FVIS: Uma Ferramenta de Verificação de Integridade de Software

Cristiano G. de Castro*, Flávio P. Moraes*, Davidson R. Boccardo*, Raphael C. S. Machado*, Paulo C. Brandão*, Luiz F. R. C. Carmo*

*Instituto Nacional de Metrologia, Qualidade e Tecnologia
{cgcastro,fpmoraes,drboccardo,rcmachado,pcbrandao,lfrust}@inmetro.gov.br

***Resumo* - O advento de dispositivos com software embarcado para aplicações em sistemas de medição demanda maior controle por parte da Autoridade Metrológica a fim de garantir a integridade destes sistemas. O problema de controle de integridade destes dispositivos é um desafio, pois muitas vezes tais medidores estão expostos a operar em ambiente hostil e de forma autônoma, sendo sujeitos à captura, engenharia reversa e manipulação. No presente trabalho, é proposta uma ferramenta de verificação de integridade arquitetada para ser independente de protocolos ou tecnologia empregada nos variados sistemas de medição e também para manter a confidencialidade do código sob análise.**

Palavras-chave — software, segurança da informação, sistemas de medição.

I. Introdução

Cada vez mais, sistemas de medição [1] fazem uso de processamento computacional via eletrônica digital e software embarcado. Denominamos inteligentes aos medidores que, por serem dotados de módulos de software e processamento de dados, apresentam um comportamento potencialmente complexo. Um exemplo de sistemas desses tipos são os medidores inteligentes de energia elétrica, os quais surgiram primeiramente da necessidade de leitura remota de medição, porém hoje possuem um ampliado leque de funcionalidades.

A validação de medidores inteligentes pela Autoridade Metrológica envolve um sofisticado processo que culmina com a caracterização do código que deverá estar em execução nos medidores em campo. Um dos requisitos mais importantes a respeito de medidores inteligentes de energia é que eles devem possuir mecanismos que permitam verificar se, de fato, o software que está em execução é exatamente aquele previamente aprovado pela Autoridade Metrológica. Tal requisito é denominado controle de integridade e garante que não haja quaisquer modificações não-autorizadas, quer sejam acidentais, o que pode comprometer a acurácia dos resultados de medição, quer sejam intencionais, poderia acarretar em fraudes na medição.

Apesar da importância do controle de software, o regulador optou por não definir padrões rígidos para a verificação e integridade em campo, o que levou a um cenário onde, atualmente, coexistem em campo softwares de diversos fabricantes, em várias versões e comunicando-se através dos mais diversos protocolos. No presente trabalho, apresentamos uma ferramenta prática de verificação de integridade capaz de comunicar-se com vários sistemas de medição tratando a complexidade inerente aos variados protocolos de comunicação e tecnologia utilizados nos diferentes sistemas. Ao mesmo tempo em que a solução cumpre os desafios de escalabilidade e gerenciamento, a mesma baseia-se em técnicas de verificação de integridade por meio de introspecção [2], mantendo o código protegido de acesso público.

O artigo está organizado da seguinte forma. Na Seção II, apresentamos as técnicas de identificação unívoca de software utilizadas para efetuar o verificação de integridade. Na Seção III apresentamos uma ferramenta de verificação de integridade e mostramos como ela trata os diversos protocolos de comunicação e gerencia diferentes versões software. A Seção IV apresenta nossas conclusões a respeito da importância de uma harmonização nos protocolos e técnicas de verificação de integridade de software, e descreve nossos atuais esforços e trabalhos futuros.

