Uma arquitetura de segurança para medidores inteligentes – verificação prática de dados de energia multitarifada

Sérgio Câmara[1,2], Raphael Machado[2], Luci Pirmez[1], Luiz F.R.C. Carmo[1,2]

[1]PPGI iNCE/IM, Universidade Federal do Rio de Janeiro (UFRJ), RJ – Brasil

[2]Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro),
Av. Nossa Senhora das Graças, 50, Xerém, Duque de Caxias, 25250-020, RJ – Brasil

{smcamara,rcmachado,lfrust}@inmetro.gov.br, luci@nce.ufrj.br

Abstract. *Energy meters have become functionally complex over time, on the other hand, the correctness of their behavior is still questionable and the effort involved on new smart meters type approval is growing huge. Aiming at establishing trust on the smart meter, this paper proposes a security architecture based on a consumption authenticator. This authenticator is generated inside a secure device, placed on the metering module, using ECPVS signature scheme. Our approach considers different Time-Of-Use scenarios and presents three composing techniques for the authenticator's embedded message. Our concerns are the final size of this message and the necessary data for validating the Time-Of-Use rates consumption values.*

Resumo. *Medidores de energia elétrica tornaram-se equipamentos complexos, por outro lado, seu correto funcionamento ainda é questionável e o esforço envolvido na aprovação de novos modelos aumentou imensamente. Objetivando estabelecer confiança no medidor inteligente, este artigo propõe uma arquitetura de segurança baseada em um autenticador de consumo. Este autenticador é gerado por um dispositivo seguro, localizado no módulo de medição, usando o esquema ECPVS. Nossa abordagem considera diferentes cenários de energia multitarifada e apresenta três técnicas de composição da mensagem contida no autenticador. Nossas preocupações incluem o tamanho total desta mensagem e os dados necessários para validação do consumo em cada faixa de preço.*

1. Introdução

As Redes Elétricas Inteligentes (*Smart Grids*) são uma realidade atualmente em diferentes localidades e continuam a demandar um grande número de pesquisas, estabelecimento de novos conceitos e esforços de desenvolvimento em busca de melhorias e confiança na rede. No Brasil, o primeiro passo em direção à Rede Inteligente foi a adoção, no começo dos anos 2000, do Sistema de Medição Centralizado (SMC), que foi desenvolvido com um objetivo específico brasileiro – reduzir as altas taxas de perda não-técnicas devido ao roubo de energia [Boccardo et al. 2010]. A arquitetura SMC determinava que os módulos de medição não mais estivessem localizados em cada residência e, sim, agrupados em um concentrador, geralmente localizado no alto do poste de luz em meio às instalações

*Artigo financiado com recursos da MCT/FINEP PLATCOG (01.10.0549.00).

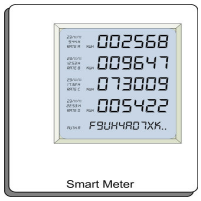

Figura 1. Exemplo de um dispositivo mostrador exibindo 4 valores de kWh por faixa de preço e o Autenticador de Consumo Distribuído.

elétricas de média tensão, para dificultar o acesso físico. Cada módulo de medição é associado ao seu respectivo dispositivo mostrador, localizado na residência do consumidor, no qual são atualizados constantemente os valores de consumo de energia através de uma comunicação sem-fio.

Se, por um lado, a adoção desta arquitetura tornou o roubo de energia uma tarefa mais difícil, por outro, vários medidores começaram a se comportar de maneira incorreta com o passar do tempo, apresentando típicas falhas associadas a *software*. Tal problema revelou a necessidade de um controle estrito do *software* embarcado no SMC. Em 2009, a Portaria Inmetro 011/2009 [Inmetro 2009] foi publicada, a qual estipulava requisitos específicos para esse *software* e tornava obrigatória a abertura de seu código-fonte para o Inmetro. Desde então, os medidores inteligentes são aprovados de acordo com esta portaria juntamente com a Portaria Inmetro 371/2007, relacionada à parte metrológica.

A crescente complexidade da arquitetura do medidor e de seu *software* embarcado apresentam um novo desafio para a autoridade metrológica brasileira: como validar e confiar em uma solução de medidor, ao passo que o processo de aprovação de novos modelos se torna cada vez mais dispendioso? A ideia aqui apresentada é a de assinar digitalmente os dados o mais próximo possível do tempo de "consolidação dos dados de medição". Uma vez que estes dados sejam assinados, qualquer manipulação posterior seria detectada e, dessa forma, os elementos de *hardware* e *software* não mais precisariam ser validados metrologicamente – isto é, o processo de validação de *software* pela autoridade metrológica será restrita apenas aos módulos que manipulam os dados "antes da assinatura digital". Ademais, as preocupações relacionadas aos riscos de vazamento de propriedade intelectual associados à abertura e armazenamento de código-fonte dos fabricantes seriam eliminadas.

Enquanto a adoção desta solução simplifica o processo de validação do *software* metrológico, o aumento de confiança nos medidores pela sociedade também se faz necessário. Para isso, a solução requer o estabelecimento de um padrão de assinatura seguro – que nós chamaremos de *Autenticador de Consumo Distribuído* – que poderá ser usado pelo consumidor para verificação da integridade dos valores de kWh consumidos e cobrados em sua conta. O presente artigo propõe algumas técnicas para a composição deste autenticador levando em conta seu tamanho máximo, e considerando uma abordagem prática: o autenticador ACD será lido do dispositivo mostrador (Fig. 1) e transcrito pelo consumidor durante o processo de verificação de integridade dos dados de medição. Tal abordagem permite um baixo custo no processo de verificação, uma vez que não há a necessidade de um segundo dispositivo para possibilitar a leitura dos dados.

