Uma ICP baseada em certificados digitais autoassinados

Cristian Thiago Moecke[1], **Ricardo Felipe Custódio**[1]
Jonathan Gehard Kohler[1], **Marcelo Carlomagno Carlos** *[2]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Campus Universitário – Trindade – Florianópolis – SC – Brasil – CEP 88040-970

[2]Royal Holloway – University of London
London – UK

{cristiantm,custodio,jonathan}@inf.ufsc.br, marcelo.carlos.2009@rhul.ac.uk

Abstract. *Public Key Infrastructures have been used in some scenarios were there is a need to stabilish trust between two entities. Specially, its use is common in the stabilishment of trust to access the so called "secure websites" using SSL/TLS. However, with new iniciatives like ICP-Brasil (Brazilian PKI) and the growing use of digital certification to sign eletronic documents, some limitations of PKI have become more clear. This paper discusses the inversion of some concepts of Public Key Infrastructures (PKI) to simplify the process of digital signature validation. Changing the form that the user certificate is issued and modifying the responsibilities of a Certification Authorities by creating a Validation Authority, that also replaces the main function of Time Stamping Authority on digital signatures, we can reduce the effort spent on the process of digital signature validation. We also propose a simple protocol to interact with the Validation Authority.*

Resumo. *Infraestruturas de Chaves Públicas tem sido amplamente utilizadas em alguns contextos onde se deseja estabelecimento de confiança entre duas entidades. Em especial, seu uso é difundido no estabelecimento de confiança para acesso aos chamados "sites seguros", através do SSL/TLS. Entretanto, com o advento de iniciativas como a ICP-Brasil e o uso crescente da certificação digital para assinatura digital de documentos eletrônicos, algumas limitações e dificuldades de implementação de ICPs tornaram-se mais evidentes. Este artigo discute a inversão de alguns conceitos de Infraestruturas de Chaves Públicas (ICP) para simplificar o processo de validação de uma assinatura digital. Alterando a forma que um certificado digital é emitido e modificando as responsabilidades de uma Autoridade Certificadora através da criação da Autoridade de Validação, que também substitui a principal função da Autoridade de Carimbo de Tempo, podemos reduzir o esforço de validação de uma assinatura digital. Propomos também um protocolo para interação com a Autoridade de Validação.*

1. Introdução

Infraestruturas de Chaves Públicas (ICPs, também conhecidas amplamente pelo termo inglês *PKI – Public Key Infrastuctures*) são uma alternativa já consolidada para for-

*Apoiado pelo CNPq/Brazil

necer a capacidade de estabelecimento de relações de confiança entre entidades envolvidas em uma transação em meio digital. As ICPs tem sido amplamente utilizadas, por exemplo, no estabelecimento de confiança na navegação em sítios de internet (SSL/TLS) [Dierks and Rescorla 2008], e mais recentemente tem ganho espaço para a autenticação entre pessoas físicas e jurídicas. No Brasil, destaca-se a Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil) [CG ICP-Brasil 2010] que provê presunção de veracidade jurídica a documentos assinados digitalmente com certificados sob sua cadeia.

Entretanto, o crescimento do uso das ICPs nestes novos contextos trouxe a tona uma série de limitações e dificuldades relacionadas à sua implantação e uso. Concebidas para serem flexíveis, as ICPs são difíceis de implementar, necessitam de demasiados recursos humanos e computacionais para mantê-las e mesmo na presença desses recursos não conseguem prover serviços confiáveis a longo prazo. Por isso, o mercado acaba muitas vezes preferindo soluções alternativas mais eficientes, deixando de lado importantes requisitos de segurança prometidos pelas ICPs [Gutmann 2002].

Linn [Linn 2004] resume as razões que limitam o crescimento do uso das ICPs em três categorias:

- A pequena oferta de serviços que demandam os recursos oferecidos por ICPs;
- A forma que as ICPs atuais foram projetadas tornam-las difíceis de serem implementadas e dificultam a prestação de serviços pela ICP, tornando-a pouco atrativa diante de outras soluções alternativas existentes;
- A implantação de uma ICP implica em níveis de segurança muito mais elevados do que os que seriam apropriados ou com boa relação de custo/benefício em muitos contextos.

Nota-se que os dois últimos grupos de razões dizem respeito a características intrínsecas de uma ICP. A solução destes naturalmente minimizam o primeiro. Ou seja, a diminuição das dificuldades de implantação e uso, e o menor custo operacional de uma ICP tendem a tornar atraentes os recursos oferecidos por uma ICP.

Desde a descoberta da chaves públicas na década de 1970, vários modelos de ICPs tem sido propostos, destacando-se o PGP[Zimmermann 1995], X.509[Cooper et al. 2008], SPKI[Ellison et al. 1999b] e IBC[Boyen and Martin 2007]. O mais proeminente desses modelos é o X.509 por ter sido escolhido pelas empresas e governo para a emissão de certificados digitais para pessoas, instituições, sistemas e equipamentos. Apesar dos esforços em se implantar ICPs X.509, percebe-se que não é fácil sua implantação e que ainda há muitas dúvidas quanto aos serviços que podem ser ofertados, principalmente diante de um cenário de longo prazo.

Um dos aspectos de maior complexidade em ICPs é a construção e validação de caminho de certificação. Para a validação de um certificado é necessária: a identificação de um caminho de certificados até uma âncora de confiança; a obtenção destes certificados; a validação das respectivas assinaturas digitais e verificação das informações de expiração e situação de revogação de cada certificado deste caminho. A complexidade é maior quanto maior for o tamanho da cadeia de certificação.

