# Aspectos de Segurança da Informação em Redes de Medidores de energia Elétrica

L. F. R. C. Carmo[1], E. L. Madruga[2], R. C. S. Machado[3]

[1]DIMCI/Inmetro, Rio de Janeiro, Brasil, lfrust@inmetro.gov.br
[2] DIMCI/Inmetro, Rio de Janeiro, Brasil, elmadruga@inmetro.gov.br
[3] DIMCI/Inmetro, Rio de Janeiro, Brasil, rcmachado@inmetro.gov.br

**Resumo:** No presente trabalho, apresentamos um modelo para a organização de Redes de Medidores de Energia Elétrica e apresentamos abordagens para a avaliação de segurança de tais medidores. Nossa abordagem envolve três etapas: (1) avaliação da arquitetura, protocolos e algoritmos, (2) validação do software da rede de medidores, e (3) avaliação dos mecanismos de verificação de integridade do software. Enquanto as etapas (1) e (2) são tratadas com abordagens convencionais, propomos, para a etapa (3), técnicas inéditas de verificação de integridade, as quais são apresentadas em maiores detalhes.

## 1. INTRODUÇÃO

Com a evolução da computação e das telecomunicações, os instrumentos de medição deixam de ser unidades autônomas isoladas e passam a constituir redes integradas de medição, dotadas de lógica e comunicação complexas. Na área de medição de energia elétrica, observa-se uma evolução que foi iniciada com os tradicionais medidores eletromecânicos, passando por medidores eletrônicos isolados, até chegar aos atuais modelos nos quais medidores são dotados de lógica computacional e capacidade de comunicação, estabelecendo-se em rede. Tal modelo, que aqui denominamos Rede de Medição de Energia Elétrica (RMEE), traz uma série de novas possibilidades para o processo de medição, divulgação e cobrança do consumo de energia elétrica.

A principal mudança das RMEE em relação aos sistemas tradicionais é a separação entre unidade responsável pela medição e elemento responsável pela divulgação da informação de consumo, seja à fornecedora de energia, seja ao consumidor. Tal separação vem acompanhada da necessidade de troca de informações entre unidades de medição, central de tarifação e terminais de consulta, tornando a RMEE um complexo sistema de comunicações.

São inúmeras as possibilidades trazidas pelas RMEE, dentre as quais destacamos:

- tarifação e cobrança automatizadas;

- desligamento e religamento de energia à distância;

- mapeamento do furto de energia;

- redução da potência elétrica fornecida a determinados clientes (conhecido como “corte social”);

- tarifação diferenciada por horário.

Tais possibilidades, entretanto, trazem consigo uma série de riscos potenciais à confiabilidade da medição de consumo informada ao consumidor [2]. O papel da Metrologia Legal, através da Autoridade Regulamentadora, é o de impedir que tais riscos se concretizem, garantindo a integridade da informação de consumo, desde a sua geração na unidade de medição, até a sua divulgação ao cliente. No presente trabalho, apresentamos um modelo para o funcionamento de uma RMEE genérica e investigamos os riscos que surgem quando se migra de um sistema baseado em medidores eletromecânicos ou eletrônicos isolados para uma complexa rede de medidores interconectados.

Desenvolvemos um arcabouço para a avaliação da segurança dos dados legalmente relevantes manipulados por redes de medidores de energia elétrica. Desenvolvemos um modelo para o funcionamento de redes de medidores e, a partir deste modelo, elaboramos uma metodologia para a referida avaliação de segurança. Tal avaliação envolve três etapas

1) Avaliação da arquitetura e do funcionamento da RMEE. Esta primeira etapa de avaliação é fortemente baseada em análise de documentação, e buscam-se os seguintes objetivos:
    a. Completo entendimento do funcionamento da rede de medidores e de seus principais protocolos e algoritmos.
    b. Identificação das interfaces de comunicação de todos os participantes da rede de medidores.
    c. Identificação de toda a cadeia de comunicação através do qual são transmitidos os dados metrológicos legalmente relevantes.
    d. Identificação do software legalmente relevante, ou seja, aquele que tem a

possibilidade de modificar alguma informação de medida.

e. Avaliação dos protocolos e algoritmos utilizados, verificando se estes atendem aos requisitos de segurança preestabelecidos.

A não-adequação da rede de medidores a requisitos mínimos de segurança pode ser verificada já nesta primeira etapa.