Em resumo, nossa proposta apresenta uma arquitetura de segurança, apropriada

para a autoridade metrológica, facilitando o processo de aprovação de modelos de medidores inteligentes, e para o consumidor final, permitindo a verificação da autenticidade e integridade dos dados de consumo. A proposta se concentra especificamente no cenário de energia multitarifada, onde não apenas o total do consumo, mas também o "momento do consumo", é importante no cálculo da conta, devido a diferentes preços de energia ao longo do dia.

O restante deste artigo está organizado da seguinte maneira. A Seção 2 apresenta esforços estabelecidos e em andamento que abordam a segurança, de forma geral, nos medidores elétricos inteligentes. A Seção 3 explica de forma detalhada a contextualização e definição do problema abordado. Os elementos e mecanismos da arquitetura de segurança proposta são descritos na Seção 4. Na Seção 5, três técnicas de composição do autenticador ACD são descritas e, na Seção 6, essas técnicas são avaliadas e os resultados apresentados. A Seção 7 expõe algumas considerações de segurança da arquitetura e, finalmente, fazemos nossa conclusão na Seção 8.

2. Trabalhos Relacionados

A segurança da informação desempenha um papel fundamental nas Redes Inteligentes, portanto é necessário que o *software* embarcado nos medidores inteligentes atenda os requisitos de segurança estipulados pela autoridade metrológica de cada localidade e siga por uma avaliação estrita antes de sua aprovação. No Brasil, as portarias lançadas pelo Inmetro – Portaria Inmetro 011/2009 e Portaria Inmetro 366/2011 – cumprem esse papel normativo. No âmbito internacional, podemos destacar dois documentos chave que serviriam de base para as portarias do Inmetro e que abordam os problemas envolvendo a avaliação de *software* embarcado, são eles: (i) o OIML D31/2009: *General Requirements of Software Controlled Measuring Instruments* [OIML 2009] e (ii) o WELMEC 7.2: *Software Guide – Measuring Instruments Directive* 2004/22/EC [WELMEC 2008]. Ambos os documentos estabelecem requisitos para *software* embarcado em instrumentos de medição, abordando temas como identificação, separação e atualização de *software*, proteção de interfaces, proteção contra mudanças, recuperação de falhas entre outros.

Na Alemanha, o processo de avaliação de medidores inteligentes é baseado nas diretivas estabelecidas pelo WELMEC e pelo PTB – *Physikalisch-Technische Bundesanstalt* – o Instituto Metrológico Nacional alemão. Além dos medidores, a infraestrutura de distribuição alemã conta com um elemento a mais, um *gateway*, que compreende as funcionalidades relacionadas à segurança da informação. Dessa forma, ao separar as particularidades dos medidores MID das funções de segurança da informação, o *gateway* pode ser submetido a uma avaliação de Common Criteria, de acordo com um Protection Profile apropriado. Além dessa vantagem, essa abordagem permite que os Organismos Notificados lidem de forma melhor com as limitações em envolver especialistas de segurança em TI em avaliações de medidores.

Outras soluções de segurança para os medidores inteligentes estão sendo propostas fazendo o uso de módulos criptográficos integrados a arquitetura de medição. Em Teller *et al.* [Treytl et al. 2004], uma versão compacta do TPM – Trusted Platform Module – é implementada, a qual consome poucos recursos e requer um conjunto mínimo de comandos. Este módulo criptográfico é integrado ao medidor e provê proteção de propriedade intelectual, corretude do comportamento e atualização segura do *software*. Em

outro projeto, a Infineon, uma fabricante de semicondutores, desenvolveu a família de chipset UMF11x0 [Infineon 2012] para medidores inteligentes, que pode ser conectados a um *smartcard* ou a um módulo de segurança de hardware. Esse *chip* oferece funções para criptografia simétrica (AES-128/256) e para uma Infraestrutura de Chave-Pública, além de proteção das chaves criptográficas e um gerador real de números aleatórios.

Apesar dos trabalhos citados anteriormente apontarem preocupações e soluções sobre a segurança e corretude do comportamento dos medidores inteligentes, o presente trabalho é o primeiro a propor uma arquitetura de segurança que disponibiliza um procedimento prático de verificação de integridade dos dados gerados por um medidor inteligente e capaz de ser realizado por um usuário comum.

3. Definição do Problema

O *software* embarcado nos medidores inteligentes não podem ser, *a priori*, considerados confiáveis, isto é, não há garantias que este irá se comportar corretamente ou que este é exatamente o *software* validado na aprovação de modelos. A autoridade metrológica parte do princípio que os medidores inteligentes poderiam ser modificados posteriormente pelos consumidores, pelos fabricantes ou pela distribuidora de energia, para que ajam de maneira maliciosa de acordo com seus interesses.

Medidores convencionais, sem multitarifação de energia, apenas medem acumulativamente o total de energia consumida, logo alguém poderia forçar, por exemplo, um comportamento malicioso que incremente o contador total de energia mais rápido do que o normal, para tirar vantagem desse falso consumo extra todo mês. No entanto, medidores com suporte a multitarifação (horo-sazonal) podem ser adulterados de forma mais sutil – o total de consumo de energia continuaria inalterado, porém uma (pequena) porcentagem da faixa de preço mais barata de energia poderia ser adicionada às faixas de preço de pico, dessa forma, gerando contas mais caras.

Logo, para a validação em um cenário de multitarifação de energia, uma verificação de consumo deve assegurar os valores consumidos em cada faixa de preço. Além disso, se o mecanismo não sabe a configuração destas faixas, que é o nosso caso, ele deverá consolidar dados relevantes para esse propósito e, de alguma forma segura, disponibilizar essa informação para o usuário final. Neste trabalho, isso será feito através de um código autenticador.

Por motivo da não necessidade de outros dispositivos se comunicarem com o medidor para realizar a leitura do autenticador (comentado na Seção 4.1), este será exibido no próprio dispositivo mostrador. Esta condição estipula restrições sobre o tamanho, a integridade e autenticidade do autenticador: ele deve ser tão curto quanto possível e a prova de falsificações.