II. TÉCNICAS DE VERIFICAÇÃO DE INTEGRIDADE

A presente seção apresenta duas soluções frequentemente utilizadas para o verificação de integridade de software de medidores de energia elétrica. Tais soluções são baseadas em um mecanismo de desafio-resposta e fundamentadas no conceito de introspecção, através do qual o software em análise é responsável pela leitura do o seu próprio código armazenado em memória. Ambas as técnicas apresentadas podem ser divididas em três etapas: "configuração" (ou "preparação"), na qual a técnica de verificação bem como os parâmetros aleatórios utilizados nos desafios são definidos, e também os resultados esperados dos respectivos desafios são coletados de uma cópia reconhecidamente íntegra do software; "consulta", na qual os desafios preparados na primeira etapa são apresentados ao software sob análise; e "verificação", na qual se compara os valores obtidos na etapa de "consulta" aos resultados esperados obtidos na "preparação".

A.Semente aleatória

O chamado método de semente aleatória consiste no envio de um dado (a semente) que cumprirá o papel de chave criptográfica para a execução de um algoritmo denominado *Message Authentication Code* [3], o qual será executado sobre o próprio código do software sob avaliação (para maiores detalhes sobre o método, sugerimos consultar [4]). A Fig. apresenta um diagrama das etapas de verificação de integridade segundo a técnica de semente aleatória. Na etapa "configuração" é criado um conjunto suficiente de pares do tipo chave-valor, onde a chave corresponde a uma semente gerada aleatoriamente e o valor é calculado como o resultado de uma função de extrato (o *message authentication code*), aplicada a imagem de memória do software e tendo como chave a respectiva semente. Na etapa de "consulta", um subconjunto de sementes quaisquer retirado do conjunto construído na etapa anterior é enviado (desafio) ao módulo sob análise, o qual deve comunicar (resposta) um valor de extrato do seu código, de modo idêntico ao utilizado para construção do conjunto na etapa 1. Finalmente, no terceiro passo, há a comparação da resposta recebida do módulo com o valor previamente armazenado no respectivo par ordenado. A integridade é confirmada se os valores forem iguais.

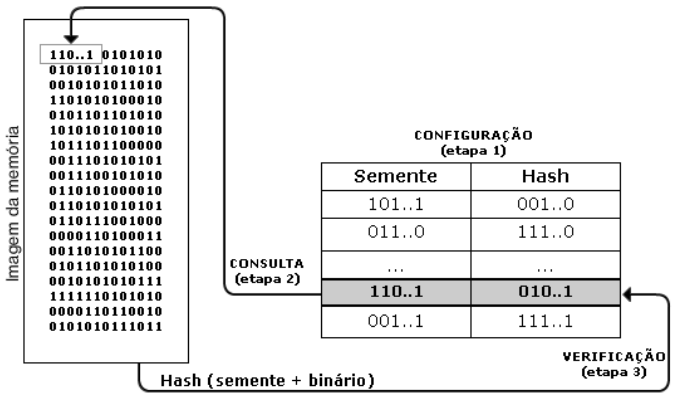

Fig. 1. Técnica da semente aleatória.

B. Intervalos de memória aleatórios

Há situações em que as plataformas nas quais os softwares executados permitem tão somente o cálculo de extrato de intervalos de memória com comprimento máximo definido, e a aplicação da técnica de sementes aleatórias não é possível. Evitar a leitura da imagem completa da memória é uma restrição dessas plataformas, pois se trata de um mecanismo de proteção da confidencialidade de código. A técnica de intervalos de memória aleatórios contorna esse tipo de situação para efetuar a verificação de integridade. A Fig. 2 apresenta os passos da técnica. Na etapa "configuração", é gerado um ou mais conjuntos de pares intervalo-extrato, em que intervalo representa um par início-fim de posições de memória, e o extrato representa a aplicação de uma função de *hash* [3] nos valores de memória retirados do intervalo correspondente. Os intervalos gerados devem, em conjunto, abranger toda a área de memória utilizada para o armazenamento do programa. Na "consulta", cada intervalo de um conjunto de pares gerado anteriormente, é enviado ao módulo do sistema de medição (desafio), o qual deve comunicar (resposta) o extrato correspondente ao intervalo, de acordo com o mesmo procedimento adotado na "configuração". Por fim, na etapa de checagem os valores de extratos correspondentes às posições de memória são comparados. No presente caso, o parâmetro aleatório do desafio é, portanto, o intervalo de memória e não uma semente.

