UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Alexsandra Carvalho da Silva

UMA SOLUÇÃO DE ASSINATURA DIGITAL CURTA ESPECIAL BASEADA EM UMA VARIAÇÃO DO DSA GERADA EM DISPOSITIVO PESSOAL

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Luiz Carlos Zancanella, Dr.
Orientador

Florianópolis, setembro de 2006

UMA SOLUÇÃO DE ASSINATURA DIGITAL CURTA ESPECIAL BASEADA EM UMA VARIAÇÃO DO DSA GERADA EM DISPOSITIVO PESSOAL

Alexsandra Carvalho da Silva

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Raul Sidnei Wazlawick, Dr.
Coordenador do Curso

Banca Examinadora

_____

Luiz Carlos Zancanella, Dr.
Orientador

_____

Carlos Barros Montez, Dr.

_____

Daniel Santana de Freitas, Dr.

_____

Ricardo Felipe Custódio, Dr.

Aos meus pais Antônio e Fátima.
Aos meus irmãos Maurício, Soraia e seu esposo Willian.
Ao meu noivo Paulo e seus pais José Paulo e Conceição.
A todos os meus familiares e amigos
que estão sorrindo comigo diante dessa nova conquista.

AGRADECIMENTOS

É uma tarefa difícil conseguir citar todos que de alguma maneira contribuíram para o desenvolvimento do presente trabalho. Porém, deixo registrado aqui meus agradecimentos:

- A Deus, pela Graça dessa conquista que sem dúvida foi por Ele abençoada;
- Ao meu orientador Prof. Luiz Carlos Zancanella, por me selecionar para ser sua orientanda, por me direcionar nos caminhos do Mestrado e por entender meus momentos. Suas orientações, explicações e compreensão foram decisivas para o êxito desse trabalho;
- Ao Prof. Daniel Santana de Freitas, pelas orientações sábias e indispensáveis e pela disseminação de idéias em nossas várias reuniões e troca de e-mails;
- À WMW Systems, empresa onde trabalho há 6 anos e que me proporcionou todo o apoio necessário para que o Mestrado fosse uma realidade;
- Ao Robson da Silva Rodrigues e Andrey Morais Brüggemann, que implementaram o protótipo para validação da presente proposta. Ao Marcos de Souza Marcon, que contribuiu na edição de figuras e interface do protótipo. Foi simplesmente essencial o trabalho de vocês;
- Ao Programa de Pós-Graduação em Ciência da Computação desta Universidade e à secretária Vera Lúcia Sodré Teixeira, pelo apoio concedido durante o Mestrado.

A todos que deram sua contribuição e que não se encontram nessa lista, meus sinceros sentimentos de gratidão.

RESUMO

O presente trabalho apresenta uma solução para geração de assinatura digital curta com características especiais, baseada em uma variação do algoritmo DSA. A assinatura é gerada com apenas 32 bits em um dispositivo pessoal que opera sem conectividade com o computador, sendo curta o suficiente para ser informada manualmente pelo usuário, através do teclado, para validação. A aparente vulnerabilidade ocasionada pela redução no tamanho da assinatura é contornada por um protocolo criptográfico especialmente projetado. O contexto da solução aqui apresentada visa garantir, através da assinatura digital, o não-repúdio nas transações bancárias realizadas de forma eletrônica pelo correntista através do *Internet Banking*.

Palavras-Chave: assinatura digital curta; autenticação do usuário; DSA.

ABSTRACT

This work presents a solution for generation of short digital signature with special features based on a variation of the DSA algorithm. The signature, generated in a personal device without any connectivity with the client's computer, is only 32-bit long, being short enough to be manually informed by a human through a keyboard for validation. The apparent weakness caused by the exceptionally large reduction in the size of the signature is overcome by a specially designed cryptographic protocol. The context of our solution aims to assure, through the use of digital signatures, the non-repudiation requirement in Internet Banking transactions.

Key-Words: short digital signature; user authentication; DSA.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Historicamente, a tecnologia de autenticação remota no acesso a sistemas tem apresentado soluções pouco satisfatórias. Nos últimos anos, a crescente onda de ataques a sistemas de *Internet Banking* tem provocado uma concentração de esforços e recursos em busca de soluções que apresentem alto grau de segurança e custos reduzidos.

Basicamente, as necessidades de autenticação visam satisfazer duas situações distintas: *Autenticações de curto prazo,* utilizadas quando a necessidade é verificar a presença física de alguém em um instante específico e *Autenticações de longo prazo*, quando se faz necessário provar um ato praticado, mesmo após passagem de longo tempo do evento ocorrido.

Nos sistemas de *Internet Banking*, até bem pouco tempo o único fator de autenticação utilizado nas transações era um segredo associado à conta corrente do titular. Mais recentemente, o uso de um segundo fator de autenticação tem sido utilizado para aumentar o nível de segurança na identificação do correntista. Uma solução amplamente utilizada hoje como segundo fator de autenticação no acesso ao *Internet Banking* está baseada em sincronismo de tempo. Produzidos em *hardware*, esses dispositivos atendem satisfatoriamente os requisitos de autenticação de curto prazo, mas em nada contribuem para autenticação de longo prazo.

Autenticações de longo prazo possuem importância significativa para os bancos, visto representar a possibilidade do banco ter condições de certificar se transações ocorridas em algum momento do passado foram efetivamente realizadas pelo titular da conta. Qualquer um entre os três fatores de identificação[1] pode ser utilizado para autenticação de curto prazo. Todavia, autenticação de longo prazo somente pode ser obtida através do fator *algo que se tem*, relacionado a um par de chaves assimétricas para ser utilizado na geração de assinaturas digitais.

Soluções de uso de assinatura digital no acesso ao *Internet Banking* são de ampla aceitação e têm conquistado espaço em grandes bancos. A solução mais comumente adotada é o uso de certificado digital instalado no próprio computador do usuário ou em dispositivos do tipo *smart cards*. Instalar certificados digitais em computadores não apresenta nível de segurança aceitável para uso em *Internet Banking*. A alternativa

[1] 1 - algo que se sabe, 2 - algo que se tem, 3 - algo que se é.

recomendada é armazenar a chave privada em dispositivos do tipo *smart cards*, que operam através de leitoras conectadas aos computadores, ou em dispositivos apresentados na forma de *token* USB. Apesar da aceitação, uma das dificuldades na utilização de certificados digitais pelos bancos é o custo associado, não somente com os dispositivos como cartões e leitoras ou *tokens*, mas também com o próprio certificado digital que necessita ser gerenciado e renovado periodicamente.

A solução apresentada neste trabalho visa satisfazer os dois aspectos de autenticação de curto e longo prazo, atendidos através de assinatura digital. A assinatura é produzida por um dispositivo de baixo custo, sem conexão ao computador e sem a utilização de certificado digital. A assinatura digital gerada pelo dispositivo é informada manualmente pelo usuário ao computador para validação pelo sistema do banco.

Para viabilizar este cenário, é necessária a geração de assinaturas digitais curtas, passíveis de serem digitadas. Muitas são as aplicações que podem ser beneficiadas com o uso de assinaturas digitais curtas. Conforme citado por Boneh, Lynn e Shacham (2002), assinaturas curtas são especialmente necessárias onde um indivíduo é solicitado a digitar a assinatura manualmente. Assim sendo, o uso de assinaturas curtas torna viável a geração de assinatura digital em dispositivos sem conectividade com o computador.

Para a geração de assinatura digital curta apresentada como proposta deste trabalho, foi realizada uma análise sobre o *Digital Signature Algorithm* (DSA) no intuito de encontrar meios de reduzir a assinatura gerada sem, contudo, comprometer sua segurança. Algumas suposições se fizeram necessárias de modo a garantir o nível de segurança exigido para seu uso em *Internet Banking*, ainda que a assinatura digital tenha sido reduzida a nível extremamente baixo.

Os resultados alcançados e o nível de segurança obtido demonstram a viabilidade de utilização de assinaturas curtas geradas em equipamentos sem a exigência de conexão.

1.1 Objetivos

1.1.1 Objetivo Geral

Apresentar uma solução de assinatura digital curta de 32 bits com características especiais baseada no algoritmo DSA, gerada em um *token* sem conectividade com o computador, dispensando a instalação de software e hardware específicos no computador do usuário.

1.1.2 Objetivos Específicos

- Apresentar um protocolo especialmente projetado para o uso seguro da assinatura digital curta que está sendo proposta;
- Realizar a análise de segurança da assinatura digital curta apresentada a fim de destacar a viabilidade de sua utilização;
- Contextualizar o uso da solução para prover não-repúdio em transações efetuadas através do *Internet Banking*, permitindo a autenticação de curto e longo prazo;
- Possibilitar o uso de assinatura digital em ambientes onde há a necessidade da digitação da assinatura;
- Diminuir os custos envolvidos no uso de assinatura digital.

1.2 Justificativa

As atuais soluções de algoritmos de assinatura digital requerem estruturas complexas para viabilizar a sua utilização de maneira segura. Uma dessas complexidades diz respeito à proteção da chave privada utilizada na geração da assinatura, que normalmente é obtida através da armazenagem em dispositivos seguros como *token* USB ou *smart card*. Esses dispositivos fazem uso de software e hardware específicos instalados no computador do usuário. Além disso, as soluções de assinatura digital fazem uso dos certificados digitais, necessitando vários níveis de gerenciamento para sua manipulação de maneira confiável. Essas necessidades restringem o uso de

assinatura digital em diversas aplicações, devido ao custo e complexidade envolvidos no processo.

Uma solução mais viável para geração de assinatura digital é a utilização de dispositivos sem conectividade com o computador. Porém, os algoritmos de assinaturas digitais para serem seguros geram assinaturas digitais longas, inviabilizando sua utilização em dispositivos sem conectividade com o computador e que necessitem que a assinatura seja digitada manualmente. Algoritmos reconhecidos como seguros como o RSA (RIVEST; SHAMIR; ADLEMAN, 1978) e o DSA (NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY - NIST, 2000) geram assinaturas longas (mínimo recomendado de 1024 bits para o RSA e 320 bits para o DSA). A assinatura gerada pelo DSA ainda que curta comparada àquela gerada pelo RSA, inviabiliza o seu uso em ambientes onde há interesse que a assinatura digital seja informada manualmente.

Dessa forma, a motivação deste trabalho está direcionada à apresentação de uma solução para geração de assinatura digital curta de 32 bits produzida em dispositivo pessoal que não possua conectividade com o computador. A informação a ser assinada é informada de maneira manual ao dispositivo, que procede com a geração da assinatura. Posteriormente, o usuário informa de maneira igualmente manual ao computador a assinatura gerada. A proposta apresentada não fará uso de certificados digitais, nem tampouco de hardware ou software especiais instalados no computador para gerenciar o dispositivo de assinatura, dado que esse funcionará de forma independente do computador. A solução busca simplificar o processo, reduzir custos e ainda permitir uma maior mobilidade por parte do cliente, que não necessitará de hardware e software instalados no seu computador para geração da assinatura.

O enfoque do trabalho será direcionado à aplicação de assinatura digital curta em transações bancárias efetuadas através do *Internet Banking*, uma vez que esses tipos de sistemas requerem um alto nível de segurança no processo de autenticação dos seus clientes.

1.3 Trabalhos Correlatos

Alguns esquemas de geração de assinaturas curtas foram encontrados na literatura. Naccache e Stern (2000) sugeriram uma variação no DSA que resultou em assinatura digital de 240 bits. Boneh, Lynn e Shacham (2002), especificaram uma proposta de

assinatura digital que gira em torno da metade do tamanho da assinatura gerada pelo DSA, ou seja, 160 bits. Courtois, Finiasz e Sendrier (2002) apresentaram um trabalho para a obtenção de assinatura de 81 bits baseado no algoritmo McEliece.

Puente, Sandoval e Hernández (2003) descreveram uma solução para autenticação do usuário e integridade dos dados em aplicações de *Internet Banking* fazendo uso de um dispositivo seguro, chamado assinador, que não possui conectividade com o computador. Os dados a serem assinados são codificados em cores preta e branca na tela do computador, representando os valores 0 e 1, e então, enviados de maneira codificada ao assinador que possui sensores óticos. No assinador, o usuário poderá visualizar as informações que serão assinadas. Para gerar a assinatura digital, o assinador quebra os dados a serem assinados em blocos de 64 bits e posteriormente esses são cifrados com o Triple-DES. O resultado final dos dados cifrados é passado por uma função de *hash,* reduzindo a assinatura a 24 bits. A assinatura é então informada manualmente ao computador pelo usuário para posterior validação. Essa solução implementa portanto, criptografia simétrica.

No cenário nacional existe uma solução com assinatura digital curta utilizada no licenciamento de veículos do Departamento Estadual de Trânsito de São Paulo (Detran-SP). A solução é baseada no esquema de assinatura descrito em Boneh, Lynn e Shacham (2002) e foi especificada pelo Laboratório de Arquitetura de Redes e Computadores (LARC) da Universidade de São Paulo, sendo implementada pela empresa Scopus (www.scopus.com.br). Para efetuar o pagamento dos impostos e taxas, o proprietário do veículo se dirige a uma instituição financeira credenciada. A instituição gera sua assinatura digital sobre as informações do veículo e taxas quitadas e a assinatura é então impressa no comprovante de pagamento, sendo formatada em 64 caracteres alfanuméricos. A emissão do documento do veículo é somente liberada após o órgão emissor de documentação do veículo, que possui a chave pública da instituição financeira, realizar a validação da assinatura digital impressa no comprovante de pagamento.

As soluções de assinaturas digitais curtas discutidas aqui não possuem tamanho tão reduzido a ponto de serem informadas manualmente de forma trivial. Ainda que a solução apresentada por Puente, Sandoval e Hernández (2003) gere apenas 24 bits,

consiste no uso de técnicas de criptografia puramente simétricas, o que não agrega a propriedade de não-repúdio à assinatura gerada.

1.4 Estrutura da Dissertação

O presente trabalho está organizado como descrito a seguir.

O capítulo 2 apresenta conceitos e vulnerabilidades relacionados a diferentes técnicas de autenticação de usuário existentes.

O capítulo 3 descreve o uso de métodos de autenticação que vêm sendo aplicados nos sistemas de *Internet Banking*.

No capítulo 4 são apresentadas informações relacionadas à assinatura digital com enfoque no algoritmo de assinatura digital DSA. Fundamentos Matemáticos serão apresentados para um melhor entendimento desse capítulo.

No capítulo 5 será descrita a proposta deste trabalho para geração de assinatura digital curta baseada no DSA com características especiais, visando garantir não-repúdio nas transações bancárias através do *Internet Banking*. Serão especificados os protocolos da proposta, efetuada uma análise de segurança e apresentado um protótipo da solução descrita.

Por fim, o capítulo 6 relaciona as considerações finais do trabalho e sugestões para pesquisa futura.

2 ESQUEMAS DE AUTENTICAÇÃO

2.1 Introdução

A autenticação, isto é, a garantia de que a entidade (usuário, computador) que está do outro lado realmente é quem diz ser, tem sido tema de estudos há alguns anos. Os métodos de autenticação têm como desafio provar a identidade de quem está do outro lado. Esses métodos podem ser divididos em autenticação mútua, onde ambas as partes que irão participar da comunicação certificam a identidade do outro e autenticação de único caminho, onde apenas uma das partes é autenticada.

Este capítulo apresenta uma descrição e vulnerabilidades de diversos métodos utilizados nos processos de autenticação.

2.2 Descrição

A autenticação é o processo pelo qual uma parte tem garantias da identidade de uma segunda parte. O modo como os usuários podem ser autenticados pode ser dividido em categorias chamadas fatores de autenticação, como descrito a seguir (FEDERAL FINANCIAL INSTITUTIONS EXAMINATION COUNCIL - FFIEC, 2005):

- Algo que o usuário sabe: como senha ou *Personal Identification Number* (PIN);
- Algo que o usuário possui: uma identificação magnética como uma chave privada ou um gerador de identificação matemática, normalmente associados a um dispositivo físico como *smart card* ou *token* USB;
- Algo que o usuário é: características humanas reconhecidas através de leituras biométricas como impressão digital, reconhecimento de íris, entre outros.

Na busca por um mais alto nível de segurança, os métodos de autenticação buscam utilizar a combinação de dois ou mais fatores acima descritos. Um exemplo muito comum de combinação de dois fatores de autenticação é o caso do cartão (algo que se possui) utilizado nos terminais bancários e que são ativados por uma senha (algo que se sabe).

2.3 Ameaças à Segurança em Esquemas de Autenticação

As tentativas de ataques aos esquemas de autenticação, isto é, descobrir a identificação de outro para acessar indevidamente um sistema, são temas de estudos tão importantes quanto à própria solução de segurança. O objetivo de um atacante é quebrar o esquema de autenticação mais rápido do que o método da força bruta. A seguir, será apresentado um resumo dos ataques praticados em quebras de protocolos de autenticação (MAO, 2003; MENEZES; OORSCHOT; VANSTONE, 1996):

- Ataque de Repetição: o atacante obtém a informação de autenticação utilizada em uma execução anterior do protocolo. Posteriormente, utiliza a mesma informação para forjar o processo de autenticação;
- Ataque de Reflexão: o atacante envia o desafio do protocolo de autenticação de volta à entidade que o emitiu, no intuito de obter a resposta ao desafio e utilizá-la como sua própria resposta;
- Ataque de texto escolhido: o atacante estrategicamente escolhe mensagens e se passa pelo verificador a fim de obter informações a respeito do segredo utilizado na autenticação;
- Ataque de dicionário: utilizado para quebra de senhas, onde um adversário mantém uma lista de prováveis senhas, normalmente palavras comuns, no intuito de tentar descobrir a senha do usuário.

2.4 Métodos de Autenticação

Nesta seção, serão descritos alguns métodos utilizados no protocolo de autenticação. A fim de exemplificação, citaremos os personagens Alice e Bob que dizem respeito às duas partes que participam do processo de autenticação.

2.4.1 Senhas Estáticas

O termo “senhas” inclui palavras, frases e PINs que devem ser mantidos secretos para uso em autenticação (GORMAN, 2003). A autenticação baseada em senhas estáticas funciona como uma espécie de segredo compartilhado entre o usuário e o sistema. Uma senha é normalmente uma seqüência de caracteres passível de ser

memorizada pelo seu proprietário. Todavia, por questões de segurança, o sistema nunca deve conhecer a senha do usuário, mas apenas ser capaz de validá-la. Para evitar que a senha seja armazenada em texto claro, uma função de caminho único (*hash*) é aplicada sobre a senha e o resultado é então armazenado (resumo da senha). Neste sentido, para certificar a autenticidade do usuário, o sistema aplica a função de caminho único sobre a senha fornecida pelo usuário e compara o resultado com o valor armazenado. Se forem iguais, a autenticação do usuário foi efetuada. Na Fig. 1 é possível visualizar o processo de autenticação baseada em senha estática.

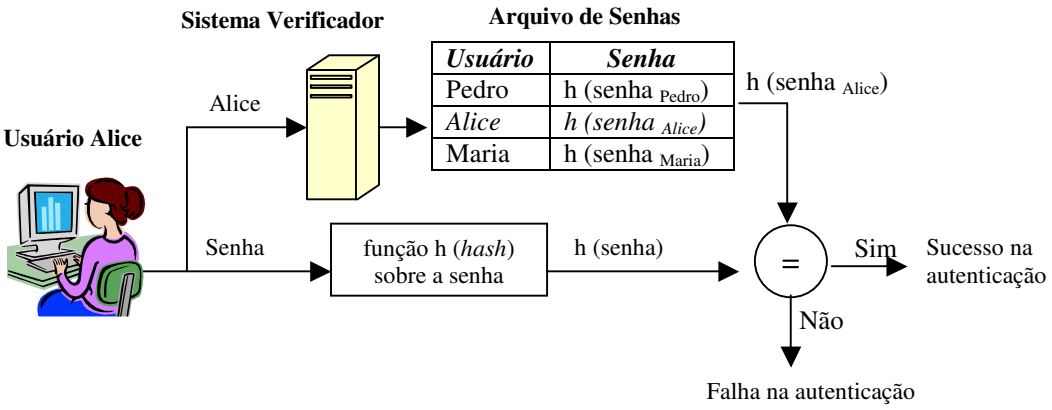


Figura 1 - Processo de autenticação através de senhas

Fonte: Adaptado de Menezes, Oorschot e Vanstone (1996)

A autenticação baseada em senhas estáticas possui diversas vulnerabilidades. Se a senha for curta, ela pode ser descoberta por um atacante através de busca exaustiva. Por outro lado, uma senha longa e randômica é difícil de ser memorizada por seu proprietário (GORMAN, 2003). A senha difícil de ser memorizada faz com que o seu proprietário a anote em papel ou arquivo de computador, abrindo espaço para roubo da senha. Além disso, a maioria dos usuários utiliza a mesma senha para se autenticar em diferentes aplicações. Se a senha for comprometida, todos os sistemas autenticados por ela estarão igualmente comprometidos (ULUDAG et al., 2004).

Schneier (1996) cita o risco do ataque do dicionário ser efetuado sobre as senhas chamadas fracas, onde um usuário seleciona a senha de um universo pequeno de possibilidades, como palavras do dicionário ou nomes próprios. Neste ataque, o atacante mantém uma lista das possíveis senhas submetidas à função de único caminho.

Através do roubo do arquivo de senhas ou da obtenção do *hash* da senha informada pelo usuário, o atacante procura em sua lista a entrada correspondente ao *hash* da senha.

A autenticação baseada em senhas requer a existência de um canal seguro para sua transmissão (MENEZES; OORSCHOT; VANSTONE, 1996). Caso a senha trafegue em texto claro, um atacante poderá obtê-la e, posteriormente, efetuar autenticação em nome do usuário legítimo. Este ataque é conhecido como ataque de repetição.

Um outro ataque que pode ser efetuado neste tipo de autenticação é a possibilidade de um atacante capturar a senha digitada pelo usuário através da recuperação de eventos oriundos do seu próprio teclado, conhecidos como ataques de *keylogger* (HILTGEN; KRAMP; WEIGOLD, 2006; KIRDA; KRUEGEL, 2005).

2.4.2 Desafio-Resposta

A idéia da autenticação baseada em desafio-resposta é que, para autenticar Alice, o verificador Bob seleciona um desafio (tipicamente um número randômico secreto) e o envia a Alice. Alice, de posse de um segredo, envia a resposta a Bob referente ao desafio recebido. Bob verifica se a resposta recebida de Alice é a esperada, confirmando ou não a autenticidade de Alice (STALLINGS, 2003).

Caso a resposta ao desafio enviada por Alice seja monitorada por um adversário, esse não poderá utilizá-la em uma futura tentativa de se passar por Alice, uma vez que o desafio será outro, e, por conseqüência, a resposta também. Essa prática evita o ataque de repetição.

Importante notar que este método está baseado na existência de um segredo previamente conhecido entre as partes para troca segura da resposta ao desafio. Este segredo está relacionado a técnicas criptográficas, simétricas ou assimétricas. Um dos problemas relacionados à autenticação utilizando tais técnicas diz respeito à segurança da guarda da chave utilizada no processo de autenticação. Uma alternativa é mantê-la no disco rígido, porém, esse recurso abre várias possibilidades para um atacante capturar a chave. Segundo Puente, Sandoval e Hernández (2003), no sistema operacional Windows, que é amplamente utilizado por usuário final, até mesmo uma macro do Word pode executar tal tarefa. Ainda que o arquivo que contém a chave esteja protegido por uma senha, após capturado esse arquivo poderá ser atacado, como vimos anteriormente a respeito da vulnerabilidade relacionada ao uso de senhas.

Uma alternativa para proteção de chave é armazená-la em um hardware *token* (HOOVER; KAUSIK, 1999), como por exemplo, um *smart card*. Veremos alguns detalhes relacionados a *token*s na seção 2.4.5.

Nos próximos tópicos, analisaremos a autenticação baseada em desafio-resposta obtida por técnicas simétricas e assimétricas encontrada em (MENEZES; OORSCHOT; VANSTONE, 1996; SCHNEIER, 1996; STINSON, 1995). Os protocolos descritos aqui apenas trafegam o desafio-resposta propriamente dito, embora na prática algumas outras informações possam ser enviadas juntamente, como o identificador das partes envolvidas.

