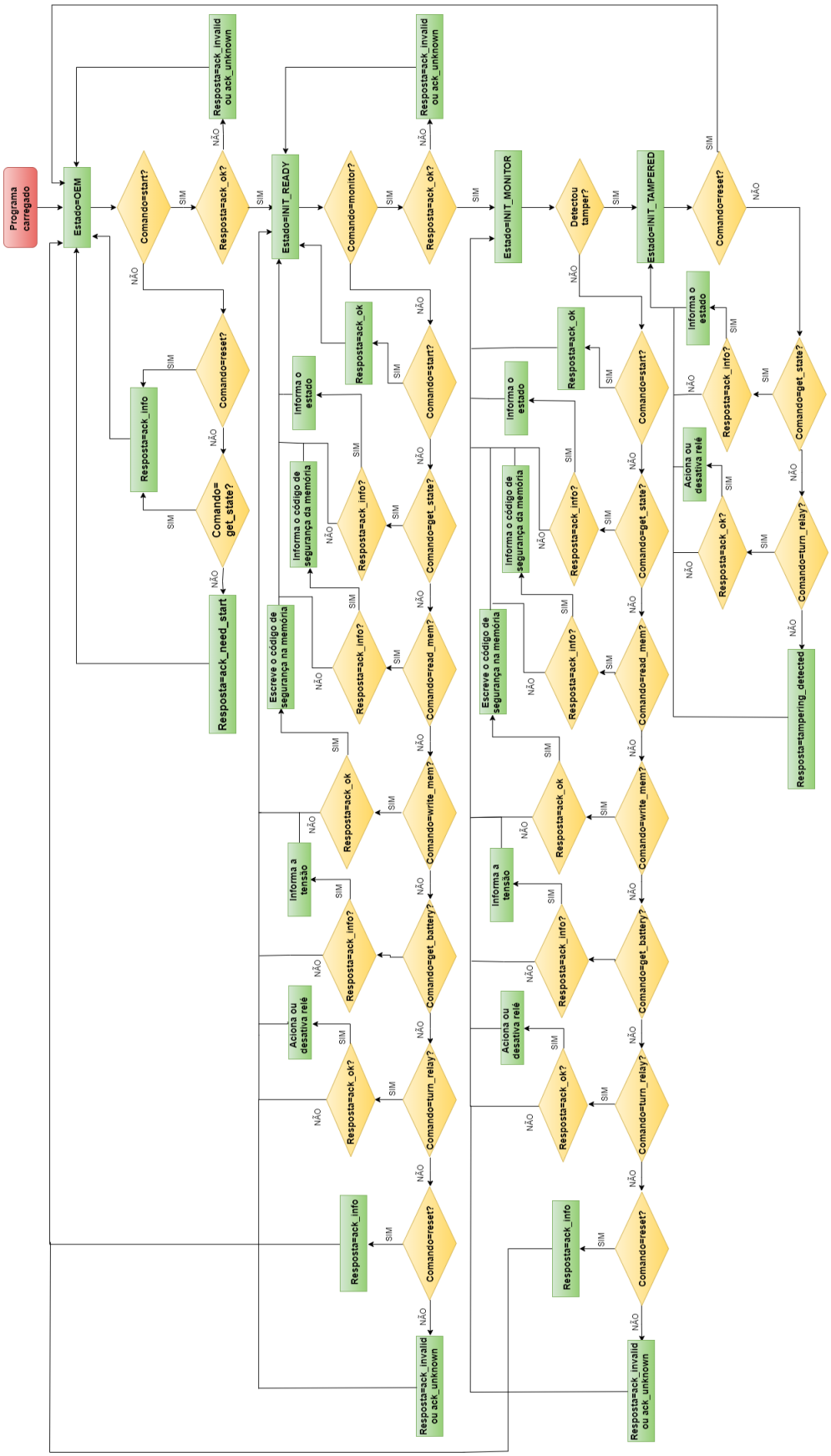


Figura 4.8: Fluxograma do funcionamento geral do sistema.

4.3.2 Modelo Operacional

O funcionamento do controlador é baseado na máquina de estados apresentada na figura 4.5, cada estado é definido por um valor conforme indicado na tabela 4.2. Ademais, a cada mensagem enviada do hospedeiro para o controlador, uma mensagem ACK será entregue como resposta, com o status da operação e dados, quando for aplicável.

Tabela 4.2: Definição do formato de cada mensagem.

Estado	Valor
OEM	0x01
INIT TAMPERED	0x02
INIT READY	0x03
INIT MONITOR	0x04

Além das mensagens utilizadas para as transições, também há uma troca de mensagens específicas entre hospedeiro e controlador em cada estado de tal modo que não ocorra transições, apenas a entrega e o recebimento de informações. A figura 4.8 apresenta o funcionamento geral do sistema caracterizando as trocas de mensagens. Os comandos são as mensagens enviadas pelo Módulo de Segurança Criptográfico e as respostas são enviadas pelo circuito supervisor. Cada mensagem do protocolo será descrita a seguir:

1. start: inicialmente o hospedeiro deve enviar uma mensagem start, após isso o controlador sai do estado OEM para o INIT READY e responde com um ack_ok. Além disso, caso o controlador esteja no estado INIT MONITOR e receber um start do hospedeiro, há um retorno para o estado INIT READY.

2. reset: esse comando faz com que o controlador retorne para OEM, independentemente de qual estado ele esteja. O controlador responde com um ack_info e o novo estado é passado como informação no campo BYTE_VECTOR [0...255], caso ocorra um erro um ack_invalid é enviado de volta ao hospedeiro.

3. monitor: essa mensagem faz com que o controlador mude do estado INIT READY para o INIT MONITOR, assim o controlador entra no modo de detecção de *tamper*.

4. tampering_detected: é a mensagem de resposta do controlador para o hospedeiro caso uma violação seja detectada, além disso ocorre a transição para o estado INIT TAMPERED.

5. ack_ok: é a mensagem de resposta do controlador para informar ao hospedeiro que a mensagem foi recebida com sucesso.

6. ack_unknown: é a resposta do controlador para mensagens desconhecidas.

7. ack_need_start: é a resposta do controlador no estado OEM para as mensagens enviadas pelo hospedeiro que sejam diferentes de get_state ou start.

8. read_mem: é usado para recuperação de dados na memória e não causa mudança de estado. O modo de endereçamento em duas partes representa uma posição de memória linear dentro do espaço de endereço de 16 bits. Essa memória de 4K é tratada linearmente no protocolo, sendo [0x00 00] o endereço do primeiro byte, [0x01 00] do segundo, e assim por diante, em *little endian*. O controlador deve responder a esta solicitação com uma mensagem ack_info informando qual é o valor, em byte, da posição de memória requerido, em caso de erro um ack_invalid é enviado de volta ao hospedeiro.

9. write_mem: não causa mudança de estado e é usado para escrita de dados na memória. Segue o mesmo modelo de endereçamento do read_mem, o terceiro campo do byte_vector informa qual dado deve ser escrito na posição de memória especificado. Em caso de êxito o controlador deve responder a essa requisição com um ack_ok, já em caso de erro com um ack_invalid.

10. ack_info: é a mensagem utilizada pelo controlador para enviar alguma informação requisitada pelo hospedeiro.

11. get_state: o controlador deve informar seu estado com o ack_info após o hospedeiro solicitar.

12. turn_relay: o controlador deve ativar ou desativar o relé do teclado com base na mensagem do hospedeiro. O primeiro campo do byte_vector informa qual o relé deve ser ligado ou desligado, como na primeira versão havia um para o sinal do monitor e outro para o do teclado esse campo podia ser 0x00 ou 0x01, na atualidade como apenas o relé do teclado é implementado esse valor só pode ser 0x01. Já o campo de status indica se será ativado ou desativado. Assim, o controlador deve responder ao hospedeiro com um ack_ok em caso de sucesso e com um ack_invalid em caso de erro.

13. get_battery_status: o controlador informa a tensão de sua bateria baseado em porcentagem, ou seja 0x00 == 0%, 0x10 == 16% e assim por diante com uma mensagem ack_info, em caso de erro um ack_invalid é enviado de volta para o hospedeiro.

Além disso, para essa primeira versão com o DS3645 foi decidido que não haveria mudanças no protocolo e que inicialmente deveria funcionar igual a versão com componentes obsoletos com as mesmas trocas de mensagens, entretanto para uma segunda versão novas mensagens serão acrescentadas, como o hospedeiro poder enviar para o controlador o horário no qual ocorreu o *tamper*. Outra decisão de projeto está relacionada ao programador do firmware não desenvolver o programa do circuito supervisor, isso ocorre em decorrência da necessidade de manter duas equipes trabalhando independentemente em segmentos diferentes.

4.4 Descrição do Software

O código foi desenvolvido na IDE do Arduino, foram utilizadas as bibliotecas Wire.h e DueFlashStorage.h. A primeira permite que os dispositivos se comuniquem pelo protocolo I2C, já

a segunda é utilizada para salvar dados na memória flash do Arduino Due. Essa biblioteca é semelhante a EEPROM.

Na função *setup* as variáveis e as bibliotecas são inicializadas, o modo dos pinos são configurados e é verificado o valor da memória flash caso esse valor seja o padrão, o modo de fábrica do DS3645 é desativado e o valor do estado OEM é escrito na memória flash, caso contrário o valor do estado da memória é lido e armazenado. Além disso, nessa função é verificado se ocorreu *tamper* quando o equipamento estava desligado para isso confere-se o bit que indica violação no DS3645. Por fim, uma interrupção é definida caso seja detectada uma invasão pelo microswitch quando o HSM está ligado.

Na função *loop* são determinadas as funções que definem o funcionamento do protocolo e a máquina de estados. Assim que a primeira mensagem é identificada o estado é conferido, para cada há um tratamento específico em relação às mensagens. O estado OEM aceita os comandos get_state, reset e start, o INIT READY e INIT MONITOR permitem get_state, start, reset, monitor, read_mem, write_mem, get_battery_status, turn_relay, a diferença entre eles está relacionada ao INIT READY não detectar a violação e o INIT MONITOR sim, já o estado INIT TAMPERED aceita reset, turn_relay e get_state. Além disso, para cada comando enviado pelo hospedeiro o controlador responde com um dos ack explicitados.