Nossos esforços a seguir concentram-se na descrição do mecanismo de segurança usado, em como ele consolida os dados relevantes para a validação das faixas de consumo e em como o autenticador pode ser encurtado.

4. Proposta

A proposta deste trabalho é a concepção de uma arquitetura de segurança que estabeleça confiabilidade nos dados gerados pelo medidor inteligente para todas as partes interessadas – o consumidor, a distribuidora de energia elétrica e a autoridade metrológica. A

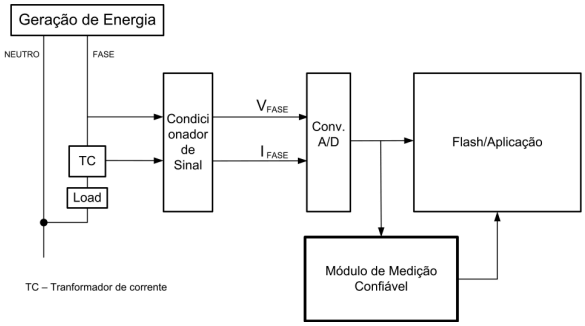

Figura 2. Diagrama de blocos dos componentes de um módulo de medição incluindo o TMM.

arquitetura é composta por mecanismos de segurança, como criptografia de chave-pública e assinatura digital, juntamente a procedimentos externos de verificação de dados.

A arquitetura tem sua "raiz de confiança" baseada em um *hardware* que chamaremos de *Trusted Metering Module*. O TMM é um módulo criptográfico acoplado ao começo da cadeia de medição com o objetivo de armazenar dados referentes à energia consumida e repassá-los à frente, para o resto da cadeia de medição, assinados digitalmente. Por definição, a cadeia de medição legalmente relevante é a sequência de elementos de um instrumento, ou sistema de medição, que constitui o trajeto do sinal de medição desde o estímulo até exibição do resultado ao consumidor [VIM 2008]. É mandatório que o TMM esteja posicionado na cadeia de medição antes do *software* proprietário embarcado, que possui seu próprio algoritmo de medição (Fig. 2). Essa restrição assegura que, se dados originais forem alterados de alguma forma após serem assinados digitalmente, este erro, intencional ou não, poderá ser detectado posteriormente através da rastreabilidade provida pela assinatura.

Essa assinatura digital, o Autenticador de Consumo Distribuído (ACD), é uma sequência alfa-numérica disponibilizada no visor do dispositivo mostrador. O autenticador ACD é gerado dentro do TMM e é, então, repassado para à aplicação principal do medidor, que fica encarregada de mandá-lo, juntamente com os valores de consumo de energia, para o dispositivo mostrador. Um novo autenticador ACD é exibido para cada atualização de valores no mostrador, o que acontece a cada unidade de kWh consumido em qualquer faixa de preço.

Chamaremos de Processo de Validação da Medição (PVM) o processo pelo qual o consumidor pode checar a integridade dos dados de consumo. O PVM consiste basicamente da transcrição e inserção de dados relevantes em um sistema computacional, além da execução do algoritmo de validação. As entradas que o algoritmo precisa saber são: o autenticador ACD e os valores de consumo em cada faixa horária no momento da leitura, a configuração das faixas de preço e a chave-pública do TMM.

4.1. Objetivos e Requisitos

Nesta Seção, os requisitos mandatórios que a arquitetura de segurança deve atender são descritos. Seguem dois objetivos gerais do projeto:

1. Diminuir o esforço empregado durante o processo de aprovação de modelos de medidores inteligentes;

2. Prover confiança no comportamento do medidor inteligente para as partes interessadas, garantindo a autenticidade e integridade dos dados de consumo.

Durante o processo de aprovação de modelos de medidores, a autoridade metrológica restringiria sua análise apenas aos componentes legalmente relevantes que se encontram antes do TMM na cadeia de medição, uma vez que seria apenas necessário garantir que os dados de entrada do TMM sejam genuinamente dos sensores de corrente e voltagem. Dessa forma, o TMM garante a rastreabilidade dos dados originais. Como consequência do uso dessa arquitetura, a autoridade metrológica mitigaria os riscos associados à abertura e armazenamento do código proprietário da aplicação metrológica dos fabricantes de medidores.

Em relação ao item 2, através do Autenticador de Consumo Distribuído e do Processo de Validação da Medição, os consumidores terão meios de estabelecer confiança nos dados de consumos exibidos no visor e cobrados em sua conta de luz. Para as distribuidoras de energia elétrica, sua confiança nos medidores inteligentes em operação também aumenta, uma vez que elas poderão recuperar os autenticadores ACD remotamente e atestar se seus medidores sofreram algum tipo de violação que comprometa a integridade de seus dados.

Além dos objetivos gerais, os requisitos atendidos pelo projeto são:

- Estabelecer as mudanças necessárias na arquitetura de um medidor inteligente, priorizando a mitigação de vulnerabilidades.
- Dar suporte à funcionalidade de Postos Tarifários (*Time-Of-Use*).
- Manter um baixo custo para a solução.
- Estabelecer maneiras de compor o Autenticador de Consumo Distribuído.

O esboço do TMM e sua posição são descritos de forma a dificultar a inserção de *backdoors* no mecanismo. Mais detalhes serão discutidos na Seção 4.2. É mandatório no projeto que não haja necessidade de um segundo dispositivo se comunicar com o medidor para possibilitar o PVM, logo a leitura dos dados deverá ser possível e factível de forma manual. Dessa forma, o autenticador ACD deverá apresentar o menor número de caracteres e ainda capaz de oferecer um alto nível de acurácia e segurança para a informação que carrega. Algumas técnicas de composição do ACD serão apresentadas na Seção 5.