2.4.2.1 Desafio-Resposta por Técnicas de Chaves Simétricas

Neste caso, Alice e Bob compartilham uma chave simétrica *K*, que é utilizada com um algoritmo simétrico escolhido por eles. Na autenticação de caminho único, para que Alice possa ser autenticada, Bob envia a ela um número randômico r_B. Posteriormente, Alice envia a Bob o número randômico r_B cifrado com a chave *K* compartilhada entre eles. Bob decifra a informação enviada por Alice e verifica se o número decifrado é o mesmo número randômico r_B enviado anteriormente. Se forem iguais, Alice foi autenticada.

No protocolo de autenticação mútua, além de enviar a Bob o número r_B cifrado para se autenticar, Alice envia ainda um número r_A para que Bob também possa ser autenticado. Bob checa se o número randômico r_B recebido é o mesmo enviado anteriormente e recupera o valor randômico r_A, enviando-o de volta a Alice. Neste sentido, Alice terá a confirmação de que está se comunicando com Bob e Bob com Alice.

2.4.2.2 Desafio-Resposta por Técnicas de Chaves Assimétricas

No protocolo de desafio-resposta por técnicas de chaves assimétricas, Bob possui a chave pública de Alice, e somente Alice possui sua chave privada. Em uma autenticação de caminho único, Bob envia a Alice um número randômico r_B. Alice envia a Bob o número randômico r_B cifrado com sua chave privada, utilizando algum algoritmo assimétrico (o mesmo a ser utilizado por Bob). Bob decifra a informação recebida com a chave pública de Alice. Em seguida, verifica se o número recebido diz respeito ao

mesmo número randômico r_B enviado anteriormente. Se forem iguais, Alice foi autenticada.

No protocolo de autenticação mútua, Alice envia a Bob um número randômico r_A cifrado com a chave pública de Bob. Bob decifra a informação recebida com sua chave privada utilizando algum algoritmo assimétrico (o mesmo utilizado por Alice). Em seguida, Bob envia a Alice o número r_A recuperado, juntamente com um novo número randômico r_B cifrados com a chave pública de Alice. Alice decifra a informação recebida com sua chave privada, recuperando r_A e r_B. Alice então verifica se o número r_A recebido diz respeito ao mesmo número randômico r_A enviado anteriormente. Se forem iguais, Bob está autenticado e Alice envia a Bob o número r_B recebido para também ser autenticada por ele.

As chaves públicas dos usuários podem ser distribuídas por meio dos certificados digitais, os quais consistem de um conjunto de informações do usuário e sua chave pública digitalmente assinado por uma terceira entidade confiável chamada Autoridade Certificadora. Através dos certificados digitais, as chaves públicas podem ser armazenadas, distribuídas ou encaminhadas sobre um canal inseguro. O objetivo é fazer com que a chave pública da entidade esteja disponível para outros usuários de modo que sua autenticidade possa ser validada e verificada (MENEZES; OORSCHOT; VANSTONE, 1996).

2.4.3 One-Time Password (OTP)

Existem diversas variações de autenticação baseada em OTP, mas a essência desse método é que a cada autenticação uma senha diferente deve ser informada pelo usuário ao sistema. Um exemplo desse tipo de autenticação é o caso onde o usuário e o sistema compartilham uma lista de senhas pré-definidas (cada uma válida para apenas uma autenticação). As senhas podem ser utilizadas seqüencialmente ou não. No caso de autenticação com senhas seqüenciais, no momento da autenticação o sistema verifica se o usuário informou determinada senha (a próxima a ser utilizada). No caso de autenticação com senhas não seqüenciais, o sistema verifica se a senha informada pelo usuário faz parte do grupo de senhas não utilizadas, ou ainda, o sistema poderá solicitar uma senha referente a uma seqüência específica (MENEZES; OORSCHOT; VANSTONE, 1996).

Uma variação mais sistemática do método OTP ocorre quando as aplicações de geração de OTP possuem um valor de entrada e uma chave secreta compartilhados com o sistema de autenticação. No momento de se autenticar, a aplicação de geração de OTP utiliza o valor de entrada cifrado com a chave secreta compartilhada, sendo que o mesmo processamento é efetuado no sistema de autenticação. Esse valor de entrada, que pode ser baseado em sincronismo de tempo ou evento, deve sofrer alguma mutação no intuito de que, a cada tentativa de autenticação, a informação de autenticação gerada seja diferente, caracterizando um OTP (FFIEC, 2005; NIST, 2004). No sincronismo de tempo, o usuário e o sistema de autenticação possuem o tempo sincronizado, e este participa da geração do OTP. Uma nova senha é gerada a cada período de tempo pré-determinado (como por exemplo, a cada 60 segundos). Isso significa que quando o usuário tenta se autenticar, o sistema de autenticação espera por uma determinada senha, válida para aquele determinado período. No sincronismo de evento, o sistema gerador de OTP utiliza um contador como valor de entrada. Quando o sistema é inicializado pela primeira vez, o contador interno está configurado com o valor zero. Cada vez que um evento ocorrer, como por exemplo, sempre que o usuário solicitar nova senha, o contador incrementa seu valor e esse é cifrado gerando o OTP. Do mesmo modo, no servidor de autenticação, a conta do usuário possui um contador que é incrementado sempre que o usuário se autentica. No caso de sincronismo de eventos, deve haver uma tolerância a falhas para permissão de autenticação, uma vez que os contadores de eventos podem não estar sincronizados. Exemplos desses casos são situações onde o usuário solicitou senhas à aplicação de OTP que não foram efetivamente utilizadas no servidor de autenticação, ou devido a erros de conexões, fazendo com que o contador de eventos do usuário esteja diferente do contador de eventos do servidor de autenticação.

2.4.4 Biométrica

A autenticação biométrica diz respeito à verificação de identidade através de características humanas físicas ou comportamentais lidas por dispositivos biométricos (ULUDAG et al., 2004). Essas características podem ser obtidas pelo formato da face, impressão digital, leitura da íris, timbre da voz, assinatura de punho entre outros. A autenticação biométrica adiciona um paradigma diferenciado nas formas de autenticar o

usuário: com algo que você é (como impressão digital ou face) ou algo que você faz ou produz (como assinatura ou voz).

Embora tenha suas vantagens sobre outros esquemas de autenticação, os sistemas biométricos possuem taxas de erros superiores às taxas de erros em hardware como leitoras de cartões inteligentes (GORMAN, 2003). Erros biométricos podem ocorrer devido à sujeira no dispositivo de captura, luminosidade inadequada, ajuste irregular com relação ao ambiente (frio, chuva, sol), inicialização incoerente do registro da informação biométrica, entre outros.

Um dos ataques que pode ser praticado na autenticação baseada em biometria é o ataque de repetição, onde o atacante envia ao sistema biométrico o dado (informação biométrica) submetido anteriormente pelo usuário legítimo (XIAO, 2005). Neste caso, o atacante informa ao sistema a informação biométrica capturada, sem passar pelo sensor (através do qual é capturada a informação biométrica). Uma das formas de evitar esse tipo de ataque é incluir um mecanismo de desafio-resposta. O desafio lançado pelo sistema de autenticação deve ser adicionado à informação biométrica indicada pelo usuário.

Outro ataque que pode ser praticado, apesar de haver uma certa complexidade no processo, é através da apresentação de uma característica biométrica falsa ao sensor, copiada do usuário legítimo (XIAO, 2005). Um exemplo é a utilização de um dedo falso, quando a autenticação é realizada por impressão digital. Segundo cita Gorman (2003), apesar de ser difícil copiar ou forjar uma característica biométrica, uma vez atacada não pode ser substituída, como acontece com senhas ou *token*s.

2.4.5 Token

Um *token* é algo que a entidade que deseja provar sua identidade (ou seja, que deseja se autenticar) possui e controla, caracterizando assim o fator de autenticação "algo que se tem". Para obter um segundo fator de autenticação, os *token*s podem ser ativados pelo usuário através da indicação de um PIN ou identificação biométrica.

A seguir, é apresentado um resumo de categorias dos *token*s encontradas em (FFIEC, 2005; NIST, 2004).

- USB *token*: um dispositivo do tamanho de um chaveiro que conecta diretamente na porta USB do computador. Não necessita de hardware especial instalado no computador do usuário;
- *Smart Card*: *token* em formato de cartão inteligente, que requer uma leitora de cartão instalada no computador do usuário;
- OTP *token*: dispositivos que geram OTP sempre que é utilizado. O OTP é apresentado ao usuário através de uma tela;
- Software *token*: sistema instalado no computador que simula o comportamento de um hardware *token*.

Forjar uma identidade utilizando um hardware ou software *token* requer que o atacante tenha posse do *token*, e dependendo da configuração deste, o atacante precisará ainda indicar a senha ou informação biométrica para poder ativá-lo. Deste modo, a autenticação utilizando *token* é mais segura do que a autenticação baseada apenas em senhas. Adicionalmente, o hardware *token* apresenta-se mais seguro se comparado ao software *token*, uma vez que a perda ou roubo do hardware *token* é percebido pelo seu proprietário, o que não acontece com o software *token*.

Os *token*s podem armazenar certificados digitais, habilitando sua utilização em um ambiente de infra-estrutura de chaves públicas (ICP).

2.4.6 Código de Autenticação de Mensagem

Um código de autenticação de mensagem (MAC), é uma função que permite autenticar mensagens a partir de técnicas de criptografia simétrica ou funções *hash*. Um algoritmo MAC, a partir da mensagem a ser autenticada e de uma chave secreta compartilhada entre a origem e destino da mensagem, produz um texto de saída de tamanho fixo (o MAC), de tal modo que é impraticável para um atacante produzir a mesma saída sem o conhecimento da chave secreta (MENEZES; OORSCHOT; VANSTONE, 1996). Ao receber a mensagem, o receptor calcula o MAC a partir da mensagem recebida e compara com o MAC recebido. Se forem iguais, dois requisitos de segurança são obtidos (STALLINGS, 2003):

- Integridade dos Dados: o receptor constata que os dados não foram alterados. Se um atacante alterar a mensagem, o cálculo do MAC realizado pelo receptor será diferente do MAC recebido;
- Autenticidade da mensagem: o receptor tem garantia de que a mensagem veio de quem diz ter sido o remetente.

2.5 Conclusão

No meio digital, autenticar o outro constitui-se um desafio. Citamos neste capítulo alguns métodos de autenticação, cada qual com suas características e alguns requisitos para que possam se mostrar mais seguros. Entretanto, na medida em que se aumenta o quesito segurança na autenticação, aumenta-se também a complexidade envolvida no processo.

O método de autenticação mais adequado depende da aplicação onde será utilizado. Nível de segurança, custo, viabilidade técnica e praticidade de uso são alguns dos itens que devem ser analisados na escolha do método mais apropriado.

3 AUTENTICAÇÃO EM INTERNET BANKING

3.1 Introdução

Internet Banking é o sistema pelo qual o cliente efetua operações bancárias pela Internet. Através dele, o cliente pode realizar transações diversas como consultas a saldos, pagamentos de contas e movimentações bancárias.

Na busca pela garantia da autenticidade do cliente, os bancos vêm utilizando diferentes métodos de autenticação em suas aplicações de *Internet Banking*. Devido ao aumento de roubo de identidade, novos métodos e alternativas de autenticação vêm sendo implantados. Nos sistemas de *Internet Banking*, a autenticação segura do usuário é uma preocupação constante, dado que o risco de identificação incorreta dos usuários pode resultar em perdas financeiras e de confiança no banco por parte dos clientes.

Há uma grande variedade de soluções para autenticação de clientes disponíveis no mercado. Custo, confiança e conveniência são propriedades levadas em consideração na busca pela identificação segura do usuário.

O presente capítulo apresenta algumas soluções utilizadas nos sistemas de *Internet Banking* na autenticação de clientes, a fim de evitar fraudes em transações bancárias pela Internet.

3.2 Segurança em Internet Banking

Os sistemas de *Internet Banking* proporcionam comodidade, agilidade e facilidade de acesso aos serviços das instituições financeiras utilizando como meio a Internet. Entretanto, a Internet oferece uma variedade de aberturas para ataques de autenticação, e dessa forma, para manter um nível adequado de segurança, é necessário observar algumas práticas contra ataques maliciosos.

Os Estados Unidos, através do regulamento emitido pelo Federal Financial Institutions Examination Council - FFIEC (2005), está exigindo que os bancos utilizem no mínimo dois fatores de autenticação até o final de 2006. O FFIEC é formado por um grupo de entidades que emite recomendações e princípios direcionados às instituições financeiras. O regulamento especifica que os ataques em esquemas de autenticação

ocorrem mais freqüentemente em métodos de um único fator, como o uso de *login* e senha.

Um estudo apresentado pelo Federal Deposit Insurance Corporation – FDIC (2005), que é uma agência membro do FFIEC, indica que os clientes estão mais preocupados com relação à segurança digital e por esse motivo estão mais receptivos ao uso de dois fatores de autenticação. Ressalta também que os clientes podem parar de utilizar o *Internet Banking* devido à preocupação com segurança e inclusive, podem mudar de banco caso sua instituição financeira venha a falhar no quesito segurança de suas informações. O estudo diz ainda que muitos usuários de Internet que não utilizam *Internet Banking* revelaram que gostariam de fazer uso desse serviço desde que a segurança seja aperfeiçoada.

Além do custo direto em perdas financeiras relacionado ao roubo de identidade nos sistemas de *Internet Banking*, os indiretos também devem ser observados. Os custos indiretos incluem taxa reduzida de adesão ao *Internet Banking* e, portanto, uma maior utilização dos canais bancários, perda de confiança no banco e aumento na preocupação da falta de segurança na instituição bancária em geral. Como citam Jin e Cheng (2005), uma análise cuidadosa dos riscos potenciais e a probabilidade deles ocorrerem devem ser levados em consideração na escolha de soluções de autenticação segura.

Uma das formas de roubo de identidade em acesso ao *Internet Banking* que vem ocorrendo com freqüência é o chamado *phishing*, cujo objetivo é obter as credenciais de acesso ao banco do usuário. Os ataques de *phishing* utilizam uma combinação de engenharia social e pretextos para convencer o usuário a revelar suas informações de acesso ao banco (KIRDA; KRUEGEL, 2005). Em um típico ataque de *phishing*, o atacante envia um e-mail malicioso em nome da instituição bancária para o cliente do banco. O e-mail traz um texto com o objetivo de convencer o cliente a clicar em um *link* para acessar o web site do banco. Porém, ao clicar no *link*, o cliente é na verdade direcionado a um site malicioso cuja interface é uma cópia do site oficial da instituição bancária do cliente e, no entanto, é controlado pelo atacante. Posteriormente, o usuário é solicitado a indicar suas informações pessoais (conta e senha), permitindo que o atacante obtenha suas credenciais bancárias, e assim, possa acessar sua conta no site legítimo do banco para executar ataques diversos.

Alguns ataques de *phishing* fazem uso de certas vulnerabilidades existentes nos *browsers* que permitem instalar programas maliciosos no computador do usuário. Um exemplo é o *keylogger*, que registra todas as teclas pressionadas pelo usuário em determinado site bancário, enviando posteriormente esta informação para o atacante. Outra forma de ataque pode ocorrer quando o atacante altera as configurações de *proxy* do *browser* do usuário de tal modo que todo o tráfego da web passe através do servidor do atacante (KIRDA; KRUEGEL, 2005).

Uma pesquisa realizada pela Forrester Research (GIOVANNINI; BENJAMIN; CONDON, 2006) mostra que 70 milhões de europeus utilizam *Internet Banking*. Entretanto, aproximadamente 80 milhões de usuários de Internet nunca utilizaram o *Internet Banking* ou já utilizaram, mas posteriormente desistiram. Para entender o motivo de um número tão expressivo de pessoas que não utilizam os serviços *on-line* dos bancos, 23.000 clientes de instituições financeiras e usuários de Internet na Europa foram entrevistados. A pesquisa mostra que os bancos estão perdendo usuários do *Internet Banking* por motivos que vão desde a desconfiança na segurança do serviço até problemas com perda de senhas. O gráfico apresentado na Fig. 2 mostra os motivos pelos quais os usuários pararam de utilizar o *Internet Banking* ou nunca utilizaram esse serviço (múltiplas respostas foram aceitas). A desconfiança com relação à segurança é uma das grandes barreiras para adoção ao *Internet Banking*.

Sendo assim, fica evidente a necessidade de estudos no quesito segurança em autenticação no sistema *Internet Banking*. A autenticação do usuário é um item que deve ter especial atenção, uma vez que se a autenticação do usuário falhar, todo o sistema estará comprometido.

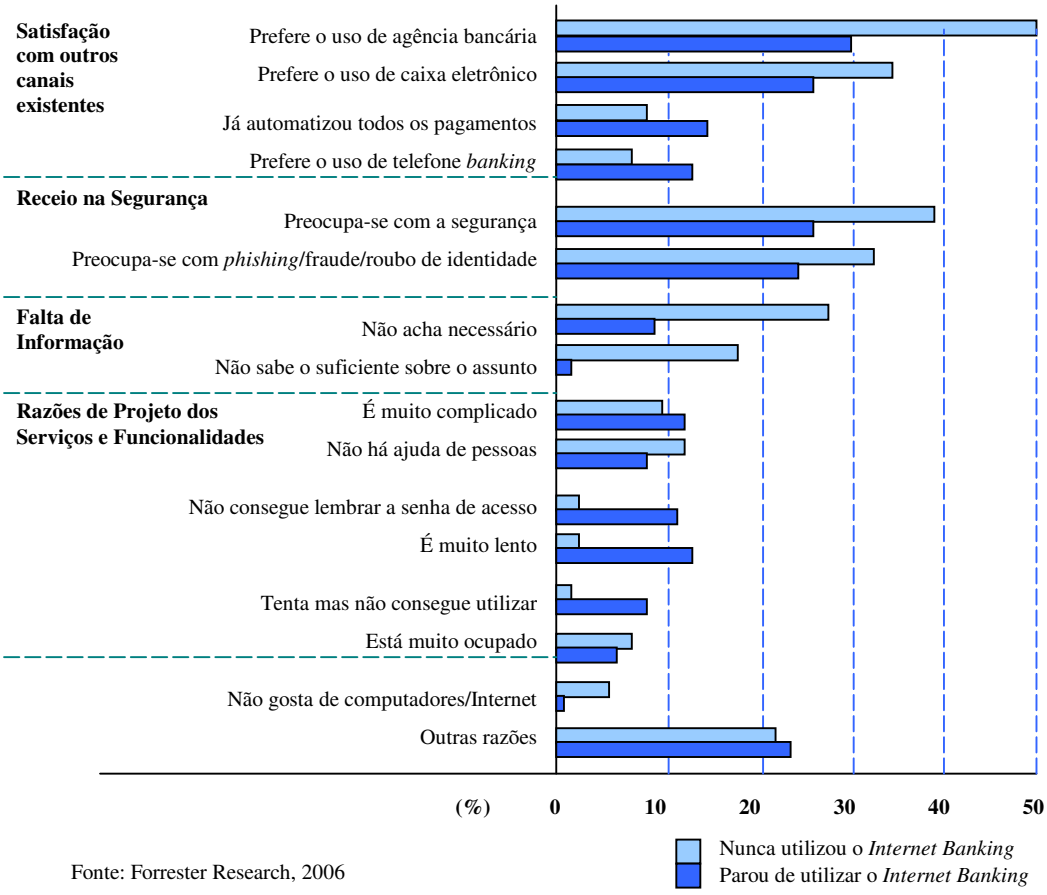
3.3 Soluções de Autenticação em Internet Banking

Nesta seção, serão apresentadas algumas soluções de aplicação de métodos de autenticação de usuário utilizadas nos sistemas *Internet Banking*.

3.3.1 Senhas Estáticas

Apesar das fragilidades relacionadas ao uso de senhas citadas no Capítulo 2, autenticação utilizando *login* e senha é o mecanismo mais difundido nas aplicações

bancárias pela Internet (PUENTE; SANDOVAL; HERNÁNDEZ, 2003). Alguns bancos gerenciam duas senhas para cada usuário que utiliza o sistema de *Internet Banking*: uma é utilizada nas operações de consultas, e outra para efetivação de transações.



Fonte: Forrester Research, 2006

Figura 2 - Razões para o não uso do *Internet Banking* ou interrompimento do uso

Na tentativa de evitar a captura de senha por atacante quando o cliente estiver informando-a ao *Internet Banking,* alguns sistemas vêm utilizando os teclados virtuais, onde uma imagem de teclado é apresentada ao cliente com as teclas assumindo posição randômica a cada acesso (Fig. 3). Para informar sua senha, o cliente clica sobre as teclas dos caracteres que correspondem a ela. Apesar de evitar o *keylogger,* essa solução é vulnerável a alguns tipos de *trojans* que capturam o local da tela onde foi pressionado o mouse (PIETRO; STRANGIO, 2005).

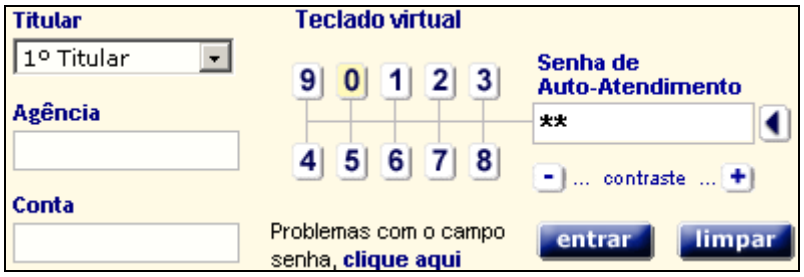


Figura 3 - Teclado virtual

Uma prática utilizada em conjunto com o uso de senhas no processo de autenticação é permitir que os clientes acessem o *Internet Banking* somente de dispositivos previamente cadastrados (computadores, PDA's, telefones celulares). Os dispositivos podem ser reconhecidos por um identificador no formato de *cookie* ou outra forma de identificação derivada de suas características. Para cadastrar um novo dispositivo, o usuário deve acessar o site do banco a partir dele, sendo gerado um código de cadastramento. O cliente deve posteriormente informar o código de cadastramento ao banco via telefone, terminal de auto-atendimento ou diretamente à agência bancária para que o cadastramento do novo dispositivo possa ser finalizado. Entretanto, essa prática restringe a mobilidade no uso do *Internet Banking,* uma vez que o cliente terá sua autenticação aceita somente a partir de dispositivos cadastrados. Além disso, não há meios de prevenir o ataque por pessoas que também tenham acesso físico aos dispositivos registrados pelo cliente.

Outra opção é o PassMark (www.passmarksecurity.com) que se apresenta como uma solução de dois fatores de autenticação e de dois caminhos. Dois fatores de autenticação porque autentica o cliente baseado na senha e no cadastramento do computador de acesso ao sistema. Neste caso, para cadastrar novo computador, o sistema faz alguns questionamentos ao cliente (cujas respostas já foram previamente cadastradas), e posteriormente realiza a confirmação do cadastro do novo computador através do envio de e-mail ou mensagem para o celular. Com relação à característica de dois caminhos, é pelo fato do sistema autenticar o cliente ao site e também autenticar o site ao cliente através do chamado PassMark. No momento de seu registro, o cliente cadastra uma imagem (o PassMark). Quando acessar o site e informar o login, o cliente visualiza a sua imagem recuperada da base de dados (Fig. 4). O objetivo da solução é que, se a imagem apresentada ao cliente for a mesma cadastrada no momento do seu registro, significa que ele pode confiar no sistema que está acessando.