Quando uma violação ocorre no estado INIT MONITOR o chip supervisor apaga a sua memória de chave, congela os seus registradores e o controlador muda para o estado INIT TAMPERED, caso isso ocorra no estado INIT READY o chip apresesenta o mesmo comportamento, porém o controlador permanece no estado INIT READY, assim que ele receber um comando monitor, a violação é detectada, pois o bit de *tamper* estará congelado no valor que indica a invasão.

No estado INIT TAMPERED, quando o comando reset é usado, além do retorno para OEM ocorre a retirada do *tamper*, ou seja, o chip supervisor é resetado, sai do seu estado de congelamento e volta a monitorar os sinais. A função que faz o tratamento da mensagem write_mem é utilizada para escrever o código de segurança de 17 bytes na memória de chave protegida do chip supervisor. O usuário consegue verificar se a máquina foi violada quando consulta esse código e identifica que ele foi apagado da memória, essa informação é fornecida pela mensagem read_mem. Esse código só pode ser escrito novamente na fábrica.

4.5 Desenvolvimento da *Shield* Cryptuino

A *shield* Cryptuino foi desenvolvida para ser plugada no Arduino Due e desempenhar todas as funções especificadas, ela foi baseada no sistema de teste da figura 4.4. Além disso, foi projetada para reduzir todas as interferências e os mal contatos causados pela quantidade de fios e pela *protoboard*. A figura 4.9 mostra o projeto da PCB desenvolvida no software *Eagles*. Essa placa foi desenvolvida com duas camadas, pois foi necessário realizar o roteamento dos 49 pinos do DS3645, além disso a *shield* foi projetada para realizar acréscimos de novas tecnologias, como a manta *mesh*, desse modo vários pinos que apesar de ainda não serem utilizados foram deixados disponíveis para serem usados no futuro.

Um dos problemas críticos foi em relação a fuga de corrente que estava fazendo a bateria usada no sistema da *protoboard* e ser descarregada em poucos meses, sendo que deveria durar cerca de 3 anos. A criticidade dessa situação está relacionada com a detecção da violação pelo chip supervisor caso a bateria esteja descarregada, assim um *tamper* é identificado mesmo que não tenha ocorrido a tentativa de invasão. Foram realizadas uma série de medidas em diferentes cenários para verificar quanto de corrente estava vazando pela bateria, todos os valores e conclusões serão descritas no capítulo de resultados.

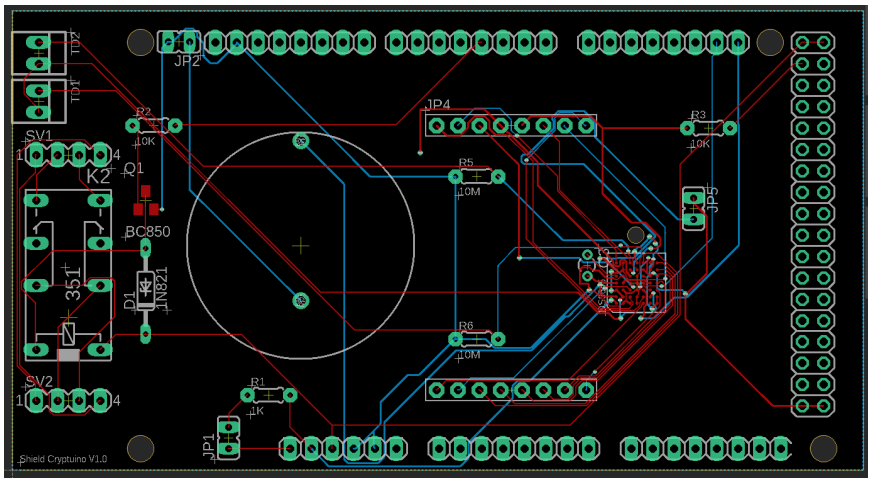


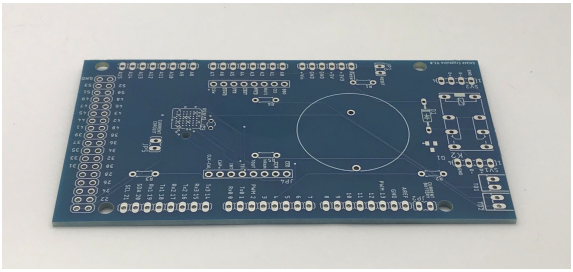
Figura 4.9: Projeto da *shield*.

Capítulo 5

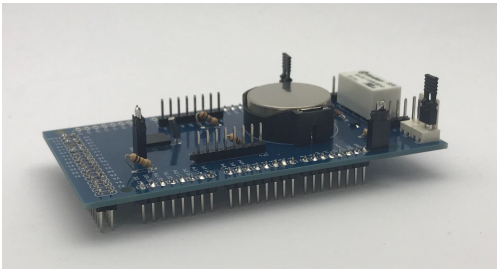
Resultados

5.1 *Shield* Cryptuino

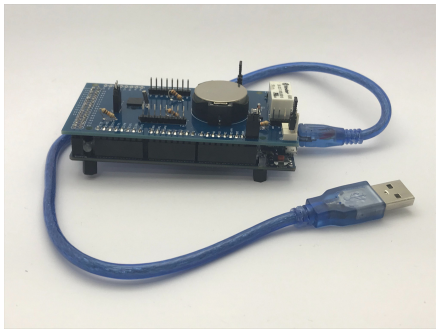
O projeto apresentado na seção 4.5 foi enviado para produção e resultou na figura 5.1a, em seguida, os componentes, como o chip supevisor DS3645, resistores, barras de pino macho e fêmea, cristal, suporte para bateria, diodo, transistor e relé foram soldados na placa, esse sistema é mostrado na figura 5.1b. Já a figura 5.1c apresenta o circuito supervisor acoplado ao Arduino.



(a) Placa de circuito impresso.



(b) *Shield* com os componentes.



(c) *Shield* acoplada ao arduino.

Figura 5.1: Circuito supervisor de violação de fronteira.

5.1.1 Instalação no HSM

Com o sistema plenamente funcional, o passo seguinte consistiu na instalação da *shield* no HSM que é um equipamento de 1U e com dimensão 420 x 44 x 370 mm. Foram instalados os 4 sensores em uma configuração de pares em paralelo em dois pontos estratégicos do HSM de modo que a retirada do parafuso da tampa gere uma violação, em seguida a placa acoplada ao Arduino Due foi montada no interior do gabinete.

Desse modo, com o auxílio de suportes de nylon parafusados ao gabinete, a placa foi instalada no Módulo de Segurança Criptográfico . A figura 5.2 mostra o sistema *shield*-Arduino devidamente instalado no equipamento, juntamente com o posicionamento de um par de sensores.

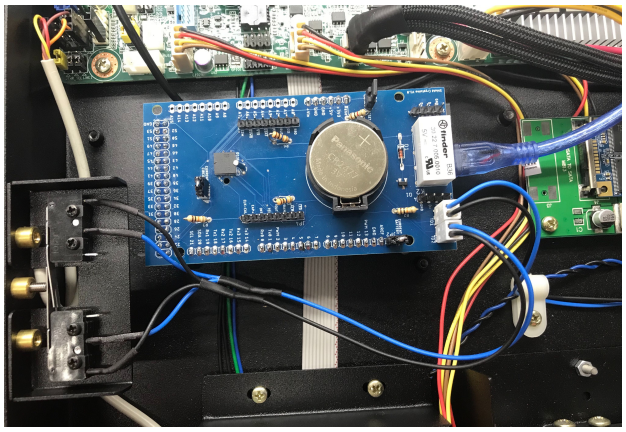


Figura 5.2: Instalação do sistema supervisor no HSM.

5.2 Validação do Funcionamento

Foram realizados dois testes para validar o funcionamento. O primeiro foi feito com o circuito desenvolvido na *protoboard* e o programa que simula o comportamento do HSM, já o segundo foi executado na *shield* integrada ao firmware do equipamento.

5.2.1 Programa de Teste

O primeiro teste realizado com o circuito da *protoboard* e o programa que simula o comportamento do HSM na *Virtual Machine* teve como intuito de verificar se o software estava funcionando como deveria. Foram realizadas todas as trocas de mensagens especificadas no fluxograma da figura 4.8. Os resultados obtidos validaram o correto funcionamento do sistema. O mesmo procedimento foi realizado para a placa proprietária com os componentes soldados.

A seguir serão explicitadas as respostas obtidas pelo programa a partir das trocas de mensagens definidas pelo fluxograma simplificado da figura 5.3. Esse primeiro cenário apresenta os procedimentos realizados no HSM durante a fabricação, de modo a deixar o circuito supervisor em modo de monitoramento, bem como realizar a escrita do código de segurança na memória,

verificar se realmente foi escrito a partir da leitura, obter o valor de tensão da bateria e validar o funcionamento dos relés. Assim, após o programa ser carregado, as figuras 5.4a a 5.4g mostram a sequência de mensagens trocadas relacionada ao fluxograma.