Ainda há mais um ponto de complexidade envolvido. Usualmente uma assinatura digital de um documento eletrônico contém carimbos do tempo para fornecer uma evidência temporal confiável de que a assinatura foi produzida quando o caminho de

certificação era válido. Isso é fundamental para documentos eletrônicos cujas assinaturas devem ser verificáveis mesmo após a expiração ou revogação do certificado do signatário e até mesmo da sua cadeia de certificação. Entretanto o carimbo do tempo é, essencialmente, um documento assinado. E desta forma, tem os mesmos problemas e desafios da assinatura digital comum. A Figura 1 representa um documento assinado digitalmente sob uma ICP tradicional com carimbo de tempo.

A determinação e a validação do caminho de certificação é a fonte de muitos dos problemas das ICPs. Para a implantação de uma ICP que tratasse adequadamente estes problemas, uma série de soluções tiveram que ser desenvolvidas, que por sua vez aumentaram ainda mais a complexidade da validação dos certificados.

Além dos problemas já citados, há uma série de limitações de modelo de negócio impostas pela arquitetura tradicional. A hierarquia de ACs, inicialmente concebida para permitir a distribuição geográfica das atividades de emissão de certificados, mostrou-se extremamente cara devido aos custos de operação de cada AC. Para minimizar estes custos, a parte do processo de validação de dados dos titulares dos certificados digitais é repassada para Autoridades de Registro (AR). O peso da confiança sobre a identidade do detentor do certificado, portanto, recai sobre estas ARs.

Há ainda a questão da definição de quais são as âncoras de confiança. Em diferentes contextos distintas âncoras de confiança são consideradas válidas. Por exemplo, a AC Raiz ICP-Brasil não é aceita por outros países como válida. Para ser aceita, são necessários acordos internacionais, implementados tecnicamente por exemplo através de certificação cruzada entre ICPs, criando o que Gutmann chama de "spaghetti de dúvida" [Gutmann 2002], em alusão ao não determinismo da construção do caminho de certificação.

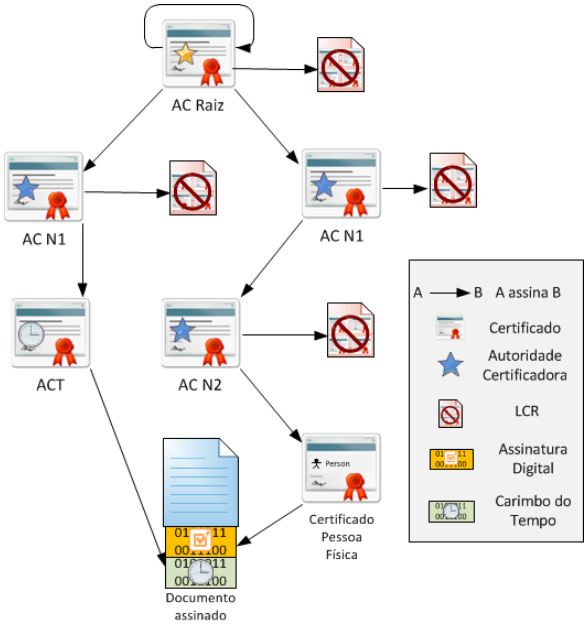

Figura 1. Um documento assinado em uma ICP típica

Este artigo propõe uma novo modelo de certificação digital, através do uso de certificados digitais autoassinados e a substituição das ACs finais por Autoridades de Validação (AV). Apesar de já existirem outros modelos de certificação que utilizam certificados autoassinados [Zimmermann 1995] ou de curta duração [Ellison et al. 1999a] e ainda sistemas de validação online [Freeman et al. 2007], o que é proposto neste trabalho é completamente diferente, conforme demonstraremos. Em sistemas como o PGP, o certificado é autoassinado, mas existe a necessidade da assinatura do certificado por terceiros, estabelecendo a chamada "teia de confiança".

Quando uma AC emite um certificado, pressupõe-se que ele seja válido por um

determinado período do tempo. Entretanto, isso não é correto uma vez que pode ser revogado a qualquer momento. Portanto, é necessária uma prova de que continua válido. Normalmente utilizam-se como prova listas de certificados revogados. Em virtude disso, defendemos que o certificado não deve ser considerado válido quando emitido, mas sim somente na presença da prova. Assim, não há necessidade do certificado ser emitido por uma autoridade certificadora. Por razões práticas e de simplicidade, propomos que os certificados sejam emitidos por seus titulares, ou seja, autoassinados. Isso quer dizer que não haveria mais uma autoridade certificadora para a emissão dos certificados para o usuário final. Propomos ainda que as provas sejam de curta duração, ou vinculadas às assinaturas digitais dos documentos eletrônicos, tornando desnecessário o uso de carimbos do tempo. Neste último caso a prova é válida para um determinado instante de tempo. Isso muda completamente o modelo de certificação e, conforme será arguido neste artigo, vários do problemas das ICPs tradicionais deixam de existir, além de tornar a ICP mais flexível e de menor custo de implantação. Tudo isso sem prejuízo dos serviços esperados por ICPs.

A Seção 2 apresenta os trabalhos correlatos. A Seção 3 discute nossa abordagem e na seção 4 o protocolo é descrito. A Seção 5 discute benefícios e limitações da proposta. E por fim, concluímos com as considerações finais em 6.