2) Validação do software legalmente relevante. Ao final desta etapa, a Autoridade Regulamentadora possuirá o exato código do(s) programa(s) que deverá(ão) estar em execução nas unidades da rede de medidores. Em geral, duas atividades são conduzidas nesta etapa:

   a. Verificação do(s) código(s) fontes das unidades da rede de medição.

   b. Mapeamento do(s) código(s) fonte(s) aprovado para o(s) código(s) executável(is) que deverá(ao) ser executado(s) nas unidades da rede de medição.

3) Avaliação dos mecanismos de garantia da integridade do software legalmente relevante. Existem dois níveis de garantia de integridade de software.

   a. Garantia que software adulterado não será executado.

   b. Garantia de que software adulterado será identificado.

   Deve-se proporcionar, pelo menos, o nível mínimo de garantia de integridade descrito no item b, ou seja, um software adulterado até poderá ser executado na rede de medidores, no entanto, ele será detectado em situações de auditoria e outras verificações periódicas.

O presente trabalho foi desenvolvido a partir da necessidade de avaliação da segurança lógica do processo de medição e transmissão de informações metrologicamente relevantes em RMEE. Ao longo do processo de avaliação de tais redes, verificou-se a necessidade das três etapas de avaliação descritas. As etapas de avaliação da arquitetura (1) e de validação do software (2) são conduzidas usando técnicas convencionais, e descritas nas Seções 2 e 3, respectivamente. Para a etapa de verificação da integridade (3) usamos uma abordagem inovadora, a qual descrevemos em maiores detalhes na Seção 4. A Seção 5 contém nossas considerações finais.

## 2. AVALIAÇÃO DA ARQUITETURA

Como descrevemos na introdução, a primeira etapa para a avaliação da segurança de uma rede de medidores envolve o entendimento do seu funcionamento. De uma forma geral, uma RMEE é composta de quatro partes: unidade de medição (UM), rede de concentradores de dados (RCD), central de processamento da fornecedora de energia (CPF) e terminal de consulta do cliente (TCC). A unidade de medição (UM) é o elemento ligado à rede elétrica responsável pela medida de consumo de energia. O RCD possui uma série de concentradores de dados (CD), os quais podem estar organizados hierarquicamente e são responsáveis pela coleta e distribuição dos dados gerados pela UM. O RCD comunica-se com o TC e o CPF. O CPF é o sistema da fornecedora de energia que receberá os dados de consumo e dará início ao processo de tarifação e cobrança. Em ambientes mais complexos, o CPF poderá fazer processamentos mais sofisticados, como, por exemplo, a verificação em tempo real do consumo de seus clientes. É também, do CPF, que partem comandos de configuração da Rede, tais como comandos de manutenção e de desligamento/religamento de energia. Finalmente, o TC é o equipamento onde o cliente consultará o seu consumo. O diagrama da Figura 1 ilustra a organização de uma RMEE genérica:

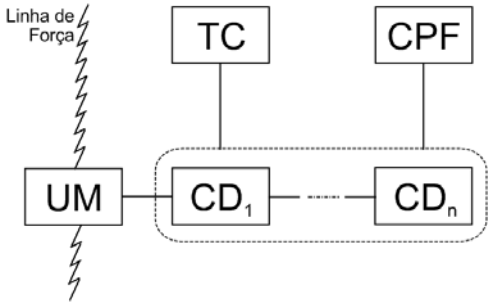


Fig. 1. Arquitetura de uma RMEE – visão de um cliente

Do ponto de vista do cliente, seu consumo é registrado por uma determinada UM ligada à linha de força que abastece sua residência. A informação de consumo é transmitida a um conjunto de CI responsáveis pela coleta de informações de consumo e sua posterior transmissão ao TC e à CPF. A organização de uma RMEE, considerando-se o conjunto de seus clientes, é apresentada na Figura 2.

