Um Data Diode com Hardware Criptográfico para Redes Industriais Críticas

Gabriel Carrijo B. Teixeira[1], **Eduardo L. Feitosa**[1]

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
CEP 69.077-000 – Manaus – AM – Brasil

{gabrielcarrijo,efeitosa}@icomp.ufam.ed.br

Abstract. *This paper presents a security scheme capable of addressing the problems encountered in the integration of critical industrial networks with insecure corporate networks, aiming to guarantee data integrity and reliability between the devices. For this purpose, it is proposed the use of a Data Diode in the interconnection of the networks for the protection of the industrial plant and a TPM cryptographic hardware (Trusted Platform Module) to guarantee the integrity and reliability of the devices involved. As a way of proving the effectiveness of this architecture, tests were performed, which, at the end of the work, show that it is possible to achieve superior results to those already existing in the literature.*

Resumo. *Este artigo apresenta um esquema de segurança capaz de tratar os problemas encontrados na integração de redes industriais críticas com redes corporativas inseguras, objetivando garantir integridade dos dados e confiabilidade entre os dispositivos. Para esse fim é proposto a utilização de um Data Diode na interligação das redes para a proteção da planta industrial e um hardware criptográfico TPM (Trusted Platform Module) para garantia de integridade e confiabilidade dos dispositivos envolvidos. Como forma de provar a efetividade dessa arquitetura, foram realizados testes, que ao final no trabalho, mostram que é possível alcançar resultados superiores aos trabalhos já existentes na literatura.*

1. Introdução e Motivação

Redes industriais são geralmente relacionadas a infraestruturas críticas (geração de energia elétrica, gás e petróleo, tratamento e fornecimento de água, transporte, entre outras) e devido à sua aplicabilidade, demandam alta disponibilidade na operação e proteção contra incidentes deliberados e inadvertidos. Antigamente, o isolamento era a forma mais simples e utilizada para garantir a segurança dessas redes. Contudo, a necessidade por aumentar a produtividade e o desempenho financeiro fizeram com que essas redes fossem integradas à redes corporativas com acesso a Internet. Consequentemente, problemas de segurança tornaram-se mais frequentes e ganharam notoriedade em 2010, quando o Worm Stuxnet infectou várias industrias no Irã, inclusive uma planta de produção nuclear [Moreira et al. 2016].

Uma solução para o problema é o uso de gateways unidirecionais que permitem a transmissão de dados em uma única direção. Esta tecnologia é comercializada e denominada como *Data Diode*. Entretanto, ela apresenta problemas de segurança e confidencialidade, uma vez que não permite nenhuma sinalização de controle do tráfego,

reconhecimento dos pacotes e outros mecanismos de segurança dos protocolos orientados à conexão.

Neste cenário, tomando como hipótese que um *Data Diode* com um hardware criptográfico integrado, para autenticação mútua e controle de retransmissão de dados em tempo real, pode resolver certos problemas, este artigo propõe um novo *Data Diode* utilizado como gateway de comunicação em infraestruturas críticas, possibilitando a comunicação segura entre as redes envolvidas e garantindo a confiabilidade dos dispositivos e a integridade dos dados trafegados. A ideia é integrar um *Data Diode* a um hardware criptográfico TPM (*Trusted Platform Module*), visando garantias de confiabilidade e integridade na transmissão de dados.

Este artigo está organizado da seguinte forma: as Seções 2 e 3 apresentam conceitos básicos sobre *Data Diode* e TPM respectivamente. A Seção 4 descreve os trabalhos relacionados. Em seguida, as Seções 5 e 6 detalham a arquitetura e a prototipação realizada. A Seção 7 apresenta e discute os resultados experimentais da arquitetura proposta. Em sequência a Seção 8 descreve as dificuldades encontradas. Por fim, a Seção 9 apresenta as considerações finais.

2. Data Diode

Data Diodes, também conhecidos como *One-Way Data Transfer* ou *Unidirectional Security Gateway*, podem ser implementados de várias formas. As três mais usuais são [Jeon and Na 2016]: (1) **Simple One-way Communication**, onde o tráfego segue uma única direção, em tempo real, impossibilitando retransmissão e prejudicando a integridade da informação; (2) **Bilateral Communication using Multiple Data Diodes**, que utiliza dois gateways unidirecionais em paralelo, cada um enviando dados no sentido contrário; (3) **Control Signal transmission by Feedback Node**, que emprega três nós de comunicação (transmissor, receptor e *feedback*), interligados por gateways unidirecionais, onde o nó *feedback* recebe a informação sobre a verificação de integridade do nó receptor. A Figura 1 ilustra esses três tipos de implementação.

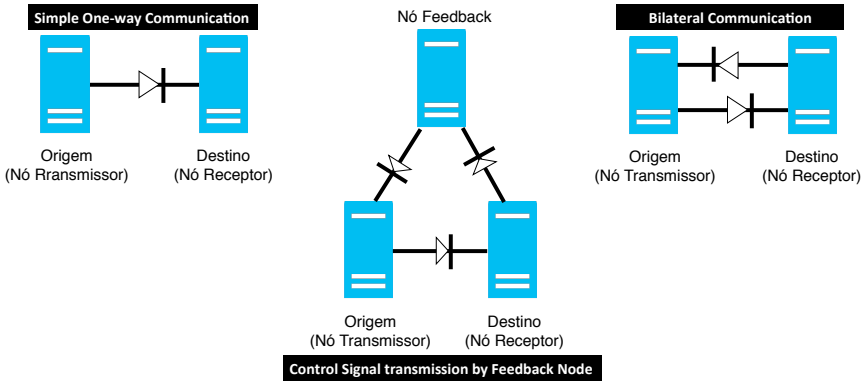

Figura 1. Estratégias para Implementar e Implantar *Data Diodes*
[Jeon and Na 2016].