4.2. *Trusted Metering Module* (TMM)

Nesta Seção descrevemos, como segue, um esboço do *Trusted Metering Module*. O TMM é um módulo criptográfico capaz também de contabilizar o consumo de energia, além de realizar procedimentos particulares ao projeto. Seus componentes internos são: (i) o processador, (ii) o código do programa, o qual inclui as técnicas de composição do autenticador ACD, (iii) o motor de execução, (iv) a memória persistente e protegida, para a guarda de chaves criptográficas, (v) a memória versátil, que armazena variáveis e valores de energia consumidos, (vi) o motor de criptografia e assinatura digital, (vii) o motor de *hash* (resumo criptográfico) e (viii) o *Real Time Clock*, RTC.

O TMM possui duas entradas de dados: uma referente às grandezas de voltagem e corrente, provenientes das fases do circuito elétrico da instalação, amostrados por um conversor Analógico/Digital, e a outra aos comandos para correção de sincronização do

RTC, localizado no TMM, com o relógio da aplicação. Essa sincronização é de responsabilidade da aplicação do fabricante, onde apenas serão permitidas correções de acordo com a deriva máxima em um ano do relógio interno do TMM. O RTC usado deve ser bem preciso, com erro abaixo de 5ppm (partes por milhão) [Arora 2011]. Em princípio, nenhuma outra entrada é considerada para qualquer propósito, de forma que as vulnerabilidades em potencial sejam minimizadas.

Na memória protegida, o TMM armazena uma chave-privada única, gerada em *hardware* em tempo de fabricação, juntamente com sua chave-pública. Esse par de chaves é baseada em curvas elípticas. Além disso, um calendário é também armazenado na memória protegida.

De fato, o TMM desconhece a configuração das faixas de preço, isto é, ele não sabe determinar o começo/término dos períodos de pico, fora-de-pico e intermediário. Portanto, o módulo lida com a multitarifação de energia da forma descrita na sequência. O TMM converte os dados de corrente e voltagem para valores de mesma magnitude de quilowatt-hora e, então, calcula e armazena os valores acumulados de energia total consumidos por períodos fixados de uma hora de duração (por exemplo, 11:00:00h – 11:59:59h). Dessa forma, existem 24 registradores armazenando valores acumulados de consumo desde o tempo de instalação do medidor até o momento presente. Além disso, existe um registrador a mais para armazenar valores de energia fora-do-pico, consumidos durante os fins-de-semana e feriados. Esses registradores precisam ser mantidos atualizados para que o autenticador ACD seja composto corretamente.

4.3. Autenticador de Consumo Distribuído (ACD)

Tecnicamente, o autenticador ACD é uma assinatura ECPVS (*Elliptic Curve Pintsov-Vanstone Signature*) codificado em base64. Uma vez que o esquema ECPVS, especificado em ISO/IEC 9796-3:2006, permite assinaturas de tamanho reduzido – em torno de seis vezes menor do que o RSA e metade do tamanho do ECDSA [Modares 2009] – e recuperação parcial ou total de mensagem, essas características se encaixam idealmente aos requisitos do ACD. Um exemplo de aplicação real deste esquema é o uso de assinaturas ECPVS em *Digital Postage Marks* [Certicom 2004].

De acordo com os algoritmos de assinatura digital e funções de *hash* recomendados no FIPS (*Federal Information Processing Standards Publications*) publicado pelo NIST [Gallagher et al. 2009, Barker and Roginsky 2011], os parâmetros de domínio escolhidos para o processo de assinatura do ACD são:

- Curvas elípticas de 224 bits sobre conjuntos finitos primos;
- Função *Hash* SHA-224;
- XOR Encryption Scheme (Aqui usamos o XOR como cifrador simétrico uma vez que não há real necessidade de confidencialidade da mensagem-ACD, porém a previne de ataques passivos, adicionando uma camada fraca de confidencialidade [Certicom 2000])

O processo de geração da assinatura ECPVS é descrito na sequência [ISO/IEC 2006]. A entrada para o processo consiste dos parâmetros de domínio, a chave-privada x_A e uma mensagem M para ser assinada. Então, o assinante divide a mensagem M em $M_{clr} \mid\mid M_{rec}$, onde M_{clr} é a parte da mensagem enviada em

claro e M_{rec} é a parte recuperável da mensagem. M_{clr} e M_{rec} devem ser formatadas e codificadas de maneira acordada anteriormente entre o assinante e o verificador. Seja d o dado de entrada M_{rec} (com redundância natural ou adicionada), n um número primo, k um inteiro aleatório no intervalo $[1, n-1]$, Π a chave simétrica computada por uma *Key Derivation Function* (KDF) a partir da chave-pública do assinante, a assinatura ECPVS deve ser computada pela seguinte, ou equivalente, sequência de passos:

1. Computar $r = Sym(d, \Pi)$;
2. $u = Hash(r||M_{clr})$;
3. Converta $t = OS2IP(u)$; (octet-string-to-integer primitive function), note que $t \in [0, n-1]$;
4. Se $t = 0$, então o processo de assinatura deve ser repetido com um novo valor aleatório k;
5. Computar $s = (k - x_A t) \mod n$;
6. Se $s = 0$, então o processo de assinatura deve ser repetido com um novo valor aleatório k;
7. Apague k;

Gere como saída a assinatura (r, s) e a mensagem parcial M_{clr} (que pode ser nula).

Para o autenticador ACD, chamamos M de mensagem-ACD, onde M_{rec} é exatamente M e M_{clr} é nula. O tamanho final do autenticador ACD depende do tamanho de s, o *overhead* criptográfico, e r. Uma vez que o tamanho de r está diretamente associado ao tamanho da mensagem-ACD, otimizar a composição desta mensagem se faz necessário, como apresentaremos na próxima seção.