2. Modelos de certificação digital e suas dificuldades

Os modelos tradicionais de certificação mais conhecidos e usados são: PGP, SPKI/SDSI, IBC e X.509.

O PGP [Zimmermann 1995] também apresenta uma arquitetura de ICP onde são utilizados certificados digitais autoassinados. A confiança nestes certificados é estabelecida através das "redes de confiança", ou seja, os usuários do PGP assinam os certificados em quem confiam e desta forma pode-se estabelecer em quais certificados pode-se ou não confiar. Para determinados contextos a abordagem é útil, mas não é escalável, pois torna-se muito fácil qualquer nodo mal intencionado da rede comprometer a confiabilidade do sistema.

O SPKI [Ellison et al. 1999b] propõe uma arquitetura simplificada de ICP, mais voltada a autorização do que a autenticação [Ellison et al. 1999a]. Um certificado de SPKI (*Simple Public Key Infrastructure*) são mais centrados na chave e que autorização essa chave tem, do que entre a chave e uma entidade (apesar de prover suporte a isso). A limitação está no próprio objetivo com que foi desenvolvido: embora mais simples para autorização, é limitado para uso em autenticação, e em especial para assinatura digital. Na mesma linha está o SDSI (*Simple Distributed Security Infrastructure*) [Rivest and Lampson 1996], outra abordagem simplificada de ICP onde o foco está na definição de grupos de acesso e emissão de certificados para estes grupos.

A IBC (*Identity Based Criptography*) [Boyen and Martin 2007] é uma abordagem onde as chaves não são geradas randômicamente, mas sim calculadas. A chave pública é calculada a partir de uma identidade e a interação com um Gerador de Chaves, e a chave privada é gerada a partir da chave pública e a interação com este mesmo Gerador de Chaves. Nota-se que existe uma forte dependência de confiança com o Gerador de Chaves.

O X.509 foi proposto como certificado digital em diretórios X.500. Esse modelo

de certificação consiste em uma ou mais árvores de certificados, cada uma com sua raiz, e autoridades certificadoras finais que emitem certificados para os usuários.

Todos esses modelos de certificação digital apresentam uma série de dificuldades que precisam ser tratadas por serviços auxiliares. A maior parte das dificuldades estão relacionadas a validação do certificado do usuário, tanto para determinação e validação do seu caminho de certificação quanto da gestão de revogação dos certificados.

O maior problema quanto a revogação está na busca constante de LCRs atualizadas e o tamanho destas listas. Para contornar estes problemas, Kocher propôs a Árvore de Revogação de Certificados (*Certificate Revocation Tree – CRT*) [Kocher 1998], que provê respostas de tamanho levemente menor a consultas sobre revogação de certificados. O problema é minimizado, mas não eliminado. Outra alternativa é a validação de Certificados em Árvore-Lista (*Tree-List Certificate Validation – TLCV*) [Lim et al. 2008] que usa uma estrutura do tipo árvore-lista para promover um ganho de performance sobre a proposta de Kocher.

Quanto a complexidade da construção da cadeia de certificação Levi propôs o uso de certificados aninhados (*Nested-certificate-based PKI – NPKI*) [Levi et al. 2004]. Tais certificados criam uma infraestrutura especial ligando o ponto de confiança ao certificado do usuário através de uma lista de hashes. A idéia é substituir operações criptográficas complexas por operações de cálculo de hash.

Uma outra proposta para resolver esta mesma questão é a apresentada por Custódio [Custódio et al. 2008], que propõe que o problema da validação de assinaturas digitais pode ser resolvido usando um tipo especial de certificado, chamado Certificado Otimizado (*Optimized Certificates – OC*). Esta proposta minimiza o número de operações aritméticas na validação de um certificado. Além disso outro problema, também tratado por Custódio, está na conservação a longo prazo de documentos eletrônicos, que consiste na emissão periódica de carimbo de tempo que leva ao aumento do tamanho do arquivo de assinatura digital.

Outra dificuldade desses modelos tradicionais de certificação é a necessidade de muitos recursos e poder de processamento para validação do caminho de certificação. Vários dispositivos atuais, tais como celulares e PDAs não dispõem de recursos e/ou informações suficientes para efetuar esta operação. Para contornar essa dificuldade são usados serviços especializados tal como o SCVP [Freeman et al. 2007]. O SCVP é uma serviço (terceira parte) que pode ser invocado para construção e/ou validação de cadeias de certificação em uma ICP usual. Contudo este esquema adiciona mais uma parte confiável no sistema, sujeito a alta demanda de carga, e não elimina os problemas existentes na ICP tradicional, pois apenas delega a outra entidade a resolução do desafio.

Outras pesquisas tem proposto diferentes abordagens [Laih and Yen 1995, Satizábal et al. 2007, Rivest 1998, Cooper 1999, Hunter 2002]. Todas tentam melhorar o tempo de resposta das consultas a informações de revogação, criar mecanismos mais eficientes de validação, ou até mesmo eliminar a necessidade de revogação de certificados. Entretanto nenhuma das propostas tenta mudar os fundamentos da validação de certificados. Nosso trabalho seguirá uma abordagem diferente, propondo uma nova modelagem de ICP, adequada para assinatura digital de documentos, sem perder a aplicabilidade para autenticação e demais usos de uma ICP tradicional.

3. Invertendo os paradigmas da validação de certificados

Como vimos, no esquema tradicional de certificação digital, quando um certificado é emitido, ele é assumido como válido por um determinado tempo. Entretanto, certificados digitais podem ser revogados a qualquer momento. Portanto, ele não é válido até que seja obtida uma prova fornecida por uma terceira parte confiável de que ele continua válido. Usualmente usam-se LCRs como prova.