2.1. Desafios com Data Diodes

A operação de um *Data Diode* em ambientes reais apresenta três (03) dificuldades. A primeira é o tráfego de protocolos de comunicação bidirecionais por meio de um canal

unidirecional [Oh et al. 2015]. Protocolos baseados em TCP (*Transmission Control Protocol*) não se adequam ao uso em gateways unidirecionais por se tratarem de protocolos orientados à conexão. Embora protocolos baseados no UDP (*User Datagram Protocol*) pareçam ser os mais indicados, eles não possuem mecanismos que garantam o ordenamento dos pacotes e a integridade dos dados transmitidos.

A segunda é a confiabilidade da transmissão unidirecional dada a falta de retransmissão de dados. O uso de código Reed-Solomon ou FEC (*Forward Error Correction*) para recuperar partes da informação sem a necessidade de retransmissão tem sido proposto em trabalhos como [Heo et al. 2016], mas o código Reed-Solomon permite detectar n caracteres incorretos em uma transmissão e recuperar apenas $n/2$ caracteres. Assim, em transmissões ruidosas ou com alto índice de perda de quadros a integridade dos dados estará totalmente comprometida.

A terceira dificuldade é falta de relação de confiança na comunicação, ou seja, não há garantia de que a origem é realmente o ativo responsável por transmitir os dados para o outro segmento de rede ou que o destinatário é o sistema correto para receber as informações e processá-las. As propostas e implementações de *Data Diodes* disponíveis na literatura são completamente suscetíveis a ataques *man-in-the-middle*. Os mecanismos de estabelecimento de conexão e verificação mais simples (ainda que falhos) das comunicações bidirecionais não estão presentes no cenário unidirecional. Portanto, se faz necessária a implementação de um mecanismo seguro de autenticação mútua e garantia de confidencialidade da informação e confiabilidade entre dispositivos, como TPM.

3. TPM

TPM (*Trusted Platform Module*) é uma tecnologia de segurança de dados emergente na última década e alvo de vários estudos que buscam comprovar sua efetividade ou encontrar falhas em sua arquitetura [Winter and Dietrich 2013]. Fisicamente, TPM é um chip composto por um coprocessador criptográfico. A especificação dessa tecnologia é gerida e atualizada pelo TCG (*Trusted Computer Group*)[1]. A versão atual é a 2.0.

O TPM possui três importantes funcionalidades [Osborn and Challener 2013] aplicadas à este trabalho. A primeira é a inicialização confiável (*Measured Boot*), um processo no qual o software de pré-inicialização do dispositivo utiliza o TPM para verificar se não houveram mudanças no ambiente de inicialização, através de hashes obtidos previamente. A ideia é que durante a inicialização do sistema haja uma medição de todos os componentes utilizados (BIOS, *bootloader* e sistema operacional) gerando um valor de hash armazenado no TPM. Esses valores são salvos pelo TPM em um conjunto de registradores chamado PCR (*Platform Configuration Registers*), que não podem ser ajustados ou diretamente introduzidos por nenhum hardware ou software, a não ser o próprio TPM. Os valores registrados no PCR recebem uma assinatura criptográfica com uma chave interna do TPM, que servirá como um atestado da configuração de software do dispositivo, que poderá ser enviada posteriormente a um sistema solicitante, com uma prova do estado e integridade do sistema. Em outras palavras, toda vez que o sistema é iniciado, um hash é gerado e comparado com o armazenado no TPM, garantindo assim a confiança e integridade do sistema.

[1] http://www.trustedcomputinggroup.org

Após a configuração correta dos valores no PCR, ocorre o processo de armazenamento selado (*Sealed Storage*) - segunda funcionalidade, onde as chaves criptográficas disponíveis no dispositivo são seladas, tornando-se acessíveis apenas se o processo de inicialização confiável for finalizado. Assim, se um malware se instalar alterando o processo de inicialização do sistema, as chaves ficarão inacessíveis.

Por fim, a terceira funcionalidade é a atestação de rede (*Network Attestation*), que consiste na prova criptográfica a um host remoto de que a plataforma está em um estado particular (seguro), para que possa se comunicar com outros dispositivos. Por exemplo, um servidor cria uma chave criptográfica aleatória e envia ao cliente. Essa chave aleatória é enviada ao TPM, que gera um hash com os valores presentes na PCR juntamente com a chave enviada, que é assinado por uma chave de identificação. Esses dados são enviados ao servidor, onde é verificado o estado da plataforma e a chave de identificação para liberação de acesso à rede.

Vale destacar que essas funções, além de garantir a integridade do sistema local, fornecem um mecanismo confiável de autenticação remota totalmente baseado em hardware.

4. Trabalhos Relacionados

No que tange os trabalhos relacionados, uma revisão sistemática sobre o assunto foi realizada e ao final apenas seis (06) artigos foram selecionados e estudados. Contudo, apenas três (03) são apresentados aqui, pois tratam efetivamente da proposta e implementação de um *Data Diode*.