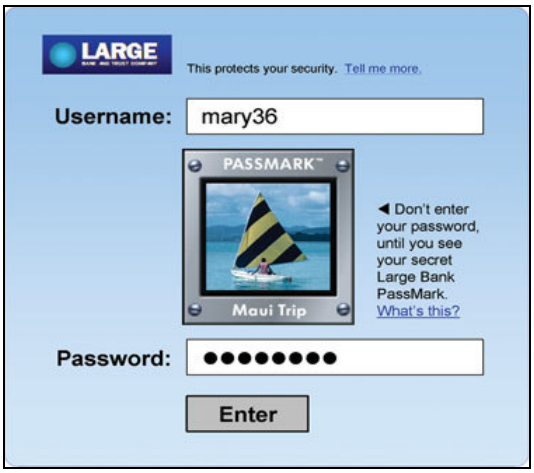


Figura 4 - Autenticação de dois caminhos utilizando PassMark

3.3.2 Hardware Token

Existem diversas soluções disponíveis no mercado que fazem uso de hardware *token*. A empresa RSA (www.rsasecurity.com) possui uma variedade de modelos como *token* USB e *smart card*. Os modelos de hardware *token* da RSA utilizam o algoritmo DES para gerar um OTP a cada 60 segundos (chamado *token code*), que é baseado no tempo sincronizado e em uma chave simétrica compartilhada entre o *token* e o servidor de autenticação. O modelo *token* USB pode operar conectado no computador ou desconectado. Quando desconectado, o usuário lê o *token code* apresentado no visor do *token* e informa manualmente ao computador para proceder com a autenticação. Quando conectado ao computador, o modelo *token* USB é acessado diretamente pela aplicação de autenticação, não necessitando que o usuário digite o *token code*. Esses modelos, além de gerar o OTP são capazes de armazenar certificados digitais, permitindo autenticação, assinatura digital e cifragem de informações. Na Fig. 5 podemos visualizar o modelo *token* USB RSA SecurID SID800.

Figura 5 - RSA SecurID SID800

Na Tabela 1, são apresentadas algumas características técnicas do modelo RSA SecurID SID800.

Tabela 1 - Especificação técnica RSA SecurID SID800

Característica	Medida
Dimensões	86mm x 27mm x 10mm
Peso	21g
Temperatura de Operação	-20°C até 65°C
Tempo de vida da Bateria	Até 5 anos

A empresa Vasco (www.vasco.com) também oferece várias opções de *tokens*, chamados DigiPass, para autenticação em ambientes remotos. Os DigiPass não possuem conexão com o computador e são projetados para realizar duas funções de segurança:

- Calcular OTP que pode ser baseado em sincronização de tempo ou evento para prover autenticação;
- Calcular MAC sobre transações eletrônicas, para garantir a integridade de seu conteúdo (a Vasco chama essa função de assinatura eletrônica ou *e-signature*).

Os produtos DigiPass calculam os OTPs e MACs utilizando o algoritmo DES ou Triple-DES, podendo operar em três modos:

- Somente resposta
- Desafio/resposta
- Assinatura eletrônica

No modo somente resposta (Fig. 6), após o usuário informar seu PIN, o DigiPass calcula um OTP utilizando uma chave secreta compartilhada com o servidor de autenticação em combinação com o tempo ou evento sincronizado. Neste modo, o usuário pode selecionar o uso de OTP baseado em tempo, evento ou uma combinação de ambos para gerar as senhas dinâmicas.

No modo desafio/resposta (Fig. 7), sempre que o usuário deseja se autenticar, o servidor de autenticação apresenta a ele um desafio. Após indicar seu PIN ao DigiPass, o usuário informa-lhe o desafio. O DigiPass, utilizando uma chave secreta compartilhada com o servidor de autenticação, calcula o OTP (resposta) baseado no desafio informado e no tempo ou evento de sincronismo com o servidor.

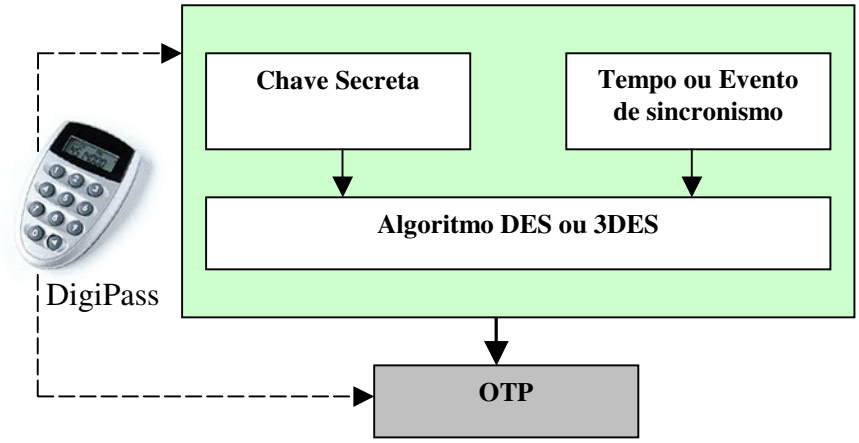


Figura 6 - Processo de OTP somente resposta com DigiPass

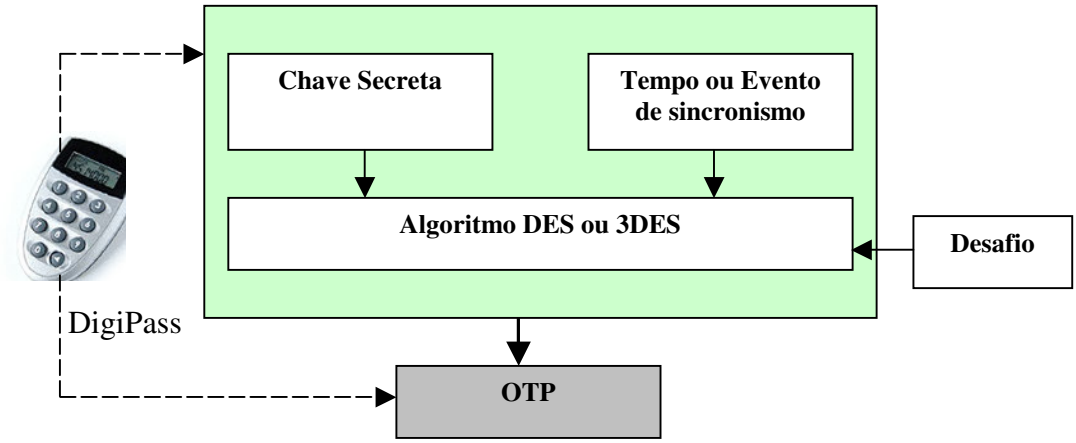


Figura 7 - Processo de OTP desafio/resposta com DigiPass

No modo assinatura eletrônica (Fig. 8), após indicar seu PIN ao DigiPass o usuário entra com as informações que participarão do MAC (no máximo 8 informações). O DigiPass calcula uma senha dinâmica com base nas informações indicadas pelo usuário (em uma operação de transferência de fundos entre contas, usuário pode escolher gerar assinatura sobre a conta origem, conta destino e valor a ser transferido, por exemplo). Posteriormente, usuário informa ao sistema a senha dinâmica (MAC) gerada pelo DigiPass bem como as informações que participaram do MAC, sendo enviados ao servidor de autenticação para validação. No modo de operação de assinatura eletrônica, além de autenticar o usuário, o sistema no servidor também checa a integridade dos dados transmitidos. Essa solução pode ser configurada para que o

próprio sistema do servidor apresente previamente ao usuário quais informações participarão da geração do MAC pelo DigiPass.

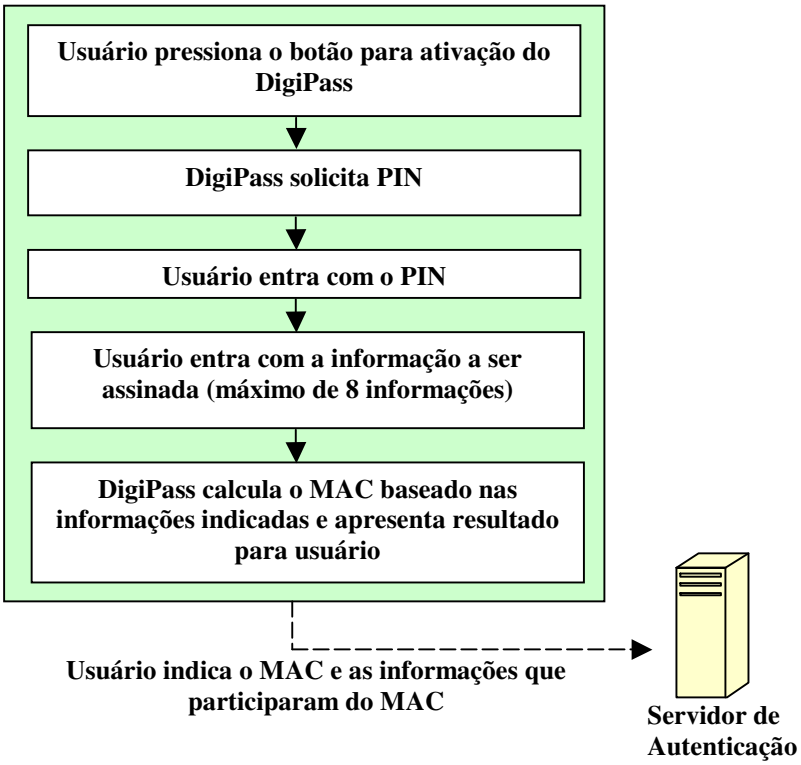


Figura 8 - Processo de assinatura eletrônica com DigiPass

Os dispositivos DigiPass possuem um visor que pode apresentar até 8 algarismos decimais e alguns possuem teclado embutido. O DigiPass GO3 (Fig. 9) não possui teclado. Neste caso, sua ativação não é realizada pela indicação de um PIN, e sim através do pressionamento de um botão, podendo operar nos modos de autenticação de sincronismo de evento e tempo.

Figura 9 - DigiPass GO3

Outra opção de hardware *tokens* são os modelos oferecidos pela empresa CryptoCard (www.cryptocard.com), os quais não possuem conexão com o computador. O fabricante cita que os *tokens* podem operar no modo de autenticação baseado em desafio/resposta ou sincronismo de evento e ainda podem efetuar assinatura digital sobre transações eletrônicas. Os algoritmos utilizados são o 3DES e o AES, sendo que a

assinatura digital citada, assim como acontece com o DigiPass, é na verdade a geração de um MAC.

3.3.3 Software Token

Software *token* apresenta-se como uma opção extra para uso em sistemas de autenticação. A funcionalidade é a mesma de um hardware *token*, entretanto, todo o processo é gerenciado por um programa instalado no computador pessoal, PDA ou celular. Os software *token*s executam a mesma funcionalidade dos hardware *token*s, porém, a chave simétrica utilizada (o *token* propriamente dito) é criado por um sistema central e armazenada no próprio dispositivo pessoal do usuário ao invés de ser armazenada no hardware *token*.

As empresas citadas aqui como fornecedoras de soluções de hardware *token* possuem também versões correspondentes em software *token* para desktop, PDA ou celular. Na Fig. 10 é apresentado o software *token* da RSA para desktop, chamado *RSA SecurID Software Token*. Neste caso, as chaves simétricas RSA SecurID podem ser armazenadas em um *smart card* ou dispositivo USB e utilizadas em conjunto com o *RSA SecurID Software Token*.

Figura 10 - Software token da RSA para desktop

3.3.4 OTP via Mensagem para Celular

Com a grande popularização dos telefones celulares, uma das soluções que vem sendo utilizada é o uso de mensagens SMS para o envio de senhas temporárias. Após entrar com suas credenciais de *login* e senha no site, o cliente recebe um OTP via

mensagem SMS em seu telefone celular. O OTP deve então ser digitado pelo usuário no site para completar o processo de autenticação.

Uma variação desta solução é o produto PINsafe da empresa Swivel (www.swiveltechnologies.com) que utiliza um algoritmo aplicado sobre um PIN registrado previamente e uma *string* de segurança dinâmica (seqüência numérica). Após o usuário informar seu *login* para identificação no sistema da Swivel, uma mensagem SMS é enviada para o seu telefone celular contendo a string de segurança dinâmica. O usuário extrai da *string* de segurança os dígitos que estão dispostos na posição correspondente ao seu PIN. Os dígitos extraídos formam o chamado *One-Time-Code* (OTC) o qual deve ser digitado ao sistema da Swivel para que a autenticação possa ser completada. No exemplo visualizado na Fig. 11, o PIN do cliente é 2, 4, 6 e 8. O OTC é portanto, o segundo, quarto, sexto e oitavo dígitos da *string* de segurança, ou seja, 1, 3, 2 e 6.

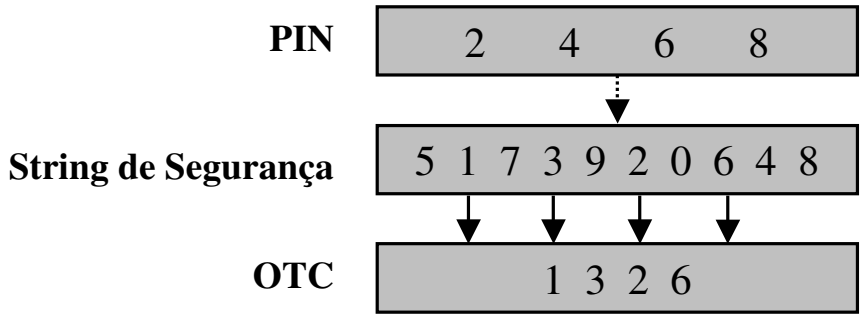


Figura 11 - Funcionamento do PINsafe

3.3.5 Certificado Digital

Autenticação com o uso de certificados digitais também vem sendo aplicada pelos bancos. Com a utilização do certificado digital, os bancos podem determinar a identidade do cliente e ainda validar as transações por ele efetuadas com o uso de assinatura digital. Os certificados podem ser criados pela instituição financeira para uso específico com seus clientes, ou podem ser emitidos por uma Autoridade Certificadora aceita pela instituição financeira.

No Brasil, a Federação Brasileira de Bancos (Febraban) possui um protocolo assinado com a Secretaria da Receita Federal e o Instituto de Tecnologia da Informação (ITI) para viabilizar a assinatura de documentos eletrônicos e transações bancárias com a utilização dos certificados digitais seguindo o padrão do e-CPF para pessoas físicas

(Fig. 12) e e-CNPJ para pessoas jurídicas. Os bancos habilitados podem atuar como Autoridades Registradoras, onde o cliente poderá solicitar o seu e-CPF ou e-CNPJ. Posteriormente, o banco solicita à Autoridade Certificadora credenciada na Receita Federal, a emissão do Certificado Digital de seu cliente.

Figura 12 - e-CPF

3.4 Conclusão

As facilidades proporcionadas pelo sistema de *Internet Banking* são inquestionáveis. Entretanto, os ataques a esses sistemas vêm aumentado tanto em número quanto em complexidade, fazendo com que o risco de fraudes em transações bancárias gere uma resistência ao uso desses sistemas por parte dos clientes. Quando uma falha na segurança ocorre, não é somente o custo direto em termos de valores monetários perdidos que deve ser analisado. A confiança no banco acaba sendo questionada por parte do cliente fazendo com que por vezes, o cliente inicie uma análise de outras opções de bancos. Além disso, a adesão de novos clientes também pode ser afetada.

Muitos clientes de bancos não utilizam o *Internet Banking* por não confiarem na segurança oferecida por essa aplicação. Para as instituições financeiras, obter segurança na autenticação de seus clientes no *Internet Banking* continua sendo tema de várias pesquisas, no intuito de obter um nível satisfatório em tal quesito. Existem no mercado diversas soluções em busca de autenticação segura dos clientes bancários no acesso *online* ao banco, seguindo a tendência no uso de mais de um fator de autenticação.

4 ASSINATURA DIGITAL

4.1 Introdução

No meio digital, a necessidade de provar um ato ocorrido no passado é atendida pelo uso de assinaturas digitais. A assinatura digital permite criar uma relação única entre uma mensagem eletrônica e um par de chaves assimétricas vinculadas à entidade que efetivou a assinatura.

O presente capítulo apresenta conceitos relacionados à assinatura digital, com ênfase no *Digital Signature Algorithm* (DSA), descrevendo suas características e possíveis ataques. Para um melhor entendimento do capítulo, uma fundamentação matemática será apresentada.

4.2 Conceituação

O objetivo da assinatura digital, assim como da assinatura manuscrita, é indicar que o usuário que assinou a mensagem (signatário) concorda com o seu conteúdo. Conforme citado por Menezes, Oorschot e Vanstone (1996), as assinaturas digitais permitem verificar a autenticidade e integridade do documento assinado, bem como prover não-repúdio, no sentido de que o signatário não poderá negar que utilizou sua chave privada para produzir uma assinatura. Para obter tais propriedades, os algoritmos de assinatura digital fazem uso dos algoritmos de chaves públicas.

Um esquema de assinatura digital consiste de um algoritmo de geração e um algoritmo de verificação de assinatura (STINSON, 1995). O signatário assina a mensagem utilizando sua chave privada e o receptor da mensagem verifica a assinatura utilizando a chave pública do signatário. As duas chaves estão matematicamente relacionadas. Como somente o signatário é possuidor de sua chave privada, se a assinatura for válida significa dizer que foi o proprietário da chave privada que a gerou. A assinatura digital de uma mensagem é, portanto, um resultado matemático que depende do documento eletrônico assinado e da chave privada do signatário.

4.3 Fundamentos Matemáticos

Esta seção apresenta fundamentos matemáticos que estão relacionados ao presente trabalho. As informações citadas a seguir podem ser encontradas com maiores detalhes em (FÄRNQVIST, 2005; MACLANE; BIRKHOFF, 1993; MENEZES; OORSCHOT; VANSTONE, 1996; SCHNEIER, 1996; STALLINGS, 2003).

4.3.1 Definições e Teoremas

Aritmética Modular: A operação a mod n denota o resíduo (resto) de a quando dividido por n, tal que o resíduo é algum inteiro entre 0 e $n - 1$. Esta operação é chamada redução modular. Os inteiros a, b são congruentes mod n se, divididos por um número n, resultam no mesmo resto. A notação utilizada é $a \equiv b \bmod n$.

Os algoritmos criptográficos utilizam com freqüência a aritmética modular, dado que alguns cálculos quando realizados mod n são problemas de difícil resolução e agregam segurança ao sistema. A aritmética modular é também mais simples de trabalhar com os computadores uma vez que restringe a faixa de valores que são manipulados (sempre entre 0 e $n - 1$).

Número Primo: um inteiro $p > 1$ é primo se for divisível somente por 1 e por ele mesmo. Schneier (1996) cita que para números menores que n a probabilidade de um número randômico ser primo é aproximadamente 1 em ln n. Assim, o limite superior para a quantidade total de números menores que n e que sejam primos é obtida pela fórmula $n/(\ln n)$.

Maior Divisor Comum: o maior divisor comum entre a e b é o maior inteiro positivo que divide tanto a quanto b. A notação utilizada é mdc (a,b).

Relativamente primo: dois inteiros a e b são ditos relativamente primos se o maior divisor comum entre ambos for 1, ou seja, mdc $(a,b) = 1$.

Inverso Multiplicativo Modular: o inverso multiplicativo de a modulo n é um inteiro $x < n$, tal que $ax \bmod n = 1$. O inteiro a possui um inverso multiplicativo módulo n somente se mdc $(a,n) = 1$, ou seja, se a e n forem números relativamente primos. Se n for um número primo, então todos os números entre 1 e n - 1 são relativamente primos a n e portanto, possuem um inverso mod n. A notação utilizada para o inverso multiplicativo de a é a^{-1}.

Função Totiente de Euler: número de inteiros menores que n e relativamente primos a n. A notação utilizada é $\phi(n)$. Para casos onde n é primo, então $\phi(n) = n - 1$.

Definição de Grupo: Um grupo G, denotado por $\{G, \bullet\}$ é um conjunto de elementos onde $\bullet$ é uma operação binária, genericamente de adição ou multiplicação, originando grupos respectivamente aditivos ou multiplicativos cujas propriedades listadas a seguir são existentes:

- Elemento Identidade: $\forall a \in G, \exists e \in G: a \bullet e = a$
  - Para as operações binárias de adição e multiplicação, o elemento identidade é definido como 0 e 1 respectivamente.
- Elemento Inverso: $\forall a \in G, \exists a' \in G: a \bullet a' = e$
  - Para as operações binárias de adição e multiplicação, o elemento inverso é escrito como $-a$ e a^{-1} respectivamente.
- Associatividade: $\forall a,b,c \in G, a \bullet (b \bullet c) = (a \bullet b) \bullet c$
- Comutatividade: $\forall a,b \in G, a \bullet b = b \bullet a$
  - O grupo que possui esta propriedade é chamado de grupo abeliano.

Definição de Corpo: Um corpo F denotado por $\{F,+, .\}$ é um conjunto de elementos com duas operações binárias chamadas adição e multiplicação, cujas propriedades são existentes:

- {F,+} é um grupo abeliano;
- Distributividade: $a\ (b + c) = ab + ac$

Se o corpo possui um número finito de elementos é então chamado de corpo finito.

O conjunto de inteiros módulo n, denotado Z_n, é o conjunto de inteiros menores que n cujas operações de adição e multiplicação são realizadas módulo n. O conjunto Z_n é um corpo se n for primo.

O grupo multiplicativo Z_n é definido como Z_n^*. A ordem de Z_n^* é definida como sendo o número de elementos de Z_n^*.

4.3.2 Geradores

Um número g é gerador mod n se para cada número $b < n - 1$, existe um número a tal que $g^a \equiv b \bmod n$.

Por exemplo: para $n = 11$, os números 2, 6, 7 e 8 são geradores. A Tabela 2 mostra que todo número de 1 até 10 pode ser escrito na forma $2^a \bmod 11$.

Tabela 2 - Número 2 gerador mod 11

a	10	1	8	2	4	9	7	3	6	5
$2^a \bmod 11$	1	2	3	4	5	6	7	8	9	10

Se Z_n^* possui um gerador então ele é chamado de cíclico. Se Z_n^* é cíclico, então o número de geradores existentes é $\phi(\phi(n))$. Para Z_p^* com p sendo primo, então necessariamente Z_p^* é cíclico. Neste caso o número de geradores é obtido pela fórmula $\phi(p - 1)$. Se g é gerador de Z_n^*, então $b = g^a \bmod p$ é também um gerador de Z_n^* se gcd $(a,\phi(n)) = 1$.

4.3.3 Funções One-Way

A noção de função *one-way* é o centro da criptografia de chave pública (SCHNEIER, 1996). Essas funções são fáceis de serem calculadas, porém, o seu inverso é de difícil resolução. Em outras palavras, dado x é fácil calcular $f(x)$, porém, dado $f(x)$ é difícil calcular x. O termo "difícil resolução" é definido como um problema computacionalmente impraticável de ser resolvido, ou seja, mesmo com todos os computadores do mundo disponíveis, o problema levaria muitos anos para ser resolvido.

Uma função *trapdoor one-way* é um tipo especial de função *one-way* com um segredo adicionado, de modo que seja fácil calcular x a partir de $f(x)$ tendo conhecimento do segredo. Algoritmos de chave pública são baseados em funções *trapdoor one-way*.

4.3.4 Logaritmo Discreto em Corpos Finitos

A segurança de muitos algoritmos de criptografia de chave pública baseia-se na intratabilidade do problema do logaritmo discreto: dado um primo p, um gerador g de

Z_p^*, encontrar um inteiro x, $0 \leq x \leq p - 1$, tal que $g^x \equiv y \bmod p$. A equação $y = g^x \bmod p$, que é uma função modular, é fácil de ser calculada. O inverso da função modular é encontrar o logaritmo discreto e constitui-se de um problema de difícil resolução (SCHNEIER, 1996).

Alguns algoritmos criptográficos como o DSA têm sua segurança firmada também na intratabilidade do problema do logaritmo discreto em subgrupos de Z_p^*, sendo descrito a seguir:

Seja um primo p, um primo q divisor de p - 1, um subgrupo cíclico G de Z_p^* de ordem q e g um gerador de G. O problema do logaritmo discreto é encontrar um inteiro x, $0 \leq x \leq q - 1$, tal que $g^x \equiv y \bmod p$.