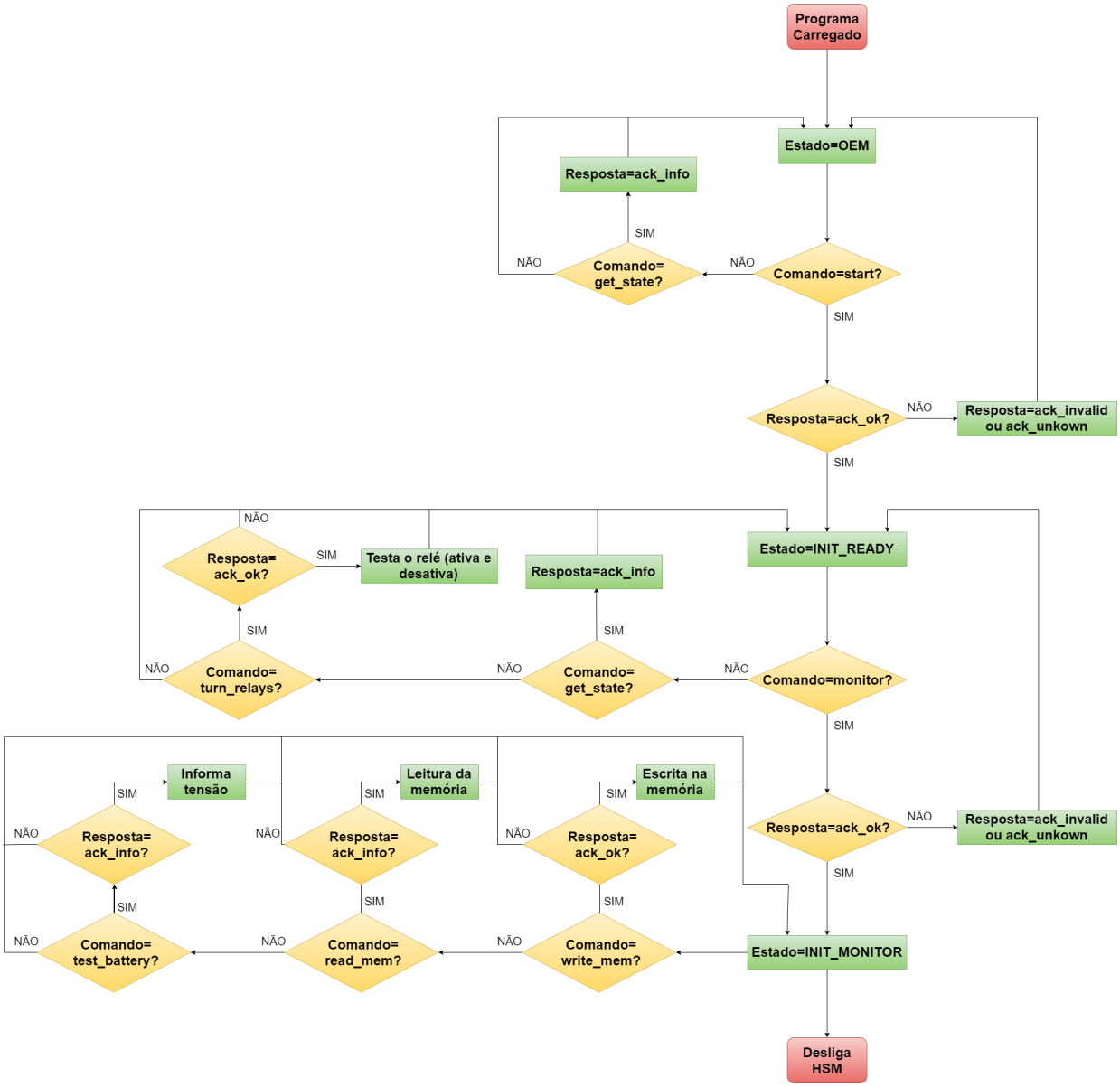
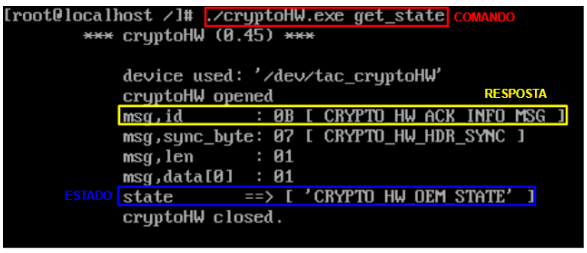
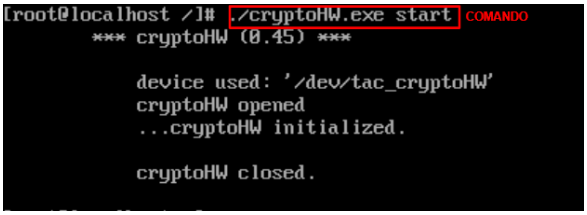


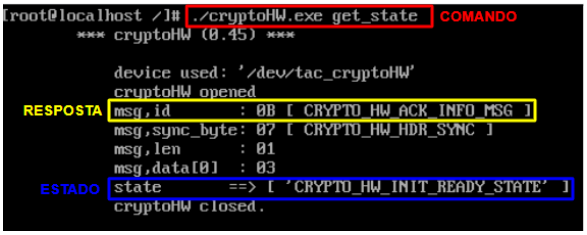
Figura 5.3: Fluxograma do primeiro cenário de testes.



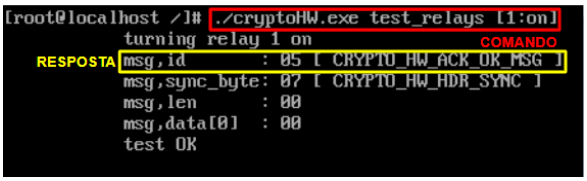
(a) Comando=get_state e Resposta=ack_info retornando o estado=OEM.



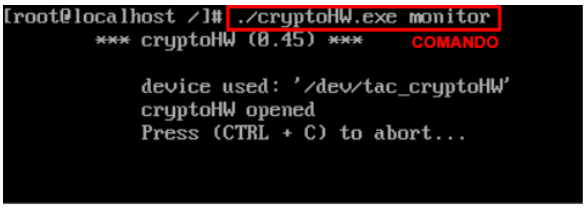
(b) Comando = start e Resposta=ack_ok ocorrendo a transição para o estado INIT_READY.



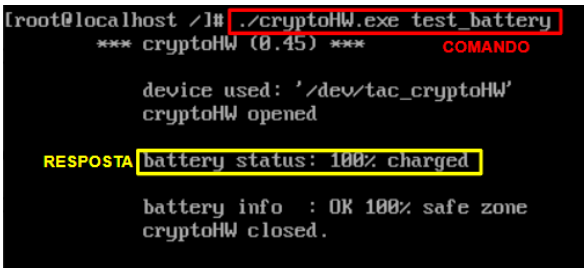
(c) Comando=get_state e Resposta=ack_info retornando o estado=INIT_READY.



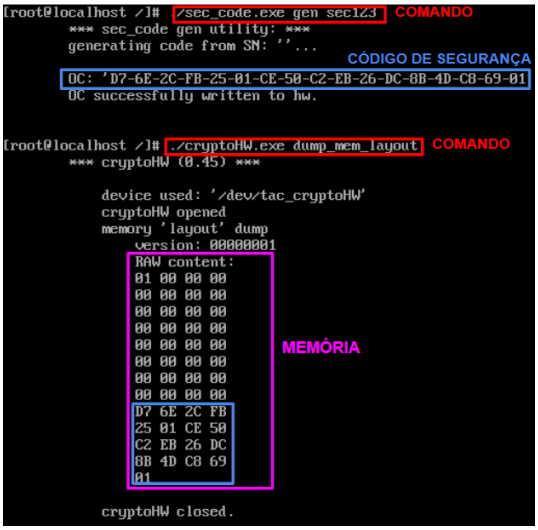
(d) Comando=test_relays(1:on) e Resposta=ack_ok retornando teste ok.



(e) Comando=monitor e ocorre a transição para o estado=INIT_MONITOR.



(f) Comando=test_battery e Resposta=ack_info retornando a porcentagem de tensão da bateria.



(g) Escrita do código de segurança na memória e leitura.

Figura 5.4: Resultados obtidos ao executar o programa de teste para o primeiro cenário.

Os testes são realizados com o auxílio do programa cryptoHW.exe, exceto para o teste de escrita do código de segurança na memória que é utilizado o programa sec_code.exe. Os dois programas de testes foram desenvolvidos pela empresa e herdados da versão anterior do sistema supervisor. Após o equipamento ser desligado, o circuito supervisor aciona o seu modo secundário de energia e passa a ser sustentado pela bateria, desse modo ao ser ligado retorna para o modo primário e o sistema supervisor verifica se ocorreu *tamper* enquanto estava desligado, em caso negativo o circuito permanece no mesmo estado de antes de ser desligado, ou seja, no INIT_MONITOR, caso contrário um alerta de *tamper* é emitido e para retornar ao estado INIT_MONITOR é necessário enviar as mensagens reset->start->monitor.

O procedimento descrito é caracterizado pelo fluxograma da figura 5.5 e foi usado para executar o segundo e terceiro cenário de teste. A figura 5.6 mostra a resposta obtida pelo programa para o cenário dois, ou seja, quando ocorre *tamper* com o HSM ligado. Já a figura 5.7 apresenta o resultado encontrado para o cenário três, no qual a violação ocorre quando o sistema está desligado, para isso é enviado uma mensagem de get_state assim que o sistema é ligado e o estado retornado é INIT_TAMPERED. Após retornar ao estado monitor foi realizada novamente a leitura na memória apresentada na figura 5.8 e verificou-se o código de segurança escrito tinha sido apagado como era esperado.