Arkhangelskii et al. [Arkhangelskii et al. 2016] desenvolveram um *Data Diode* baseado em um FPGA (*Field Programmable Gate Array*). A ideia foi criar um dispositivo único, onde a direção do tráfego entre as interfaces de rede é provida pelo FPGA, permitindo a transferência de dados apenas em uma direção. Os autores montaram uma prova de conceito utilizando um computador com processador Core i7, 16Gb de Ram, disco SSD de 1Tb e interfaces de 1Gbps, bem como um algoritmo para gerenciamento de dados e alteraram o protocolo UDP (registro de sequência de pacotes, campos de informação e campos de confirmação) para assegurar a transferência das informações.

Heo et al. [Heo et al. 2016] propuseram um gateway unidirecional, chamado UNIWAY, capaz de fornecer mecanismos (correção de erros, gerenciamento de sessão, número de sequência de pacotes, filtro de IP/porta e filtros de conteúdo) para garantir confiabilidade e segurança dos dados. O UNIWAY é integralmente baseada no projeto de um hardware para permitir comunicação em apenas uma única direção e que utiliza proxies TCP, UDP e de aplicação para permitir comunicação com sistemas legados. Em outro trabalho [Heo and Na 2016], Heo et al. implementaram sua proposta de UNIWAY em uma arquitetura Intel 82580EB, onde a comunicação do cliente com o Diodo é TCP, convertida para UDP e transferida unidirecionalmente para a segunda unidade, que entrega os dados ainda via UDP a um destino final.

Embora os trabalhos apresentados lidem com o conceito de *Data Diodes*, propondo e implementando novas soluções, o grande problema encontrado na comunicação unidirecional consiste na impossibilidade de retransmissão oferecida tradicionalmente pelo protocolo TCP e a não utilização dos protocolos tradicionais de transmissão de da-

dos, impactando diretamente na integridade dos dados transmitidos entre segmentos e na arquitetura de comunicação proposta pelos fabricantes.

Este artigo apresenta justamente um *Data Diode* que resolve os problemas encontrados nos trabalhos propostos até aqui. A falta de confiabilidade entre os dispositivos envolvidos na troca de transmissões, a ausência de garantias de integridade dos ativos, os problemas de confidencialidade dos dados que partem da rede sensível e a interoperabilidade com protocolos legados, são exemplos de problemas encontrados nos modelos propostos pelos artigos lidos e resolvidos pelo dispositivo apresentado neste trabalho.

5. Proposta

O *Data Diode* proposto é composto por quatro (04) componentes. As **Interfaces de Conexão** fazem a ligação da origem e destino com o *Data Diode*, permitindo a entrada de dados por parte da origem e a entrega de dados para o destino. Também são responsáveis pela interação TCP/UDP com os dispositivos de origem e destino da comunicação. A **Unidade de Processamento** coordena todo processo de escrita/leitura na região da Unidade de Armazenamento destinada à guarda dos dados. A **Unidade de Armazenamento** guarda todos os dados trocados entre origem e destino. O último componente é o **TPM**, responsável por implementar segurança no dispositivo através de funções de verificação de integridade, atestação remota e autenticação de dispositivos. É importante destacar que os dispositivos de origem e destino dos dados também possuem como requisito estrutural para seu funcionamento uma unidade TPM para atender requisitos de atestação e autenticação no dispositivo principal.

O diferencial da proposta reside no fato de que antes da transferência de dados entre origem e destino, as garantias de integridade dos dados, confiabilidade entre os dispositivos e autenticação precisam ser estabelecidas. Para isso, três (03) etapas ocorrem envolvendo o *Data Diode* e os dispositivos conectados a ele (origem e destino). A primeira etapa é o *Measured Boot*, que mede os valores de inicialização do dispositivo, para comparar esses valores adquiridos com os valores armazenados da última inicialização considerada segura. Ela ocorre em todos os dispositivos envolvidos (Figura 2) durante o processo de inicialização do sistema.

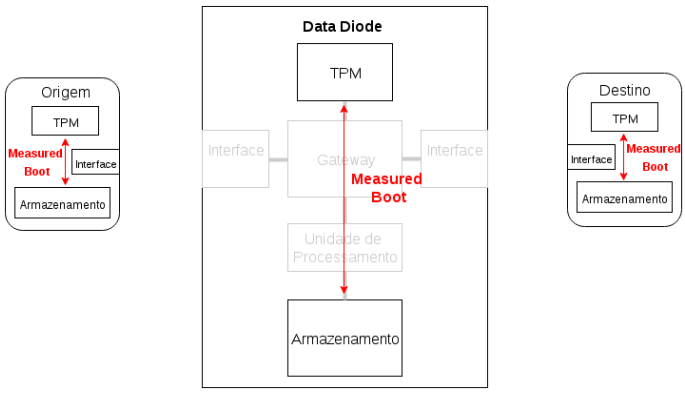

Figura 2. *Measured Boot*

A cada reinicialização do sistema, o *Measured Boot* é executado e o funcionamento dos dispositivos depende diretamente da integridade do processo de inicialização.

No caso da mensuração apresentar um valor insatisfatório por conta de alterações indevidas na inicialização ou arquivos do sistema, o dispositivo em questão não poderá avançar para a próxima etapa, ficando assim incomunicável, pois as chaves necessárias para o *Network Attestation* estarão seladas, impossibilitando seu acesso.