Então, não é necessário pressupor que o certificado seja válido quando emitido. O certificado pode ser emitido sem ser considerado válido e somente quando for necessário busca-se a prova que o torna válido.

Assim, propomos que o certificado do usuário só seja considerado válido quando houver uma prova de que ele é válido. Partindo do pressuposto que isso é uma característica do modelo, ou seja, sempre é preciso obter uma prova da validade do certificado, não há a necessidade que o certificado seja emitido por uma Autoridade Certificadora. A prova de sua validade pode fornecer a evidência necessária para a comprovação das informações do certificado. Neste modelo não há caminho de certificação, e todos os problemas relacionados a isso são automaticamente eliminados.

Na nossa proposta, o usuário gera seu próprio certificado autoassinado e realiza uma autenticação segura com uma Autoridade de Registro (AR) para provar sua identidade e posse da chave privada. A AR verifica os dados do certificado e a posse da chave, e os envia para uma terceira parte confiável, denominada de Autoridade de Validação (AV), através de uma mensagem que vincula o certificado ao usuário. A AV é responsável por emitir provas, quando solicitado, de que o certificado do usuário é válido num determinado instante de tempo.

A autenticação segura deve ser feita de forma presencial. Neste modo de autenticação, o usuário precisa ir a uma instalação técnica da AR para se apresentar, provar (através de documentos) que ele é quem alega ser e provar a posse da chave privada ou dizer-se responsável pela mesma. O certificado autoassinado será então assinado pela AR e enviado para uma ou mais Autoridades de validação.

Como não existe uma AC para assinar o certificado do usuário, então este não precisa mais esperar por sua emissão. O certificado pode ser emitido a qualquer momento por conta própria, e o usuário apenas precisa ir até a AR quando desejar que seu certificado possa ser validado por terceiros. As assinaturas feitas pelo usuário a partir do momento que uma AV passa a responder por sua validação podem ser verificadas.

No nosso modelo, o certificado autoassinado é usado apenas para distribuição da chave pública correta do usuário e para vincular a mesma ao seu detentor. Não há cadeia de certificação entre este e a âncora de confiança, já que isto não é necessário. A prova de validade evidenciará a validade do certificado. Com estas modificações, o verificador não precisa mais construir o caminho de certificação para verificar a integridade e posse do par de chaves do signatário.

Quando o verificador precisar validar a integridade de um certificado de usuário, este precisa obter uma prova de validade de uma AV. Esta prova pode ser obtida pelo detentor do certificado e enviada ao verificador, ou solicitada pelo próprio verificador. Quando recebe uma requisição de verificação de validade, a AV verifica a situação do cer-

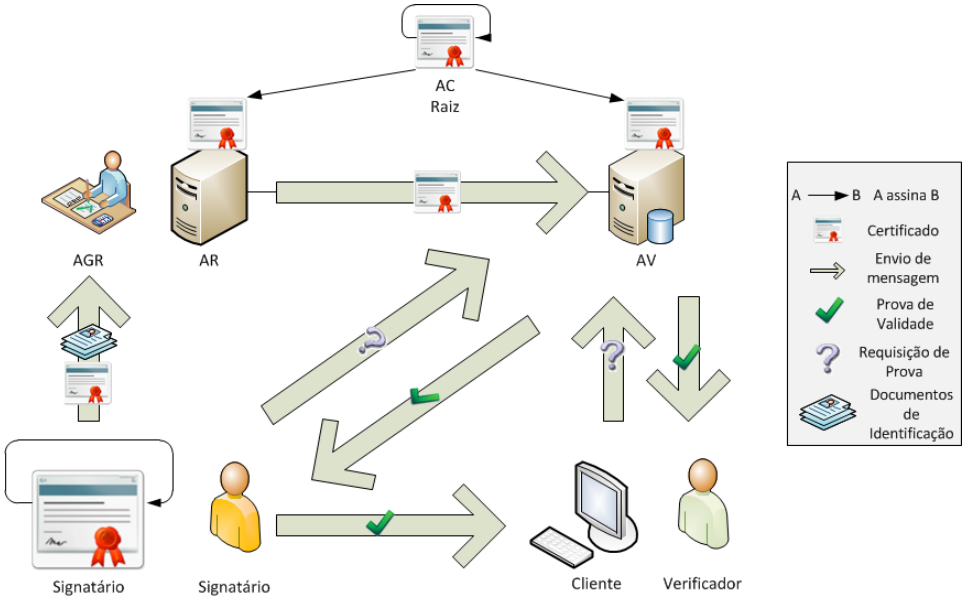

Figura 2. Possíveis mensagens envolvidas na validação de um certificado

tificado em questão na sua base de dados e retorna um *token*, a prova, com a situação do certificado. Este *token* contém a prova de que o certificado é válido por um determinado período de tempo. Propõe-se que o período de validade do *token* seja o mais curto possível, desta forma pode-se dispensar a utilização de um mecanismo de revogação para validar o *token*, conforme argumentam os trabalhos de Rivest e Elisson [Rivest 1998, Ellison et al. 1999a]. A Figura 2 ilustra as possíveis mensagens envolvidas na validação de um *token*.