4.4 Função de Hash

Uma função de *hash* é uma função criptográfica pública que, a partir de uma mensagem de tamanho arbitrário, produz uma mensagem de tamanho fixo chamada resumo da mensagem ou *hash* (STINSON, 1995). As funções de *hash* são utilizadas em conjunto com algoritmos de assinaturas digitais, como o DSA, os quais aplicam a função de *hash* sobre a mensagem a ser assinada. O *hash*, que representa a mensagem, é então assinado no lugar desta. Além de reduzir o volume de informações a participar do processo de assinatura, o que agiliza o processo, as funções de *hash* evitam que mensagens assinadas e assinaturas sejam modificadas voluntária ou involuntariamente (CHANG; CHANG, 2004). Na validação da assinatura digital, a mensagem recebida que corresponde à assinatura, será submetida novamente à mesma função *hash,* participando do processo de validação. Qualquer alteração na mensagem será identificada no momento da verificação da assinatura, uma vez que o *hash* obtido não será o mesmo criado na geração da assinatura.

Uma função *hash h* com *n* bits de entrada deve satisfazer as seguintes propriedades (MENEZES; OORSCHOT; VANSTONE, 1996; SCHEINER; KELSEY, 2005):

1. Resistência à colisão forte: um atacante não deveria ser capaz de encontrar duas mensagens *M, M'* tal que o *hash* gerado por ambas seja igual, de modo que $h(M) = h(M')$ com um esforço inferior a $2^{n/2}$;

2. Resistência ao cálculo de pré-imagem ou função de caminho único: um atacante, dado um *hash y*, não deveria ser capaz de encontrar uma mensagem *M* tal que h(*M*) = *y* com um esforço inferior a 2^n;
3. Resistência ao cálculo de segunda pré-Imagem ou resistência à colisão fraca: um atacante, dado uma mensagem *M*, não deveria ser capaz de encontrar uma segunda mensagem *M'* tal que *hash* (*M*) = *hash* (*M'*) com um esforço inferior a 2^n.

A quebra de uma função *hash* que produz *n* bits de resumo diz respeito ao ataque à colisão forte com um esforço inferior a $2^{n/2}$ ou ataque de primeira ou segunda pré-imagem com um esforço inferior a 2^n. Stinson (1995) cita que é suficiente que uma função *hash* atenda a propriedade de resistência à colisão forte, uma vez que isso implica em satisfazer as duas outras propriedades citadas.

4.4.1 Ataque do Aniversário

O ataque de colisão nas funções *hash* é também conhecido como ataque do aniversário. O ataque do aniversário segue um padrão de problema de estatística e especifica que 253 pessoas devem estar em uma sala para que pelo menos uma delas compartilhe a mesma data de aniversário pré-selecionada. Porém, para que no mínimo duas pessoas compartilhem a mesma data de aniversário, apenas 23 pessoas precisam estar nesta sala.

Com relação aos ataques às funções *hash*, Schneier (1996) especifica que o ataque à colisão forte, isto é, encontrar duas mensagens aleatórias (*M* e *M'*) que gerem o mesmo *hash*, é análogo ao ataque do aniversário. É mais fácil achar duas mensagens que resultem em um mesmo *hash* randômico, do que achar uma mensagem que resulte em um *hash* de um valor determinado. Isto é chamado o "paradoxo do aniversário".

Menezes, Oorschot e Vanstone (1996) citam que um signatário desonesto poderia utilizar o ataque do aniversário às funções de *hash*, apresentando sua assinatura sobre a mensagem M_1 e mais tarde negar (repudiar) essa assinatura dizendo que sua assinatura era sobre a mensagem M_2 e não sobre M_1. Outra forma de uso do ataque do aniversário é por um verificador desonesto que é capaz de convencer o signatário a assinar uma mensagem M_1 e posteriormente dizer que o signatário assinou na verdade, M_2. Por esse motivo a importância de uma função *hash* ser resistente ao ataque de colisão forte.

4.4.2 Secure Hash Standard (SHS)

O *National Institute of Standards and Technology* (NIST) é uma agência federal que emite padrões criptográficos para serem adotados pelos sistemas computacionais dos EUA. Muitos dos padrões são direcionados às agências governamentais, porém o setor privado também costuma adotá-los. Os padrões emitidos pelo NIST são publicados como *Federal Information Processing Standards* (FIPS).

Para compor um padrão de função de *hash* seguro, o NIST projetou o *Secure Hash Standard* (SHS) através da publicação FIPS PUB 180-1. Esse documento traz especificações da função *Secure Hash Algorithm* (SHA-1) que recebe uma mensagem de qualquer tamanho $< 2^{64}$ bits como entrada e produz um *hash* de tamanho fixo com 160 bits. Em 2002, o NIST publicou o FIPS PUB 180-2 (NIST, 2002), apresentando uma nova família de funções *hash* conhecidas como SHA-2. Nesse documento, o NIST adicionou três novos algoritmos *hash* para o padrão SHS, os quais são SHA-256, SHA-384 e SHA-512, gerando resumos de 256, 384 e 512 bits respectivamente. O objetivo indicado pelo NIST na criação de novas funções *hash* seria de fornecer uma compatibilidade ao aumento de segurança proporcionado por novos algoritmos criptográficos que surgiram (STALLINGS, 2003). Em fevereiro de 2004, o NIST publicou uma mudança no documento FIPS 180-2 chamado FIPS 180-2 *CHANGE NOTICE 1*. Essa mudança adiciona a função *hash* SHA-224 que gera resumo de 224 bits a partir do resumo gerado pelo SHA-256.

Na Tabela 3, podemos visualizar um resumo comparativo das principais diferenças entre os algoritmos citados.

Tabela 3 - Propriedades Secure Hash Algorithm

Algoritmo	Mensagem (bits)	Resumo Gerado (bits)	Segurança (bits)
SHA-1	$<2^{64}$	160	80
SHA-224	$<2^{64}$	224	112
SHA-256	$<2^{64}$	256	128
SHA-384	$<2^{128}$	384	192
SHA-512	$<2^{128}$	512	256

Fonte: Adaptado de NIST (2002)

A segurança referenciada na Tabela 3 diz respeito ao fato de que o ataque do aniversário no resumo gerado produz uma colisão com um esforço de aproximadamente $2^{n/2}$, onde *n* é o tamanho do resumo gerado.

Em fevereiro de 2005, Xiaoyun Wang, Yiqun Lisa Yin e Hongbo Yu anunciaram um ataque que poderia encontrar colisão no SHA-1 requerendo não mais que 2^{69} operações, ao invés de 2^{80} operações que deveria ser requerido para a quebra do SHA-1. A versão completa do artigo foi publicada em agosto na conferência Crypto 2005 (WANG; YIN; YU, 2005). Na sessão *Rump Session* desta mesma conferência, uma melhoria neste ataque foi apresentada por Xiaoyun Wang, Andrew Yao e Frances Yao reduzindo o esforço de encontrar a colisão no SHA-1 para 2^{63} operações (WANG; YAO; YAO, 2005).

A partir do anúncio de tais ataques, o NIST vem tomando algumas atitudes como orientando a transição do uso do SHA-1 para a família de funções *hash* SHA-2, as quais são assumidas como sendo funções *hash* mais forte do que SHA-1. O NIST vem também encorajando os pesquisadores a entender melhor o projeto de funções *hash* e possíveis ataques, para que possam estar preparados para a escolha de novas funções de *hash*. O NIST realizou em novembro de 2005 o *Cryptographic Hash Workshop*, com o objetivo de realizar discussões a respeito dos padrões de funções *hash* existentes e planeja realizar vários outros para fortalecimento da pesquisa nessa área. E ainda, o NIST pretende realizar no futuro uma competição de função *hash* para a adoção de um novo padrão.

Com relação à migração da função *hash* SHA-1 para alguma função da família SHA-2, em alguns casos é necessário uma atenção especial. Existem algoritmos como o DSA que requerem *hash* de tamanho inferior ao gerado pela família SHA-2. Devido a tal necessidade, novas pesquisas buscam alternativas de reduzir o tamanho do resumo gerado por funções *hash* mais fortes para se adaptarem a tamanhos de resumos menores esperados por determinados algoritmos.

O documento FIPS 180-2 *CHANGE NOTICE 1* cita que em casos de aplicações que necessitem um tamanho de saída inferior à saída fornecida pela função a ser utilizada, pode-se efetuar o truncamento do *hash.* O truncamento é efetuado selecionando-se um número apropriado de bits mais à esquerda. Exemplifica ainda que, se um resumo de 96 bits é desejado, pode-se utilizar o SHA-256 e os 96 bits mais à

esquerda do resumo gerado pelo SHA-256 devem ser selecionados (descartando os 160 bits mais à direita).

4.5 Noção de Segurança em Esquemas de Assinatura Digital

Um atacante de esquema de assinatura digital tem como objetivo forjar assinaturas, ou seja, produzir assinaturas digitais em nome do signatário. Os objetivos de quebra de um esquema de assinatura digital podem ser classificados como apresentado a seguir (MENEZES; OORSCHOT; VANSTONE, 1996; NACCACHE; POINTCHEVAL; STERN, 2001):

- Quebra total: um atacante é capaz de descobrir a chave privada do signatário;
- Falsificação seletiva: um atacante é capaz de criar uma assinatura válida para uma certa mensagem escolhida anteriormente;
- Falsificação existencial: um atacante é capaz de falsificar uma assinatura para no mínimo uma mensagem. O atacante não tem nenhum controle sobre a mensagem cuja assinatura será obtida (muito provavelmente, sendo uma mensagem sem significado).

Para um atacante falsificar assinaturas, os seguintes meios podem ser utilizados:

- Ataque de somente chave: o atacante conhece apenas a chave pública do signatário;
- Ataque de mensagem conhecida: o atacante obtém uma lista de assinaturas e mensagens conhecidas, mas não escolhidas por ele;
- Ataque de mensagem escolhida: o atacante obtém uma lista de assinaturas sobre mensagens escolhidas por ele. As mensagens são escolhidas antes de qualquer assinatura ter sido analisada;
- Ataque de mensagem escolhida adaptável: o atacante pode solicitar ao signatário que gere assinaturas sobre mensagens que foram escolhidas previamente e cuja seleção foi adaptada baseada em assinaturas anteriores.

Naccache, Pointcheval e Stern (2001) citam que o mais alto nível de segurança de um esquema de assinatura é ser seguro contra falsificação existencial em um ataque de mensagem escolhida adaptável.

4.6 Digital Signature Algorithm (DSA)

O *Digital Signature Algorithm* (DSA) é o algoritmo de assinatura digital proposto pelo NIST em 1991, especificado como o padrão de assinatura digital conhecido como *Digital Signature Standard* (DSS) através da publicação FIPS 186 de 1994. Posteriormente em 1998, o NIST fez uma revisão e publicou o FIPS 186-1, o qual adicionou ao DSS o algoritmo *Digital Signature Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA),* sendo um algoritmo baseado no RSA e especificado pelo padrão ANSI X9.31. Em 1999, nova alteração foi realizada originando o FIPS 186-2, que acrescentou o *Elliptic Curve Digital Signature Algorithm (ECDSA)* especificado pelo padrão ANSI X9.62.

O algoritmo DSA especificado em tais publicações utiliza chaves que variam de 512 a 1024 bits. Um *Draft* do FIPS 186-3 foi publicado pelo NIST em março de 2006 que, entre outras alterações, especifica o uso de chaves no DSA de 1024, 2048 e 3072 bits. O NIST está em fase de avaliação dos comentários públicos recebidos sobre o *Draft* do FIPS 186-3. Como o documento *Draft* do FIPS 186-3 ainda não havia sido publicado como um documento oficial até a finalização deste trabalho, a descrição do algoritmo DSA apresentada neste é baseada no documento FIPS 186-2 (NIST, 2000).

O DSA tem sua segurança baseada na dificuldade de computar logaritmos discretos em Z_p^* e em subgrupos de ordem q em Z_p^* (STALLINGS, 2003; STINSON, 1995).

4.6.1 Os Parâmetros

A assinatura digital gerada pelo DSA é calculada utilizando um conjunto de parâmetros de domínios, uma chave privada, um número secreto gerado por mensagem e a função *hash* SHA-1. A assinatura digital é verificada utilizando o mesmo conjunto de parâmetros de domínio, uma chave pública (matematicamente associada à chave privada) e a função *hash* SHA-1. Os parâmetros do DSA são especificados como a seguir:

***p*:** um módulo primo tal que $2^{512} < p < 2^{1024}$

***q*:** um primo divisor de p - 1, tal que $2^{159} < q < 2^{160}$

g: um gerador do subgrupo de ordem q mod p. É obtido pela fórmula $h^{(p-1)/q}$ mod p, tal que h é um inteiro $1 < h < p - 1$ e $h^{(p-1)/q} \bmod p > 1$

***x*:** um número inteiro randômico ou pseudo-randômico tal que $0 < x < q$ (chave privada)

y: $g^x \bmod p$ (chave pública)

k: um número inteiro randômico ou pseudo-randômico que deve ser secreto e único por mensagem, tal que $0 < k < q$

Os parâmetros p, q e g são os parâmetros de domínio que podem ser públicos ou comuns a um grupo de usuários.

4.6.2 O Algoritmo de Geração da Assinatura

O algoritmo de geração da assinatura sobre uma mensagem M consiste de um par de números inteiros r e s que é calculado conforme as equações abaixo:

$$r = (g^k \bmod p) \bmod q$$

$$s = (k^{-1}(\text{SHA-1}(M) + xr)) \bmod q$$

A assinatura é gerada com 320 bits (160 para r e 160 para s).

Se r ou s resultarem em valor igual a zero, um novo valor de k deve ser gerado e a assinatura deve ser calculada novamente.

O valor de r pode ser pré-computado juntamente com os valores de k e k^{-1}, uma vez que para efetuar o cálculo de r, a mensagem a ser assinada não é requerida.

4.6.3 O Algoritmo de Verificação da Assinatura

Antes de efetuar o processo de validação da assinatura, os parâmetros de domínio e chave pública do signatário devem ter sido disponibilizados ao verificador de uma maneira segura (como por exemplo via certificado digital assinado por uma entidade confiável).

Sejam M', r', s' as versões recebidas pelo verificador e correspondentes a M, r e s. O verificador deve inicialmente checar se $0 < r' < q$ e $0 < s' < q$. Se ambas as condições não forem verdadeiras, então a assinatura é considerada inválida. Se as condições forem satisfeitas, o verificador calcula:

$$w = (s')^{-1} \bmod q$$

$$u1 = ((\text{SHA-1}(M'))w') \bmod q$$

$$u2 = (r'w) \bmod q$$

$$v = (((g)^{u1} (y)^{u2}) \bmod p) \bmod q$$

Se $v = r'$, então a assinatura é válida.

Se $v \neq r'$, então uma das possibilidades abaixo ocorreu:

- A mensagem ou a assinatura foram modificadas (problemas de integridade);
- Pode haver um erro no processo de geração de assinatura do signatário;
- Um atacante pode ter tentado forjar uma assinatura.

Neste caso, a assinatura é considerada inválida.

4.6.4 Análise de Segurança do DSA

A segurança do DSA é fundamentada na intratabilidade do logaritmo discreto. Entretanto, ainda que o problema do logaritmo discreto seja considerado intratável, outros requisitos devem ser observados para que o DSA mantenha um nível de segurança adequado. Veremos a seguir, questões de segurança a respeito do problema do logaritmo discreto bem como outros possíveis ataques relacionados ao DSA, descrevendo meios de serem evitados.

4.6.4.1 Ataques Relacionados ao Problema do Logaritmo Discreto

Uma das primeiras tentativas de ataque ao DSA é tentar obter a chave privada x pelo ataque ao problema do logaritmo discreto (DLP) em $y = g^x \bmod p$. O ataque mais elementar diz respeito à busca exaustiva (força bruta) que consiste em calcular exaustivamente $g^{x1} \bmod p$, $g^{x2} \bmod p$ até que o valor procurado y seja encontrado. Entretanto, é um ataque extremamente ineficiente.

Existem outras formas de ataque ao DLP, como o de *Pohlig-Hellman* (POHLIG; HELLMAN, 1978), que é evitado selecionando p tal que $p - 1$ tenha fatores primos grandes. Há também o de *Shanks*, que utiliza excessivo espaço de armazenamento e o de *Pollard-rho*, todos os quais funcionando em tempo exponencial (MENEZES; OORSCHOT; VANSTONE, 1996). Métodos mais elaborados são os chamados Métodos de Cálculo de Índice que são muito mais eficazes que os citados anteriormente, porém, também são ineficientes para valores suficientemente grandes de p.

A segurança dos esquemas que se baseiam no DLP contra os ataques cuja meta é resolver o problema do logaritmo discreto, depende da dificuldade dos métodos citados acima e dos métodos de busca de colisão (IEEE P1363 WORKING GROUP, 2000, seção D.4.1). A colisão, no caso de assinaturas, ocorre quando mensagens distintas resultam em uma mesma assinatura. O tamanho do corpo p e da ordem do subgrupo q deveria ser selecionado de tal forma que ambos os ataques (quebra do logaritmo discreto e busca de colisão de assinatura) sejam impraticáveis. Para ser seguro, o documento IEEE P1363 Working Group (2000) especifica que o tamanho mínimo do corpo p e do subgrupo de ordem q devam ser de 1024 e 160 bits respectivamente.

Neste sentido, tais ataques ao logaritmo discreto e ao subgrupo são se aplicam ao DSA, uma vez que os parâmetros, por definição do algoritmo, possuem tamanho considerado seguro contra tais ataques.

4.6.4.2 Ataques sobre o Parâmetro *k*

Por definição do algoritmo DSA, o parâmetro k deve ser secreto. Stinson (1995) ressalta que, se o parâmetro k for conhecido, a chave privada x poderá ser facilmente quebrada. Com o conhecimento do k, um atacante poderia obter a chave privada x através da equação $x = (ks - \text{SHA-1}(M))\ r^{-1}$ (originada da equação de s). Como s, SHA-1(M) e r são conhecidos (e neste caso, também o parâmetro k), a chave privada x será quebrada diretamente por substituição de valores.

A definição do algoritmo cita ainda que o parâmetro k deve ser único por assinatura. Caso duas mensagens sejam assinadas com mesmo parâmetro k, através da equação de s poderá ser quebrada a chave privada x (STINSON, 1995). Neste caso, x pode ser obtido resolvendo-se um sistema de equações lineares.

Em 2001, o pesquisador Daniel Bleichenbacher (BLEICHENBACHER, 2001) propôs um ataque ao DSA, aproveitando uma falha na rotina dos geradores de pseudo-números randômicos (PRNG) especificada no Apêndice 3 do documento NIST FIPS 186-2. A falha está relacionada a não uniformidade na geração do parâmetro k (que é gerado para cada assinatura) oriundo da rotina PRNG, isto é, os números gerados pela rotina são selecionados com mais freqüência dentro de uma faixa específica. No ataque citado, utilizando a rotina proposta pelo NIST, a chave secreta x pode ser obtida com 2^{22} assinaturas, 2^{64} de tempo computacional e 2^{40} de espaço de memória. Diante desse

ataque, o NIST apresentou uma modificação na rotina PRNG que foi apresentada no documento FIPS 186-2 *CHANGE NOTICE 1*. Segundo este documento, esse ataque pode ser evitado limitando-se o número de assinaturas criadas com um específico par de chaves, a dois milhões de assinaturas (se for utilizada a PRNG especificada em FIPS 186-2), ou então, modificando-se a PRNG conforme sugerido no FIPS 186-2 *CHANGE NOTICE 1*.

Caso alguns bits do parâmetro k sejam conhecidos, a segurança da chave privada x pode estar comprometida. É o que indica o trabalho apresentado por Howgrave-Graham e Smart (2001), o qual descreve que dentro de certas condições, se para um determinado número de assinaturas 8 bits do parâmetro k são conhecidos, a chave privada x poderá ser descoberta. Nguyen e Shparlinski (2002) avançaram na pesquisa e validaram um ataque onde, em alguns minutos, descobriram a chave secreta x de 160 bits do DSA quando os 3 bits menos significativos do parâmetro k eram conhecidos em 100 assinaturas. Bleichenbacher (2005) apresentou algumas experiências sobre certas implementações do DSA que deixam um bit do parâmetro k ser conhecido a cada assinatura, o que segundo sua observação, é suficiente para quebrar a chave privada x em um computador normal. Um exemplo citado é a biblioteca AT&T's CryptoLib, onde o parâmetro k gerado é sempre ímpar, e portanto, deixa em aberto o bit menos significativo de tal parâmetro. Segundo Nguyen e Shparlinski (2002) talvez isso aconteça porque as mesmas rotinas citadas são utilizadas na implementação de esquemas de assinaturas El Gamal para o qual o parâmetro k deve ser relativamente primo a p - 1. Para prevenir os ataques de Bleichenbacher, Nguyen, Shparlinski, Howgrave-Graham e Smart, o parâmetro k gerado para cada assinatura deve ser uniformemente distribuído entre 1 e q - 1.

4.6.4.3 Ataque de Vaudenay - Colisão de Assinaturas

Vaudenay (1996) apresentou um ataque que atua na geração dos parâmetros do DSA, no intuito de obter a colisão de assinaturas. Dessa forma, o usuário assina uma mensagem e um atacante poderia substituir tal mensagem por outra, uma vez que a assinatura é a mesma. O ataque citado parte da seleção do parâmetro q de tal modo que SHA-1(M) ≡ SHA-1(M') mod q. No ataque sugerido, o parâmetro q é obtido pela

equação q = SHA-1(M) – SHA-1(M'), tendo sucesso desde que o parâmetro q gerado seja de 160 bits.

Para evitar a colisão de assinatura citada, Vaudenay comenta que pode-se adicionar um número randômico antes de efetuar o *hash* da mensagem. Dessa forma, o ataque não teria sucesso uma vez que o atacante não saberia previamente o valor final do *hash* que será gerado sobre determinada mensagem.

Caso os parâmetros públicos não sejam gerados por entidade confiável, o usuário deve checar se esses foram gerados de maneira adequada.

4.7 Tipos Especiais de Assinatura Digital

Em algumas aplicações, certas propriedades especiais são desejadas e não são obtidas com o uso de assinatura digital convencional. Para esses casos, o esquema básico de assinatura digital é combinado com protocolos específicos com o objetivo de obter funcionalidades adicionais (MENEZES; OORSCHOT; VANSTONE, 1996). Alguns tipos de assinaturas especiais serão descritos nas próximas seções.

4.7.1 Assinatura Cega

Assinatura cega é uma forma especial de assinatura digital em que o signatário não tem conhecimento sobre a informação que irá assinar. O conceito foi introduzido por Chaum no Crypto'82 (CHAUM, 1983) e é de grande importância em aplicações como voto eletrônico e esquemas de dinheiro eletrônico. Chaum propôs um esquema de assinatura cega baseada no algoritmo de assinatura RSA. O esquema é descrito a seguir, onde *A* recebe de *B* a assinatura sobre a mensagem *M* (LAI; CHANG, 2005):

Os Parâmetros

$n = p * q$, tal que p e q são primos

e = um número inteiro randômico tal que $1 < e < \phi(n)$ e mdc $(e, \phi(n)) = 1$

$d = e^{-1} \bmod \phi(n)$

A chave pública é o par (e,n). A chave privada é o par (d,n)

O Algoritmo de Geração da Assinatura

- *A* seleciona um número randômico k, onde $k < n$ e mdc $(k,n) = 1$;

- *A* calcula $M' = h(M)k^e \mod n$, onde *h* é uma função *hash*. *A* envia *M'* para *B* assinar;
- *B* calcula $s' = M'^d \mod n$ e envia *s'* para *A*;
- *A* calcula $s = k^{-1} s' \mod n$, onde *s* é efetivamente a assinatura de *B* sobre *M*.

O Algoritmo de Verificação da Assinatura

Calcular $s^e \mod n$. Se o resultado for igual a h(*M*), então a assinatura é válida.

Neste esquema de assinatura, o signatário *B* não tem conhecimento sobre a mensagem *M* que está sendo assinada, uma vez que a mensagem está mascarada com um número randômico *k,* o qual é removido antes do último passo da fase de geração da assinatura.