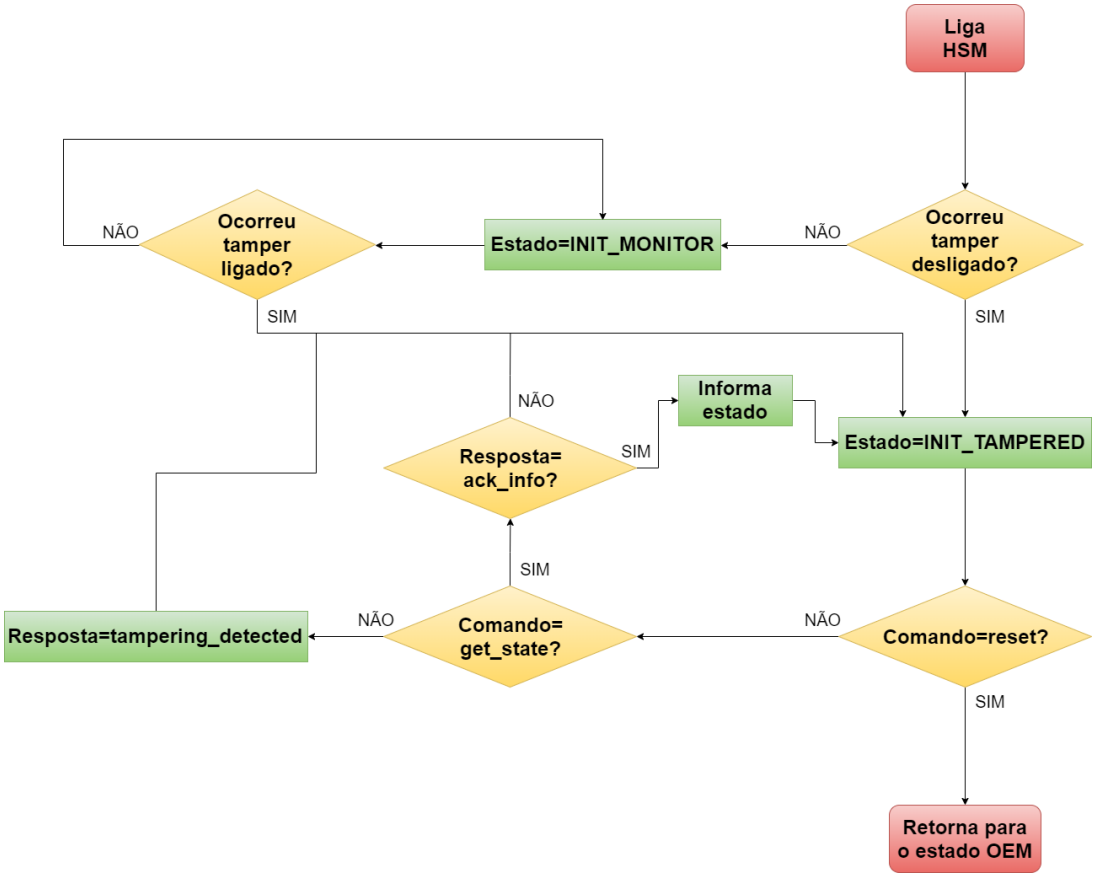


Figura 5.5: Fluxograma do segundo e terceiro cenário de testes.

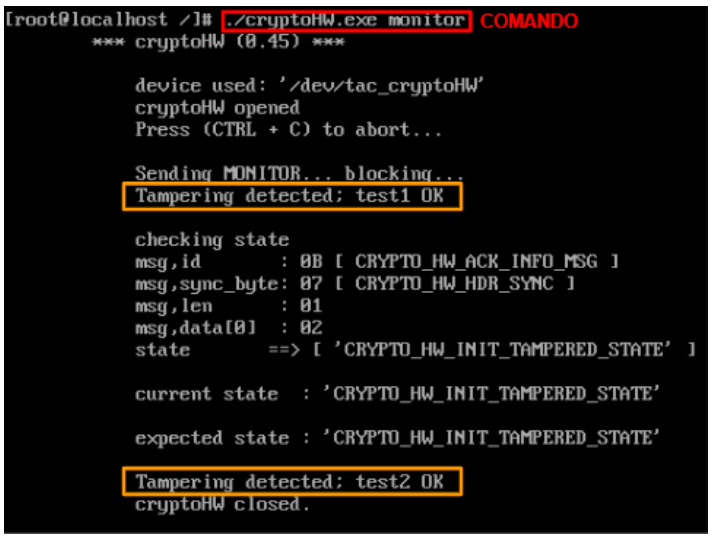


Figura 5.6: Resposta do programa quando ocorre *tamper* ligado.

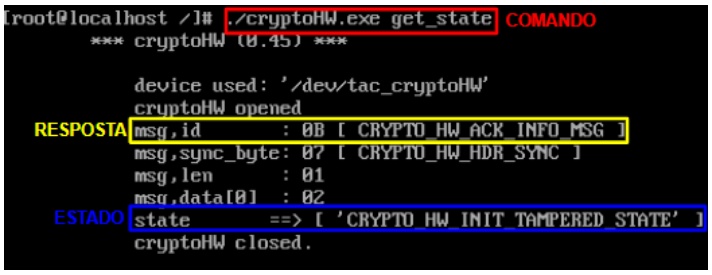


Figura 5.7: Resposta do programa quando ocorre *tamper* desligado.

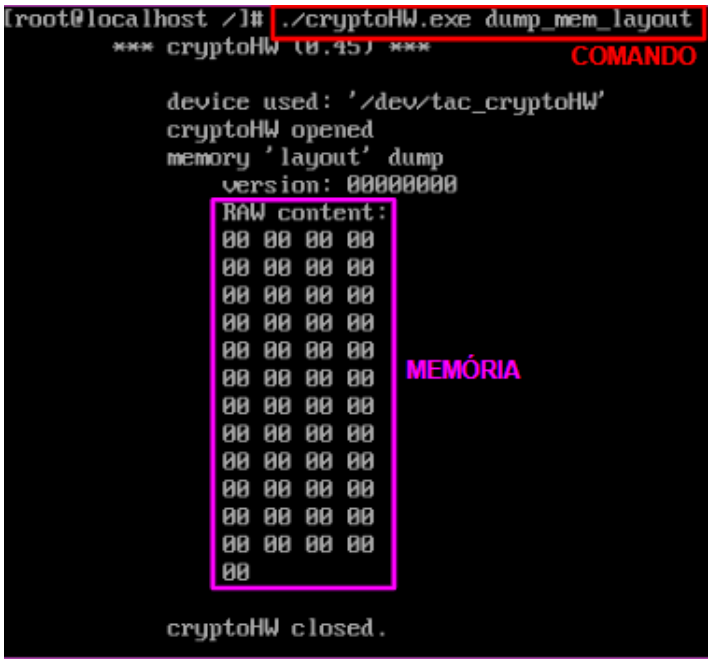


Figura 5.8: Valor da memória após uma violação.

5.2.2 Integração com o Firmware

O teste de integração com o firmware do HSM foi realizado com o sistema *shield*-Arduino. Inicialmente foi necessário criar um link da porta USB do HSM que que se comunica via um protocolo serial para a porta que o Arduino envia os comandos do protocolo de comunicação. A condição inicial determinada para os testes foi de que o circuito supervisor estava no estado INIT_MONITOR e com a memória zerada, nesse estado é possível escrever o código de segurança na memória do DS3645.

Desse modo, após a instalação, o equipamento foi ligado, assim que o processo de *start up* do HSM é concluído, o firmware libera as interfaces para configuração e foi possível verificar o acionamento do relé. Após esse procedimento a tela inicial é apresentada, como mostrado na figura III.1 dos anexos.

Durante a inicialização do HSM é definida a chave de ativação denominada *Server Master Key*, gerada e armazenada em *smart cards* a partir de um esquema de compartilhamento M de N. Durante a ativação do equipamento essa chave é lida a partir do conjunto de *smart cards* e escrita na memória volátil do HSM, assim ao ser desligado ou reiniciado essa chave é apagada e deve ser inserida novamente na próxima ativação. Todas as chaves mantidas em memória persistente no HSM são cifradas pela *Server Master Key*.

O código de segurança é gravado na memória protegida do DS3645 durante a fabricação a partir do programa ./sec_code.exe, após esse procedimento consulta-se esse valor na console do HSM para verificar se o código realmente foi escrito e o resultado encontrado é apresentado na figura III.2 dos anexos.

A remoção não autorizada da tampa ativa o circuito supervisor de violação física que desativa o relé e desliga automaticamente o HSM. As informações sobre garantia de integridade do equipamento e o código de segurança também são destruídos da memória volátil do circuito. Desse modo, o cliente sabe que a sua máquina foi violada pelo alerta apresentado pela figura III.3, o qual é emitido assim que a máquina é ligada, bem como ao consultar o código segurança e verificar que o mesmo foi apagado como mostra a figura III.4. Somente com a intervenção do fabricante é possível gravar novamente o código de segurança.

Ao selecionar a opção Y (*Yes*) da figura III.3, o firmware já envia imediatamente os comandos reset->start->monitor para que assim que o sistema seja inicializado o circuito supervisor já esteja monitorando a fronteira criptográfica. Por fim, a tensão da bateria foi consultada na console do HSM e é apresentada na figura III.5. O valor encontrado foi o mesmo medido e obtido pelo programa de teste.

A resposta à violação do HSM depende do modo de operação. Em modo *FIPS* o equipamento não poderá mais ser utilizado e deve ser enviado ao fabricante para reparo e reativação, já em modo *non-FIPS* o administrador pode resetar o registro de *tamper* e continuar operando o equipamento, entretanto este não poderá mais ser colocado em modo *FIPS*. De um modo geral, caso ocorra um *tamper* e o HSM estiver ligado, a *Server Master Key* é apagada da memória volátil e o equipamento é desligado imediatamente, se o HSM estiver desligado o sistema supervisor registrará a violação.

5.3 Fuga de Corrente

Foram levantadas várias hipóteses com o intuito de identificar qual seria a causa da bateria estar descarregando rapidamente. A primeira análise foi realizadas no protótipo na *protoboard*, já a segunda foi feita com a placa.

5.3.1 Análise no Protótipo

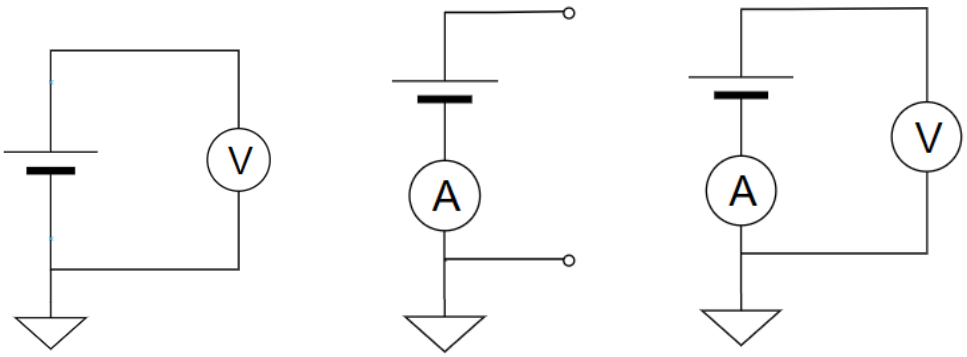
A tensão da bateria foi medida duas vezes ao dia com um multímetro minipa ET-1400, a primeira tensão era obtida às 13:00 após o circuito passar 18 horas desligado alimentado pela bateria e a segunda às 19:00 após passar 6 horas ligado alimentado pela fonte. Essas medidas foram realizadas por 10 dias e os resultados são apresentados na tabela 5.1.