A etapa seguinte (segunda) é o processo *Network Attestation*, onde os dispositivos atestam remotamente a sua integridade, oferecendo uma garantia baseada em hardware e se autenticam mutuamente (Figura 3). A atestação ocorre entre o *Data Diode* e os

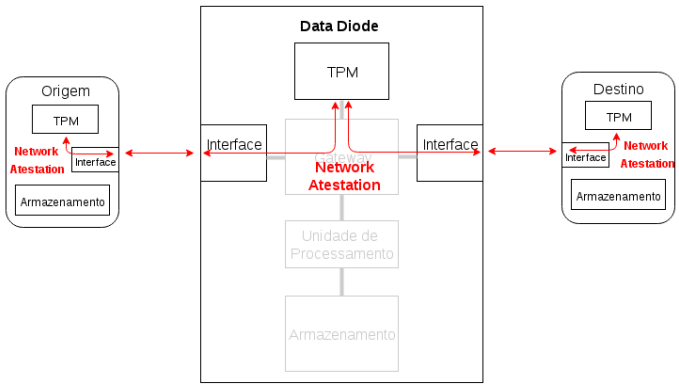

Figura 3. *Network Attestation*

dispositivos de origem e destino dos dados separadamente, ou seja, são dois processos que ocorrem simultaneamente. Não há atestação direta entre origem e destino, porém o *Data Diode* é o servidor de autenticação e realiza o processo nas duas direções, assim garantindo a integridade de todos os dispositivos envolvidos.

A última etapa é a transferência de dados. O dispositivo de origem transfere dados através de protocolo pré-determinado com o *Data Diode*, que realiza todas as interações como sendo o destino final da conexão através do Intermediário de Conexão. No caso de protocolos baseados em TCP, todo estabelecimento e negociação dos parâmetros de conexão é realizado entre origem e Intermediário de Conexão. A partir do recebimento de dados, o *Data Diode* armazena essas informações, descartando informações desnecessárias e pertinentes ao estabelecimento de conexões anteriores. Em seguida são enviadas pelo próximo Intermediário de Conexão, que estabelece conexão e define os parâmetros de comunicação com o destino (no caso de conexões TCP) e envia os dados recebidos pela conexão de entrada anteriormente.

No tange a implementação, graças a separação dos elementos funcionais, o *Data Diode* proposto pode ser implementado de diferentes formas sem perder suas principais características e apresentando resultados satisfatórios na comunicação unidirecional. Devido a restrições de espaço, elas não serão explicadas aqui, mas todos os possíveis modelos pensados para implementação estão descritos em https://tede.ufam.edu.br/handle/tede/6209.

5.1. Modelo de Ameaças

O modelo de ameaças descrito aqui é o mais ajustado para o protótipo implementado, considerando os trabalhos relacionados e o cenário proposto.

Em primeiro lugar, é assumido um modelo de sincronia parcial, ou seja, o sistema pode se comportar de forma assíncrona por algum tempo, até que se torne síncrono, com limites de tempo de processamento e comunicação. Este é um requisito essencial para garantir o término do processo de inicialização confiável, um alicerce fundamental do *Data Diode* proposto.

Em relação às ameaças, assume-se que as políticas de acesso físico da rede crítica não permitem que um atacante seja capaz de ter o controle de qualquer elemento. Porém, os elementos da rede externa não possuem garantias de acesso, sendo os mesmos acessíveis através da Internet e, portanto, suscetíveis a ataques. Também considera-se um cenário de ameaça onde o atacante tem por objetivo: (i) o controle dos canais de comunicação entre origem e destino e o acesso à rede crítica é feito a partir da Internet; (ii) a obtenção das informações trafegadas entre os elementos de rede presentes. Entre as ameaças possíveis nesse ambiente estão: *Man-in-the-middle*, Ataques de Replay e Força Bruta.

6. Prototipação e Decisões de Implementação

Como prova de conceito, o *Data Diode* proposto foi prototipado, utilizando o modelo de implementação par de mini computadores de placa única. O hardware empregado foi um Raspberry Pi 1 modelo B+, que possui uma interface ethernet, quatro interfaces USB 2.0, uma interface HDMI, saída de áudio, uma entrada para cartão Micro SD - funciona como unidade de armazenamento principal do dispositivo - e uma interface GPIO de 40 pinos para a comunicação serial. O sistema operacional utilizado foi o Raspbian, uma versão baseada no sistema Debian GNU/Linux Jessie, com kernel 4.9 e lançado em 5 de julho de 2017.

As funcionalidades do chip TPM foram emuladas via software, tendo sido utilizado o *TPM-Emulator* versão 0.7 que emula a especificação 1.2 do TPM (representando a unidade TPM apresentada na proposta). O software *TrouSerS* foi utilizado em conjunto com o *TPM-Emulator* com o objetivo de acessar as funções disponíveis no TPM emulado.

A Figura 4 ilustra o protótipo do *Data Diode* usando duas Raspberry Pi, interligadas entre si através da interface serial UART presente nos pinos GPIO de ambos.