4.7.2 Assinatura Inegável

A assinatura digital convencional possui a propriedade de ser universalmente verificável, isto é, a validade de uma assinatura digital pode ser verificada por qualquer entidade. Entretanto, tal característica em algumas aplicações não é desejável, como exemplo, em situações em que o signatário assina digitalmente determinados documentos e não deseja que terceiras partes tenham conhecimento ou validem sua assinatura sem a sua devida autorização (CHOW et al., 2003). Outra aplicação prática diz respeito ao controle de cópias piratas de software. O signatário assina digitalmente o software e não deseja que tal assinatura possa ser validada por qualquer um, e sim, que seja validada apenas com o seu consentimento. Para suprir essa necessidade, em 1989, Chaum e van Antwerpen introduziram o conceito de assinaturas inegáveis. Assim como na assinatura digital convencional, uma assinatura inegável é uma seqüência de bits gerada pelo signatário, que depende da mensagem assinada e da chave privada do signatário. Entretanto, ao contrário da assinatura digital convencional, uma assinatura inegável não pode ser verificada sem a cooperação do signatário (CHAUM; van ANTWERPEN, 1990). Dessa forma, a assinatura inegável é um tipo especial de assinatura digital.

O processo de verificação é obtido por meio do protocolo de desafio-resposta. A prova da assinatura pode ser efetuada por qualquer um que gere um desafio ao

signatário e valide sua resposta. Se o teste obteve sucesso, então a assinatura é provavelmente válida. Se o teste falhar, há duas possibilidades: ou a assinatura não é válida, ou o signatário está tentando negar uma assinatura válida.

Para prevenir que o signatário negue a validade de uma assinatura verdadeira, o protocolo de assinatura inegável incorpora um protocolo de negação pelo qual será identificado se a assinatura é realmente falsa ou se o signatário está tentando negar uma assinatura legítima. Assim sendo, o esquema proposto por Chaum e van Antwerpen consiste do algoritmo de geração dos parâmetros, algoritmo de geração da assinatura, algoritmo de verificação e algoritmo de negação, os quais serão descritos a seguir (MENEZES; OORSCHOT; VANSTONE, 1996; STINSON, 1995):

Os Parâmetros

$p = 2q + 1$, tal que q é primo

$g =$ gerador de ordem de q em Z_p^*

$x =$ um número inteiro randômico ou pseudo-randômico tal que $0 < x < q$

$y = g^x \bmod p$

A chave pública é composta por p, g, y. A chave privada é o parâmetro x.

O Algoritmo de Geração da Assinatura

O signatário calcula $s = M^x \bmod p$, onde M é a mensagem a ser assinada e c é a assinatura.

O Algoritmo de Verificação da Assinatura

- O verificador seleciona dois inteiros randômicos x_1 e $x_2 < q$;
- O verificador calcula $z = s^{x1} * y^{x2} \bmod p$ e envia z para o signatário (onde z é o desafio);
- O signatário calcula $w = z^{x^{-1}} \bmod p$ e retorna w para o verificador (que é a resposta para o desafio);
- O verificador calcula $w' = M^{x1} g^{x2} \bmod p$.

Se $w = w'$ então a assinatura é verdadeira. Senão, ou a assinatura é falsa, ou o signatário está tentando negar uma assinatura legítima. Para certificação, deve ser executado o algoritmo de negação.

O Algoritmo de Negação da Assinatura

- O verificador seleciona dois inteiros randômicos x_1 e $x_2 < q$;
- O verificador calcula $z_1 = s^{x1} * y^{x2} \bmod p$ e envia z_1 para o signatário;
- O signatário calcula $w_1 = z_1^{x^{-1}} \bmod p$ e retorna w_1 para o verificador;
- O verificador calcula $w_1' = M^{x1} g^{x2} \bmod p$. Se $w_1 = w_1'$ então o verificador aceita a assinatura e o protocolo finaliza, senão continua;
- O verificador seleciona dois inteiros randômicos $x_{1'}$ e $x_{2'} < q$;
- O verificador calcula $z_2 = s^{x1'} * y^{x2'} \bmod p$ e envia z_2 para o signatário ;
- O signatário calcula $w_2 = z_2^{x^{-1}} \bmod p$ e retorna w_2 para o verificador;
- O verificador calcula $w_2' = M^{x1'} g^{x2'} \bmod p$. Se $w_2 = w_2'$ então o verificador aceita a assinatura e o protocolo finaliza, senão continua;
- O verificador calcula $c = (w_1 * g^{-x2})^{x1'} \bmod p$ e $c' = (w_2 * g^{-x2'})^{x1} \bmod p$. Se $c = c'$ então o verificador conclui que a assinatura é falsa. Se $c \neq c'$ então o verificador conclui que a assinatura é legítima e que o signatário está tentando negar uma assinatura verdadeira.

4.7.3 Assinatura Fail-Stop

A assinatura *fail-stop* é um esquema que proporciona uma característica de segurança adicional contra a possibilidade de um adversário *B* ser capaz de falsificar assinaturas em nome de *A*. Neste tipo de assinatura, *A* pode provar que as assinaturas geradas por *B* são falsas (STINSON, 1995). Assinaturas *fail-stop* tem a vantagem de, ainda que um adversário falsifique uma assinatura, a falsificação poderá ser detectada e o mecanismo de assinatura não ser mais utilizado.

Um esquema de assinatura *fail-stop* deve respeitar as seguintes propriedades:

- Se a entidade *A* assinar uma mensagem de acordo com o mecanismo previsto, então o verificador *B* deveria aceitá-la;
- Se um atacante gerar uma assinatura válida, passando assim com sucesso pelo processo de verificação, o verdadeiro signatário *A* deve ser capaz de provar que a assinatura gerada é falsa;
- Um signatário *A* não deve ser capaz de gerar assinaturas e mais tarde negar a legitimidade delas.

No esquema de assinatura convencional, uma chave pública está relacionada a uma chave privada. A idéia básica da assinatura *fail-stop* é que para toda chave pública existam muitas possibilidades de chaves privadas (mas uma entre elas é a escolhida, que é de posse do signatário). Mesmo que o atacante descubra uma chave privada válida, haverá muitas possibilidades entre elas, e a probabilidade do atacante descobrir a chave privada verdadeira é muito pequena (SCHNEIER, 1996). Ainda que o atacante falsifique uma assinatura com a chave privada que gerou, essa assinatura será diferente da assinatura gerada com a chave privada verdadeira. Para o signatário provar que a assinatura gerada pelo atacante é falsa, duas assinaturas devem ser geradas: uma com a chave privada verdadeira e outra com uma chave privada falsa, o que permitirá o cálculo de um segredo que somente o possuidor da chave privada verdadeira poderia efetuar.

4.8 Conclusão

Assinatura digital permite agregar as propriedades de autenticidade, integridade e não-repúdio à mensagem assinada. Nesse capítulo foi enfatizado o algoritmo de assinatura digital DSA, cuja segurança é firmada na intratabilidade do logaritmo discreto. Entretanto, para ser considerado seguro, vimos que alguns outros itens devem ser observados no uso do algoritmo DSA.

Na seção de Assinaturas Digitais Especiais, foi possível identificar que sempre que o modelo de assinatura digital convencional não consegue satisfazer necessidades específicas, variações sobre este modelo são combinadas com protocolos especiais.

5 PROPOSTA DE ASSINATURA DIGITAL CURTA ESPECIAL COM O DSA GERADA EM DISPOSITIVO PESSOAL

5.1 Introdução

Como apresentado no Capítulo 3, os bancos estão implantando uma variedade de soluções para autenticação segura do cliente nos sistemas de *Internet Banking*. Algumas soluções fazem uso de *tokens* com conectividade ao computador (USB *token* ou *smart card*), havendo a necessidade de instalação de software específico no computador do cliente. No caso do *smart card*, é necessário ainda um hardware adicional (a leitora do cartão). Outra alternativa que vem sendo implantada é a utilização da estrutura de telefonia celular como meio para o envio da informação de autenticação, criando assim uma dependência da disponibilidade do serviço telefônico. Em outros casos, o sistema de geração da informação de autenticação é gravado no próprio celular do cliente (operando de maneira *off-line*), requerendo para isso aparelhos mais avançados, capazes de executar programas específicos.

Uma opção que vem sendo utilizada nos sistemas de *Internet Banking* são as soluções de autenticação que utilizam *tokens* sem conectividade com o computador, os quais geram OTP ou MAC. Em alguns casos são intitulados de *tokens* capazes de gerar assinatura digital sobre a transação bancária e, entretanto, utilizam criptografia simétrica, a qual não adiciona a característica de não-repúdio às transações assinadas. Para contemplar a geração de assinatura digital, o uso de criptografia assimétrica deve ser empregado. Porém, as soluções de assinatura digital existentes geram assinaturas longas, inviabilizando a sua geração em dispositivos sem conectividade com o computador.

Neste capítulo será descrita a proposta do presente trabalho, que é a apresentação de uma solução para geração de assinatura digital especial curta de 32 bits, através de uma variação do algoritmo de assinatura DSA. A assinatura digital é curta o suficiente para ser gerada por meio de um *token* sem conectividade com o computador. Serão especificadas as entidades envolvidas no processo, apresentada a descrição da solução proposta e definidos os protocolos. Posteriormente será realizada a análise de segu-

rança, bem como a demonstração de um protótipo da solução especificada. O contexto da solução apresentada visa garantir, através da assinatura digital, o não-repúdio nas transações realizadas através do *Internet Banking*.

5.2 Entidades

As principais entidades envolvidas na proposta apresentada são descritas a seguir e o relacionamento entre elas é apresentado na Fig. 13.

- **Dispositivo Pessoal (DP):** *token* sem conectividade com o computador, que é de uso do Cliente e gera assinatura digital curta sobre as transações bancárias;
- **Cliente**: correntista do banco. É o signatário e proprietário do Dispositivo Pessoal;
- ***Internet Banking* (IB):** sistema do banco utilizado pelos Clientes para realizar transações diversas pela Internet;
- **Sistema Gerenciador de Assinaturas (SGA)**: sistema criado e mantido por uma entidade externa e de confiança do banco. O SGA é instalado na própria instituição bancária e realiza a validação das assinaturas geradas pelo Dispositivo Pessoal;
- **Base de Dados (BD):** armazena os parâmetros de validação de assinaturas do DSA de cada Cliente.

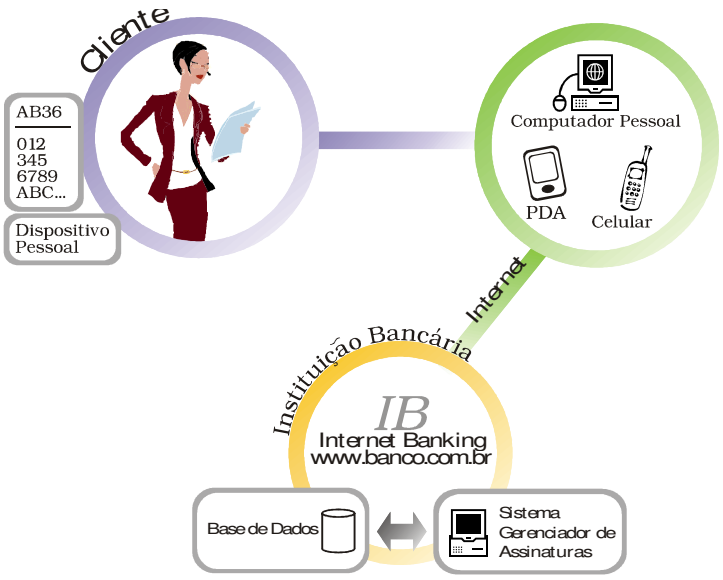


Figura 13 - Principais entidades e seus relacionamentos

5.3 Descrição do Esquema de Assinatura

O Cliente utiliza o *Internet Banking* para efetuar transações bancárias remotamente, as quais somente são efetivadas mediante a geração de sua assinatura digital sobre essas transações. Para isso, o Cliente faz uso de um Dispositivo Pessoal (*token*) que não possui conectividade com o computador. Dessa forma, para efetuar a assinatura sobre a transação, o Cliente deve digitar manualmente em seu Dispositivo Pessoal a informação a ser assinada e, posteriormente, também de forma manual, deve indicar ao *Internet Banking* a assinatura gerada pelo Dispositivo Pessoal. O *Internet Banking* encaminha a assinatura para ser validada pelo Sistema Gerenciador de Assinaturas e retorna o resultado da validação ao Cliente. A transação somente será efetivada se a assinatura digital do Cliente for válida.

Como o Dispositivo Pessoal não possui interface de conexão com o computador, não requer qualquer hardware ou software especiais instalados no computador do Cliente. Essa característica simplifica o uso do sistema e permite uma maior mobilidade por parte do Cliente.

O Dispositivo Pessoal gera assinatura digital curta de 32 bits, viabilizando a digitação manual por parte do Cliente. A assinatura é gerada através da aplicação do algoritmo de assinatura digital DSA com algumas alterações, que são listadas a seguir:

- **Parâmetro *q* reduzido**: no algoritmo DSA, *r* e *s* (resultados da assinatura) são reduzidos ao módulo *q*, o qual é de 160 bits. Dessa forma, a assinatura gerada possui 320 bits. Na proposta atual, o *q* é especificado com tamanho de 16 bits, gerando assinatura digital de 32 bits;
- **Parâmetros *p, q, g, y* ocultos**: para manter a segurança do DSA mesmo com a redução do parâmetro *q*, os parâmetros de validação de assinaturas, *p, q, g* e *y,* não são de acesso público, sendo armazenados cifrados na Base de Dados da instituição bancária e manipulados unicamente pelo Sistema Gerenciador de Assinaturas;
- **Truncamento do *hash***: no algoritmo DSA, a função *hash* utilizada na geração e validação da assinatura é o SHA-1, que gera resumo de 160 bits (do tamanho do parâmetro *q*). Na proposta atual, o *hash* é gerado com 16 bits através da aplicação da técnica de truncamento sobre a saída da função *hash* SHA-256;

- **Assinatura Cifrada**: para aumentar o nível de segurança e evitar possíveis ataques, após gerar a assinatura digital o Dispositivo Pessoal a cifra com criptografia simétrica. Assim sendo, a assinatura já é informada de maneira cifrada pelo Cliente ao *Internet Banking* para posterior validação. A cifragem da assinatura é efetuada com o algoritmo simétrico AES (NIST, 2001a) e chave de 128 bits.

O esquema apresentado é um tipo de aplicação de assinatura digital curta em um ambiente de *closed* PKI, uma vez que somente a entidade confiável, neste caso o banco, poderá proceder com a validação da assinatura digital (BRANDS, 2000; KWON, 2001). Na proposta atual, o banco é a única entidade que possui a chave pública do signatário e, portanto, único verificador de assinaturas.

A presente solução não requer o uso de certificados digitais, uma vez que a chave privada do usuário fica armazenada no Dispositivo Pessoal e os parâmetros públicos ficam armazenados diretamente na Base de Dados do banco. Isso reduz o custo da solução e proporciona simplicidade ao processo, já que não há necessidade de diversos gerenciamentos inerentes ao uso de infra-estrutura de chaves públicas.

5.4 Detalhes do Cenário

As operações bancárias através do *Internet Banking* são sempre concluídas baseadas na validade da assinatura digital do Cliente. Deste modo, o cenário aqui descrito apesar de possuir enfoque na assinatura digital sobre a transação bancária, pode ser utilizado também no processo de identificação do Cliente. Assim, tanto na identificação do Cliente quanto na confirmação de transação bancária, as operações efetuadas são as seguintes:

1. O Cliente, através do *Internet Banking,* efetua uma solicitação de identificação (*login*) ou transação bancária;
2. O *Internet Banking,* antes de aceitar a solicitação, apresenta ao Cliente o *hash* da solicitação;
3. O Cliente digita o *hash* no Dispositivo Pessoal;
4. O Dispositivo Pessoal assina o *hash*;
5. O Cliente digita no *Internet Banking* a assinatura digital sobre a solicitação;

6. O *Internet Banking* submete a validação da assinatura ao Sistema Gerenciador de Assinaturas;
7. O *Internet Banking* informa ao Cliente se a solicitação foi aceita ou não.

Os dados trafegam entre o Cliente e *Internet Banking*, e vice-versa, através de um canal seguro (SSL).

A seguir, serão descritos características e detalhes do funcionamento da solução com relação às duas partes envolvidas no processo: o Cliente e o Banco.

5.4.1 Lado do Cliente

Para efetuar assinaturas digitais sobre as transações bancárias, o Cliente faz uso do Dispositivo Pessoal, o qual possui um visor e um teclado hexadecimal. Não possui interface de conexão com o computador e é acionado através de um PIN de conhecimento do Cliente. Após a indicação de *n* tentativas incorretas de PIN, o Dispositivo Pessoal é bloqueado.

Os parâmetros do DSA (*p, q, g ,y, x*) do Cliente bem como a chave simétrica K_{DP} utilizada para cifrar a assinatura, são gerados pelo Dispositivo Pessoal. Ao receber o Dispositivo Pessoal no banco, o Cliente deve acionar opção para inicialização dos parâmetros de assinatura e chave simétrica. Os parâmetros (*p, q, g, x, y*) e chave simétrica K_{DP} gerados ficam armazenados no Dispositivo Pessoal cifrados com AES e chave simétrica K_{PIN} que é derivada do PIN do Cliente.

Para efetivar as transações bancárias requisitadas, o *Internet Banking* apresenta ao Cliente o *hash* sobre a transação, sendo gerado com 16 bits através da aplicação de truncamento sobre a saída da função SHA-256. O Cliente, em seu Dispositivo Pessoal, digita manualmente o *hash* a ser assinado. Para proceder com a assinatura, o Dispositivo Pessoal decifra os parâmetros (*p, q, g, x, y*) e chave simétrica K_{DP} do Cliente utilizando o AES e a chave simétrica K_{PIN}, gerando a assinatura utilizando *p, q, g, x* com o DSA. Posteriormente, o Dispositivo Pessoal efetua a cifragem da assinatura gerada utilizando o algoritmo criptográfico AES e a chave simétrica $K_{DP,}$ e apresenta o resultado ao Cliente. O Cliente então procede com a digitação manual da assinatura ao *Internet Banking* que a encaminha para validação.

Além de gerar assinatura digital, o Dispositivo Pessoal fornece opções para o Cliente gerar novos parâmetros de assinatura e chave simétrica K_{DP}, bem como alterar seu PIN. Na Fig. 14 é apresentado o processo de geração de assinatura que ocorre no interior do Dispositivo Pessoal.

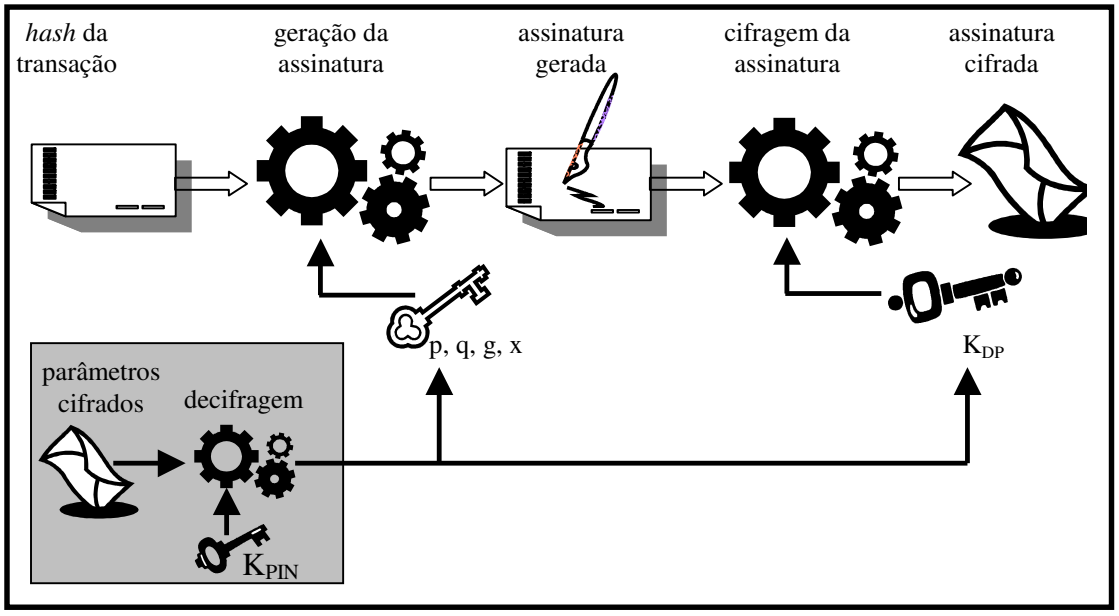


Figura 14 - Geração da assinatura no Dispositivo Pessoal

5.4.2 Lado do Banco

Uma vez gerados pelo Dispositivo Pessoal, os parâmetros de validação de assinaturas (*p, q, g, y*) e a chave simétrica K_{DP} do Cliente são informados manualmente ao banco para que as assinaturas possam ser validadas posteriormente. Essas informações ficam armazenadas na Base de Dados do banco, cifrados com RSA utilizando a chave pública KU_{SGA} do Sistema Gerenciador de Assinaturas.

Após receber a assinatura digital do Cliente, o *Internet Banking* a encaminha juntamente com a transação ao Sistema Gerenciador de Assinaturas para que seja efetuada a sua validação. Para proceder com a validação, o Sistema Gerenciador de Assinaturas decifra os parâmetros do DSA (*p, q, g, y*) e chave simétrica K_{DP} do Cliente utilizando o RSA e sua chave privada $KR_{SGA.}$ Em seguida, decifra a assinatura recebida utilizando o AES e a chave simétrica K_{DP} do Cliente e então, efetua a validação da assinatura com o DSA e parâmetros de validação (*p, q, g, y*). O Sistema Gerenciador

de Assinaturas retorna posteriormente o resultado da validação ao *Internet Banking*, o qual somente efetiva a transação do Cliente caso a assinatura digital seja válida.

O par de chaves assimétrico KR_{SGA} e KU_{SGA} fica armazenado em um dispositivo seguro dentro do banco sendo acessado apenas pelo Sistema Gerenciador de Assinaturas.

Para permitir validações futuras de assinaturas geradas com parâmetros do DSA não mais utilizados pelo Cliente, o Sistema Gerenciador de Assinaturas armazena na Base de Dados o histórico dos parâmetros de validação (*p*, *q, g, y*) e a chave simétrica K_{DP} de cada Cliente, juntamente com a data e hora de geração (*time*). Assim, ao ser solicitado para validar assinatura, o Sistema Gerenciador de Assinaturas recupera o grupo de parâmetros relativo à data da respectiva transação.

Na Fig. 15 é apresentado o processo de validação de assinatura que ocorre no Sistema Gerenciador de Assinaturas.

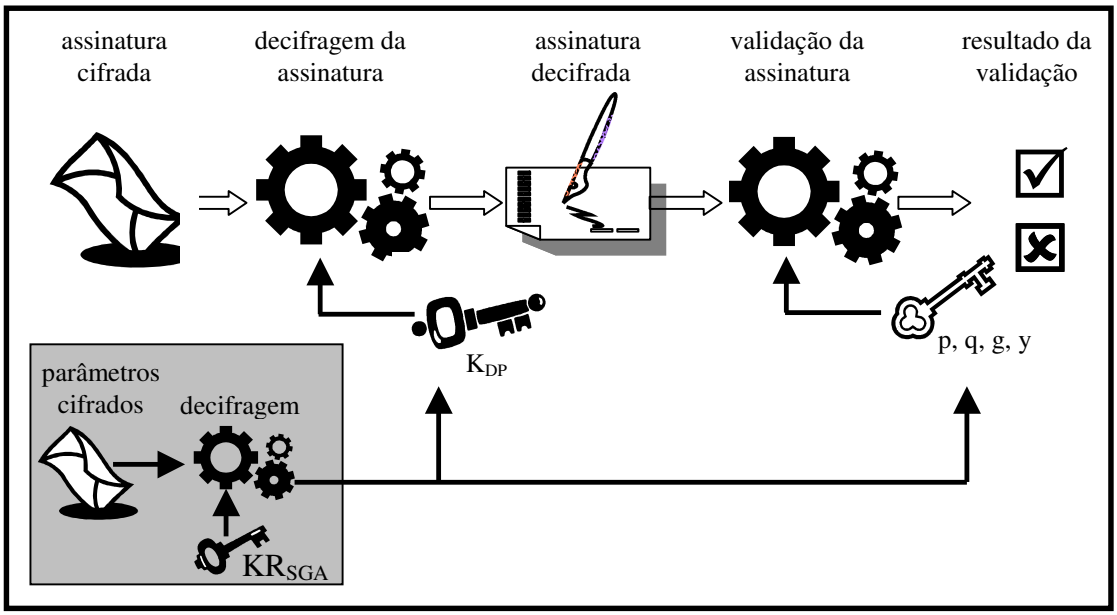


Figura 15 - Validação da assinatura no Sistema Gerenciador de Assinaturas

5.5 Notação

Na Tabela 4 é apresentada a notação utilizada no protocolo proposto.