Entretanto, o *token* é um documento assinado pela AV, que por sua vez possui um certificado emitido por uma AC Raiz. A AC Raiz, além de emitir os certificados das AVs, também emite os certificados das ARs. Essa é a única hierarquia que existe. Como a AR só assina mensagens destinadas a AVs, não haveria maiores problemas em se manter o esquema de LCRs para verificar, por parte da AV, se um certificado de AR continua válido. Por outro lado, os verificadores precisam verificar a assinatura dos *tokens*, estes assinados pelas AVs. Como esses *tokens* são válidos por um curto período de tempo, e quanto emitidos eram válidos, pode-se considerar com razoável margem de segurança, conforme defendido por Rivest, que esses sempre serão válidos e não necessitariam ser revogados. O conceito é o mesmo usado por Carimbos do Tempo [Adams et al. 2001].

Há ainda o certificado da AV. Para evitar o uso de LCRs como prova de que o certificado da AV é válido, propõe-se usar o as provas do Novomodo do protocolo de Micali [Micali 1995], conforme descrito por Custódio [Custódio et al. 2008], o qual implica na inserção de uma prova de validade do certificado dentro do *token* emitido. Tal prova, produto de uma lista de hashes, é produzido diretamente pela AC Raiz.

A mesma técnica poderia ser usada para provar a validade dos certificados das ARs. Assim, eliminamos a lista de certificados revogados de nosso modelo de certificação. E com isso, elimina-se a maior parte dos problemas que essas trazem para a ICP.

Além da própria simplificação do processo de validação de certificados, propomos que a AV possa atuar também como uma fornecedora de prova de confiabilidade temporal para assinaturas digitais de longo prazo. Para melhor analisar a questão da aplicação deste modelo em cada situação, classificamos o uso dos certificados em assinaturas digitais em dois tipos: as de curta duração (comumente usadas para autenticação) e as de longa duração (comumente usadas para assinatura de documentos). A aplicação da proposta em ambos os casos é melhor detalhada nas próximas duas sessões.

3.1. Autenticação

Para processos de autenticação, o *token* emitido pela AV apenas prova que o certificado é válido para um determinado (e curto) período. Esta opção pode ser útil para assinaturas de curto prazo e especialmente mecanismos de autenticação. Um único *token* pode ser utilizado inúmeras vezes para quantas autenticações forem necessárias, sem necessidade de obtenção de mais dados externos pelo verificador.

Para uma autenticação, a aplicação do usuário obtém o *token* de curta duração com a AV que detém prova que seu certificado é válido. Após isso, a aplicação do usuário envia o certificado e o *token* para o servidor de autenticação. O servidor pode então confirmar os dados do usuário no certificado, bem como a validade do mesmo através do *token*. Se a aplicação cliente não tem recursos suficientes para enviar o *token*, o servidor pode realizar o procedimento de obtenção do *token* por conta própria. A aplicação cliente apenas envia o certificado, e o servidor consulta uma autoridade de validação para confirmar a validade do mesmo.

3.2. Assinaturas de Documentos

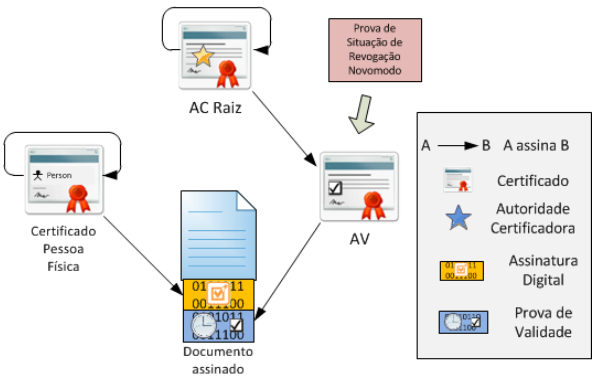

Figura 3. Um documento assinado sob nossa abordagem de ICP

Quando um usuário assina um documento, a aplicação pode encaminhar à AV o certificado, o resumo criptográfico do documento e a assinatura. A AV verifica a validade do certificado e da assinatura, e retorna para a aplicação um *token* de validação de assinatura. Este *token* fornece uma prova de validade não apenas da certificado, mas também da assinatura. A sessão 4.2 detalha a estrutura proposta para este *token*. O *token* pode ser usado para provar que determinada assinatura era válida para a data em que foi gerado.

O *token* é um atributo não-assinado da assinatura digital e é apenas útil quando alguém quer validar a assinatura de um documento eletrônico. Note que se o signatário não anexar o *token*, o primeiro verificador pode fazê-lo. Se o certificado for válido quando esta solicitação é feita, a AV envia o *token* ao verificador que pode anexa-lo à assinatura. Um próximo verificador ao receber o documento assinado poderá verificar a validade desta sem mais consultas externas.

Na verificação de assinatura nos modelos tradicionais de certificação digital, o verificador precisa primeiro verificar a integridade do carimbo do tempo e depois a integridade e validade da assinatura digital. Para verificar a integridade do carimbo do tempo, o usuário precisa construir e validar o caminho de certificação entre o certificado da Autoridade de Carimbo do Tempo (ACT) emissora do carimbo do tempo e a âncora de confiança. E depois, construir e validar o caminho de certificação do certificado do signatário. Como o modelo de certificação que está sendo proposto é preciso somente verificar a validade do *token*. Se o *token* for válido, o verificador verifica a integridade da chave pública do signatário, mas sem precisar construir ou validar cadeias de certificação, uma vez que trata-se de um certificado autoassinado. E ainda, devido a presença do hash e da assinatura do documento no *token*, não precisa realizar a operação de decifragem do hash. Precisa apenas comparar o hash do documento incluído no *token* com o hash do documento assinado. Ainda existe a possibilidade do usuário verificar a aplicação da chave privada do signatário à assinatura, caso deseje.