Tabela 5.1: Medidas de tensão da bateria do circuito da figura 4.4.

Data	Horário	Medida
09/09/2019	19:00	3,28 V
10/09/2019	13:00	3,28 V
10/09/2019	19:00	3,28 V
11/09/2019	13:00	3,24 V
11/09/2019	19:00	3,24 V
12/09/2019	13:00	3,23 V
12/09/2019	19:00	3,23 V
13/09/2019	13:00	3,22 V
13/09/2019	19:00	3,22 V
16/09/2019	13:00	3,19 V
16/09/2019	19:00	3,19 V
18/09/2019	13:00	3,18 V

Verificou-se que a tensão da bateria só diminuía no período em que estava desligado, portanto não havia descarga quando o circuito estava alimentado pela fonte. Considerando uma descarga linear de 0,01 V/dia e sabendo que a capacidade de armazenamento da bateria usada é de 1000 mAh, em cerca de 2,5 meses com o HSM desligado a bateria descarregaria.

Com a medida de tensão não foi possível concluir se realmente havia fuga de corrente, pois só ao medir a corrente é possível realizar essa dedução. Com o auxílio de um multímetro digital Agilent - Hp 34401a foram realizados uma série de testes. Os primeiros tiveram como intuito verificar se havia fuga de corrente pela *protoboard*, para isso utilizou-se uma bateria nova com a tensão de 3,2639 V nas três disposições apresentadas nas figuras 5.9a a 5.9c.



(a) Teste 1 de fuga pela *protoboard*.

(b) Teste 2 de fuga pela *protoboard*.

(c) Teste 3 de fuga pela *protoboard*.

Figura 5.9: Circuitos para teste de fuga de corrente pela *protoboard*.

Na figura 5.9a colocou-se a bateria isolada na *protoboard* com o voltímetro e não houve alteração de tensão durante os 10 minutos de medida. Já na figura 5.9b o multímetro foi utilizado como amperímetro e o circuito estava em aberto, a corrente medida foi nula. No terceiro teste da figura 5.9c foram usados dois multímetros, um como voltímetro e o outro como amperímetro e os resultados obtidos foram de uma tensão DC estável e uma corrente de 0,4 µA. Concluiu-se que a fuga não estava sendo causada pela *protoboard*.

O segundo teste teve um caráter mais qualitativo e foi utilizado circuito da figura 4.4. A disposição do amperímetro na figura 5.10 caracteriza a medida de corrente do circuito.

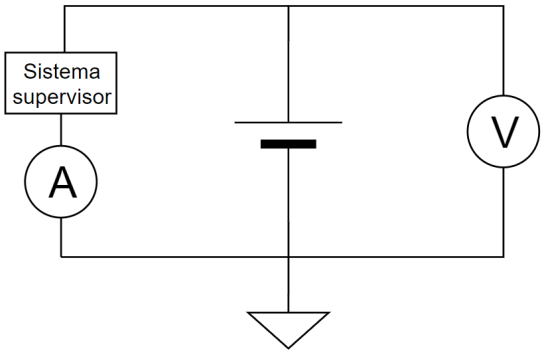


Figura 5.10: Medida de corrente no sistema supervisor.

O primeiro valor encontrado para corrente foi com o circuito no estado INIT_TAMPERED, ou seja após ocorrer uma violação, tanto para o sistema ligado quanto para ele desligado a corrente foi praticamente nula, como esperado, uma vez que o *chip* fica em um estado de congelamento e não há monitoramento.

O segundo valor obtido foi com o sistema no modo de energia fornecido pela fonte e no estado INIT_MONITOR, verificou-se que a corrente de operação era cerca de 300 µA. Nesse modo a tensão da bateria não é consumida, portanto se o sistema estiver ligado não haverá descarga.

O sistema foi desligado, dessa forma passou a ser energizado pela bateria e o seu estado é o INIT_MONITOR. Notou-se uma variação brusca de corrente dependendo das condições do

circuito, qualquer movimento na *protoboard* gerava uma alteração, a quantidade de fios produzia capacitâncias parasitas e quando era colocada a mão próxima aos fios o valor de corrente disparava, desse modo havia fuga devido ao acoplamento capacitivo.

Além disso, percebeu-se que o circuito ficava em contato direto com uma manta antiestática sem qualquer tipo de isolamento entre o chip, o Arduino e a bancada. Assim, a causa da fuga poderia ser por esse cenário, já que a manta conectada ao sistema de aterramento fornecia uma área condutiva que proporcionaria caminhos de descarga.

5.3.2 Análise na Placa

A placa foi projetada com dois pontos para medida de corrente, um representado pela figura 5.10 e o outro pela figura 5.11.

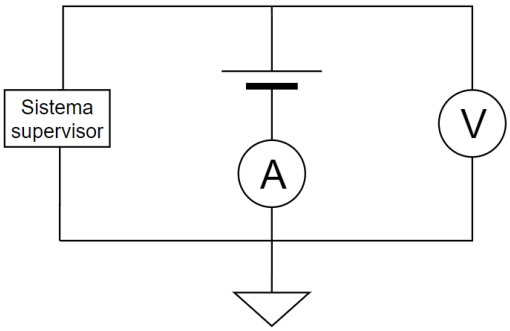


Figura 5.11: Medida de corrente na bateria.

Visando não cometer os mesmos erros com o sistema na *protoboard*, assim que os componentes da placa foram soldados também foi acrescentado espaçadores de nylon de modo que o sistema não ficasse em contato com a manta antiestática. A primeira medida foi obtida com o sistema desligado no estado de monitoramento com o amperímetro na posição da figura 5.10, a corrente encontrada foi de 4,3 µA.

Ao medir a corrente da bateria com o amperímetro na posição especificada na figura 5.11 o resultado obtido foi de 11,2 µA, essa medida corresponde ao valor especificado no *datasheet* do chip DS3645. Sabendo que a capacidade da bateria é de 1000 mAh, a equação 5.1 apresenta o tempo de vida da bateria.

$$Tempo = \frac{1Ah}{11,2 * 10^{-6}A} \approx 10anos \tag{5.1}$$

A partir da corrente medida a bateria deveria durar 10 anos, o que não é verdade pelas medidas de tensão, uma vez que esse valor diminui cerca de 0,01V por dia, assim ao considerar essa medida e uma descarga linear a bateria duraria apenas 80 dias. Ao analisar o circuito o único lugar que poderia haver fuga é no lugar onde está sendo medida a corrente.

Outra hipótese levantada foi a confiabilidade da bateria. Desse modo, foram analisados o comportamento da tensão de duas baterias uma com 3,2635 V e a outra com 3,0907 V por um

período de tempo de 15 minutos. Na primeira verificou-se que a tensão aumentava na quarta casa decimal a cada minuto, já na segunda diminuía. Não foi possível realizar conclusões acerca dessa medida, pois para que esse valor lido possa ser utilizado para uma boa estimação da carga, é necessário aguardar um período de tempo no qual a carga estabiliza a sua tensão. Esse tempo costuma ser muito longo para obter essa precisão [19]. Outra medida realizada foi a de resistência entre os terminais do *socket* da bateria com o intuito de verificar se ele era a causa do vazamento, entretanto o valor encontrado foi mínimo.

A figura 5.12 apresenta a curva de descarga fornecida pelo fabricante utilizada no sistema supervisor. Verifica-se que no momento inicial assim que a carga é ligada há uma rápida queda da tensão devido à resistência interna da bateria. Posteriormente ocorre uma queda de tensão quase linear, lenta, ou seja, a bateria apresenta uma pequena redução de tensão durante um longo período de tempo. Já no final de sua vida ocorre uma redução acentuada em um curto período de tempo.

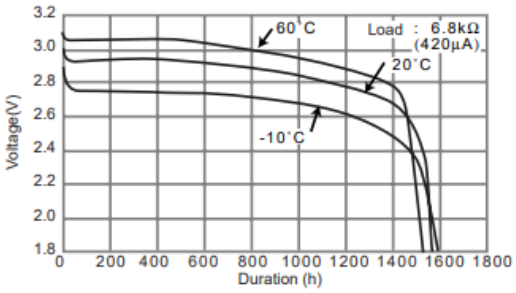


Figura 5.12: Curva de descarga da bateria de lítio do tipo moeda da Panasonic [31].

No teste em progresso com essa bateria na placa, o circuito foi colocado no modo econômico de energia e a corrente da bateria quando o circuito está desligado passou de 11 µA para 6 µA. A tensão inicial era de 3,26 V, após 24 horas mediu-se esse valor e houve uma queda de 0,03V que pode ser explicado pela resistência interna da bateria. Em seguida, pelo monitoramento dos 8 dias seguintes ocorreu uma redução de 0,05 V. Devido ao tempo de medida ter sido apenas de 10 dias, não foi possível determinar em qual momento a bateria alcançará o final de sua vida e se essa descarga é linear, ou se em algum momento ocorre a redução da queda em relação a quantidade de dias ou se realmente há uma descarga precoce.

Capítulo 6

Conclusão

Visando a proteção contra ataques físicos foi desenvolvido um sistema integrado ao Módulo de Segurança Criptográfica que faz o monitoramento da fronteira física. A principal motivação foi em decorrência dos componentes utilizados no circuito supervisor desenvolvido anteriormente estarem obsoletos, o que torna difícil encontrá-los. Desse modo, foram escolhidos materiais que ainda não possuem *end of life* definido e fornecem possibilidades para incrementar as funcionalidades.

A primeira contribuição foi o projeto do hardware capaz de detectar e responder a violação, para isso foi desenvolvido um protótipo em *protoboard* e uma placa de circuito impresso utilizada como uma *shield* acoplada ao Arduino Due. Nesse sistema foram utilizados os sensores *micro switches* para detectar a tentativa de retirada da tampa superior e com o auxílio do chip supervisor DS3645 foi possível detectar e responder a violação, além disso a partir do sensor de temperatura no chip são identificados ataques de impressão de temperatura.

Também foi desenvolvido um sistema de gerenciamento de energia capaz de manter o código de segurança escrito na memória protegida do DS3645 em caso de falha de energia, no qual ocorre o chaveamento para o modo de energia fornecido pela bateria. Por outro lado em caso de violação esse dado é apagado da memória e o sistema fica em um estado de *tamper*.