5. Técnicas de Composição da mensagem-ACD

A mensagem-ACD deve conter dados suficientes para ser possível validar os valores de consumo em kWh em cada faixa de preço. Uma vez que o TMM não conhece as faixas definidas, a mensagem-ACD representará, de alguma forma, os valores dos registradores do TMM tornando possível o Processo de Validação da Medição.

Para escolher o número apropriado de bits para cada dado na mensagem, tomaremos em consideração as seguintes suposições: (i) o consumo mensal máximo de uma residência é de 4.000 kWh. No Brasil, o consumo médio é bem abaixo disso [Francisquini 2006], portanto estamos considerando a grande maioria de classes de consumidores e (ii) a vida útil máxima de operação de um medidor inteligente é de 20 anos. Neste caso, estamos também considerando um valor razoavelmente alto em relação à vida útil de um medidor eletrônico, que é de 12 anos aproximadamente [Guimarães].

5.1. Técnica por Valores Absolutos

A Técnica por Valores Absolutos é a nossa primeira tentativa de construir a mensagem-ACD. Sejam os valores nos registradores do TMM Val_i, para $i \in [0, 23]$, logo suas representações (Rep_i) na mensagem-ACD são determinadas por

$$Rep_i = Val_i \tag{1}$$

De acordo com as suposições feitas acima, sobre consumo mensal máximo de uma residência e vida útil máxima de um medidor, o valor de Val_i não deverá ultrapassar

40.000 kWh. Logo, fixaremos $k = 16$, onde k é o número de bits necessários para Rep_i (podendo variar de 0 a 65.535) e compor a mensagem-ACD como segue:

- Soma dos períodos fora-de-pico (fins-de-semana e feriados): 20 bits;
- As 24 representações dos valores dos registradores ($Rep_0 - Rep_{23}$): 16 bits para cada representação;
- *Hash* dos 24 valores dos registradores concatenados ($Hash_{abs}$): 32 bits;

A primeira parte da assinatura, chamada r, compreende a mensagem-ACD criptografada com o cifrador simétrico. A mensagem é o resultado da concatenação dos três *bit strings* apresentados acima. A segunda parte, s, é o *overhead* criptográfico de 28 bytes (esperado de uma chave-privada ECC-224). Logo, o tamanho final da assinatura (r, s) é 54,5 + 28 = 82,5 bytes, gerando 110 caracteres codificados em base64.

Como uma primeira manobra para reduzir o tamanho da assinatura, sugerimos que uma compactação de dados seja realizada. Uma vez que, aos nossos conhecimentos, todas as modalidades existentes atualmente de multitarifação de energia consideram o período entre 23:00h e 5:59h como fora-de-pico, poderemos somar todos os valores dos sete registradores correspondentes ao bloco de bits para períodos fora-de-pico (antes apenas reservado para fins-de-semana e feriados) e, dessa forma, reduzir o tamanho da mensagem-ACD sem prejudicar o resultado final. Com essa alteração, o novo tamanho do autenticador ACD (r, s) seria de 40,5 + 28 = 68,5 bytes, gerando 92 caracteres codificados em base64 acrescido de dois caracteres de *padding* ("=="). Com a Técnica por Valores Absolutos, essa é a menor assinatura esperada. Nas seções seguintes, apresentamos técnicas que utilizam compressão lógica de dados.

5.2. Técnica por Contadores em Módulo

Essa técnica de compressão lógica estabelece que o valor do registrador, Val_i, é representado na mensagem-ACD por um contador sequencial que incrementa ao passo que Val_i é incrementado, porém restringido ao intervalo de inteiro $[0, 2^{k_{mod}} - 1]$, $k < 16$. Portanto, seja k_{mod} o número de bits usados para representar os valores dos registradores, Rep_i é calculado pela seguinte fórmula:

$$Rep_i = Val_i \ mod \ 2^{k_{mod}} \tag{2}$$

A mensagem-ACD é composta, portanto, pelas seguintes *bit strings* concatenadas: (i) soma total dos períodos fora-de-pico (incluindo 23h-5:59h), (ii) as 17 representações

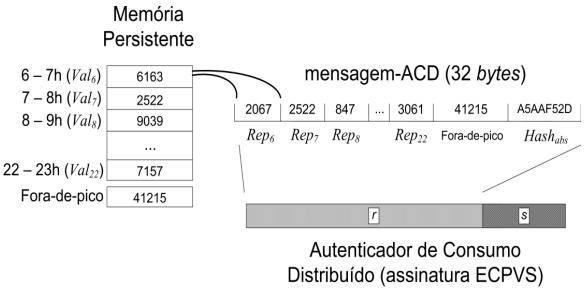

Figura 3. Exemplo de composição da mensagem-ACD pela Técnica por Contadores em Módulo ($k_{mod} = 12$).

dos valores nos registradores ($Rep_6 - Rep_{22}$) e (iii) ***Hash*** (por exemplo, SHA-224) dos 17 valores originais dos registradores concatenados e separados por ","($Hash_{abs}$).

No Processo de Validação da Medição (PVM), o algoritmo de descompressão recupera os dados originais ($Val_6 - Val_{22}$) a partir das representações ($Rep_6 - Rep_{22}$) na mensagem-ACD. Este algoritmo opera da seguinte forma:

1. Computar lista $Lcand_i$ de todos os possíveis candidatos para Rep_i, $i \in [6, 22]$;
2. Computar todas as combinações de elementos de ($Lcand_i, \cdots, Lcand_{i+j}$), o qual este grupo pertence a uma faixa de preço e a soma dos seus elementos é igual ao consumo da faixa de preço apresentado como entrada;
3. Incluir a combinação de elementos em uma lista de Sequências Candidatas, $Lseq_{cand}$;
4. Concatenar todas as listas $Lseq_{cand}$ de todas as faixas de preço, comparando com $Hash_{abs}$;
5. Quando houver igualdade, os valores de consumo originais foram recuperados.