Se a assinatura do *token* é válida, então a assinatura era válida na âncora de tempo especificada pelo *token*. Para verificar a validadade do *token*, o usuário usa a abordagem comum, montando e validando o caminho de certificação da AV, que utiliza o método de Micali, conforme descrito por Custódio [Custódio et al. 2008], para sua validação. Após isso, o usuário verifica a assinatura do *token* e está pronto para determinar a sua confiança no documento.

A mesma assinatura digital pode ser validada por mais de uma AV. O verificador pode validar a situação do certificado em qualquer uma das AVs que receberam informações da AR.

3.2.1. Manutenção a longo prazo da assinatura

Para a conservação da validade de uma assinatura, é necessário obter provas atualizadas emitidas por novas AVs, antes que a prova anterior expire. Neste sentido, o comportamento da AV é bem parecido com o de uma ACT do modelo tradicional. Entretanto, é importante destacar que não é necessário adicionar um novo *token*, mas sim substituir o antigo. A AV, ao receber um *token* para ser revalidado, confirma as informações do *token* antigo e gera um novo *token*, com as mesmas informações do anterior, mas com sua assinatura.

No modelo tradicional de ICP, para a conservação de longo prazo de um documento eletrônico, é necessário sempre adicionar novos carimbos do tempo sobre os carimbos do tempo anteriores, antes que estes percam sua validade. Cabe destacar que se comprometida a Autoridade de Carimbo do Tempo, todos os carimbos por ela já emitidos tornam-se inválidos, caso não exista um segundo carimbo contraposto ao primeiro. Isto é um problema, pois não há como prever a violação de uma ACT. Essa abordagem leva ao crescimento contínuo do arquivo de assinatura.

No nosso modelo a assinatura tem sempre o mesmo tamanho. E também possibilita a atualização de algoritmos de maneira extremamente simples. Os novos *tokens* podem fazer uso de algoritmos mais seguros e assim manter a confiabilidade da validação mesmo após a quebra dos algoritmos anteriormente usados.

4. Protocolo Proposto

Apresentamos a seguir a estrutura básica para a proposta de um protocolo de requisição de provas de validade para uma AV.

4.1. Requisição de Prova de Validade

A Figura 4 apresenta a estrutura ASN.1 da requisição.

```
ValidityProofRequest ::= SEQUENCE {
    version            INTEGER { v1(1) },
    signerCertificateDigestAlgorithm
                       AlgorithmIdentifier
    signerCertificateDigest   OCTET STRING
    document           DocumentInfo OPTIONAL }

DocumentInfo ::= SEQUENCE {
    digestAlgorithm    AlgorithmIdentifier,
    documentDigest     OCTET STRING,
    documentSignature  BIT STRING         }
```

Figura 4. Estrutura ASN.1 da Requisição

O campo *version* identifica a versão da requisição. A versão atual é v1.

O campo *signerCertificateDigest* contém um identificador único do certificado do usuário, correspondente ao resumo criptográfico do certificado. O algoritmo utilizado é identificado em *signerCertificateDigestAlgorithm*.

O campo *document* é composto por três sub-campos: *hashAlgoritm*, *documentDigest* and *documentSignature*. Estes três campos são apenas utilizados quando é solicitado um *token* de assinatura de documento. Quando um *token* de autenticação é solicitado estes campos não devem ser preenchidos.

O campo *digestAlgorithm* identifica o algoritmo utilizado para gerar o resumo criptográfico do documento. O campo *documentDigest* contém o valor do hash do documento assinado. E o campo *documentSignature* contém a assinatura digital do documento.

4.2. Resposta de Validade

A Figura 5 apresenta em ASN.1 o *token* de resposta de validade do certificado. O campo *version* identifica a versão da requisição. A versão atual é v1.

O campo *tokenValidity* indica o intervalo de tempo em que a AV garante que o certificado descrito no campo *userCert* está de acordo com a situação definida no campo *userCertStatus*. Este campo é composto por duas datas: a data em que a validade do *token* inicia (*notBefore*) e a data em que a validade do *token* encerra (*notAfter*).

O campo *signerCertificateDigest* contém um identificador único do certificado do usuário, correspondente ao resumo criptográfico do certificado. O algoritmo utilizado é identificado em *signerCertificateDigestAlgorithm*. O campo *signerCertificateStatus* é um código numérico que informa a situação do certificado do usuário, e assinatura quando aplicável. Os possíveis estados são:

- valid (0) - O certificado é válido entre as datas especificadas em notBefore e notAfter, e a assinatura foi corretamente validada, se aplicável.
- revoked (1) - O certificado foi revogado.
- expired (2) - O certificado expirou.
- error (3) - Ocorreu um erro processando o estado do certificado. Mais informações sobre o erro são descritos no campo *responseStatus*

O campo *validPolicies* é uma sequência de OIDs que representam políticas sob as quais o certificado é válido. Se não existir uma política válida, o campo tem valor nulo.

O campo *signatureProof* é composto por uma sequencia de dois campos: *digestAlgorithm* e *signatureHash*. O primeiro descreve o algoritmo utilizado para computar o resumo criptográfico da assinatura encaminhada na requisição. O segundo contém o valor do resumo criptográfico da assinatura enviada na requisição.