Outra contribuição foi o desenvolvimento do firmware do sistema, esse programa foi implementado na IDE *(Integrated Development Environment)* do Arduino e foi baseado na máquina de estados da figura 4.5 e no protocolo de comunicação desenvolvido pela empresa usado na versão anterior do circuito supervisor.

Durante a etapa de desenvolvimento, o primeiro circuito foi montado em um Arduino Mega, após a escolha do microcontrolador Atmel SAM3X8E ARM Cortex-M3 CPU e em decorrência do circuito ter parado de funcionar após alguns testes, foi implementado um novo sistema na placa Arduino Due.

Além disso, para diminuir a quantidade de fios desabilitou-se todas as entradas digitais e comparadores do chip via firmware, em seguida os fios de aterramento dessas entradas foram retirados, porém ao ligar e desligar a luz da bancada uma violação era gerada, isso ocorria, pois as entradas estavam flutuando e havia, assim foi necessário aterrá-las novamente.

Após a realização de todos os testes, a placa foi instalada ao HSM e integrada ao firmware do equipamento. O comportamento obtido foi o mesmo do sistema da versão anterior, desse modo todos os objetivos foram alcançados e ao integrar a placa ao Módulo de Segurança Criptográfica verificou-se a compatibilidade completa do sistema.

6.1 Sugestão para Trabalhos Futuros

Uma das sugestões para trabalhos futuros é ampliar as funcionalidades do sistema supervisor. Na *shield* alguns sinais foram deixados disponíveis para permitir essas melhorias, desse modo é possível acrescentar uma manta *mesh* que envolverá a fronteira criptográfica e será capaz de detectar ataques caso o gabinete seja perfurado. Essa tecnologia permitiria que a máquina fosse homologada em um nível de segurança física 4.

Além disso, acrescentar novas mensagens ao protocolo, como um comando que pergunte ao controlador o que causou o *tamper* e a resposta ao hospedeiro indicaria se foi pelas entradas digitais, temperatura, falha de bateria, frequência do cristal muito alta ou baixa, entradas comparadoras ou tensão acima ou abaixo dos níveis especificados. Outra mensagem indicaria o horário no qual a violação ocorreu. Outra possibilidade é o desenvolvimento de um programa de teste sem intervenção humana e que gere um relatório de validação.

Outra proposta é elaborar testes para analisar as causas da fuga de corrente. No primeiro, com o auxílio do Arduino será desenvolvido um sistema que mede a tensão da bateria e armazena esse valor em um arquivo texto a cada minuto, desse modo com um valor de resistor e temperatura conhecidos será possível obter a curva de descarga da bateria. Em seguida serão adquiridas baterias de outros fabricantes para analisar o comportamento de cada e verificar se o problema é com a bateria utilizada.

A versão final que será incorporada ao equipamento não utilizará o Arduino Due, assim será desenvolvida uma placa proprietária com o processador Atmel SAM3X8E ARM Cortex-M3 CPU, o chip DS3645 e componentes auxiliares. Esse microcontrolador foi escolhido por sua tensão de operação ser de 3,3 V e por ter 32 bits possui maior capacidade de processamento em relação aos microcontroladores de 8 bits.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] METALTEX ML2RC-5V. https://http2.mlstatic.com/rele-miniatura-8-pinos-2a-30vdc-1a-125vac-ml2rc-5v-metaltex-D_NQ_NP_819859-MLB27932340120_082018-F.jpg. Acessado em 28/10/2019.

[2] U. S. Federal Standart 2017 Telecommunications: Genereal Security Requirements for Equipment Using the Data Encryption Standart.

[3] M. Aarts. Hardware attacks tamper resistance, tamper response and tamper evidence. *Date of retrieval*, 23, 2016.

[4] D. G. Abraham, G. M. Dolan, G. P. Double, and J. V. Stevens. Transaction security system. *IBM Systems Journal*, 30(2):206–229, 1991.

[5] F. Amiel, K. Villegas, B. Feix, and L. Marcel. Passive and active combined attacks: Combining fault attacks and side channel analysis. In *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2007)*, pages 92–102. IEEE, 2007.

[6] R. Anderson. *Security engineering*. John Wiley & Sons, 2008.

[7] ANSI X9.31. Public-key cryptography using RSA for the financial services industry, American National Standart for Financial Services, draft. 1995.

[8] ARDUINO. Arduino Due. https://store.arduino.cc/usa/due. Acessado em 10/10/2019.

[9] T. Caddy. Fips 140-2. *Encyclopedia of Cryptography and Security*, pages 468–471, 2011.

[10] Circuit Basics. BASICS OF THE I2C COMMUNICATION PROTOCOL. http://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/. Acessado em 27/06/2019.

[11] Digi-Key. MS0850503F055P1A. https://www.digikey.com/product-detail/en/e-switch/MS0850503F055P1A/EG5431-ND/1628125. Acessado em 12/10/2019.

[12] DS3645. 4KB Secure Memory with Tamper Protection for Network Server Applications. https://www.maximintegrated.com/en/products/embedded-security/security-managers/DS3645.html. Acessado em 26/06/2019.

[13] E. Dubrova. Anti-tamper techniques. *KTH Royal Institute of Technology, Sweden*, 2018.

[14] EAGLE. Software de Projeto de Placa de Circuito Impresso. https://www.autodesk.com.br/products/eagle/overview. Acessado em 26/06/2019.

[15] Eletodex. Relé metaltex ML2RC-5V. https://www.eletrodex.com.br/rele-metaltex-ml2rc-5v-5vdc.html. Acessado em 10/10/2019.

[16] Eletrônica em casa. Como acionar um relé com Arduino ou PIC - projeto e funcionamento. http://eletronicaemcasa.blogspot.com/2013/12/como-acionar-um-rele-com-arduino-ou-pic.html. Acessado em 10/10/2019.

[17] P. FIPS. 186-4. *Digital Signature Standard (DSS)*, 2013.

[18] J. Grand. Protecting your crown jewels: an introduction to embedded security for hardware-based products. *Computer Fraud & Security*, 2005(10):13–20, 2005.

[19] Guilherme Andriotti Momesso. Estimativa do estado de carga de baterias de íon-lítio com aplicação em sistemas isolados de geração fotovoltaica. http://www.tcc.sc.usp.br/tce/disponiveis/97/970010/tce-27032019-115547/publico/Momesso_Guilherme_tcc.pdf. Acessado em 21/11/2019.

[20] S. Harris. *Physical and Environmental Security. In CISSP Exam Guide*. USA McGraw-Hill, 6 edition, 2013.

[21] D. Hutter. Physical security and why it is important. In *SANS Institute Information Security Reading Room*. SANS, 2016.

[22] ICP-Br. MCT 7 - Volume I – Requisitos, Materiais e Documentos Técnicos para Homologação de Módulos de Segurança Criptográfica (MSC) no Âmbito da ICP-Brasil, Versão 2.2. Manual de condutas técnicas, Instituto Nacional de Tecnologia da Informação, Brasília, Setembro 2017.

[23] A. D. N. ICP-BRASIL. Visão geral sobre assinaturas digitais na icp-brasil.

[24] ITI - Instituto Nacional de Tecnologia da Informação. Entes da ICP-Brasil. https://www.iti.gov.br/icp-brasil/entes-da-icp-brasil. Acessado em 03/06/2019.

[25] M. Kuhn and O. Kömmerling. Physical security of smartcards. *Information Security Technical Report*, 4(2):28–41, 1999.

[26] M. A. P. Laureano, C. A. Maziero, and E. Jamhour. Detecção de intrusão em máquinas virtuais. *5º Simpósio de Segurança em Informática–SSI. São José dos Campos*, pages 1–7, 2003.

[27] X. Leng. Smart card applications and security. *information security technical report*, 14(2):36–45, 2009.

[28] K. Markantonakis, M. Tunstall, G. Hancke, I. Askoxylakis, and K. Mayes. Attacking smart card systems: Theory and practice. *information security technical report*, 14(2):46–56, 2009.

[29] Mouser Eletronics. CR2477. https://br.mouser.com/ProductDetail/Panasonic-Battery/CR2477?qs=1eQvB6Dk1vhM2dgP2k1\%2FFg\%3D\%3D. Acessado em 12/10/2019.

[30] D. Page. Defending against cache-based side-channel attacks. *Information Security Technical Report*, 8(1):30–44, 2003.

[31] Panasonic. LITHIUM - Coin Type. https://datasheet.octopart.com/CR2477-Panasonic-datasheet-9626328.pdf. Acessado em 23/11/2019.

[32] Proto Advantage. BGA-100 SMT Adapter (0.8 mm pitch, 10 x 10 grid). http://www.proto-advantage.com/store/product_info.php?products_id=4000001. Acessado em 26/06/2019.

[33] W. Rankl. Overview about attacks on smart cards. *Information Security Technical Report*, 8(1):67–84, 2003.

[34] Robert W. Baldwin. Design tricks for great products at FIPS-140-2 Level 2 and 3. Plus Five Consulting, Inc. http://www.plusfive.com/DEV-104_Baldwin_v7a.pd. Acessado em 06/06/2019.

[35] B. Schneier. *Applied cryptography: protocols, algorithms, and source code in C*. john wiley & sons, 2007.

[36] L. G. C. Silva, L. G. SILVA, I. J. AQUINO JUNIOR, E. M. BATISTA, H. O. HOMOLKA, and M. F. LIMA. Certificação digital-conceitos e aplicações. *Editora Ciência Moderna*, 2008.

[37] S. Skorobogatov. Physical attacks on tamper resistance: progress and lessons. In *Proc. of 2nd ARO Special Workshop on Hardware Assurance, Washington, DC*, 2011.

[38] W. Stallings. Criptografia e segurança de redes: princípios e práticas 4 ed são paulo, 2006.

[39] S. H. Standard. Fips pub 180-1. *National Institute of Standards and Technology*, 17:15, 1995.

[40] P. Tuyls, B. Škoric, and T. Kevenaar. *Security with noisy data: on private biometrics, secure key storage and anti-counterfeiting*. Springer Science & Business Media, 2007.