Tabela 4 - Notação

Notação	Descrição
PIN	*Personal Identification Number* utilizado pelo Cliente para ativar o Dispositivo Pessoal
K_{DP}	Chave secreta simétrica utilizada pelo Dispositivo Pessoal para cifrar com AES a assinatura digital
K_{PIN}	Chave secreta simétrica derivada do PIN do Cliente utilizada pelo Dispositivo Pessoal com o AES para cifrar os parâmetros do DSA (*p, q, g, x, y*) bem como a chave simétrica K_{DP}
KR_C	Chave privada do Cliente, que representa o conjunto de parâmetros *p, q, g, x* para geração de assinatura com DSA
KU_C	Chave pública do Cliente, que representa o conjunto de parâmetros *p, q, g, y* para validação de assinatura com DSA
KR_{SGA}	Chave privada do SGA para decifragem dos parâmetros do Cliente com RSA
KU_{SGA}	Chave pública do SGA para cifragem dos parâmetros do Cliente com RSA
ID	Identificador do Cliente (agência, conta)
$E_{KU_{SGA}}[M]$	Cifragem da mensagem M utilizando o algoritmo assimétrico RSA com a chave pública KU_{SGA}
$D_{KR_{SGA}}[C]$	Decifragem do texto cifrado C utilizando o algoritmo assimétrico RSA com a chave privada KR_{SGA}
$E_{K_x}[M]$	Cifragem da mensagem M utilizando o algoritmo simétrico AES e a chave simétrica K_x (que pode ser K_{DP} ou K_{PIN})
$D_{K_x}[C]$	Decifragem do texto cifrado C utilizando o algoritmo simétrico AES e a chave simétrica K_x (que pode ser K_{DP} ou K_{PIN})
r	Número randômico utilizado como desafio juntamente com outras informações a serem assinadas
time	Data e hora atual
dados_operação	Dados relacionados à operação bancária a ser realizada. No caso de transferência bancária, esses dados podem ser compostos de agência e conta origem, agência e conta destino e valor a ser transferido
T	Transação a ser assinada, composta pelas seguintes informações: ID ‖ time ‖ r ‖ dados_operação
h(M)	Função *hash* com saída de 16 bits sobre a mensagem *M*
$Sig_{KR_C}[M]$	Geração de assinatura digital de 32 bits sobre mensagem *M* utilizando DSA e a chave privada KR_C
$Ver_{KU_C}[Sig_{KR_C}[M]]$	Validação da assinatura digital utilizando DSA e a chave pública KU_C
A → B:M	*A* envia mensagem *M* para *B*

5.6 Protocolos

A seqüência dos principais eventos e troca de mensagens para cada fase do esquema proposto são especificadas a seguir:

Fase 1: Recebimento do Dispositivo Pessoal pelo Cliente e Geração Inicial dos Parâmetros

O Cliente se dirige pessoalmente ao banco, recebe o seu Dispositivo Pessoal e define seu PIN. Posteriormente, deve proceder com os passos para geração inicial dos parâmetros descritos a seguir:

1. Cliente informa seu PIN ao Dispositivo Pessoal e aciona opção para inicialização dos parâmetros:
   $Cliente \rightarrow DP:PIN$
2. Dispositivo Pessoal gera os parâmetros *p, q, g, x, y* e a chave simétrica K_{DP}:
   p,q,g,x,y,K_{DP}
3. Dispositivo Pessoal apresenta *p, q, g, y* e K_{DP} separadamente ao Cliente:
   $DP \rightarrow Cliente: p,q,g,y,K_{DP}$
4. Cliente informa ao Sistema Gerenciador de Assinaturas, o seu identificador juntamente com os parâmetros de validação de assinatura (*p, q, g, y*) e a chave simétrica K_{DP}:
   $Cliente \rightarrow SGA: ID,p,q,g,y,K_{DP}$
5. Para efetuar a validação dos parâmetros recebidos, o SGA procede com a geração do *hash* sobre o *ID* do Cliente e solicita a assinatura do mesmo:
   $SGA \rightarrow Cliente: ID,h(ID)$
6. Cliente aciona no seu Dispositivo Pessoal a opção para geração de assinatura e informa manualmente o *hash* a ser assinado:
   $Cliente \rightarrow DP: h(ID)$
7. Dispositivo Pessoal gera a assinatura de 32 bits sobre o *hash*:
   $Sig_{KR_C}[h(ID)]$
8. Dispositivo Pessoal cifra a assinatura gerada utilizando AES e chave K_{DP}:
   $E_{K_{DP}}[Sig_{KR_C}[h(ID)]]$
9. Dispositivo Pessoal apresenta a assinatura cifrada ao Cliente:
   $DP \rightarrow Cliente: E_{K_{DP}}[Sig_{KR_C}[h(ID)]]$

10. Cliente informa manualmente a assinatura cifrada ao Sistema Gerenciador de Assinaturas:
    $Cliente \rightarrow SGA: E_{K_{DP}}[Sig_{KR_C}[h(ID)]]$
11. Sistema Gerenciador de Assinaturas decifra a assinatura recebida utilizando K_{DP}:
    $D_{K_{DP}}[E_{K_{DP}}[Sig_{KR_C}[h(ID)]]]$
12. Sistema Gerenciador de Assinaturas valida a assinatura utilizando os parâmetros públicos do Cliente:
    $Ver_{KU_C}[Sig_{KR_C}[h(ID)]]$
13. Caso a assinatura seja válida, o Sistema Gerenciador de Assinaturas cifra com RSA e $KU_{SGA,}$ os parâmetros p, q, g, y e a chave simétrica K_{DP} recebidos e os envia para armazenagem na Base de Dados juntamente com o ID do Cliente e a data e hora atual:
    $SGA \rightarrow BD: ID, E_{KU_{SGA}}[p,q,g,y,K_{DP}], time$
14. Sistema Gerenciador de Assinaturas retorna mensagem ao Cliente indicando que os parâmetros foram inicializados;
15. Cliente confirma os parâmetros no seu Dispositivo Pessoal;
16. Dispositivo Pessoal cifra os parâmetros gerados utilizando AES e K_{PIN}:
    $E_{K_{PIN}}[p,q,g,x,y,K_{DP}]$

Na Fig. 16, é apresentado o protocolo de geração inicial dos parâmetros.

Fase 2: Geração da Assinatura

Os procedimentos de geração da assinatura são descritos a seguir:

1. Cliente, em um computador remoto, acessa o *Internet Banking* e requisita operação bancária como por exemplo, transferência de fundos entre contas;
2. *Internet Banking* procede com a geração do *hash* sobre a transação T a ser assinada e o apresenta ao Cliente:
   $IB \rightarrow Cliente: h(T), T$
3. Cliente informa seu PIN ao Dispositivo Pessoal e o *hash* a ser assinado:
   $Cliente \rightarrow DP: PIN, h(T)$
4. Dispositivo Pessoal decifra os parâmetros do Cliente utilizando AES e K_{PIN}:
   $D_{K_{PIN}}[E_{K_{PIN}}[p,q,g,x,y,K_{DP}]]$

5. Dispositivo Pessoal gera a assinatura de 32 bits sobre o *hash* da transação *T* utilizando parâmetros de assinatura *p,q,g,x* (KR_C):
   $Sig_{KR_C}[h(T)]$
6. Dispositivo Pessoal cifra a assinatura gerada utilizando o AES e a chave K_{DP}:
   $E_{K_{DP}}[Sig_{KR_C}[h(T)]]$
7. Dispositivo Pessoal apresenta a assinatura cifrada ao Cliente:
   $DP \rightarrow Cliente: E_{K_{DP}}[Sig_{KR_C}[h(T)]]$
8. Cliente informa manualmente a assinatura cifrada ao *Internet Banking*:
   $Cliente \rightarrow IB: E_{K_{DP}}[Sig_{KR_C}[h(T)]]$

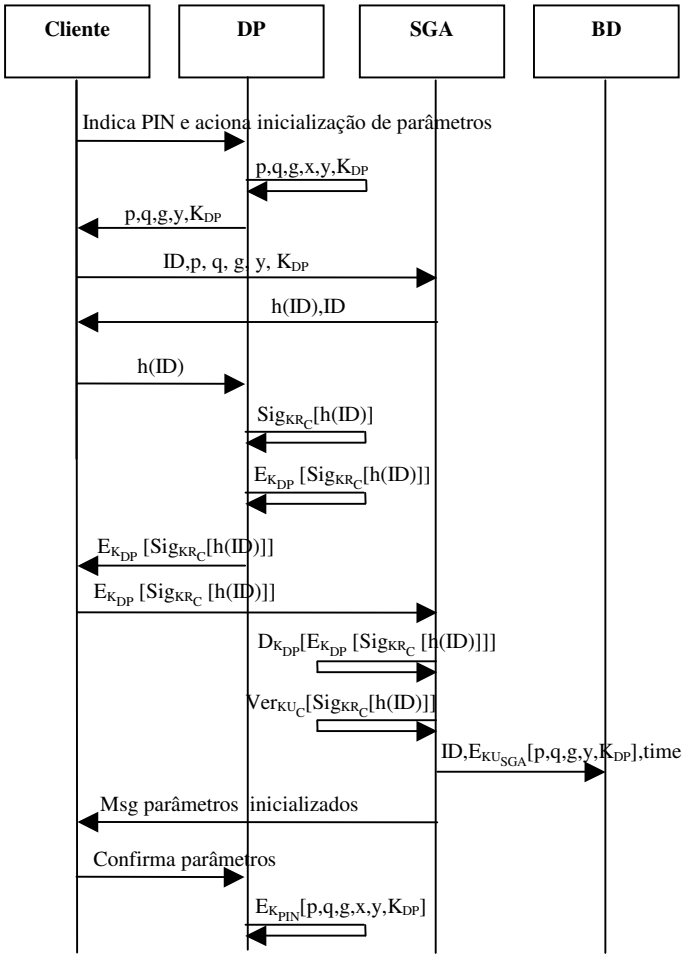


Figura 16 - Protocolo de geração inicial dos parâmetros

Fase 3: Validação da Assinatura

Após recebimento da assinatura, o banco inicia processo de validação da mesma:

1. *Internet Banking* encaminha ao Sistema Gerenciador de Assinaturas a transação e a assinatura para validação:
   $IB \rightarrow SGA: T, E_{K_{DP}}[Sig_{KR_C}[h(T)]]$
2. Sistema Gerenciador de Assinaturas extrai o *ID* do Cliente e a data e hora (*time*) a partir da transação *T*;
3. Sistema Gerenciador de Assinaturas envia *ID* do Cliente juntamente com data e hora da transação para a Base de Dados, para recuperar os parâmetros cifrados do Cliente:
   $SGA \rightarrow BD: ID, time$
4. Sistema Gerenciador de Assinaturas recupera da Base de Dados os parâmetros de validação de assinaturas *p, q, g , y* (KU_C) e chave simétrica K_{DP} do Cliente cifrados com KU_{SGA}:
   $BD \rightarrow SGA: E_{KU_{SGA}}[p,q,g,y,K_{DP}]$
5. Sistema Gerenciador de Assinaturas decifra os parâmetros do Cliente utilizando KR_{SGA}:
   $D_{KR_{SGA}}[E_{KU_{SGA}}[p,q,g,y,K_{DP}]]$
6. Sistema Gerenciador de Assinaturas decifra a assinatura recebida utilizando K_{DP}:
   $D_{K_{DP}}[E_{K_{DP}}[Sig_{KR_C}[h(T)]]]$
7. Sistema Gerenciador de Assinaturas valida a assinatura utilizando parâmetros públicos do cliente:
   $Ver_{KU_C}[Sig_{KR_C}[h(T)]]$
8. Sistema Gerenciador de Assinaturas retorna resultado da validação da assinatura ao *Internet Banking*;
9. *Internet Banking* retorna ao Cliente mensagem com a indicação da efetivação ou não da transação bancária, dependendo do resultado da validação da assinatura digital.

Na Fig. 17 é apresentado o protocolo de geração e validação da assinatura.

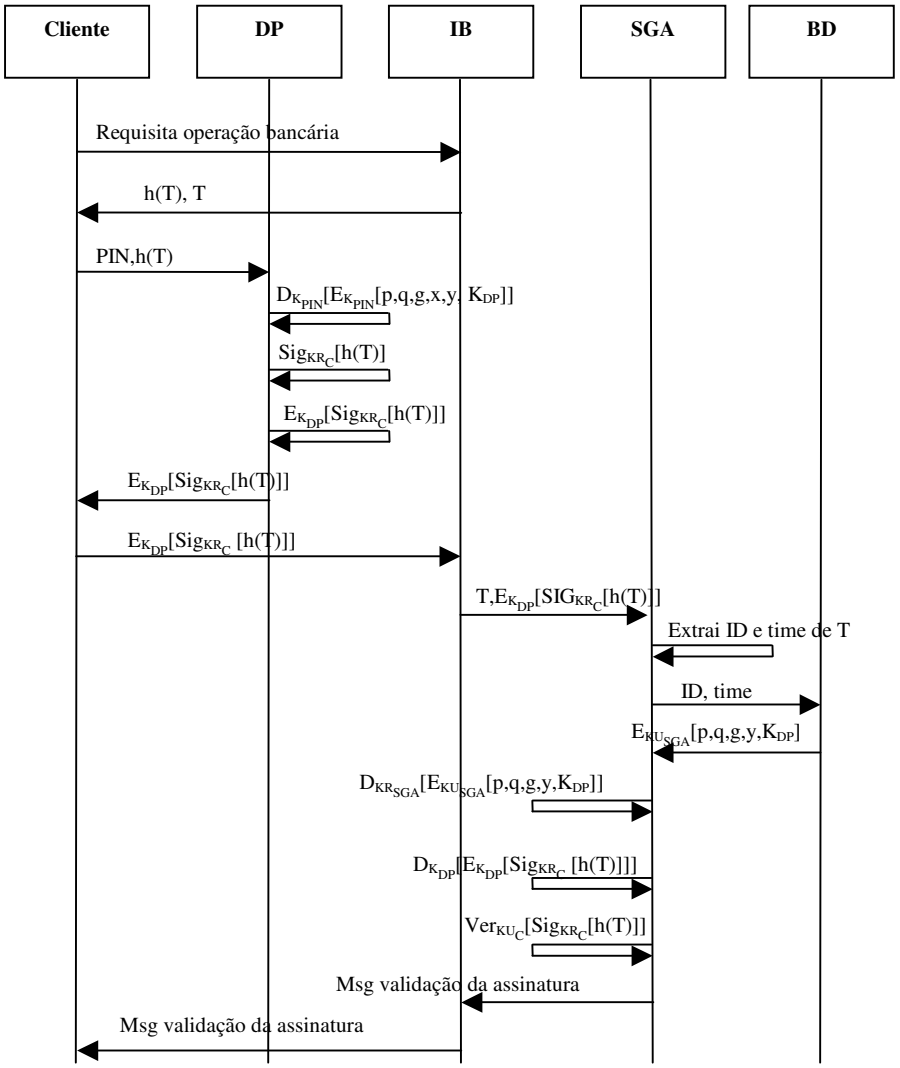


Figura 17 - Protocolo de geração e validação da assinatura

Fase 4: Alteração dos Parâmetros de Assinatura

Para gerar novos grupos de parâmetros, o Cliente remotamente acessa o *Internet Banking* e seleciona a opção para alteração dos parâmetros. Os novos parâmetros gerados serão adicionados à Base de Dados e os anteriores permanecerão, a fim de que transações efetuadas anteriormente por esses parâmetros, possam ser validadas a qualquer momento.

Os procedimentos de geração de novo grupo de parâmetros *p, q, g, x, y* do DSA bem como da chave simétrica K_{DP} são descritos a seguir:

1. Cliente informa seu PIN ao Dispositivo Pessoal e aciona opção de alteração dos parâmetros:

Cliente → *DP: PIN*

2. Dispositivo Pessoal gera os novos parâmetros *p', q', g', x', y'* e a chave simétrica K'_{DP}:
   p',q',g',x',y',K'_{DP}
3. Dispositivo Pessoal decifra os parâmetros de assinaturas atuais utilizando AES e K_{PIN}:
   $D_{K_{PIN}}[E_{K_{PIN}}[p,q,g,x,y,K_{DP}]]$
4. Dispositivo Pessoal apresenta *p', q', g', y'* e K'_{DP} separadamente ao Cliente cifrados com a chave simétrica K_{DP} atual:
   $DP \rightarrow Cliente: E_{K_{DP}}[p'], E_{K_{DP}}[q'], E_{K_{DP}}[g'], E_{K_{DP}}[y'], E_{K_{DP}}[K'_{DP}]$
5. Cliente informa manualmente e separadamente ao *Internet Banking*, os parâmetros gerados pelo Dispositivo Pessoal juntamente com sua identificação:
   $Cliente \rightarrow IB: ID, E_{K_{DP}}[p'], E_{K_{DP}}[q'], E_{K_{DP}}[g'], E_{K_{DP}}[y'], E_{K_{DP}}[K'_{DP}]$
6. *Internet Banking* solicita ao Cliente sua assinatura sobre os novos parâmetros (porém, não apresenta *hash*). Esse procedimento é importante para garantir a integridade dos parâmetros recebidos bem como a autenticidade do Cliente;
7. Cliente requisita ao Dispositivo Pessoal, a assinatura sobre novos parâmetros;
8. Dispositivo Pessoal gera *hash h(T)* sobre novos parâmetros;
9. Dispositivo Pessoal gera a assinatura de 32 bits sobre o *hash* dos novos parâmetros, utilizando parâmetros *p, q, g, x* atuais*:*
   $Sig_{KR_C}[h(T)]$
10. Dispositivo Pessoal cifra a assinatura gerada utilizando o AES e a chave K_{DP} atual:
    $E_{K_{DP}}[Sig_{KR_C}[h(T)]]$
11. Dispositivo Pessoal apresenta ao Cliente a assinatura cifrada, gerada sobre *hash h(T)* dos novos parâmetros:
    $DP \rightarrow Cliente: E_{K_{DP}}[Sig_{KR_C}[h(T)]]$
12. Cliente informa ao *Internet Banking* a assinatura cifrada:
    $Cliente \rightarrow IB: E_{K_{DP}}[Sig_{KR_C}[h(T)]]$
13. *Internet Banking* solicita validação da assinatura ao Sistema Gerenciador de Assinaturas, encaminhando o *ID* do Cliente, os novos parâmetros e a assinatura a ser validada:

$IB \rightarrow SGA: ID, E_{K_{DP}}[p'], [E_{K_{DP}}[q'], E_{K_{DP}}[g'], E_{K_{DP}}[y'], E_{K_{DP}}[K'_{DP}], E_{K_{DP}}[Sig_{KR_C}[h(T)]]$

14. Sistema Gerenciador de Assinaturas envia *ID* do Cliente à Base de Dados:
    $SGA \rightarrow BD: ID$
15. Sistema Gerenciador de Assinatura recupera da Base de Dados os mais recentes parâmetros *p, q, g , y* (KU_C) e chave simétrica K_{DP} do Cliente cifrados com KU_{SGA}:
    $BD \rightarrow SGA: E_{KU_{SGA}}[p,q,g,y,K_{DP}]$
16. Sistema Gerenciador de Assinaturas decifra os parâmetros do Cliente utilizando KR_{SGA}:
    $D_{KR_{SGA}}[E_{KU_{SGA}}[p,q,g,y,K_{DP}]\ ]$
17. Sistema Gerenciador de Assinaturas decifra, utilizando chave simétrica atual K_{DP}, os novos parâmetros recebidos:
    $D_{K_{DP}}[E_{K_{DP}}[p']], D_{K_{DP}}[E_{K_{DP}}[q']], D_{K_{DP}}[E_{K_{DP}}[g']], D_{K_{DP}}[E_{K_{DP}}[y']], D_{K_{DP}}[E_{K_{DP}}[K'_{DP}]]$
18. Sistema Gerenciador de Assinaturas decifra a assinatura recebida utilizando chave simétrica atual K_{DP}:
    $D_{K_{DP}}[E_{K_{DP}}[Sig_{KR_C}[h(T)]]]$
19. Sistema Gerenciador de Assinaturas valida a assinatura utilizando parâmetros públicos atuais do Cliente:
    $Ver_{KU_C}[Sig_{KR_C}[h(T)]]$
20. Sistema Gerenciador de Assinaturas adiciona na Base de Dados o novo grupo de parâmetros e chave simétrica, cifrados com $KU_{SGA,}$ seguidos da data e hora atual (*time*) e *ID* do Cliente:
    $SGA \rightarrow BD: ID, E_{KU_{SGA}}[p',q',g',y',K'_{DP}], time$
21. Sistema Gerenciador de Assinaturas retorna confirmação de alteração de parâmetros para o *Internet Banking*;
22. *Internet Banking* confirma a alteração dos parâmetros para o Cliente;
23. Cliente confirma parâmetros gerados no Dispositivo Pessoal;
24. Dispositivo Pessoal cifra os novos parâmetros gerados utilizando AES e K_{PIN}, substituindo os anteriores:
    $E_{K_{PIN}}[p',q',g',x',y',K'_{DP}]$

Na Fig. 18, é apresentado o processo de alteração dos parâmetros de assinatura:

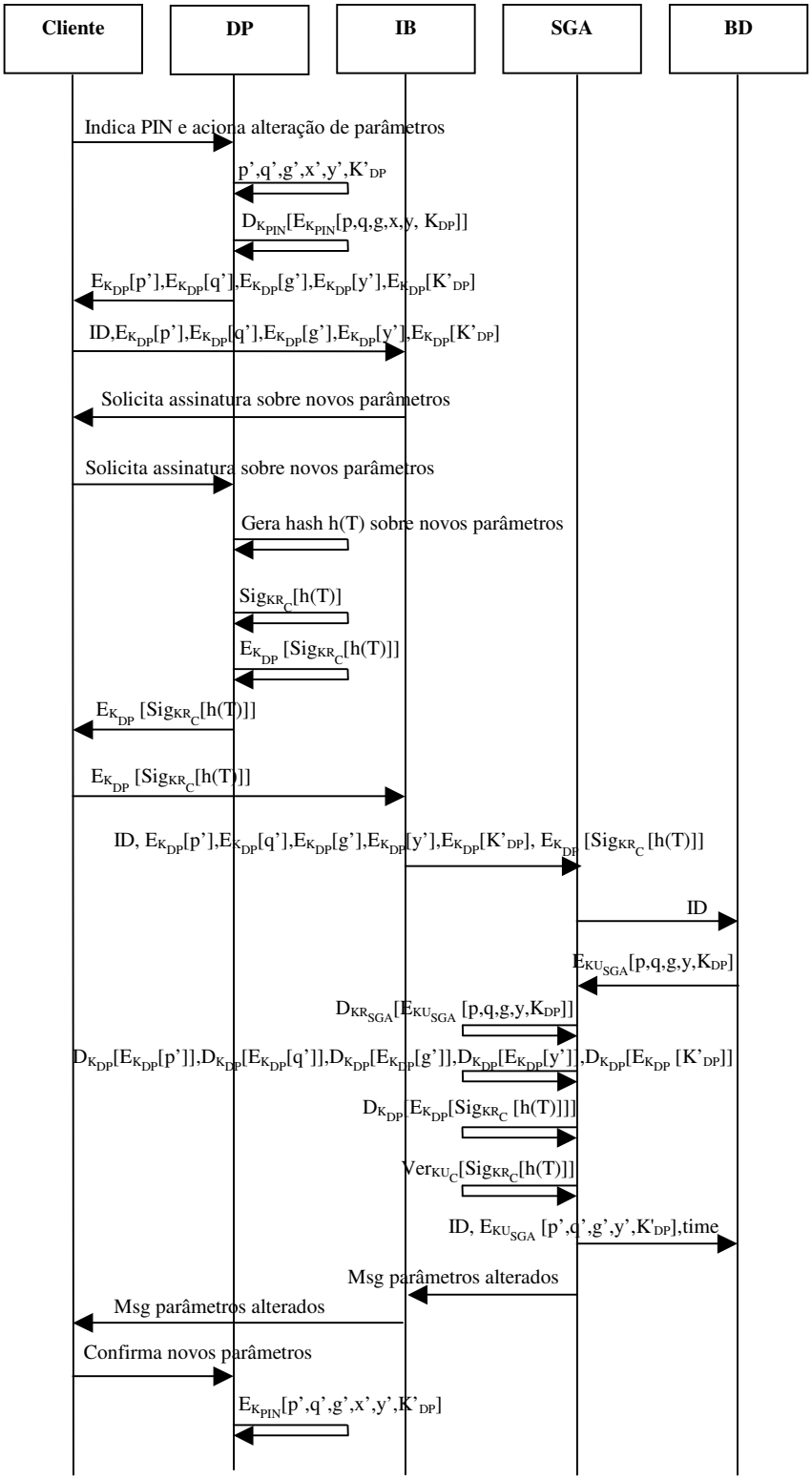


Figura 18 - Protocolo de alteração dos parâmetros de assinatura

5.7 Vantagens da Proposta sobre Esquemas Simétricos

Na proposta apresentada, os parâmetros públicos do DSA devem ser mantidos ocultos e ser de posse apenas do verificador da assinatura, que neste caso é a instituição bancária. Assim sendo, uma das primeiras questões que pode aparecer é por que não utilizar um esquema simétrico ao invés do esquema assimétrico sugerido. Algumas vantagens do esquema proposto com relação ao equivalente simétrico são apresentadas a seguir:

- No esquema simétrico há a necessidade de compartilhamento de segredo, de modo que a simples obtenção da chave simétrica implica na quebra imediata do esquema;
- Na proposta apresentada, ainda que os parâmetros públicos sejam obtidos por um atacante (usuário interno do banco), deverá haver esforço adicional até que a chave privada x seja obtida. A quebra do esquema não acontece automaticamente com a obtenção dos parâmetros ocultos armazenados no banco;
- O uso de chave simétrica não caracteriza assinatura digital. No esquema proposto, a transação é assinada digitalmente, atribuindo a propriedade de não-repúdio.