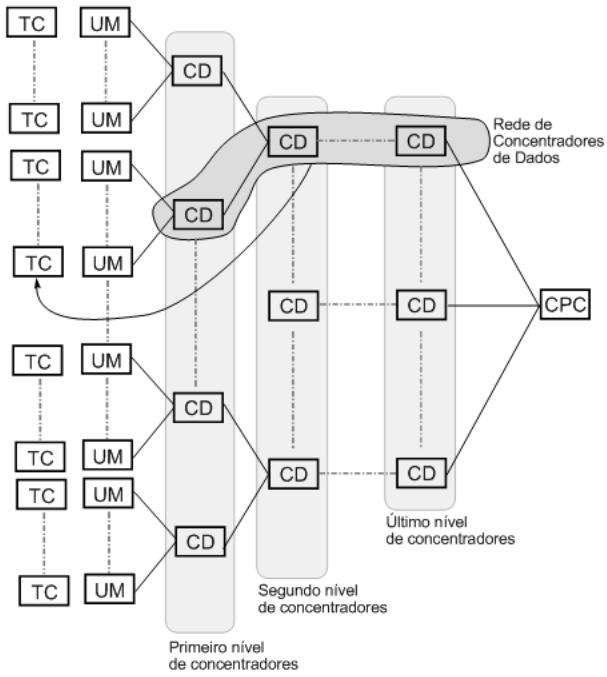


Fig. 2. Arquitetura de uma RMEE – visão do conjunto dos clientes

A Figura 2 mostra uma organização hierárquica da RMEE, na qual um conjunto de UM transmite dados a um CD de primeiro nível, e cada conjunto de CD de nível n transmite dados a um CI de nível n+1. Finalmente, os dados de toda a Rede são transmitidos à CPF por um conjunto de

CD de último nível. Observamos, no entanto, que tal organização hierárquica não se trata de requisito para a construção de uma RMEE: é perfeitamente viável um projeto cujos elementos formam uma rede *ad hoc*, conforme mostramos na Figura 3. Tal abordagem, entretanto, não foi observada, até o momento, na construção de redes de medidores elétricos.

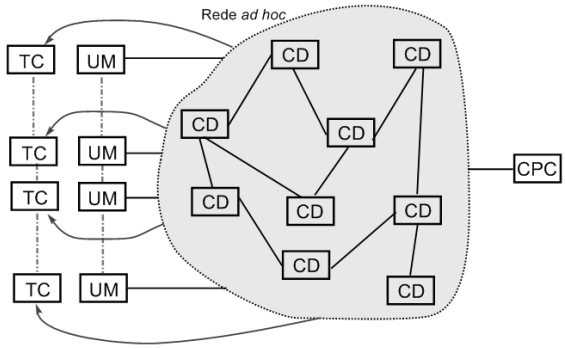


Fig. 3. RMEE com rede *ad hoc* de concentradores

Uma vez que haja entendimento da topologia da rede de medidores, das interfaces de comunicação entre as entidades da rede, e do fluxo das informações metrológicas dentro da rede, torna-se necessário definir quais são as unidades de software legalmente relevantes. O software legalmente relevante é exatamente aquele que, de alguma forma, pode manipular a informação de medida entre o momento de sua geração (na UM) e o momento de sua exibição (no TC). É exatamente este software que deverá ser validado e cuja integridade deverá estar garantida durante o funcionamento da rede.

## 3. VALIDAÇÃO DO SOFTWARE

A validação do software de medição visa verificar se a topologia, os protocolos e os algoritmos apresentados são efetivamente implementados na rede de medidores. Assim como na etapa de avaliação de arquitetura traça-se toda a cadeia de transmissão das informações legalmente relevantes, de forma a identificar o software legalmente relevante, durante a etapa de validação de software busca-se identificar quais são as funções e rotinas, dentro do software legalmente relevante, que podem efetuar modificações nas variáveis que carregam informações de medida, e essas unidades de software são as consideradas críticas – e que serão cuidadosamente verificadas. Para identificar tais funções, rotinas e variáveis, utiliza-se a técnica de *dataflow analysis* [3].

A verificação do software pode ser feita diretamente no código executável ou através de inspeção no código fonte do software. Uma inspeção em nível assembly dificilmente contemplará uma verificação detalhada de todas as instruções do software, dada a dificuldade dessa verificação. Uma alternativa mais eficiente é solicitar o código-fonte do software – em C ou Basic, por exemplo – juntamente com toda a documentação disponível, como manuais, fluxogramas e outros diagramas. A análise baseada em código-fonte traz uma questão que deve ser respondida de forma a se garantir a segurança da Rede: como assegurar que determinado código executável corresponde a um código-fonte cuja segurança foi atestada? Identificamos três estratégias possíveis para responder a essa pergunta:

**Auditoria do processo de compilação.** Trata-se de efetuar uma avaliação detalhada do ambiente de compilação do fabricante responsável pela RMEE, assegurando-se que o código-executável é realmente gerado a partir do código-fonte avaliado e aprovado pela Autoridade Metrológica.

**Reprodução da compilação.** Neste caso, o fabricante fornece informação detalhada a respeito do seu ambiente de compilação. A Autoridade Metrológica reproduz exatamente este ambiente e gera um código executável, para posterior comparação com o código do fabricante.