O campo *signatureProof* é preenchido se a requisição contém o campo *documentInfo*. Neste caso o servidor retorna uma prova que a assinatura era válida numa data específica. A assinatura do *token* prova que o certificado usado para assinar um arquivo específico era válida em uma data específica. Este *token* deve ser anexado a uma assinatura para simplificar o processo de verificação da validade da assinatura no futuro. Se o campo *documentInfo* não é preenchido na requisição, então o servidor retorna uma prova que o certificado é válido durante um determinado período. Este período é definido pelos campos *notBefore* e *notAfter*. Este *token* pode ser anexado a assinaturas para provar que um determinado certificado é válido e as assinaturas feitas por ele são válidas até que o *token* expire.

O campo *vaCertificate* contém o certificado da AV codificado em DER. A situação deste certificado é obtida pela utilização parcial do método de Novomodo. A prova de novomodo é armazenada em *vaNovomodoProof*, e permite identificar se a AV foi ou não revogada. A AV obtém esta prova da AC Raiz e insere no *token*.

```
ValidityProoftoken ::= SEQUENCE {
    version              INTEGER { v1(1) },
    tokenValidity        Validity,
    signerCertificateDigestAlgorithm
                         AlgorithmIdentifier
    signerCertificateDigest  OCTET STRING
    signerCertificateStatus   SigCertStatus,
    validPolicies        ValidPolicies,
    signatureProof       SignatureProof OPTIONAL,
    vaCertificate        Certificate    OPTIONAL,
    vaNovomodoProof      BIT STRING,
    responseStatus       ResponseStatus,
    signatureAlgorithm   AlgorithmIdentifier,
    signatureValue       BIT STRING     }

SigCertStatus ::= ENUMERATED {
    valid               (0),
    revoked             (1),
    expired             (2),
    error               (4)     }

ResponseStatus ::= ENUMERATED {
    successful          (0),
    invalidRequest      (1),
    internalError       (2),
    tryLater            (3),
    unknownCertificate  (4),
    badDigestAlgorithm  (5),
    unsupportedVersion  (6) }

ValidPolicies ::= SEQUENCE {
    policyIdentifier    OBJECT IDENTIFIER }

SignatureProof ::= SEQUENCE {
    digestAlgorithm   AlgorithmIdentifier
    signatureHash   BIT STRING   }

Validity ::= SEQUENCE {
    notBefore   Time,
    notAfter    Time   }

AlgorithmIdentifier  ::=  SEQUENCE  {
     algorithm        OBJECT IDENTIFIER,
     parameters       ANY DEFINED BY algorithm
                                 OPTIONAL  }
```

Figura 5. Estrutura ASN.1 do *token*

O campo *responseStatus* fornece informação da situação sobre a resposta de requisição. Trata-se de um código numérico, com os seguintes significados:

- successful - A requisição foi corretamente processada
- invalidRequest - Requisição inválida
- internalError - O servidor apresentou problemas internos
- tryLater - O servidor estava ocupado demais para atender o pedido

- unknownCertificate - O certificado da requisição não é conhecido pela AV
- badDigestAlgorithm - O algoritmo usado para resumo criptográfico não é conhecido pelo servidor
- unsupportedVersion - A versão de requisição não é suportada

O campo *signatureAlgorithm* representa o algoritmo de assinatura utilizado pela AV para assinar o *token*, codificado de acordo com as regras associadas com o valor do campo *signatureAlgorithm* de um certificado digital [Cooper et al. 2008]. E o campo *signatureValue* contém a assinatura digital de todos os demais campos do ASN.1 codificado em DER.

5. Benefícios e Limitações

Com o *token* de autenticação, o usuário pode usar o mesmo *token* várias vezes para se autenticar em um sistema, e o servidor não precisa com isso obter novas informações de revogação com tanta frequência. O usuário é quem envia a prova de validade ao servidor, distribuíndo assim a carga de validação que estava exclusivamente sobre o servidor de autenticação entre os usuários interessados em autenticar-se.

Como o *token* de autenticação é pequeno, o servidor pode armazenar esses *tokens* (*cache*) durante a validade do *token*. E numa próxima autenticação, o servidor não precisará mais solicitar um *token* de validade. A validação do token depente apenas da confiança na AV, e não em uma série de informações da cadeia de certificação e situação dos certificados.

Com o *token* de assinatura, não é necessária a validação de um carimbo do tempo e da assinatura digital do documento. Como o usuário precisa enviar à AV a assinatura e o resumo criptográfico do documento assinado, a AV irá também validar a assinatura digital. Desta forma, o verificador apenas precisa validar a assinatura do *token*. Se a assinatura do *token* é válida, então o verificador apenas verifica as informações do *token*, que permite verificar quando e por quem o documento foi assinado, e compara o hash contido no *token* com o hash do documento assinado. Isso significa que o verificador não precisa realizar uma operação criptográfica assimétrica para verificar a assinatura do documento.

A maior limitação da proposta é a mesma de uma Autoridade de Carimbo do Tempo [Adams et al. 2001], que é a de que se uma chave de ACT for comprometida, todos os carimbos do tempo deixam de ser válidos. De igual modo, se uma AV é comprometida, todos *tokens* por ela emitidos deixam de ser confiáveis. Por esta razão, é extremamente importante que as chaves privadas da AV sejam protegidas com segurança apropriada. No caso da violação da chave, a única forma de distinguir entre *tokens* válidos e inválidos seria com a auditoria da AV. Entretanto, cabe destacar que a manutenção dos *tokens*, através da substituição destes por novos emitidos por novas AVs, também é bastante simplificado, facilitando a manutenção da verificabilidade da assinatura digital.