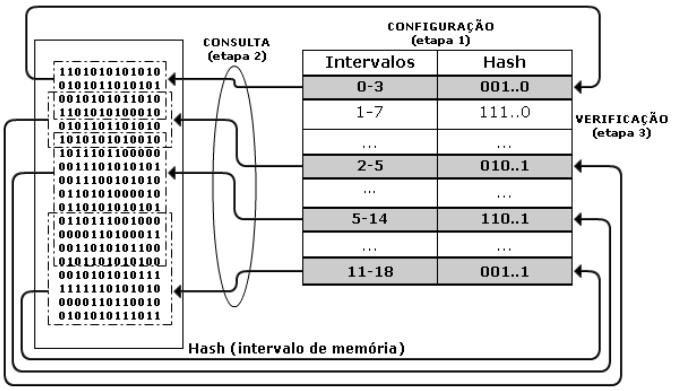

Fig. 2. Técnica de intervalos aleatórios.

III. A FERRAMENTA FVIS

Tendo definido as técnicas a serem utilizadas, a FVIS provê uma interface única para um técnico responsável por efetuar o verificação de integridade de instrumentos de medição controlados por software em campo. A FVIS objetiva

também gerenciar a coexistência de diferentes versões de software, cada uma contendo seu identificador único. Em resumo, deseja-se: ocultar, para um operador, a complexidade do estabelecimento de comunicação com os diferentes sistemas de medição; e, dessa forma, facilitar a disseminação e o treinamento da tarefa de verificação de integridade.

A FVIS é um software concebido de forma a prover compatibilidade com os instrumentos já em utilização no mercado, não tendo como premissa nenhuma normalização de protocolo utilizado para comunicação ou informação de versão. Na etapa inicial, optou-se por utilizar a disseminada interface RS-232 (serial) para a comunicação com o instrumento. A utilização da ferramenta é divida em duas fases, as quais refletem os passos de "configuração" e "consulta" das técnicas de verificação de integridade descritas na Seção II.

A. Processo de Configuração

O processo de configuração é dividido em cinco passos (ver Fig. 3): (1) descrição do protocolo de desafio-resposta e protocolo de requisição da versão de software, caso aplicável; (2) especificação de algoritmos aplicados pelos protocolos definidos no passo 1; (3) mapeamento da imagem de software na área de memória do módulo do instrumento; (4) especificação da técnica de verificação de integridade; (5) coleta dos extratos da cópia de software íntegro.

No primeiro passo, os campos que compõem as mensagens a serem comunicadas entre a FVIS e o sistema de medição devem ser informados. Esses campos são as peças básicas do protocolo de comunicação. A FVIS prevê diferentes tipos de preenchimento para esses campos, necessários para o estabelecimento da comunicação. São eles: de preenchimento predefinido (ex.: campos de cabeçalho), que são definidos manualmente durante a configuração; de preenchimento aleatório (ex.: semente); de preenchimento dinâmico, que são calculados em tempo de troca de mensagens de acordo com outros campos de uma mesma mensagem (ex. CRC, MAC); de preenchimento em campo, os quais devem ser configurados pelo operador momentos antes consulta (ex.: parâmetros específicos de cada sistema de medição, como endereço de hardware do módulo). Os algoritmos utilizados para o cálculo dos parâmetros dinâmicos devem ser também informados à ferramenta e são definidos no passo (2).

Os bytes correspondentes ao software embarcado no instrumento não ocupam necessariamente toda a área de memória disponível para sua gravação na plataforma. Surge daí a necessidade do passo seguinte da configuração (3), em que mapeia-se as áreas de memória onde o firmware será alocado no sistema de medição. Na etapa seguinte (4) especifica-se qual a técnica de verificação de integridade utilizada: semente aleatória ou faixas aleatórias.