[41] D.-C. Vasile and P. M. Svasta. Innovative conductive mesh structure for the protection of security electronic circuits. In *2018 7th Electronic System-Integration Technology Conference (ESTC)*, pages 1–6. IEEE, 2018.

[42] S. H. Weingart. Physical security for the μabyss system. In *1987 IEEE Symposium on Security and Privacy*, pages 52–52. IEEE, 1987.

[43] S. H. Weingart. Physical security devices for computer subsystems: A survey of attacks and defenses. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 302–317. Springer, 2000.

[44] J. D. Zunino. Certificação digital: assinatura digital, certificados digitais e sua utilização no mercado nacional. *Maiêutica-Tecnologias da Informação*, 2(01), 2017.

ANEXOS

I. DOCUMENTAÇÃO UTILIZADA PARA O PROJETO DA *BREAKOUT BOARD*

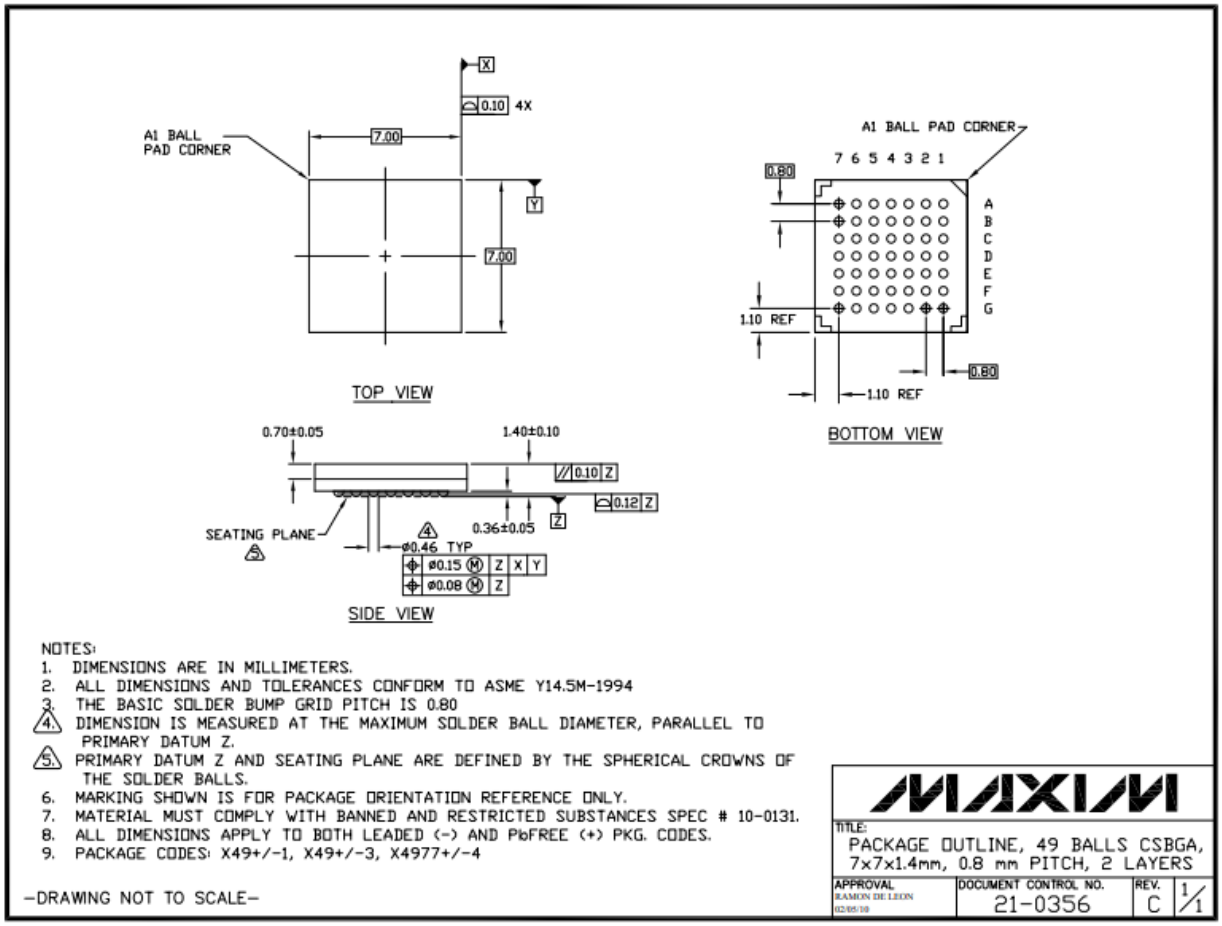


Figura I.1: Desenho de esboço [12]

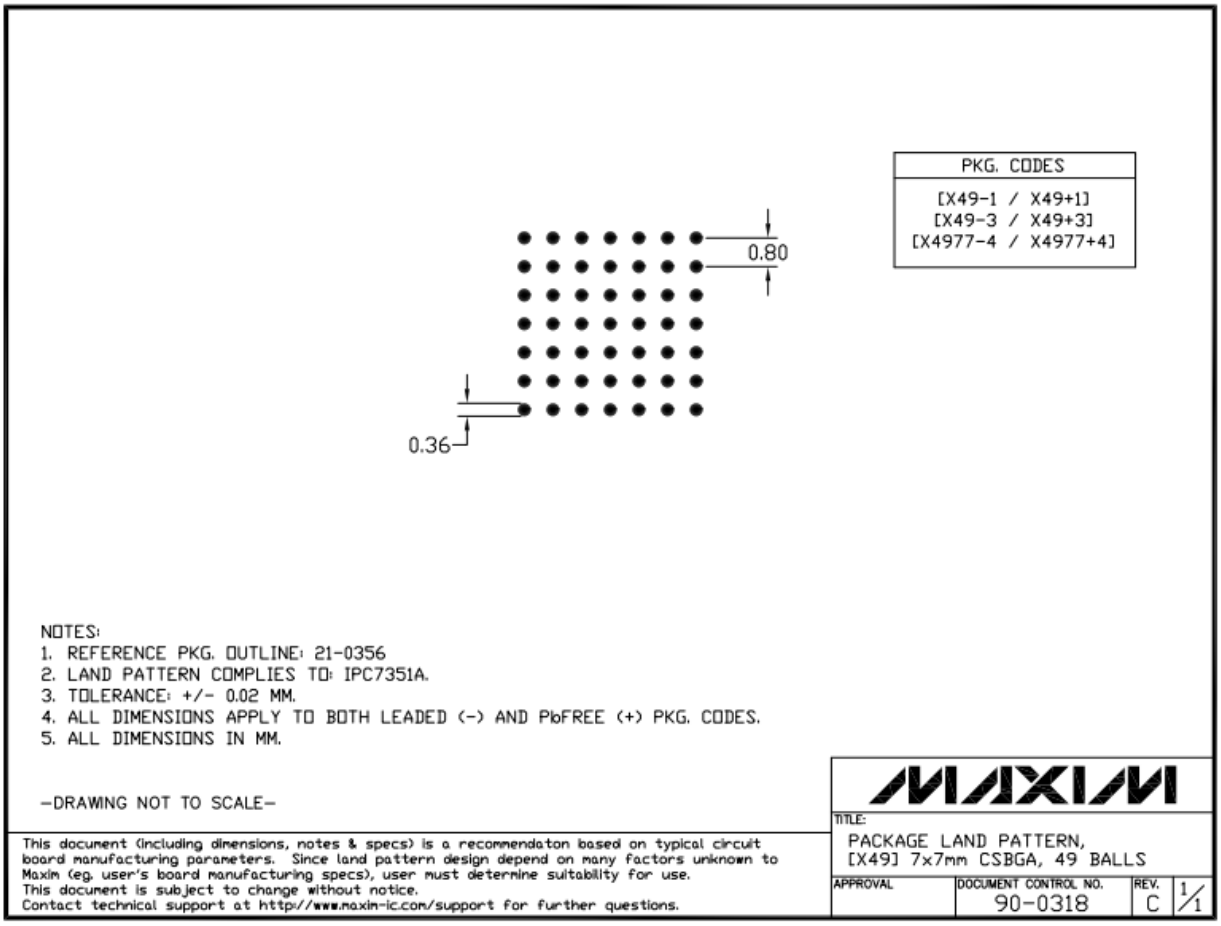


Figura I.2: *Land Pattern* [12]

II. DOCUMENTAÇÃO DO RELÉ METALTEX ML2RC-5V

METALTEX

Relé miniatura / Miniature relay

ML

- **Ideal para uso em telecomunicações**
- **2 contatos reversíveis bifurcados**
- **Comutação de cargas a partir de 10 µA**
- **Montagem direta em circuito impresso**
- **Selado**
- **Homologado UL**

- Suitable for telecom applications
- DPDT bifurcated contats
- Direct PC mounting
- Sealed
- UL recognized

Chave de código / How to order

ML2R (C2)

Tensão nominal da bobina
Coil nominal voltage

C-5V - 5VCC / VDC
C1 - 6VCC / VDC
C2 - 12VCC / VDC
C3 - 24VCC / VDC
C4 - 48VCC / VDC

Especificações de bobina / Coil specifications

Modelo Type	Tensão Nominal Nominal Voltage VCC / VDC	Máx. Tensão Contínua Max. Allowable Voltage VCC / VDC	Tensão de Operação Pick-up Voltage VCC / VDC	Tensão de Desoperação Drop-out Voltage VCC / VDC	Consumo Nominal Nominal Comsumption mW	Resistência (±10%) Resistance (±10%) Ω*
C-5V	5	10,0	≤ 3,75	≥ 0,5	200	125
C1	6	12,0	≤ 4,50	≥ 0,6	200	180
C2	12	24,0	≤ 9,00	≥ 1,2	200	720
C3	24	48,0	≤ 18,0	≥ 2,4	200	2880
C4	48	56,0	≤ 36,0	≥ 4,8	560	4000

Figura II.1: Especificações técnicas do relé [1]