**Disassembly/decompiling.** Uso de técnicas que permitem, a partir do código executável da RMEE, obter código assembly e código de relativamente alto nível, para comparação com o código apresentado pelo fabricante.

## 4. VERIFICAÇÃO DA INTEGRIDADE

Embora a verificação da integridade do software mostre-se crítica, os fabricantes de dispositivos de medição mostram-se muito pouco dispostos a manter abertos seus códigos, já que isto poderia comprometer a proteção de sua propriedade intelectual. Buscou-se, então, o desenvolvimento de uma abordagem que permitisse verificar que o software em execução é exatamente aquele esperado.

O presente trabalho apresenta uma abordagem inovadora para a verificação da integridade de um determinado software. Desenvolveu-se uma série de premissas que tornaram possível confiar no uso do conceito de "reflexão" para a verificação da integridade de um software, sem que fosse necessário ler o conteúdo daquele software.

A "reflexão", é um mecanismo de auto-verificação de software com o objetivo de assegurar sua integridade [4,6]. Em tal abordagem, o software a ser verificado deve ser capaz de responder a uma série de "perguntas" a seu respeito. Tais perguntas são elaboradas de tal forma que se torna muito difícil um determinado software fazer-se passar por outro. A base teórica para o mecanismo de reflexão é a hipótese de Smith [4], segundo a qual um software deve ser capaz de ler o seu próprio conteúdo – ou seja, os endereços de memória onde estão gravadas as suas instruções. Satisfeito este requisito, é possível desenvolver um protocolo no qual o software deverá calcular resumos criptográficos de trechos de seu próprio código – mais precisamente, o software precisará calcular Message Authentication Codes [7] sobre seu próprio código. Além do mais, é possível projetar tal protocolo de tal forma que, sob certas hipóteses, apenas o software original (íntegro) será capaz de retornar as respostas esperadas no tempo esperado.

Um Message Authentication Code (MAC) é uma função $MAC{:}M{\times}K{\rightarrow}D$ que associa a cada mensagem $m{\in}M$ e chave $k{\in}K$ um "resumo" $d{\in}D$. O MAC é construído de tal forma que qualquer pequena alteração (de apenas um bit, por exemplo) na mensagem ou chave gerará um resumo completamente não-correlacionado (o número esperado de bits diferentes é 50% do resumo). Além disso, o MAC é

construído de tal forma que é impraticável encontrar duas mensagens diferentes que, para a mesma chave, retornam o mesmo resumo. O MAC apresentou-se como ferramenta ideal para o processo de verificação de integridade de software: ele permite criar um protocolo em que o software a ser verificado recebe uma chave e calcula o resumo de seu próprio código para aquela chave. Basta ao verificador comparar o resumo recebido com o resumo esperado. Qualquer alteração mínima de software levará o software alterado a retornar um resumo completamente diferente.

Em uma situação ideal, o verificador teria a garantia de integridade, pelo menos, da própria rotina de verificação de integridade, caso em que torna-se impossível burlar a metodologia proposta. Caso não se possa garantir a integridade da rotina de verificação de integridade, identificam-se algumas possibilidades de ataque. Tais ataques, entretanto, foram avaliados e tratados, conforme descrevemos a seguir.

Os ataques à metodologia de verificação de integridade proposta consistem em fazer alterações de software conjugadas com alterações na rotina de verificação de integridade. Um ataque possível consiste em modificar a rotina de verificação de integridade de tal forma que essa rotina mantenha uma cópia do software original e efetue todos os cálculos de resumos criptográficos sobre esta cópia. Dessa forma, o software efetivamente em execução poderá ser livremente adulterado, já que ele não será lido pela rotina de verificação de integridade.

O ataque descrito anteriormente é pouco viável nos atuais dispositivos de medição elétrica, os quais, em geral, funcionam à base de software embarcado, com grandes restrições de memória. Ainda assim, se considerarmos que existe uma grande quantidade de memória disponível para manter cópia do software original, a execução de instruções adicionais (instruções de desvio para os endereços de memória onde está armazenada a cópia) acarretará impactos no tempo de resposta da rotina de integridade. Dessa forma, mesmo que a rotina maliciosa seja capaz de retornar os resumos criptográficos corretos, ela será incapaz de fazê-lo no mesmo tempo de execução que a rotina íntegra.

A abordagem para verificação de integridade desenvolvida no presente trabalho apresenta vantagens em relação a abordagens tradicionais, conforme descrevemos a seguir.