Finalmente, no último passo da configuração (5), as definições de protocolo e de algoritmos definidos nas etapas anteriores são utilizadas pela ferramenta para o efetivo

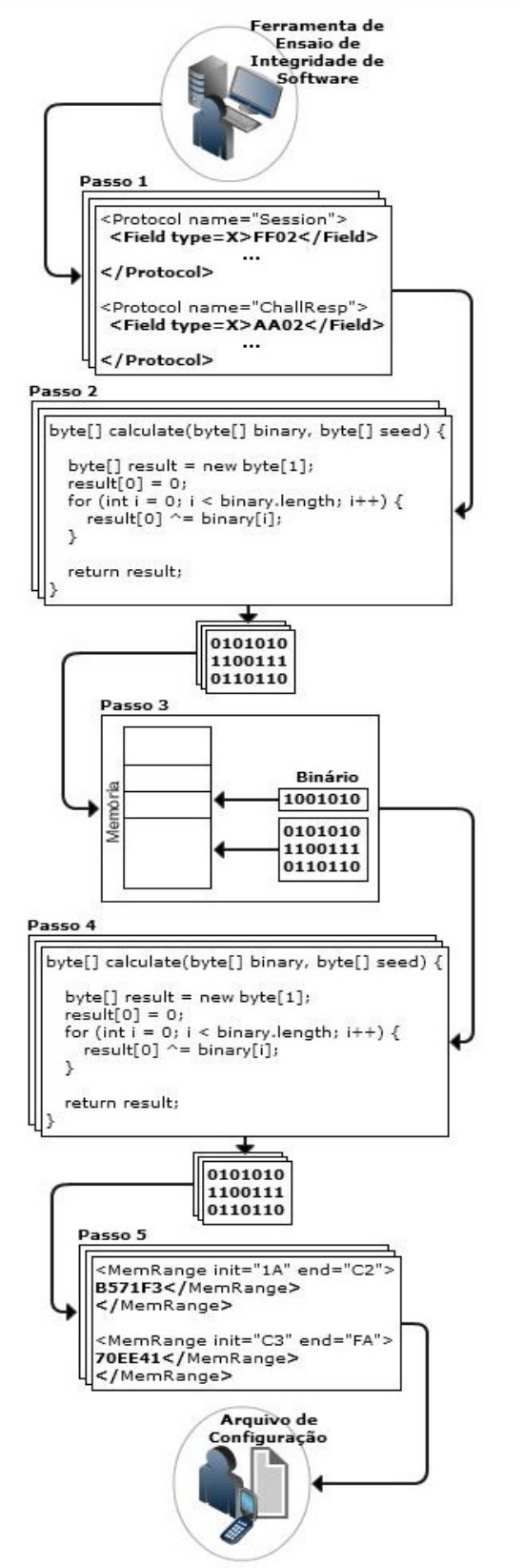

Fig. 1. Criação do arquivo de configuração.

estabelecimento de comunicação com um equipamento contendo o software confiável, e, em sequência, para a extração dos extratos de código utilizados para identificação única do software.

Os dados coletados para o estabelecimento de comunicação entre a FVIS e o instrumento de medição são organizados e armazenados em um arquivo de configuração. O formato utilizado para o armazenamento é o XML [5], um padrão utilizado comumente para intercâmbio de informações.

Importante notar que no arquivo gerado, não é armazenada informação aberta de código utilizado nos módulos do sistema de medição, e sim extratos criptográficos gerados pela aplicação de funções de *hash* em determinadas partes ou em todo o código. As propriedades de resistência à inversão (as propriedades das funções de *hash* são descritas em [3]) de tais funções proveem a confidencialidade necessária para que tais arquivos sejam amplamente distribuídos para verificação de integridade em campo sem, no entanto, comprometer os segredos industriais contidos nos software depositados para o processo de Apreciação Técnica de Modelo.