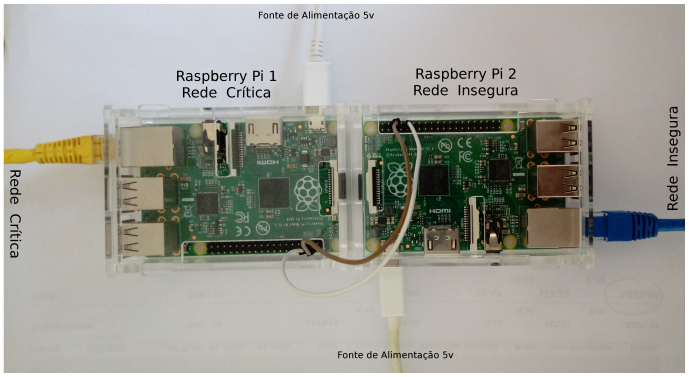

Figura 4. Interligação Física do *Data Diode*

Para cumprir o requisito de prover uma **Comunicação Unidirecional**, foi utilizada a conexão do pino TX (transmissor) do primeiro dispositivo diretamente no pino RX

(receptor) do segundo dispositivo, o que restringe o tráfego de informações no sentido contrário ao estabelecido. Na interface GPIO do Raspberry Pi modelo B+, os pinos 8 e 10 são os responsáveis pelo TX e RX da interface UART respectivamente.Para interligação segura, os dois dispositivos devem estar conectados também em uma das interfaces GND (*Graduated Neutral Density*) que funcionam como terra. O pino 6 é um GND e, por sua proximidade com a interface UART, pode ser utilizado nessa interligação. Desta forma, no protótipo foram interligados os pinos 6 dos dois Raspberry Pi B+ e o pino 8 do dispositivo conectado à rede crítica ao pino 10 do dispositivo conectado à rede insegura.

No que diz respeito à **Comunicação** do *Data Diode* com as redes crítica e insegura (origem e destino), a interface serial no sistema Raspbian foi implementada com o nome $ppp0$ e configurada com IPv4 na faixa 10.1.1.0/30, sendo 10.1.1.1/30 o transmissor e 10.1.1.2/30 o receptor, conforme Figura 5. Considerando que a interface UART funciona de forma assíncrona, em nenhum momento se faz necessária a interligação em sentido contrário para estabelecimento de conexão e sincronismo entre os dispositivos.

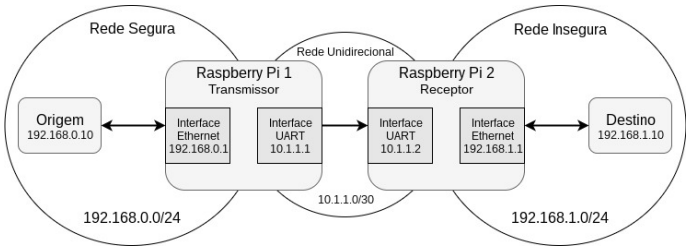

Figura 5. Esquema de Rede

O computador transmissor de dados, localizado na rede crítica, foi interligado ao primeiro Raspberry em sua interface ethernet RJ45, em modo bidirecional e velocidade máxima de 100 Mbps. Da mesma forma o computador receptor, localizado na rede insegura, foi interligado ao segundo Raspberry. Essas conexões com os dispositivos presentes nas redes são representadas pelo componente **Interfaces de Conexão**. As redes crítica e insegura possuem faixas de rede IP (versão 4) distintas, sendo 192.168.0.0/24 para a rede crítica e 192.168.1.0/24 para a rede insegura. Estas redes não possuem nenhum ponto de integração, nem rotas que as interliguem. O único ponto de integração entre as redes é o gateway implementado pelo uso dos dois Raspberry, interligados via interface UART fisicamente unidirecional.

Ainda sobre a **Comunicação**, o protótipo foi concebido para um cenário onde: (i) existe um único transmissor presente na rede crítica que possui dados a serem transmitidos via conexão TCP; (ii) existe um único destinatário presente na rede insegura; (iii) o único meio de comunicação é um gateway unidirecional; (iv) a comunicação deve ser corretamente trafegada via protocolo UDP entre os dispositivos que compõem o Diodo e posteriormente novamente convertida para protocolo TCP na sua entrega ao destinatário final. Desta forma, tal processo deve empregar uma solução de software capaz de organizar, sequenciar e bufferizar todo o fluxo de dados, minimizando o risco de perdas de dados e inconsistência nas informações transmitidas. Essa é a função das **Interfaces de Comunicação**.

Para alcançar esses requisitos de transmissão de dados foram definidas quatro (04) funções, sendo: (1) Receber a comunicação TCP que parte da origem na rede crítica; (2)

Transmitir usando UDP unidirecional até segundo dispositivo; (3) Receber a comunicação unidirecional transmitida pelo primeiro dispositivo em UDP e armazenar os dados colhidos; (4) Converter a comunicação de UDP para TCP, transmitir ao destinatário final e criptografar os dados já transmitidos.

Por fim, em termos de implementação, cada um dos dois dispositivos Raspberry Pi responsáveis por desempenhar a função de *Data Diode* recebeu dois scripts: um com funções de servidor e outro trabalhando como um cliente de comunicação.

6.1. Implementações de Segurança

Esta subseção descreve os requisitos e as implementações de segurança implementadas no protótipo para garantir a confidencialidade dos dados e a confiabilidade dos dispositivos envolvidos, através das funcionalidades providas pelo TPM e dos recursos providos pelo protocolo SSH.