Com os dados originais recuperados, isso significa que a validação da medição foi positiva e o PVM realizado com sucesso.

5.3. Técnica por Contadores em Função de Dispersão

A Técnica por Contadores em Função de Dispersão apresenta uma estrutura parecida com a Técnica por Contadores em Módulo. Sua diferença está no fato de que o valor de consumo de um registrador, Val_i, é representado por um contador aleatório, e não mais por um contador sequencial como na técnica anterior. Os valores desse contador aleatório também estão restringidos por um intervalo de inteiros $[0, 2^{k_{hash}} - 1]$, $k < 16$. O cálculo de Rep_i é dado pela fórmula:

$$Rep_i = ultimosBits(\ k_{hash}, Hash(\ concat(\ Val_i, ID_i))) \tag{3}$$

A função $Hash()$ representa qualquer função com propriedade unidirecional, com resistência a colisões e que gere um dado de saída com pelo menos k_{hash} bits (SHA-224, por exemplo). Para essa técnica, convencionou-se utilizar os últimos k_{hash} bits da saída da função $Hash()$ para preencher Rep_i, no entanto, qualquer outra combinação poderia ter sido escolhida.

A concatenação de Val_i com um identificador próprio do registrador é justificado por uma questão de melhoria na segurança do autenticador. Com esse identificador, a saída de $Hash()$ para um mesmo Val será diferente dependendo do registrador i em que o valor estiver localizado. Essa manobra atribui aos Rep_i uma aleatoriedade entre si, dificultando ainda mais um possível ataque de injeção de dados falsos.

O algoritmo de descompressão usa Tabelas $Rainbow$ durante o processo de listagem de valores candidatos para $Lcand_i$. Essas tabelas mapeiam todos os valores de ***Hash()*** para suas respectivas entradas $concat(Val_i, ID_i)$, onde $Val_i \in [0, 40.000]$ e $i \in [0, 16]$. O algoritmo, então, procede da mesma forma descrita na Seção 5.2.

6. Análise Experimental das Técnicas e Resultados

Nesta seção apresentamos os testes realizados com as técnicas de compressão descritas anteriormente e resolvemos o valor de k apropriado para cada uma. De fato, essas

Tabela 1. Tempo de descompressão x $Interval_{max}$, $k_{mod} \in [8, 14]$ – Técnica por Contadores em Módulo

$Interval_{max}$	Tempo de descompressão (seg)						
(kWh)	k_{mod} = 8	k_{mod} = 9	k_{mod} = 10	k_{mod} = 11	k_{mod} = 12	k_{mod} = 13	k_{mod} = 14
1000	8,9E+02	7,6E-02	5,6E-05	5,6E-05	5,6E-05	5,6E-05	5,6E-05
2000	1,4E+07	9,5E+02	7,7E-02	5,6E-05	5,6E-05	5,6E-05	5,6E-05
3000	1,8E+09	2,5E+05	1,6E+01	4,0E-03	5,6E-05	5,6E-05	5,6E-05
4000	-	1,4E+07	7,7E+02	8,5E-02	5,6E-05	5,6E-05	5,6E-05
5000	-	1,0E+08	1,6E+04	2,1E+00	3,5E-04	5,6E-05	5,6E-05
6000	-	-	2,2E+05	1,9E+01	4,1E-03	5,6E-05	5,6E-05
7000	-	-	2,1E+06	1,5E+02	2,1E-02	5,6E-05	5,6E-05
8000	-	-	1,3E+07	9,1E+02	8,0E-02	5,6E-05	5,6E-05
9000	-	-	4,3E+07	4,7E+03	4,6E-01	9,7E-05	5,6E-05
10000	-	-	2,0E+07	1,7E+04	2,1E+00	3,3E-04	5,6E-05
11000	-	-	-	7,2E+04	5,8E+00	1,3E-03	5,6E-05
12000	-	-	-	2,3E+05	1,5E+01	4,8E-03	5,6E-05
13000	-	-	-	7,9E+05	4,6E+01	1,2E-02	5,6E-05
14000	-	-	-	1,8E+06	1,5E+02	2,6E-02	5,6E-05
15000	-	-	-	5,6E+06	3,8E+02	4,4E-02	5,6E-05
16000	-	-	-	1,3E+07	8,0E+02	7,1E-02	5,6E-05
17000	-	-	-	2,9E+07	2,2E+03	1,3E-01	5,9E-05
18000	-	-	-	-	4,5E+03	4,7E-01	1,1E-04
19000	-	-	-	-	1,1E+04	8,3E-01	1,8E-04
20000	-	-	-	-	1,8E+04	2,0E+00	2,9E-04

técnicas utilizam algoritmos de compressão assimétricos, uma vez que o tempo gasto para recuperar os dados comprimidos pode ser bem maior do que o tempo de compressão, isso dependendo do valor usado em k_{mod}, k_{hash}. Outro conceito importante é o intervalo entre os valores maior e menor dos registradores em cada faixa de preço ($Interval_{max} = Val_{max} - Val_{min}$). Conforme $Interval_{max}$ aumenta, e devido a uma etapa de força-bruta nos algoritmos de descompressão, este processo pode demorar bastante. O tempo máximo assumido para descompressão é de até um minuto, embora este não seja um requisito mas apenas uma escolha razoável.

Para determinar o número apropriado de bits para Rep_i, e encontrar o melhor *trade-off* entre o tempo de descompressão e o tamanho total da mensagem-ACD, para cada $k_{mod}, k_{hash} \in [8, 14]$ e para cada $Interval_{max} \in [1000, 20000]$ foram simulados 100 ACDs sob a modalidade tarifária do Ontario Energy Board [OEB]. A simulação foi realizada em um PC, Intel Q6600 CPU 2.40GHz, 8GB RAM. Os autenticadores ACD simulados e as técnicas de compressão de dados foram implementados em linguagem C++.