6. Considerações Finais

Este artigo propôs um novo modelo de certificação que visa reduzir a dificuldade de validação de uma assinatura digital. Neste novo modelo sugerimos que o certificado do usuário deve ser autoassinado, e a Autoridade Certificadora seja substituída por uma Autoridade de Validação. Esta última é responsável pela emissão de *tokens* que servem como

prova de validade do certificado do usuário. Com este *token* não é mais necessário montar e validar o caminho de certificação do usuário nem utilizar uma Autoridade de Carimbo do Tempo para produzir uma assinatura de longo prazo.

O novo modelo traz uma série de vantagens com relação aos modelos de certificação anteriores, ao eliminar a cadeia de certificação do usuário e assumir como parte do modelo a obtenção de provas de validade do certificado. Conforme proposto, o modelo também apresenta características importantes, como facilitar a conservação de longo prazo de documentos, inclusive no que diz respeito a atualização de algoritmos criptográficos. A abordagem definida no artigo também reduz a quantidade de código a ser implementado em um verificador de assinaturas digitais, e pode acelerar o desenvolvimento de aplicações baseadas em ICP, em especial para dispositivos com recursos limitados como sensores e telefones móveis.

Ainda existem aspectos a serem desenvolvidos em trabalhos futuros, como por exemplo uma identificação ou ponteiro no certificado para que um verificador saiba como consultar a AV apropriada. Sugere-se por exemplo um ponteiro para a AR, que por sua vez poderia apresentar uma lista de AVs com quem tem relação de confiança. Além disso, cabe ainda a aplicação do presente modelo em outros protocolos existentes na literatura e análise dos benefícios e limitações do modelo nestes cenários. Os autores veem na evolução da abordagem proposta a possibilidade da solução de diversos problemas do modelo atual de ICP, sem perder a funcionalidade do modelo tradicional.

Esse modelo proposto é aderente aos padrões e métodos dos modelos de negócio atuais, utilizando serviços online para validação e autenticação [Freeman et al. 2007, Gutmann 2002].

Referências

Adams, C., Cain, P., Pinkas, D., and Zuccherato, R. (2001). Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161 (Proposed Standard). Updated by RFC 5816.

Boyen, X. and Martin, L. (2007). Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems. RFC 5091 (Informational).

CG ICP-Brasil (2010). Infraestrutura de Chaves Públicas Brasileira. http://www.icpbrasil.gov.br.

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. (2008). Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (Proposed Standard).

Cooper, D. A. (1999). A Model of Certificate Revocation. In *ACSAC '99: Proceedings of the 15th Annual Computer Security Applications Conference*, page 256, Washington, DC, USA. IEEE Computer Society.

Custódio, R., Vigil, M., Romani, J., Pereira, F., and da Silva Fraga, J. (2008). *Optimized Certificates – A New Proposal for Efficient Electronic Document Signature Validation*, volume 5057 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Dierks, T. and Rescorla, E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard). Updated by RFCs 5746, 5878.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T. (1999a). RFC2693: SPKI Certificate Theory. *RFC Editor United States*.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T. (1999b). SPKI Certificate Theory. RFC 2693 (Experimental).

Freeman, T., Housley, R., Malpani, A., Cooper, D., and Polk, W. (2007). Server-Based Certificate Validation Protocol (SCVP). RFC 5055 (Proposed Standard).

Gutmann, P. (2002). PKI: It's Not Dead, Just Resting.

Hunter, B. (2002). Simplifying PKI usage through a client-server architecture and dynamic propagation of certificate paths and repository addresses. In *Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on*, pages 505–510.

Kocher, P. C. (1998). On Certificate Revocation and Validation. In *FC '98: Proceedings of the Second International Conference on Financial Cryptography*, pages 172–177, London, UK. Springer-Verlag.

Laih, C.-S. and Yen, S.-M. (1995). Improved Digital Signature Suitable for Batch Verification. *IEEE Transactions on Computers*, 44:957–959.

Levi, A., Caglayan, M. U., and Koc, C. K. (2004). Use of nested certificates for efficient, dynamic, and trust preserving public key infrastructure. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):21.

Lim, T.-L., Lakshminarayanan, A., and Saksen, V. (2008). A practical and efficient tree-list structure for public-key certificate validation. In *ACNS'08: Proceedings of the 6th international conference on Applied cryptography and network security*, pages 392–410, Berlin, Heidelberg. Springer-Verlag.

Linn, J. (2004). An Examination of Asserted PKI Issues and Proposed Alternatives. *Proceedings of the 3rd Annual PKI R&D Workshop*.

Micali, S. (1995). Enhanced Certificate Revocation System. *Massachusetts Institute of Technology, Cambridge, MA*, pages 1–10.

Rivest, R. L. (1998). Can We Eliminate Certificate Revocations Lists? In *FC '98: Proceedings of the Second International Conference on Financial Cryptography*, pages 178–183, London, UK. Springer-Verlag.

Rivest, R. L. and Lampson, B. (1996). SDSI - A Simple Distributed Security Infrastructure.

Satizábal, C., Martínez-Peláez, R., Forné, J., and Rico-Novella, F. (2007). Reducing the Computational Cost of Certification Path Validation in Mobile Payment. In *EuroPKI '07: Proceedings of the 4th European PKI workshop: Theory and Practice on Public Key Infrastructure*, pages 280–296, Berlin, Heidelberg. Springer-Verlag.

Zimmermann, P. R. (1995). *The official PGP user's guide*. MIT Press, Cambridge, MA, USA.