**1) Manutenção do código executável "aberto para leitura".** Neste caso, os procedimentos, algoritmos e protocolos ficam disponíveis a qualquer um que tenha acesso ao equipamento. Além de isso poder configurar, em alguns casos, uma redução do nível de segurança, ocorre também a disponibilização do software — a qualquer um que tenha acesso ao equipamento.

**2) Testes de caixa preta.** Uma outra possível abordagem é tratar o software embarcado como uma "caixa preta" e executar uma elevada quantidade de testes, cobrindo uma grande quantidade de entradas e verificando as saídas retornadas pelo dispositivo. Tal abordagem apresenta evidente desvantagem quando deseja-se proteger o software contra alterações maliciosas, afinal, estas alterações, em geral determinam códigos (longas seqüências numéricas) que ativam o comportamento malicioso. Se o código for suficientemente grande, a probabilidade de um teste de caixa preta encontrá-lo será mínima.

**3) Hardware de segurança dedicado.** Existem soluções de segurança baseadas em hardware. Um exemplo é o uso de TPM (Trusted Plataform Model) [8]. Embora tais soluções possam alcançar bons níveis de segurança, o impacto financeiro de alterações em nível de hardware é certamente maior do que uma solução baseada em software.

A metodologia de verificação de integridade de software proposta neste trabalho apresenta diversas vantagens potenciais quando comparada com as metodologias tradicionais. As vantagens abrangem aspectos financeiros, de segurança e de proteção intelectual. Os resultados teóricos deste trabalho e os primeiros testes de bancada sinalizam para a viabilidade de seu uso em dispositivos comerciais. Muito em breve veremos a metodologia aplicada a dispositivos de medição de energia elétrica.

## 5. CONSIDERAÇÕES FINAIS

No presente trabalho apresentamos um modelo para o funcionamento das modernas Redes de Medidores de Energia Elétrica. O modelo desenvolvido é bastante abrangente, podendo ser aplicado a outros tipos de redes de medidores, e não apenas de energia elétrica. A partir desde modelo, torna-se possível identificar o software legalmente relevante de uma rede de medidores, além das interfaces de comunicação críticas, para então proceder com uma avaliação de segurança detalhada.

Desenvolvemos uma metodologia de avaliação que contempla três etapas principais: avaliação de arquitetura, validação de software e verificação de integridade. Enquanto para as duas primeiras etapas verifica-se que as técnicas convencionais aplicam-se adequadamente, para a terceira etapa tornou-se necessário desenvolver uma metodologia alternativa de verificação de integridade na qual não fosse necessário ler o conteúdo do software cuja integridade está sendo verificada. Nossa metodologia baseia-se em um protocolo no qual o software verificado deve responder a uma série de "perguntas" a respeito de si mesmo – mais precisamente, deve retornar resumos criptográficos de determinados trechos de seu próprio código. O protocolo foi desenvolvido de tal forma que é impraticável que um software adulterado possa comportar-se como o software original. Dessa forma, torna-se possível atestar a integridade de um software em execução em uma rede de medidores sem que seja necessário ler o conteúdo deste software.

## AGRADECIMENTOS

Agradecemos aos revisores anônimos pela cuidadosa revisão e valiosos comentários e sugestões.

## REFERÊNCIAS


[1] F. Douglas, *The compression cache: using on-line compression to extend physical memory*, Proceedings of the Third USENIX Conference (Anaheim, CA, Jan.), USENIX Assoc., Berkeley, CA, 519–529, 1993.

[2] Inmetro, *Portaria Inmetro nº 011 de 13 de janeiro de 2009*, In Proceedings of the Third USENIX Conference (Anaheim, CA, Jan.), USENIX Assoc., Berkeley, CA, 519–529, 1993.

[3] F. Nielson, H. R. Nielson, and C. Hankin, *Principles of program analysis*, Springer, 2004.

[4] B. C. Smith, *Procedural reflection in programming languages*, Ph.D. Dissertation. MIT Laboratory for Computer Science, Cambridge, MA, 1982.

[5] A. Seshadri, *Using Software based attestation for verifying embedded systems in cars*, Technical Report, 2004.

[6] D. Spinnelis, *Reflection as a Mechanism for Software Integrity Verification*, ACM Transactions on Information and System Security, Vol. 3, No. 1, 51–62, 2000.

[7] W. Stallings, *Cryptography and network security*, *Second Edition*, Prentice Hall, 1999.

[8] Trusted Computing Group (TCG). *http://www.trustedcomputinggroup.org*, última visita: abril de 2009.