Provisionamento

O provisionamento consiste em uma fase pré-atestação, onde os valores necessários para comprovar a integridade e autenticidade dos dispositivos são trocados para que os mesmos possam ser posteriormente validados sempre que necessário. Esse processo consiste na geração de um identificador único, uma chave de atestação e um hash baseado nos valores do PCR do chip TPM em uma inicialização considerada segura. Para tanto, inicialmente, os dispositivos passam pelo processo de provisionamento visando estabelecer valores de uma inicialização segura para que posteriormente possam ter seus estados testados. Para isso, cada dispositivo deve gerar um UUID (*Universal Unique Identifier*), o identificador único do dispositivo para o sistema remoto que fará a sua atestação. Para o UUID do dispositivo também é gerado uma AIK (*Attestation Identity Key*), chave de atestação de identidade, que será utilizada posteriormente para assinar os dados de atestação gerados no dispositivo, garantindo ao dispositivo remoto sua autenticidade. Tanto o UUID quanto a AIK são armazenados no TPM. Esses valores serão utilizados tanto na geração do hash que identifica o estado dos valores contidos no PCR quanto na atestação remota.

Com as chaves importadas, um *hash* dos valores contidos no PCR será gerado, pois havendo uma possível mudança nos valores da PCR, este indicará que o processo de inicialização foi modificado. Os PCR's utilizados nesse processos também serão os que devem ser avaliados no processo de medição de boot e atestação remota. Ao fim desse processo, os arquivos contendo o UUID do dispositivo e o hash dos PCR's devem ser enviados ao dispositivo remoto que fará a atestação. Nesse caso, sendo trocados entre o Diodo e os dispositivos de origem e destino. Esses dois arquivos devem ser armazenados, pois serão utilizados posteriormente em na atestação remota.

Atestação Remota

Após a inicialização dos sistemas e o processo de medição de *boot* realizado automaticamente pelo TPM, os dispositivos já estão prontos para iniciar o processo de atestação remota. Cada dispositivo deverá gerar um arquivo, denominado *nonce*, para cada processo de atestação, tendo o tamanho exato de 20 bytes, contendo uma sequência numérica

aleatória. Este arquivo aleatório garante que uma nova medição tenha sido gerada no dispositivo remoto, impedindo que uma medição anterior seja utilizada nesse momento, como em um Ataque de Replay. Esse arquivo será transportado ao sistema remoto via SFTP, provido pelo protocolo SSH, e servirá para a geração do arquivo que contém os dados do estado atual dos valores PCR medidos.

Com o arquivo de *nonce* gerado pelo dispositivo remoto, armazenado localmente, o TPM pode gerar um arquivo de medição (*quote*). O *quote* é gerado com base nos valores PCR atuais e no UUID do dispositivo verificado, combinado com o arquivo de dados aleatórios (*nonce*) gerado no dispositivo remoto, assinado digitalmente pela chave AIK que garante sua autenticidade e confiabilidade. O arquivo gerado nesse processo deve ser enviado para o dispositivo remoto para que haja a atestação remota. Os dois dispositivos devem realizar esse processo com o objetivo da realização de uma atestação mútua.

De posse do valor de medição do dispositivo remoto, é realizada a verificação do arquivo de medição utilizando o UUID do dispositivo remoto, seu hash dos valores PCR gerados na fase de provisionamento, o *nonce* gerado localmente no início do processo de atestação e o novo valor enviado pelo dispositivo. Caso os valores estejam íntegros, o comando não reportará nenhum erro ou saída, o que indicará que a atestação remota ocorreu com sucesso.

Criptografia de dados

Na rede insegura, o *Data Diode* estabelece um Túnel criptografado através do protocolo SSH com o dispositivo que receberá os dados da rede crítica, para garantir a confidencialidade dos dados recebidos. Esse túnel é estabelecido à partir de uma chave de criptografia trocada previamente entre os dispositivos. Somente a partir do estabelecimento do túnel SSH é que o *Data Diode* começa a receber os dados e enviá-los ao cliente. Com a aplicação dessa medida, um possível ataque de *Man in the Middle* não surtia o efeito esperado, tendo em vista que toda a comunicação entre o diodo e o servidor é criptografada, impossibilitando a quebra da confidencialidade dos dados.

Após o envio dos dados ao destinatário, os dois dispositivos Raspberry realizam a criptografia dos dados enviados utilizando o TPM, deixando-os cifrados em um diretório de armazenamento. Todo conjunto de dados transmitido ao próximo dispositivo é imediatamente criptografado para evitar uma posterior leitura dessas informações transportadas.

7. Testes e Resultados

O protótipo foi submetido a testes em ambiente simulado, com o objetivo de comprovar seu funcionamento e garantir que as ameaças descritas não afetam o cenário em questão. Para os testes, assumiu-se que a rede crítica (sem comunicação com a internet) é segura, sendo o principal alvo de testes os dispositivos conectados diretamente na rede insegura, considerando ainda que ameaças vindas da internet afetariam apenas essa rede. Em todos os testes de transmissão foi aplicado um cenário utilizando como base uma comunicação *syslog*[2], funcionando via protocolo TCP, em uma origem na rede segura e chegando a um

[2]É um protocolo de comunicação de log's via rede TCP/IP

destino na rede insegura.

7.1. Throughput e Perda

Para testar o *throughput* da solução foi utilizado um arquivo com uma carga real de dados a serem transmitidos. Foram realizadas três (03) medições com arquivos de aproximadamente 3MB, 5MB, 10MB e 20MB. Inicialmente a transmissão ocorreu na velocidade média de 658 Kbps, ocasionando uma taxa de perda de dados em cerca de 80%, devida à falta de mecanismos de controle de conexão e reconhecimento de entrega de pacotes, ausentes no protocolo UDP. Para resolver esse problema, foi adicionado um controle na velocidade de transmissão para evitar perdas de dados, chegando-se a uma taxa de transmissão média de 530 kbps, diminuindo assim a perda de dados durante a transmissão.