A Tabela 1 apresenta os resultados da simulação obtidos com a Técnica por Contadores em Módulo. Dessa tabela, devemos considerar, para o tempo de descompressão abaixo de um minuto, $k_{mod} = 11$ permitindo um $Interval_{max}$ = 6.000 kWh e $k_{mod} = 12$ permitindo um $Interval_{max}$ = 13.000 kWh. Apesar da primeira opção oferecer um tamanho menor para a mensagem-ACD, seu $Interval_{max}$ permitido é considerado pequeno para garantir um tempo de descompressão aceitável durante a vida útil do medidor, logo definiremos $k_{mod} = 12$ como a opção de melhor custo-benefício. Assim sendo, pela Técnica por Contadores em Módulo, o tamanho final do autenticador ACD (r, s) é de 32 + 28 = 60 bytes, resultando em 80 caracteres codificados em base64. A mensagem-ACD é formada com os seguintes dados (Fig. 3):

- Soma dos períodos fora-de-pico (incluindo 23h-5:59h): 20 bits;
- As 17 representações dos valores dos registradores ($Rep_6 - Rep_{22}$): 12 bits para cada representação;
- *Hash* dos 17 valores dos registradores concatenados ($Hash_{abs}$): 32 bits;

Tabela 2. Tempo de descompressão x $Interval_{max}$, $k_{hash} \in [8, 14]$ – Técnica por Contadores em Função de Dispersão

$Interval_{max}$	Tempo de descompressão (seg)						
(kWh)	k_{hash} = 8	k_{hash} = 9	k_{hash} = 10	k_{hash} = 11	k_{hash} = 12	k_{hash} = 13	k_{hash} = 14
1000	1,37E-02	3,28E-03	3,40E-03	3,25E-03	3,21E-03	3,17E-03	3,30E-03
2000	8,06E+00	5,14E-03	2,92E-03	2,75E-03	2,79E-03	2,84E-03	2,80E-03
3000	7,88E+02	7,15E-02	2,98E-03	2,69E-03	2,61E-03	2,73E-03	2,99E-03
4000	2,66E+04	8,97E-01	3,90E-03	2,77E-03	2,72E-03	2,91E-03	2,65E-03
5000	6,46E+05	1,03E+01	6,29E-03	2,80E-03	2,73E-03	2,84E-03	2,79E-03
6000	1,15E+07	9,79E+01	1,51E-02	2,95E-03	2,61E-03	2,88E-03	3,30E-03
7000	-	1,10E+03	6,03E-02	3,08E-03	2,94E-03	2,83E-03	2,69E-03
8000	-	6,81E+03	1,67E-01	3,55E-03	2,80E-03	2,74E-03	2,77E-03
9000	-	1,99E+04	8,03E-01	3,86E-03	2,88E-03	2,80E-03	2,81E-03
10000	-	8,02E+04	1,67E+00	5,09E-03	2,85E-03	2,65E-03	2,66E-03
11000	-	3,33E+05	6,81E+00	5,48E-03	2,75E-03	2,76E-03	2,87E-03
12000	-	-	1,84E+01	7,88E-03	3,00E-03	2,92E-03	3,15E-03
13000	-	-	3,84E+01	1,12E-02	2,85E-03	2,84E-03	2,84E-03
14000	-	-	9,31E+01	2,03E-02	3,03E-03	2,92E-03	3,11E-03
15000	-	-	2,35E+02	2,19E-02	3,05E-03	2,84E-03	2,88E-03
16000	-	-	4,71E+02	4,42E-02	3,27E-03	2,90E-03	2,88E-03
17000	-	-	1,19E+03	6,29E-02	3,34E-03	2,99E-03	2,80E-03
18000	-	-	1,97E+03	1,14E-01	3,50E-03	2,92E-03	2,96E-03
19000	-	-	5,83E+03	2,03E-01	3,49E-03	2,80E-03	2,68E-03
20000	-	-	1,03E+04	3,33E-01	4,13E-03	2,94E-03	3,31E-03

Da simulação, calculamos que, para $k_{mod} = 12$ e $Hash_{abs} = 32$ bits, a chance máxima de encontrar uma solução não-única no processo de descompressão é de $0,001\%$, e a chance média de solução não-única é de $1,902$ x $10^{-4}\%$.

A Tabela 2 apresenta os dados simulados referentes à Técnica por Contadores em Função de Dispersão. Pela tabela constatamos que para $k_{hash} = 10$, com um $Interval_{max}$ = 13.000, temos um tempo de descompressão médio de 38,4 segundos. Logo, para a Técnica por Contadores em Função de Dispersão, definiremos $k_{hash} = 10$ como a opção de melhor custo-benefício. Dessa forma, o tamanho do novo autenticador ACD (r, s) seria de 27,75 + 28 = 55,75 bytes, resultando em 75 caracteres codificados em base64 acrescido de dois caracteres de *padding* ("=="). Para essa técnica, a mensagem-ACD será composta pelos seguintes dados:

- Soma dos períodos fora-de-pico (incluindo 23h-5:59h): 20 bits;
- As 17 representações dos valores dos registradores ($Rep_6 - Rep_{22}$): 10 bits para cada representação;
- *Hash* dos 17 valores dos registradores concatenados ($Hash_{abs}$): 32 bits;

Com a Técnica por Contadores em Função de Dispersão, conseguimos diminuir o tamanho da mensagem-ACD para 222 bits, contra 256 bits da Técnica por Contadores em Módulo e 324 bits da Técnica por Valores Absolutos. Isso representa 68,5% (quociente de compressão) do tamanho da mensagem-ACD usada na primeira técnica apresentada. O autenticador ACD, como um todo, teve uma redução de 67,57% no tamanho desde sua primeira tentativa de composição, caindo de 110 para 75 caracteres, tornando-se mais prático para transcrição.