B. Processo de consulta

No processo de consulta e comparação é efetivamente executado a verificação de integridade. O operador deve recuperar os arquivos de configuração preparados previamente correspondentes aos softwares embarcados no sistema de medição que serão analisados. É recomendável que esses arquivos sejam armazenados em repositório de arquivos ou banco de dados de forma a organizar os arquivos por fabricante, modelo de instrumentos e versões do software de cada módulo do sistema de medição.

Os protocolos definidos no respectivo artigo de configuração são carregados na ferramenta de consulta para comunicação com o sistema sob análise. A ferramenta então extrai a identificação de cada software embarcado do sistema de medição nos mesmos moldes do aplicado no processo de configuração. Uma simples comparação dos resultados gerados no processo de configuração, que também são armazenados no arquivo de configuração, com os resultados obtidos no processo do software sob análise provê o resultado final da verificação de integridade.

Há duas formas utilizadas pelo FVIS de gerência automática do processo de verificação de integridade no caso de várias versões de software aprovado para um mesmo módulo sob análise. Caso o técnico de configuração tenha estabelecido um protocolo de questionamento da versão instalada, este é utilizado para recuperar o conjunto de desafios correspondentes àquela versão para executar o verificação de integridade. Caso não se tenha definido tal protocolo, é aplicado um método de busca exaustiva pela versão, isto é, os conjuntos de desafios configurados para cada versão de software aprovado são aplicados ao software sob análise, a busca é interrompida se a verificação for concluída com sucesso para algum dos conjuntos (integridade verificada) ou sem sucesso para todos os conjuntos (integridade não-verificada).

IV. CONCLUSÕES

Este artigo apresentou um desafio do atual processo de integridade de software com a perspectiva da inspeção periódica em mente, especialmente considerando o aumento da demanda para ATM de sistemas de medição com software embarcado. Esse aumento da demanda pode, em um futuro próximo, ocasionar um cenário não gerenciável para a verificação de integridade.

O trabalho apresentou uma ferramenta que visa contornar esse problema oferecendo uma interface única para a verificação de integridade de softwares embarcados em diferentes instrumentos de medição. Para poder tratar os diferentes aspectos arquiteturais, tecnologias e protocolos de comunicação presentes nos diferentes sistemas de medição, a ferramenta é configurável. Uma configuração gerada para um módulo de sistema de medição pode ser exportada para formato XML e transmitida para utilização em campo. Esses arquivos de configuração contêm informações necessárias para identificação de um software aprovado sem, no entanto, comprometer a confidencialidade do mesmo.

Como uma ferramenta configurável, a FVIS não necessita de que nenhuma restrição ou requisito de compatibilidade de atendimento obrigatório seja implementado pelo sistema de medição para a sua utilização; tal requisito poderia ser interpretado como uma intervenção indesejada de autoridades em requisitos de comunicação do sistema. A solução apresentada pode, portanto, ser aplicada inclusive aos sistemas já em operação. Algumas melhorias ainda devem ser implementadas, como, por exemplo, o tratamento de sistemas que necessitam do estabelecimento de sessão de comunicação antes da troca de mensagens de verificação de integridade.

REFERÊNCIAS

[1] Vocabulário Internacional de Metrologia: conceitos fundamentais e gerais de termos associados, 3rd ed. Duque de Caxias, RJ: INMETRO, 2012, p. 94.

[2] D. Spinellis, “Reflection as a mechanism for software integrity verification,” ACM Transactions on Information and System Security, vol. 3, no. 1, pp. 51–62, Feb. 2000.

[3] W. Stallings, Cryptography and Network Security: Principles and Practice. Prentice Hall, 2011, pp. 327–341.

[4] V. G. P. de SÁ, D. R. Boccardo, L. F. Rust, and R. C. S. Machado, “A tight bound for exhaustive key search attacks against Message Authentication Codes,” RAIRO - Theoretical Informatics and Applications.

[5] W3C, “Extensible Markup Language (XML) 1.0 (Fifth Edition),” 2008. [Online]. Disponível: http://www.w3.org/TR/xml/. [Acessado: 25-Mar-2013].