Como citado na seção 5.3, a solução descrita neste trabalho aplica-se no modelo de *closed* PKI. A proposta de ocultação dos parâmetros públicos aparece também no trabalho de Hoover e Kausik (1999), que tem como objetivo a armanezagem segura da chave privada em software, através de camuflagem criptográfica. Em seu trabalho, sugerem que a chave privada seja cifrada com uma determinada senha e que a decifragem com diversas senhas produza chaves que pareçam ser a verdadeira, confundindo o atacante. Para isso, Hoover e Kausik sugerem retirar das mãos de um atacante qualquer informação que o permita identificar qual chave privada é a legítima, como por exemplo, ocultando a chave pública de validação de assinatura (deixando apenas de conhecimento do verificador). Sem o conhecimento da chave pública, o atacante não é capaz de identificar qual chave privada está relacionada matematicamente àquela chave pública.

Nossa proposição de ocultar os parâmetros públicos é também encontrada no documento FIPS 186-2 que trata do DSA (NIST, 2000), o qual sugere que os

parâmetros *p*, *q* e *g* possam ser públicos e não especifica que obrigatoriamente devam ser públicos.

5.8 Especificação Técnica

Nesta seção, serão apresentados alguns detalhes técnicos relacionados à solução de assinatura digital curta especial proposta.

5.8.1 Truncamento da Função SHA-256

No DSA, o *hash* da mensagem a ser assinada é gerado com 160 bits através da aplicação da função SHA-1. Na proposta que está sendo apresentada, o *hash* é gerado com 16 bits, viabilizando a digitação manual por parte do Cliente.

Para obter a função *hash* de 16 bits, chamada aqui de SHA_256_to_16, foi seguida a instrução indicada no documento FIPS 180-2 do NIST descrita no item 4.4.2 deste trabalho. A instrução diz que se for necessária uma saída de função *hash* inferior à saída fornecida pela função aplicada, o *hash* pode ser truncado selecionando-se os bits mais à esquerda desejados. Na presente proposta, para obtenção do *hash* de 16 bits é aplicada a função SHA-256 sobre a transação a ser assinada pelo Cliente, e são selecionados os 16 bits mais à esquerda da saída da função. No *Draft* FIPS 186-3 que trata de algumas mudanças sobre o DSA, o NIST também aplica a funcionalidade de truncamento de *hash* na geração e validação da assinatura, não sendo mais citado neste documento o uso do SHA-1.

A função SHA_256_to_16 é utilizada no *Internet Banking* para gerar o *hash* da transação a ser assinada, no Sistema Gerenciador de Assinaturas para validar a assinatura da transação e ainda, no Dispositivo Pessoal para gerar o *hash* sobre os novos parâmetros gerados.

Foram realizadas várias análises sobre os resultados de *hash* de 16 bits de mensagens diferenciadas em alguns bits, no intuito de observar semelhanças de resultados nos *hash's* gerados. Na Tabela 5 são apresentadas algumas amostras.

Tabela 5 - Amostras de hash de 16 bits (informações em base 16)

Seqüência	Mensagem	*Hash* gerado
1	abcdef	bef5
2	abcdee	96dd
3	abddef	78f2
4	dabcdef0123456789aaaaabbbbb	f240
5	5abcdef0123456789aaaaabbbbb	9978

A seqüência 2 da amostra foi escolhida de tal forma que possuísse apenas o bit menos significativo diferente da seqüência 1. A seqüência 3 possui apenas o bit da 12ª posição diferente da seqüência 1. É possível perceber que independente da posição de mudança de bit, o *hash* gerado sobre essas mensagens foi diferente.

A seqüência 4 exemplifica uma mensagem mais longa. A seqüência 5 é uma cópia da seqüência 4 com o bit mais significativo passado para 0. O *hash* para a seqüência 5 não teve qualquer semelhança com o da seqüência 4.

A análise efetuada veio confirmar o que se esperava: uma vez que o *hash* de 16 bits é oriundo do resultado do SHA-256, qualquer mudança ocorrida na mensagem gera um *hash* diferente, ainda que apenas 16 bits sejam selecionados.

5.8.2 Geração dos Parâmetros do DSA

No esquema proposto, os parâmetros do DSA de cada Cliente são gerados pelo Dispositivo Pessoal, baseado no documento FIPS 186-2 (NIST, 2000) que especifica técnicas para a geração desses parâmetros. Com a redução do parâmetro q para 16 bits, as rotinas de geração do primo q, chave privada x e parâmetro por assinatura k, devem sofrer algumas alterações.

Na rotina de geração do primo q, o documento FIPS 186-2 menciona o uso do algoritmo *hash* SHA-1 em determinadas etapas. Com a proposta de reduzir o parâmetro q para 16 bits, a função SHA-1 utilizada na rotina de geração do primo q deve ser substituída pela função SHA_256_to_16.

Nas rotinas de geração de números randômicos que são os parâmetros x e k, é mencionado no documento FIPS 186-2 o uso de uma função G que pode ser construída

com o uso de SHA-1 ou com o uso do algoritmo simétrico DES. A rotina *G* gera como resultado um número de 160 bits. Também devido à redução do parâmetro *q* para 16 bits, a função *G* utilizada nas rotinas de geração de números randômicos deve ser construída a partir da função *hash* SHA_256_to_16 .

5.8.3 Cifragem da Assinatura Gerada

A assinatura gerada pelo Dispositivo Pessoal é cifrada com o algoritmo AES e chave K_{DP} antes de ser apresentada ao Cliente (*r* e *s* cifrados concatenados). O AES é um cifrador de bloco que opera em blocos de dados de 128 bits (NIST, 2001a). Caso o tamanho do texto a ser cifrado não seja múltiplo do tamanho do bloco (no caso do AES, 128 bits), os cifradores de blocos acrescentam bits extras, chamados *padding*, ao final do texto claro antes de iniciar o processo de cifragem. Para as mensagens com tamanho superior ao tamanho do bloco, os cifradores utilizam técnicas, chamadas modos de operação, que agem sobre mensagens maiores (NIST, 2001b). No caso do AES, o texto cifrado é gerado também em blocos de 128 bits.

Na presente solução, a assinatura a ser cifrada é de 32 bits (16 bits para *r* e 16 bits para *s)*. O texto cifrado resultante da cifragem da assinatura deve continuar sendo de 32 bits (e não 128 bits), a fim de viabilizar a digitação manual por parte do Cliente ao *Internet Banking*. Para suprir essa necessidade, foi utilizado o AES no modo de operação CFB, uma vez que neste modo de operação os dados podem ser cifrados em unidades menores que o tamanho do bloco, resultando em um texto cifrado com o mesmo número de bits do texto claro (NIST, 2001b).

O modo CFB faz uso de um bloco de dados de 128 bits chamado vetor de inicialização (IV) que participa do processo de cifragem do texto claro. O IV não precisa ser secreto, porém é importante que não seja uma informação previsível (NIST, 2001b).

Para gerar o IV a participar da cifragem da assinatura, o Dispositivo Pessoal replica os 16 bits do *hash* a ser assinado até completar 128 bits. Com esse valor de 128 bits é realizada uma operação XOR com os 128 bits mais à esquerda do parâmetro *y*. Dessa forma, a cada novo *hash* a ser assinado, um novo IV será gerado.

5.8.4 Derivação da Chave Secreta a partir do PIN

No Dispositivo Pessoal, os parâmetros do DSA (*p, q, g, x, y*) e chave simétrica K_{DP} são armazenados cifrados com o AES. Para obter a chave simétrica AES a participar da cifragem dos parâmetros no Dispositivo Pessoal, foi utilizada a técnica *Password Based Encryption* (PBE), que gera chave de cifragem a partir de uma senha. A técnica é especificada no documento PKCS#5 (PUBLIC KEY CRYPTOGRAPHY STANDARDS – PKCS, 1999). PKCS são os Padrões de Criptografia de Chave Pública produzidos pela *RSA Laboratories* (http://www.rsasecurity.com/rsalabs) em conjunto com desenvolvedores de sistemas seguros.

A função de derivação de chaves especificada no PKCS#5 produz chave simétrica baseada nos seguintes parâmetros:

- **Senha Base:** senha a partir da qual será derivada a chave simétrica. Para esse parâmetro foi utilizado o PIN do usuário que é informado para ativação do Dispositivo Pessoal;
- **Salt**: tem como objetivo produzir um grande conjunto de chaves a partir de uma senha, sendo que uma delas será selecionada randomicamente de acordo com o salt. O uso do salt aumenta o custo da busca exaustiva e ataque de dicionário para um grande conjunto de senhas, entretanto, não aumenta a complexidade do ataque para uma senha específica;
- **Contador de Repetição**: indica quantas vezes a função de derivação de chaves deve repetir. O objetivo do contador de repetição é aumentar o custo de produção de chaves a partir de senhas, aumentando assim o custo de busca exaustiva. O PKCS#5 recomenda no mínimo 1000 repetições, o que aumenta o custo do ataque sem ter impacto aparente no custo do processo de derivação de chaves.

5.9 Análise de Segurança

O uso do Dispositivo Pessoal de assinaturas sem conectividade com o computador oferece segurança contra roubo de chaves secretas através de sistemas maliciosos. Como não há comunicação entre o dispositivo e o computador, não há meios de um programa malicioso capturar qualquer informação do Dispositivo Pessoal.

Caso o dispositivo seja perdido ou roubado, a segurança também não estará ameaçada. Após *n* tentativas incorretas de PIN ele é bloqueado. Além disso, as informações de parâmetros do DSA bem como a chave simétrica utilizada para cifragem das assinaturas são armazenadas no Dispositivo Pessoal cifradas com o AES. Para aumentar a segurança, o *Internet Banking* pode solicitar que periodicamente o Cliente gere novo grupo de parâmetros. O atacante não consegue forjar a geração de novos parâmetros do Cliente, uma vez que o *Internet Banking* solicita a assinatura do Cliente também sobre os novos parâmetros gerados. Sem o Dispositivo Pessoal, o atacante não conseguirá informar a assinatura digital correta.

Ao solicitar assinaturas digitais do Cliente, o banco apresenta a ele os dados da transação a ser assinada e o respectivo *hash*. Dessa forma, se desejar, o Cliente tem condições de verificar o que está assinando, isto é, se o *hash* apresentado pelo banco efetivamente diz respeito à transação que está sendo assinada.

Se um atacante conseguir capturar a assinatura que está sendo informada pelo Cliente ao computador através da recuperação de eventos de seu teclado, não conseguirá avançar em seu ataque, uma vez que a assinatura é informada ao computador já cifrada pelo Dispositivo Pessoal com criptografia simétrica. O atacante também não poderá reutilizar a assinatura para forjar a realização de uma nova transação uma vez que a assinatura é única por transação.

Se o par de assinaturas (*r* e *s*) fossem informados pelo Cliente ao *Internet Banking* em texto claro, o atacante poderia recuperar tais assinaturas e efetuar algumas tentativas de ataques. Porém, descobrir a chave privada *x* sem ter conhecimento dos parâmetros *p, q, g* e *y* não é uma tarefa trivial. As informações que ele poderia obter para tentar alguma forma de ataque seriam amostras de transações e respectivas assinaturas (*r* e *s*). Tentar combinações de *p*, *q*, *g* e *k* para obter o respectivo *r* é uma tarefa inviável, uma vez que os parâmetros *p* e *g* possuem tamanho de 512 bits. Uma opção mais viável para o atacante seria gerar uma lista com combinações de parâmetros *x, k* e *q* que levam ao respectivo *s,* uma vez que tais parâmetros são de 16 bits. Com 16 bits, são aproximadamente apenas 3000 possibilidades de números primos para *q*. Entretanto, essa lista conteria diversas combinações de *x, k* e *q* gerando valores idênticos de *s*. Ainda assim, para descobrir qual combinação de parâmetros é a correta, bastaria ao atacante obter outras amostras de pares de assinaturas (*r*, *s*) com as respecti-

vas transações. O atacante aplica então o mesmo método de geração de lista com combinações de parâmetros para essas assinaturas, até que um único par *x* e *q* apareça em todas as listas. Conhecendo *x, q* e *k,* o atacante poderia utilizar o mesmo *r* para novas assinaturas de mensagens distintas (gerando o *s* correspondente).

Para *s* repetir em assinaturas de transações distintas, o *hash* das transações deve colidir, e além disso, o mesmo *k* deve ser utilizado em ambas as assinaturas. Colisão de *hash* é um ataque que não se aplica ao esquema apresentado. O *hash* é gerado com 16 bits, permitindo 65536 *hash* distintos para mensagens relacionadas ao universo restrito de transações bancárias, e, portanto, o universo de mensagens que façam sentido neste cenário é reduzido. Além disso, para compor a transação a ser assinada, o banco adiciona a data e hora atual e um número randômico, individualizando a transação. A inclusão do número randômico evita ainda que um atacante tenha conhecimento antecipado sobre o valor do *hash* que será gerado sobre tal transação.

Com a assinatura sendo informada de maneira cifrada ao *Internet Banking,* antes de tentar tais ataques é necessário a quebra da cifragem com o AES, portanto computacionalmente impraticável. Mesmo efetuando tentativas de ataque de decifragem da assinatura, o atacante não saberia se o texto decifrado é o correto, uma vez que a assinatura gerada não é uma informação que faça sentido.

O parâmetro *k* é escolhido randomicamente e, portanto, a escolha não é determinística. Neste caso, pode ocorrer pares de assinaturas do tipo (*r, s1*) e (*r, s2*), ou seja mesmo *r* em ambas as assinaturas. Lembrando que *r* é obtido através de ($g \wedge k \bmod p$) $\bmod q$, ou seja, *r* não sofre influência da transação a ser assinada. Entretanto, *r* é cifrado juntamente com *s*. Como *s1* e *s2* possuem valores distintos, o texto cifrado gerado da concatenação de *r* e *s* será diferente em ambas as assinaturas.

O ataque de Vaudenay descrito na seção 4.6.4.3 também não se aplica, já que os parâmetros são gerados pelo próprio Dispositivo Pessoal de assinaturas, e, portanto, possui um algoritmo confiável de geração de parâmetros.

5.10 Formalização

No intuito de efetuar a validação dos protocolos apresentados, foi realizada a modelagem da proposta utilizando Redes de Petri. Redes de Petri são métodos formais

do tipo diagrama de transição de estados que se adaptam bem à aplicações que requeiram a noção de eventos e evoluções simultâneas (CARDOSO; VALLETE, 1997).

A formalização gráfica através das Redes de Petri faz uso de lugares (círculos) que representam os possíveis estados do sistema, transições (retângulos) que são os eventos que permitem a mudança de estado, fichas (pequeno círculo negro) que correspondem a uma condição atendida em determinado lugar e arcos (setas) que indicam a direção dos movimentos das fichas de um lugar para outro.

A mudança de estado no sistema ocorre quando uma transição é disparada, e nesse caso as fichas passam do estado de entrada para o estado de saída referente àquela transição. Uma transição somente está habilitada a disparar se nos seus lugares de entrada existirem fichas suficientes para o seu disparo.

Na Fig. 19 é apresentada a formalização do protocolo de inicialização dos parâmetros de assinatura.

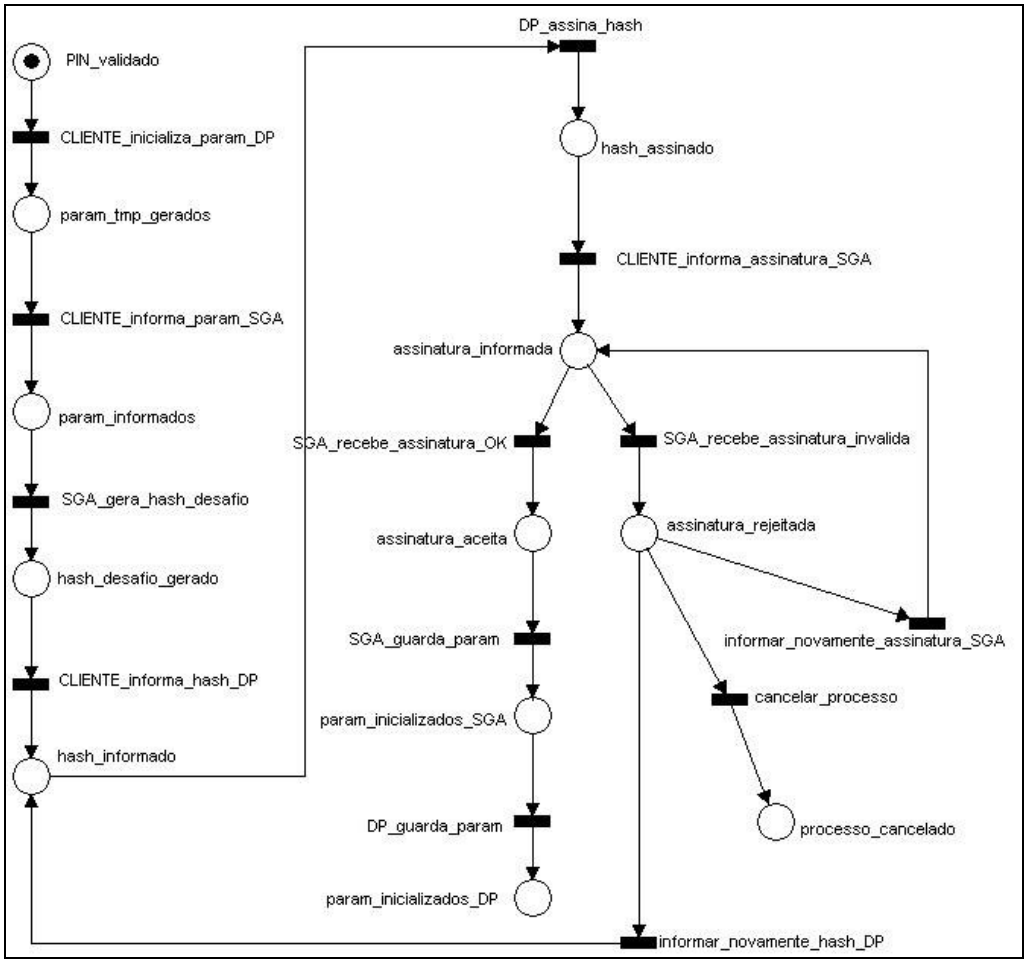


Figura 19 - Modelagem do protocolo de inicialização dos parâmetros

Na Fig. 20 é apresentada a formalização do protocolo de geração de assinatura digital sobre uma transação bancária. Essa rede possui um arco inibidor que liga o lugar tentativas_restantes à transição cancelar_processo. O arco inibidor indica que a transição somente poderá disparar se não houver fichas no lugar de origem desse arco. Nesse caso, a transição cancelar_processo somente será disparada se não houver fichas no lugar tentativas_restantes e se houver ficha no lugar assinatura_rejeitada. Na rede modelada, após o cliente indicar três assinaturas digitais inválidas, o processo é cancelado.

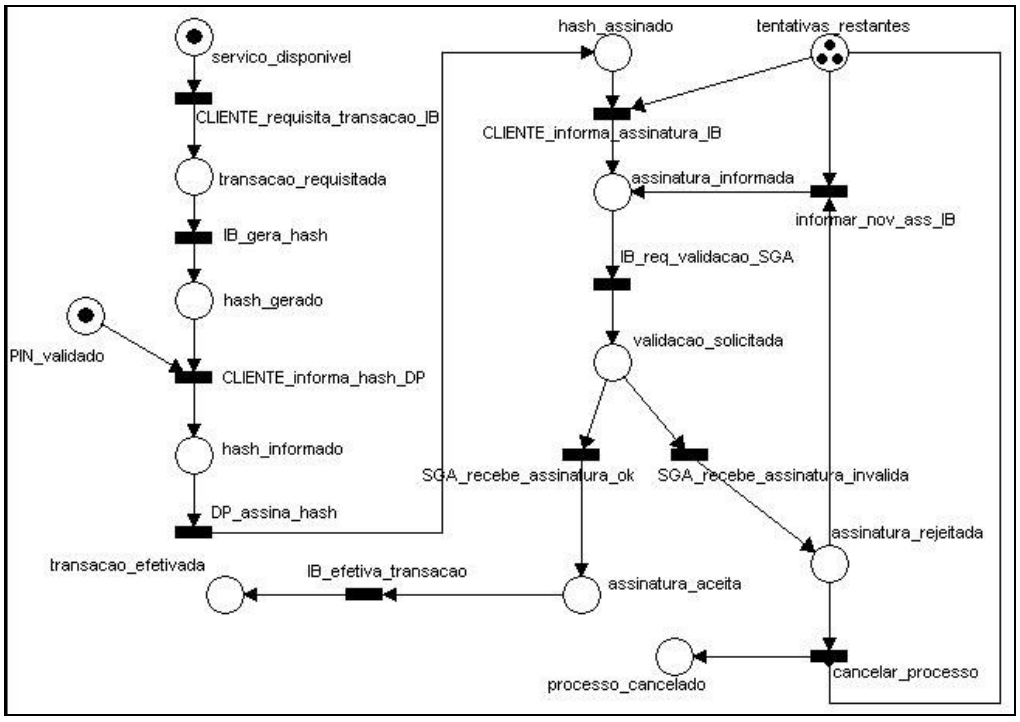


Figura 20 - Modelagem do protocolo de assinatura sobre transação

A análise das propriedades da rede de inicialização de parâmetros foi efetuada através do aplicativo ARP versão 2.4, obtido no endereço http://www.ppgia.pucpr.br/~maziero/diversos/petri/. Essa ferramenta foi criada pelo Laboratório de Controle e Microinformática (LCMI) da Universidade Federal de Santa Catarina (UFSC). As propriedades da rede analisada pela ferramenta ARP são apresentadas na Tabela 6. A rede que modela a assinatura sobre transação não foi analisada pela ferramenta ARP por possuir uma característica adicional não contemplada nessa ferramenta, que é o arco inibidor.