III. TELAS DO TESTE DE INTEGRAÇÃO COM O FIRMWARE DO HSM

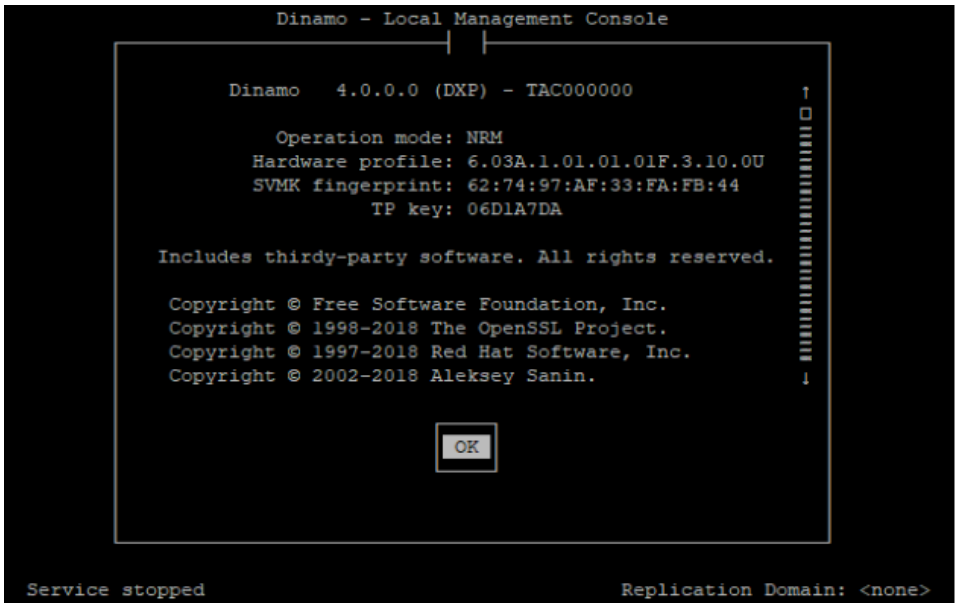


Figura III.1: Tela inicial apresentada pelo HSM quando um *tamper* não foi detectado.

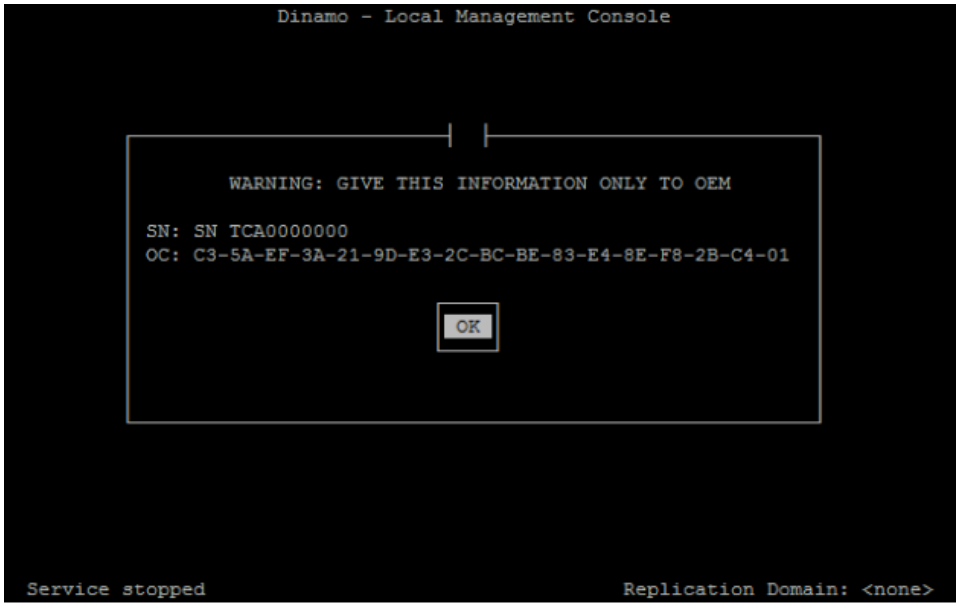


Figura III.2: Exibição do código de segurança.

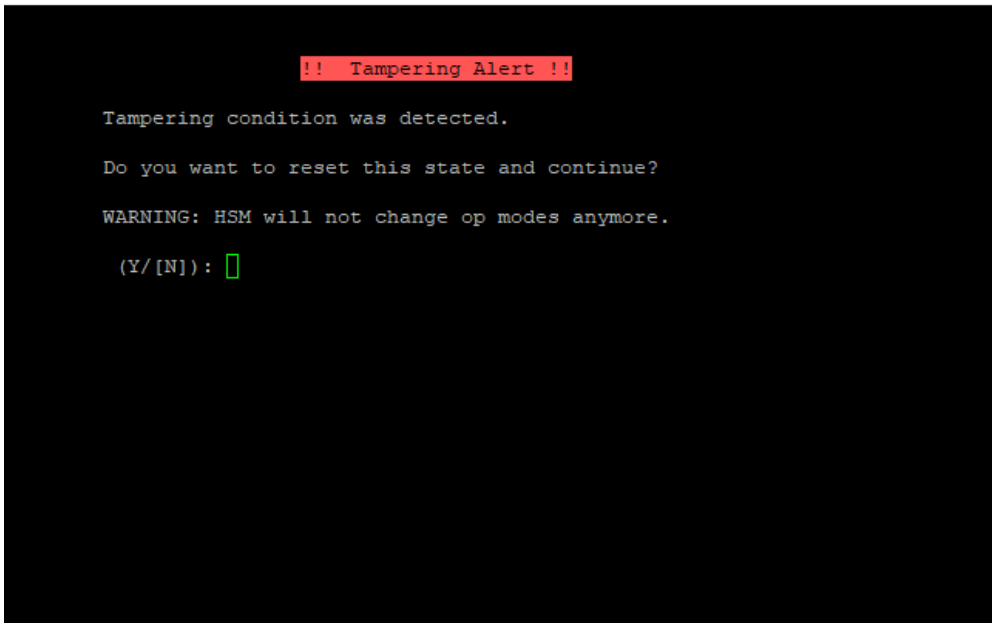


Figura III.3: Exibição do alerta de *tamper*.

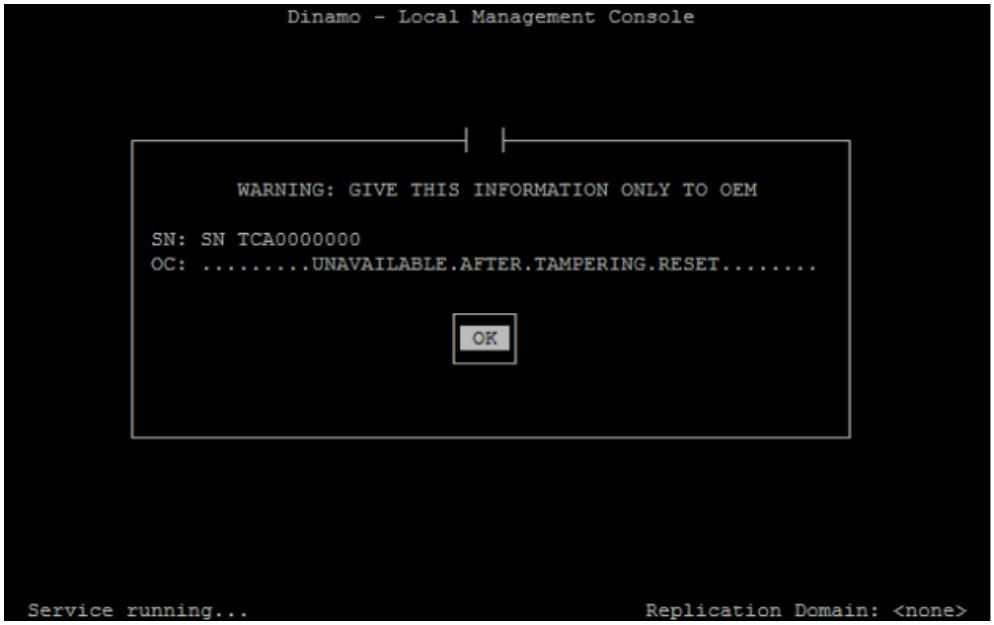


Figura III.4: Exibição do código de segurança após a ocorrência do *tamper*.

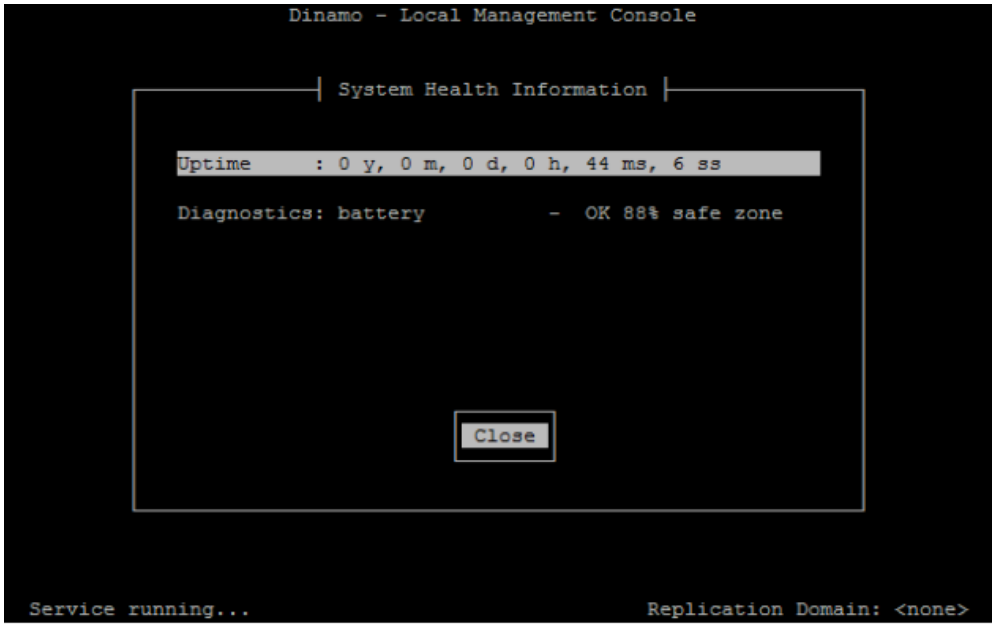


Figura III.5: Tela de informação de tensão na bateria.