A redução em aproximadamente $1/3$ do tamanho da mensagem-ACD foi conseguida por técnicas de compressão lógica elaboradas especificamente para esse problema. Algoritmos mais comuns usados para compressão de dados, como os baseados em dicionário, não teriam o mesmo sucesso ao processarem uma mensagem de tamanho pequeno como entrada. Em geral, esses algoritmos necessitam de, no mínimo, 1000 bytes de dados de entrada para tornarem-se eficientes [Chu 1996].

7. Considerações de Segurança

Para qualquer tentativa da aplicação do medidor, ou da distribuidora, de falsificar o autenticador ACD (apresentando mais kWh nos períodos de pico, por exemplo), o atacante deverá estar consciente de que todos os ACDs seguintes devem estar coerentes com os ACDs verificados anteriormente, uma vez que os autenticadores carregam valores acumulados. A probabilidade de falsificação está diretamente associada ao tamanho da chave-privada e função de *hash* usados para assinar, juntamente com a redundância conhecida na mensagem embutida. Ataques de disponibilidade, quando o autenticador não é exibido pelo dispositivo mostrador, seria facilmente detectado pelo usuário.

O *overhead* criptográfico da assinatura ECPVS poderia ser também reduzido, porém isso implicaria na mudança do tamanho da chave usada para assinar. Por exemplo, chaves ECC de 160 bits gerariam um *overhead* de 20 bytes. No entanto, de acordo com o NIST, chaves baseadas em curvas elípticas de 160 bits não estão aprovadas para uso após 2013 [Barker and Roginsky 2011].

8. Conclusão

Neste trabalho, apresentamos uma proposta de arquitetura de segurança que oferece confiança no comportamento e autenticidade dos dados do medidor inteligente para o usuário comum e a distribuidora de energia. Além disso, essa arquitetura pode ser usada para diminuir os esforços da autoridade metrológica durante o processo de aprovação de novos modelos de medidores e, como consequência, mitigar os riscos associados com a abertura e armazenamento de código proprietário dos fabricantes.

A proposta considera diferentes cenários de energia multitarifada e introduz o esboço de um mecanismo baseado em *hardware* confiável e um autenticador de consumo. Cada novo autenticador ACD é exibido pelo medidor para cada atualização de valores de consumo e carrega dados relevantes para validação das medições realizadas. Nós mostramos que, com técnicas de compressão e compactação de dados, o autenticador ACD pode ser reduzido para 67,57% de seu tamanho original, permitindo uma transcrição mais prática para o usuário.

Uma ideia considerada para trabalhos futuros é a criação de um autenticador ACD "relativo" e ainda menor, o qual poderia validar as medições de um medidor inteligente em um período curto de tempo, compreendido entre a coleta de dois autenticadores ACD diferentes.

Referências

Arora, M. (2011). Prevent tampering in energy meters. `http://www.eetimes.com/design/smart-energy-design/4014235/Prevent-tampering-in-energy-meters/`. Acessado em Novembro de 2011.

Barker, E. and Roginsky, A. (2011). NIST special publication 800-131A transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths.

Boccardo, D., Gomes dos Santos, L., da Costa Carmo, L., Dezan, M., Machado, R., and de Aguiar Portugal, S. (2010). Software evaluation of smart meters within a legal metrology perspective: A brazilian case. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pages 1 –7.

Certicom (2000). *Standards for Efficient Cryptography 1: Elliptic Curve Cryptography*. Certicom Research, Ontario, Canada. Version 1.0.

Certicom (2004). Code and cipher. Certicom's Bulletin of Security and Cryptography, 1(3), 1 -5.

Chu, K.-C. (1996). Composite dictionary compression system. Patent, US 5530645, Estados Unidos.

Francisquini, A. A. (2006). Estimação de curvas de carga em pontos de consumo e em transformadores de distribuição. Master of electrical engineering, Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista Júlio de Mesquita Filho.

Gallagher, P., Foreword, D. D., and Director, C. F. (2009). FIPS PUB 186-3 federal information processing standards publication digital signature standard (DSS).

Guimarães, L. C. Audiência pública 012/2006. `http://www.aneel.gov.br/aplicacoes/audiencia/arquivo/2006/012/contribuicao/abradee_luiz_carlos_guimaraes.pdf`. Associação Brasileira de Distribuidores de Energia Elétrica. Acessado em Abril de 2012.

Infineon (2012). Electric metering – product brief. `http://www.infineon.com/smartmeter`. Acessado em Fevereiro de 2012.

Inmetro (2009). Portaria Inmetro nº 011 de 13 de janeiro de 2009. Instituto Nacional de Metrologia, Qualidade e Tecnologia.

ISO/IEC (2006). *ISO/IEC 9796-3:2006: Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms.* International Organization for Standardization, Geneva, Switzerland, second edition.

Modares, H. (2009). A scalar multiplication in elliptic curve cryptography with binary polynomial operations in galois field. Master of computer science, The Faculty of Computer Science and Information Technology, University Malaya.

OEB. Electricity prices. `http://www.ontarioenergyboard.ca/OEB/Consumers/Electricity/Electricity+Prices`. Ontario Energy Board. Acessado em Abril de 2012.

OIML (2009). OIML D31 requirements of software controlled measuring instruments. International Organization of Legal Metrology.

Treytl, A., Roberts, N., and Hancke, G. (2004). Security architecture for power-line metering system. In *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, pages 393 – 396.

VIM (2008). *International vocabulary of metrology — Basic and general concepts and associated terms*. Joint Committee on Guides for Metrology (JCGM), third edition.

WELMEC (2008). WELMEC 7.2 software guide – measuring instruments directive 200/22/ec. European Cooperation in Legal Metrology.