Os resultados de perda de dados na transmissão são apresentados na Tabela 1. Percebe-se que a perda de informação, se tratando de uma comunicação UDP unidirecional, é mínima, e pode ser considerada satisfatória. A comparação desses valores com outros trabalhos relacionados se torna impossível, tendo em vista que os mesmos não apresentam resultados totalmente conclusivos, nem os parâmetros utilizados para os testes.

Tabela 1. Medições de Throughput

Medição	Tamanho do Arquivo	Perda	% de Perda
1	3.124.014 bytes	5.963 bytes	0,19%
2	3.124.014 bytes	3.075 bytes	0,10%
3	3.124.014 bytes	4.065 bytes	0,13%
4	5.145.565 bytes	8.263 bytes	0,16%
5	5.145.565 bytes	5.424 bytes	0,11%
6	5.145.565 bytes	7.334 bytes	0,14%
7	10.398.207 bytes	25.124 bytes	0,24%
8	10.398.207 bytes	19.696 bytes	0,19%
9	10.398.207 bytes	24.003 bytes	0,23%
10	20.796.414 bytes	66.105 bytes	0,32%
11	20.796.414 bytes	46.827 bytes	0,23%
12	20.796.414 bytes	50.309 bytes	0,24%

7.2. Negação de Serviço

Para esse teste foi adicionado à rede um dispositivo com o endereço IP 192.168.1.51, que através do uso do software hping iniciou um ataque de negação de serviço ao diodo (IP 192.168.1.1). O software hping teve seu ataque direcionado à porta 22 do Diodo, gerando, de acordo com as suas configurações, 10 mil pacotes SSH por segundo com o tamanho de 120 bytes, utilizando a flag TCP SYN. Esse ataque se mostrou bastante efetivo, tendo em vista que mesmo bloqueado no firewall do Diodo, o atacante conseguiu gerar um *overhead* no Diodo, impedindo assim sua comunicação com o servidor remoto. O processo de atestação, que demora menos de 10 segundos em condições normais, ficou indisponível gerando uma mensagem de conexão SSH perdida nos dois sentidos. Uma efetiva proteção contra esse tipo de ataque consistiria na inteligência do dispositivo de comutação, que interliga os hosts na rede insegura, para identificar o ataque e bloquear a porta do atacante, parando imediatamente o ataque.

7.3. Atestação Falsa

Em uma atestação falsa presume-se que um dispositivo não realizou o processo de provisionamento no Diodo e mesmo assim tenta se atestar como um dispositivo confiável. Vale à pena ressaltar que nesse protótipo todos os acessos via SSH são baseados em chaves RSA e na liberação do endereço IP desse dispositivo no firewall do Diodo. Portanto, a menos que haja a troca de chaves e a liberação desse endereço no firewall, o dispositivo ficaria impedido de realizar a atestação. Em um quadro assim, a garantia da não atestação desse dispositivo residiria no fato do Diodo não possuir seu UUID nem seu hash PCR para posterior análise. Portanto, mesmo que fosse gerado um *nonce* para a atestação através de um arquivo de medição (*quote*), não seria possível realizar a função de verificação, pois não haveriam os arquivos necessários para esse dispositivo.

Nos testes utilizou-se um terceiro dispositivo na rede insegura, contendo um chip TPM, scripts de atestação e simulando ter o mesmo endereço IP do dispositivo real. Através da leitura dos log's, constatou-se que o processo de atestação falhou impedindo que o mesmo chegasse ao status de dispositivo confiável pelo diodo. Um dispositivo novo somente será capaz de se atestar, caso seu UUID e seu hash PCR fossem copiados para o dispositivo responsável pela atestação.

7.4. Ataque de *Replay*

Considerando que o servidor da rede remota possa ter sido comprometido por um *malware*, que se instalou em seu processo de inicialização, o TPM terá seus valores de PCR comprometidos. A alteração das chaves durante o processo de medição de *boot*, inviabilizará uma nova atestação, já que o processo de verificação do *quote* deste, realizado remotamente no Diodo, falhará como na Atestação Falsa. Porém, através da recuperação de um *quote* gerado em uma atestação anterior, um atacante poderia tentar uma nova validação deste dispositivo.

Ao iniciar o processo de atestação e enviar o *quote* de uma atestação anterior, o Diodo reporta em seus log's falha na atestação.Essa falha se dá pela impossibilidade de utilizar o antigo *nonce* no dispositivo remoto, tendo em vista que o dispositivo que requer a atestação gera o arquivo aleatoriamente para compor a checagem do *quote* local. Sendo uma nova atestação, um novo valor aleatório será gerado, impossibilitando que uma atestação local antiga seja novamente utilizada em um ataque de *replay*. A chance de um valor de *nonce* se repetir em um processo de atestação posterior é de 1 em 10$\hat{1}$9.

7.5. Força Bruta

Considerando os dados apresentados quanto às falhas de atestação falsa e ataques de replay, a única tentativa ainda válida para quebra da confidencialidade dos dados seria utilizar um ataque de força bruta. Considerando que toda a comunicação, seja no processo de atestação ou na própria troca dos dados pela rede, está baseada no protocolo SSH, esse protocolo poderia ser o principal alvo de um ataque desse tipo. Para impedir um ataque de força bruta em senhas de autenticação via protocolo SSH, um cenário de teste foi elaborado sobre a premissa de não aceitar autenticação baseada em senhas, sendo a única forma de comunicação a troca de chaves RSA antes mesmo da atestação entre os dispositivos. A única forma para a troca dessas chaves se dá de maneira física nos dispositivos envolvidos, coletando a sua chave pública em um dispositivo de armazenamento móvel e inserindo-a também por acesso físico no dispositivo destinatário da comunicação.

Um ataque de força bruta baseado em senhas, mesmo possuindo a senha correta em sua *password list*, seria completamente inofensivo contra este cenário. Uma tentativa válida seria um ataque de força bruta sobre as chaves RSA, porém o comprimento de chave utilizado neste protótipo fora o padrão do protocolo, 2048 bit's. Um ataque de força bruta em uma chave com esse comprimento seria completamente ineficiente em um hardware comum, tendo em vista que um ataque de força bruta em chaves RSA utilizaria um algoritmo a ser executado em tempo exponencial e o tamanho das chaves impossibilitaria uma resposta válida em tempo hábil. Ainda que houvesse a possibilidade de quebra dessas chaves, um algoritmo de troca de chaves poderia ser facilmente implementado, dificultando ainda mais ataques desse tipo em um cenário idêntico.

8. Dificuldades Encontradas

Esta seção comenta sobre as dificuldades encontradas na elaboração deste trabalho.

A primeira delas foi a opção da emulação de um chip TPM via software, o que dificultou a execução dos testes. Todos os softwares que implementam as funcionalidades TPM sobre o sistema operacional (como o *TrouSerS* por exemplo) possuem dificuldades na execução de algumas funções com o TPM-Emulator ou outro emulador TPM. Pode-se perceber que o emulador TPM não consegue executar todas as funções de um chip TPM físico. A partir dessa dificuldade, os processos de atestação remota tiveram que ser refeitos e executados sem a utilização dos comandos presentes nesses softwares, o que interferiu diretamente no tempo de execução do trabalho.

Outro ponto é a baixa capacidade de processamento do Raspberry Pi 1 B+ durante a execução do experimento. O equipamento utilizado na rede insegura, que possui a maior quantidade de tarefas executadas (criptografia de arquivos enviados, estabelecimento da VPN além da transmissão dos dados) apresentou um uso constante de 100 % de sua capacidade de processamento. Esse fato inviabilizou uma melhor resposta do protótipo quanto à verificação de *throughput* do cenário, além de amplificar o dano do ataque de negação de serviço.

9. Conclusões

De forma a contribuir para o desenvolvimento de soluções contra ameaças à redes industriais críticas, este trabalho apresentou o primeiro modelo funcional para o projeto e desenvolvimento de *Data Diodes* seguros. Um modelo funcional foi apresentado e um protótipo gerado como prova de conceito, deixando claro sua capacidade de atender os requisitos de confidencialidade e integridade. Além disso, vale ressaltar que o protótipo apresentado pode ser utilizado como base de desenvolvimento de soluções de comunicação segura em quaisquer redes industriais críticas onde seja necessário enviar dados em uma única direção.

Com base nesse trabalho, novas soluções de transferência de dados em aplicações TCP, atuais ou legadas, podem ser implementadas utilizando um meio de comunicação unidirecional para o tráfego seguro entre origem e destino. Ficou demonstrado que sistemas de comunicação de dados não necessariamente precisam ser reescritos para se adaptar à comunicação unidirecional, mas o gateway de comunicação pode ser adaptado, a fim de proporcionar uma comunicação segura e o isolamento físico da rede mais crítica.

Através do protótipo foi possível provar que as grandes ameaças ao cenário de aplicação proposto não conseguiram ferir a confidencialidade e a integridade dos dados trafegados, sendo efetiva apenas a negação de serviço como base para a quebra da disponibilidade da solução. O protótipo conseguiu implementar as funções de atestação de hardware disponíveis em um chip TPM e a segurança dos dados trafegados através de criptografia.

Referências

Arkhangelskii, V., Epishkina, A., Kalmytov, V., and Kogos, K. (2016). Secure One-Way Data Transfer. pages 392–395.

Heo, Y., Kim, B., Kang, D., and Na, J. (2016). A Design of Unidirectional Security Gateway for Enforcement Reliability and Security of Transmission Data in Industrial Control Systems. pages 310–313.

Heo, Y. and Na, J. (2016). Development of unidirectional security gateway appliance using intel 82580EB NIC interface. *2016 International Conference on Information and Communication Technology Convergence, ICTC 2016*, pages 1194–1196.

Jeon, B.-s. and Na, J.-c. (2016). A Study of Cyber Security Policy in Industrial Control System using Data Diodes. pages 314–317.

Moreira, N., Molina, E., Lázaro, J., Jacob, E., and Astarloa, A. (2016). Cyber-security in substation automation systems. *Renewable and Sustainable Energy Reviews*, 54:1552–1562.

Oh, Y.-c., Han, M.-r., Shin, Y., and Kim, J.-b. (2015). A Study on the Communication Agent Model for One-way Data Transfer System. 9(10):161–168.

Osborn, J. D. and Challener, D. C. (2013). Trusted platform Module evolution. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 32(2):536–543.

Winter, J. and Dietrich, K. (2013). A hijacker's guide to communication interfaces of the trusted platform module. *Computers and Mathematics with Applications*, 65(5):748–761.