Tabela 6 - Propriedades da rede de inicialização dos parâmetros

Propriedade	Comentário
Vivacidade	Todas as transações são quase vivas, pois podem ser disparadas no mínimo uma vez
Reiniciável	A rede modelada não é reiniciável, pois descreve a geração inicial dos parâmetros de assinatura que ocorre apenas uma vez
Deadlocks	Foram identificados *deadlocks* que dizem respeito aos estados finais da rede que são os estados processo_cancelado e param_inicializados_DP, os quais não possuem nenhum estado sucessor
Conservação	A rede é conservativa visto que a soma total de fichas nas redes é constante (sempre 1)

5.11 Implementação do Protótipo

Por não haver um hardware disponível, o Dispositivo Pessoal foi implementado em simulador utilizando-se a linguagem Java, permitindo a portabilidade do mesmo código em implementações futuras para dispositivos móveis diversos, como celulares. O Dispositivo Pessoal representa a principal entidade desta proposta, visto que nele está implementado o esquema de geração dos parâmetros e assinatura curta. O *Internet Banking* e o Sistema Gerenciador de Assinaturas foram implementados como uma página WEB também utilizando-se Java.

Devido à redução do parâmetro q, não foi possível utilizar as rotinas do DSA disponíveis no Java, necessitando codificação à parte. Na implementação do protótipo, o enfoque foi direcionado às funções que deveriam representar as características especiais da solução apresentada, estando relacionadas à:

- Geração dos parâmetros de assinatura do DSA: as rotinas foram codificadas para contemplar a redução do parâmetro q e foram baseadas nos códigos fontes do projeto GNU Crypto disponíveis em http://www.gnu.org/software/gnu-crypto/;
- Geração e validação de assinaturas: com a redução do parâmetro q, foi necessário codificar as rotinas do DSA para gerar e validar assinaturas;
- Geração do truncamento da saída da função *hash* SHA-256 para 16 bits;

- Geração de cifragem sobre a assinatura gerada pelo Dispositivo Pessoal utilizando AES e K_{DP};
- Armazenagem dos parâmetros (*p, q, g, x, y*) e chave simétrica K_{DP} no Dispositivo Pessoal cifrados com o AES e K_{PIN} (que é uma chave simétrica derivada do PIN do Cliente);
- Armazenagem dos parâmetros de validação de assinatura (*p, q, g, y*) e chave simétrica $K_{DP,}$ cifrados com o RSA e KU_{SGA}. Esses parâmetros são manipulados pelo Sistema Gerenciador de Assinaturas.

Assim sendo, as funções implementadas no protótipo para demonstração da proposta são descritas a seguir:

- **Dispositivo Pessoal**:
  - Efetuar a inicialização dos parâmetros do DSA e K_{DP};
  - Gerar assinatura digital sobre as transações bancárias.
- ***Internet Banking***:
  - Gerar o *hash* de 16 bits sobre a transação a ser assinada pelo Cliente;
  - Encaminhar a assinatura informada pelo Cliente a ser validada ao Sistema Gerenciador de Assinaturas.
- **Sistema Gerenciador de Assinaturas**:
  - Receber os parâmetros gerados pelo Dispositivo Pessoal e informados pelo Cliente;
  - Efetuar a validação de assinaturas encaminhadas pelo *Internet Banking*, retornando o resultado da validação.

Para fins de demonstração, no protótipo implementado os parâmetros de geração e validação de assinatura do DSA bem como chave simétrica K_{DP} do Cliente são armazenados em arquivo XML descritos a seguir:

- **parametrosAssinaturaDSA.xml**: contém os parâmetros *p, q, g, x, y,* K_{DP} cifrados com o AES e uma chave derivada do PIN do Cliente. Esse arquivo é manipulado pelo Dispositivo Pessoal;
- **parametrosValidacaoDSA.xml**: contém o ID do Cliente (agência e conta), os parâmetros *p, q, g, y,* K_{DP} cifrados com RSA e KU_{SGA} e data/hora de geração dos parâmetros. Esse arquivo é manipulado pelo Sistema Gerenciador de Assinaturas.

A proposta apresentada neste trabalho especifica que para efetivar as transações bancárias, o Cliente deve informar a sua assinatura digital sobre elas. Para demonstração do fluxo de geração e validação de assinatura curta, no protótipo foi implementada a geração de assinatura digital sobre a transação bancária de transferência de fundos entre contas. A seguir, serão apresentadas as funcionalidades de inicialização dos parâmetros e geração de assinatura digital curta sobre transação bancária.

5.11.1 Efetuando Inicialização dos Parâmetros do DSA e K_{DP}

A Fig. 21 apresenta a interface do simulador do Dispositivo Pessoal. Possui um teclado e um visor hexadecimal. No visor do Dispositivo Pessoal são apresentados no máximo 16 símbolos hexadecimais simultaneamente.

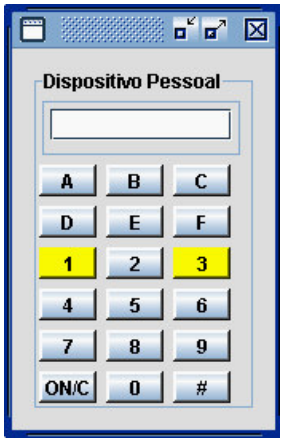


Figura 21 - Dispositivo Pessoal

Ao receber o Dispositivo Pessoal na instituição bancária, o Cliente deve inicializar os parâmetros de geração e validação de assinatura, bem como chave simétrica K_{DP}. Para isso, após indicação do seu PIN, o Cliente aciona as teclas ##A#. O Dispositivo Pessoal procede então com a geração de todos os parâmetros.

Os parâmetros devem ser informados manualmente pelo usuário (neste caso o Cliente ou funcionário do banco) ao Sistema Gerenciador de Assinatura (Fig. 22). Para visualizar os parâmetros (*p, q, g, y,* K_{DP}) que são enumerados de 1 a 5 respectivamente, o Cliente deve informar ao Dispositivo Pessoal o número do parâmetro desejado através da indicação ##<nr. parâmetro>#. Ex.: ##1# faz com que o Dispositivo Pessoal

apresente o parâmetro de número 1 ao usuário, que é o parâmetro *p*. Alguns parâmetros possuem tamanho que ultrapassa os 16 caracteres suportados pelo visor do Dispositivo Pessoal. Neste caso, o Cliente deve utilizar as teclas <1> e <3> para se movimentar entre os caracteres dos parâmetros.

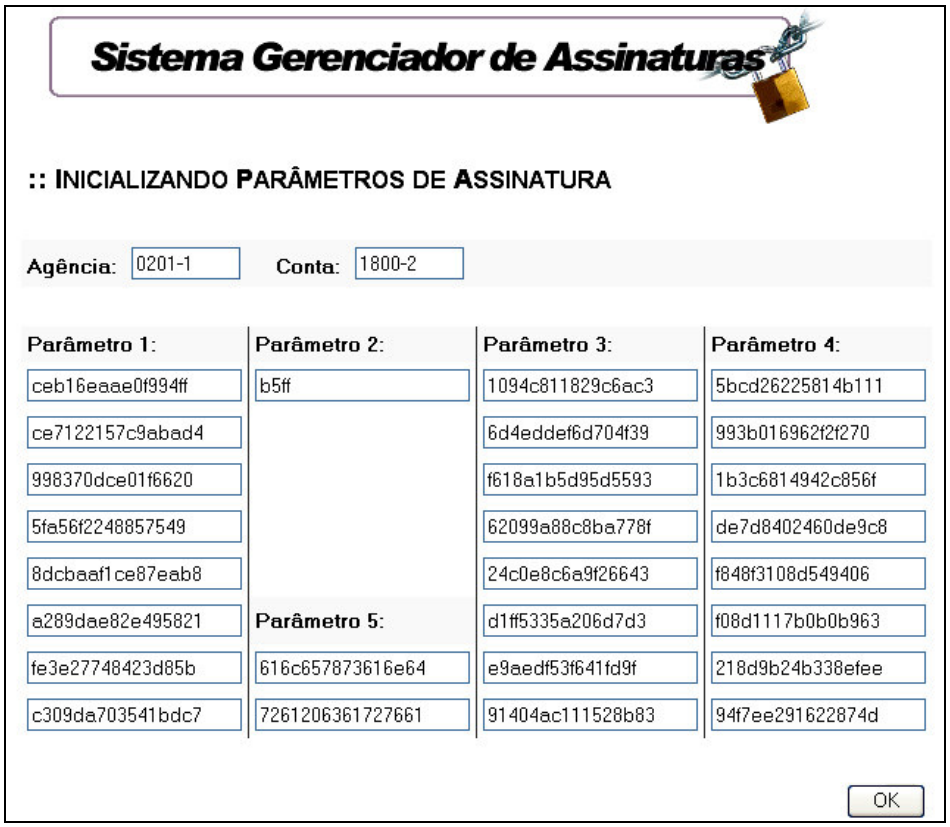


Figura 22 - Recebimento dos parâmetros no SGA

No Sistema Gerenciador de Assinaturas, após clicar no botão *OK*, o Cliente será solicitado a efetuar assinatura sobre o *hash* de um desafio (seu próprio identificador), a fim de validar se os parâmetros recebidos foram digitados corretamente (Fig. 23). O Cliente digita então o *hash* do desafio em seu Dispositivo Pessoal para geração da assinatura utilizando os parâmetros inicializados. Posteriormente, o Cliente informa manualmente a assinatura gerada ao Sistema Gerenciador de Assinatura para que seja efetuada a validação. Com o sucesso na validação da assinatura, o Sistema Gerenciador de Assinaturas armazena os parâmetros de validação de assinaturas do Cliente no arquivo parametrosValidacaoDSA.xml, cifrados com RSA e a chave pública KU_{SGA}. O Cliente confirma no seu Dispositivo Pessoal os parâmetros gerados, os quais são

armazenados no arquivo parametrosAssinaturaDSA.xml, cifrados com AES e com a chave simétrica K_{PIN} derivada do PIN do Cliente.

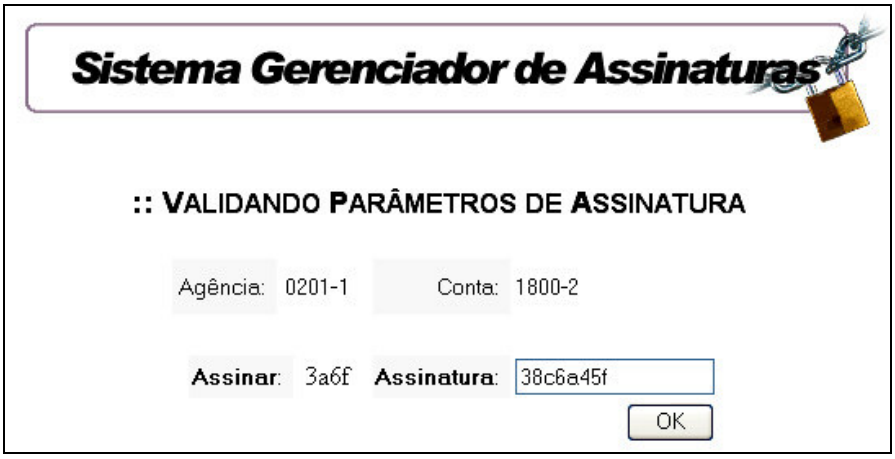


Figura 23 - Validando parâmetros recebidos no SGA

Detalhes sobre procedimentos no processo de geração e validação de assinaturas serão apresentados na seção a seguir.

5.11.2 Efetuando Transação Assinada

A partir de um computador remoto, o Cliente solicita acesso ao *Internet Banking* indicando o número de sua agência e conta e sua assinatura digital sobre um desafio (data, hora e número randômico) gerado pelo *Internet Banking*. Após validação com sucesso da assinatura do Cliente na tela de acesso, o *Internet Banking* apresenta a tela da Fig. 24, para que o Cliente possa informar os dados da transação bancária a ser realizada.

Após o Cliente indicar as informações de transferência e clicar no botão *OK*, o *Internet Banking* calcula o *hash* sobre a transação e o apresenta ao Cliente, solicitando sua assinatura para efetivação da transação (Fig. 25). Em seu Dispositivo Pessoal, o Cliente informa o *hash* da transação a assinar. O Dispositivo Pessoal efetua a assinatura com o DSA utilizando os parâmetros de assinatura *(p, q, g, x)* do Cliente recuperados do arquivo parametrosAssinaturaDSA.xml. O código fonte de geração da assinatura utilizado no Dispositivo Pessoal é apresentado na Fig. 26, que recebe o *hash* a assinar e retorna o par de assinatura *r,s* concatenados.

Figura 24 - Transferência entre contas com assinatura digital no IB – passo 1

Figura 25 - Transferência entre contas com assinatura digital no IB – passo 2

```
Public byte[] rs geraAssinatura (byte[] hash16){
  <…lista variaveis declaradas…>
  BigInteger hashBig = new BigInteger (1,hash16);
  while (!rsOK) {
    //Gera um k randômico baseado nas instruções do FIPS 186-2
    k =  randomK();
    r =  g.modPow(k, p).mod(q);
    if (!r.equals(BigInteger.ZERO)) {
      s = hashBig.add(x.multiply(r)).multiply(k.modInverse(q)).mod(q);
      if (!s.equals(BigInteger.ZERO))
         rsOK = true;
      }
    }
    rByte = (r.toByteArray());
    sByte = (s.toByteArray());
    byte [] assinatura = new byte[] {rByte[0],rByte[1],sByte[0],sByte[1]};
    return assinatura;
}
```

Figura 26 - Código fonte de geração da assinatura no DP

Posteriormente, o Dispositivo Pessoal cifra a assinatura com o AES e K_{DP} no modo CFB e apresenta a assinatura cifrada para o Cliente (8 símbolos hexadecimais). No processo de cifragem, o Dispositivo Pessoal gera o IV a partir do *hash* da transação e do parâmetro *y*, conforme código fonte apresentado na Fig. 27. Esse mesmo código de geração do IV é utilizado no Sistema Gerenciador de Assinaturas para decifrar a assinatura recebida para validação.

O Cliente informa manualmente ao *Internet Banking* a assinatura cifrada gerada pelo Dispositivo Pessoal. *Internet Banking* envia a assinatura para validação ao Sistema Gerenciador de Assinaturas juntamente com a transação solicitada. O Sistema Gerenciador de Assinaturas recupera agência, conta, data e hora do cabeçalho da transação enviada pelo *Internet Banking*. Com base nessas informações, recupera os parâmetros do Cliente do arquivo parametrosValidacaoDSA.xml, que são então decifrados com RSA e KR_{SGA}. Em seguida, decifra a assinatura recebida e realiza a validação da assinatura, conforme o código apresentado na Fig. 28.

```
public byte[] geraIV (byte[] y, byte [] hash16) {
    <…lista variaveis declaradas…>

  //Copia os 128 bits mais a esquerda do parametro y para a variavel y128
  System.arraycopy(y, 0, y128, 0, y128.length);
  //Copia os 16 bits do hash16 para a variavel hash128, alternando entre os
  //2 bytes para completar os 16 bytes da variável hash128
  byte[] hash128 = new byte[] {hash16[0],hash16[1],hash16[0],
                       hash16[1],hash16[0],hash16[1],hash16[0],hash16[1],
                       hash16[0],hash16[1],hash16[0],hash16[1],hash16[0],
                       hash16[1],hash16[0],hash16[1]};
  y128Big = new BigInteger (1,y128);
  hash128Big = new BigInteger (1,hash128);
  //Realiza o xor entre y128 e hash128 gerando o IV
  ivBig = y128Big.xor(hash128Big);
  iv = ivBig.toByteArray();
  //Retorna o IV gerado
  return iv;
}
```

Figura 27 - Código fonte de geração do IV no DP e SGA

```
public boolean validaAssinatura(byte[] transacao, byte[] r, byte[] s){
 <…lista variaveis declaradas…>

  //Gera o hash da transação
  SHA256_TO_16 sha256_to_16 = new SHA256_TO_16();
  hash16 = sha256_to_16.getHash(transacao);
  BigInteger hash16Big = new BigInteger (1,hash16);
  BigInteger rBig = new BigInteger (1,r);
  BigInteger sBig = new BigInteger (1,s);
  if (rBig.equals(BigInteger.ZERO) || sBig.equals(BigInteger.ZERO))
     return false;
  w = sBig.modInverse(q);
  u1 = (hashBig.multiply(w)).mod(q);
  u2 = (rBig.multiply(w)).mod(q);
  v = g.modPow(u1, p).multiply(y.modPow(u2, p)).mod(p).mod(q);
  if (rBig.equals(v))
     return true;
  else
    return false;
}
```

Figura 28 - Código fonte de validação da assinatura no SGA

O Sistema Gerenciador de Assinaturas retorna o resultado da validação ao *Internet Banking* que somente procede com a efetivação da transação se a assinatura for validada com sucesso.

Os processos de geração e validação de assinaturas citadas nas etapas de inicialização dos parâmetros bem como na tela de acesso ao *Internet Banking*, procedem como descrito nesta seção.

5.12 Conclusão

Neste capítulo, apresentamos a proposta de geração de assinatura digital curta em dispositivo sem conexão com o computador, dispensando a instalação de software ou hardware específicos no computador do usuário. A assinatura é gerada com 32 bits (8 símbolos hexadecimais). Para geração da assinatura digital curta, utilizamos uma versão modificada do algoritmo DSA, buscando manter um nível de segurança adequado.

A fim de demonstração prática da solução, foi apresentado um protótipo para o uso de assinatura digital curta em sistemas de *Internet Banking*.

6 CONSIDERAÇÕES FINAIS

Segurança na autenticação dos clientes de *Internet Banking* é uma preocupação constante e muitos esforços vêm sendo direcionados à pesquisa de novos métodos em busca de soluções seguras, convenientes e de baixo custo. A maioria dos métodos de autenticação existente valida o usuário apenas no processo de identificação, sendo esta uma autenticação de curto prazo. A autenticação de longo prazo é obtida com a adoção de assinaturas digitais, utilizando certificados digitais que podem ser instalados no computador do cliente, o que não possui níveis de segurança adequados, ou em dispositivos de hardware com conectividade ao computador, que agregam custo e complexidade à solução.

A proposta aqui apresentada abordou uma solução para geração de assinatura digital curta de 32 bits baseada no algoritmo DSA e gerada em dispositivo pessoal (hardware *token*) sem conectividade com o computador. Como visto na Seção 4.7, quando o modelo convencional de assinatura digital não satisfaz determinadas necessidades, variações desse modelo com protocolos específicos são combinados, gerando assinaturas com características especiais. É o caso da assinatura inegável, que permite que a validação da assinatura ocorra somente mediante o consentimento do signatário (diferentemente do modelo convencional que possui a propriedade de ser universalmente verificável). Dessa forma, nossa proposta também é um tipo de assinatura digital especial uma vez que, para obter a redução da assinatura para 32 bits, foi necessário combinar protocolos especiais ao modelo convencional, gerando assinatura curta com as seguintes características:

1. A assinatura é gerada com 32 bits (16 bits para r e 16 bits para s);
2. Segurança adicional foi aplicada cifrando a assinatura gerada com uso de criptografia simétrica antes de ser informada ao sistema de validação, evitando assim que a assinatura seja capturada por um adversário e posteriormente submetida a alguma forma de ataque;
3. Somente o verificador possui os parâmetros (p, q, g , y) do usuário necessários para proceder com a validação da assinatura;
4. O *hash* de 16 bits sobre a transação a ser assinada é obtido através do truncamento da saída da função SHA-256;

5. Devido o seu tamanho reduzido, a assinatura pode ser digitada manualmente pelo signatário, permitindo que seja gerada em Dispositivo Pessoal sem conectividade ao computador, dispensando o uso de software e hardware instalados no computador do usuário.

Com a redução dos parâmetros de assinatura, elaboramos uma análise da segurança da solução, a fim de identificar possíveis vulnerabilidades. A análise mostrou que o ataque à solução não é uma tarefa trivial, e essa segurança foi obtida devido às características especiais agregadas à solução com a cifragem da assinatura e a ocultação dos parâmetros de validação de assinatura. Devido à ocultação dos parâmetros públicos de assinatura, a solução é empregada em um ambiente de *closed* PKI, uma vez que somente o banco possui os parâmetros necessários para validação da assinatura do cliente. Esse ambiente se constitui de um cenário bastante adequado para sistemas de *Internet Banking*, já que neste contexto, o banco é a única entidade que possui interesse em validar assinaturas.

Vale ressaltar que os objetivos destacados no início do trabalho foram alcançados em sua totalidade. A solução apresentada permite a geração de assinatura digital curta de 32 bits utilizando um protocolo especialmente projetado que agrega segurança à solução ainda com a extrema redução no tamanho da assinatura. Embora a proposta aqui apresentada tenha sido contextualizada em aplicações de sistemas de *Internet Banking*, muitas outras aplicações podem ser beneficiadas com o seu uso. A solução apresentada permite a geração da assinatura em dispositivos sem conexão com o computador viabilizando a digitação manual da assinatura. Não requer utilização de certificados digitais, dispensa a necessidade de hardware e software instalados no computador do cliente e por conseqüência, aumenta a mobilidade do cliente, reduz custos e simplifica o processo.

A pesquisa desse trabalho relacionada à redução dos parâmetros de assinatura do DSA foi apresentada em formato de pôster sob o título *Análise da Vulnerabilidade do DSA com Parâmetros Curtos* no I Congresso Sul Catarinense de Computação promovido pela Universidade do Extremo Sul de Santa Catarina (UNESC). O artigo, *A Solution for Special Short Digital Signatures Generated in Personal Devices*, que contém os resultados finais da pesquisa, foi aceito no *7th International Workshop on Information Security Applications (WISA 2006)*, cujos anais são publicados em LNCS.

Como linha de pesquisa futura, podemos citar a implementação da solução apresentada em dispositivos móveis como PDA's e celulares, com um estudo relacionado ao tratamento de segurança necessário para viabilizar a geração da assinatura curta em tais dispositivos.

REFERÊNCIAS

BLEICHENBACHER, D. *On the generation of DSA one-time keys*. Trabalho não publicado, 2001.

BLEICHENBACHER, D. *Experiments with DSA*. CRYPTO 2005 - Rump Session Program, 2005. Disponível em: < http://www.iacr.org/conferences/crypto2005/r/3.pdf>. Acesso em: 01 fev. 2006.

BONEH, D.; LYNN, B.; SHACHAM, H. *Short signatures from the Weil pairing*. In: ADVANCES IN CRYPTOLOGY – ASIACRYPT'2001. Proceedings. Gold Coast, Australia: Springer-Verlag, LNCS, v. 2248, 2002. p. 514-532.

BRANDS, S. A. *Rethinking Public Key Infrastructures and Digital Certificates, Building in Privacy*. The MIT Press, 2000.

CARDOSO, J.; VALLETE, R. *Redes de Petri*. Editora da UFSC, 1997.

CHANG, C. C.; CHANG, Y. F. *Signing a Digital Signature Without Using One-Way Hash Functions and Message Redundancy Schemes***.** Communications Letters, IEEE. v. 8, n.8, 2004. p. 485-487.

CHAUM,D. *Blind signatures for untraceable payments*. In: ADVANCES IN CRYPTOLOGY - CRYPTO '82. Proceedings. D. Chaum, R. L. Rivest, and A. T. Sherman, Eds., Plenum, NY, 1983. p. 199-203.

CHAUM D.; van ANTWERPEN, H. *Undeniable signatures*. In: ADVANCES IN CRYPTOLOGY - CRYPTO '89. Proceedings. Santa Barbara, USA: Springer-Verlag, LNCS, v. 435, 1990. p. 212-216.

CHOW, S. S. M. et al. *A Secure Modified ID-Based Undeniable Signature Scheme based on Han et al.'s Scheme against Zhang et al.'s Attacks***.** Cryptology ePrint Archive, Report 2003/262, 2003. Disponível em: <http://eprint.iacr.org/2003/262>. Acesso em: 30 nov. 2005.

COURTOIS, N.; FINIASZ, M.; SENDRIER, N. *How to achieve a McEliece-based digital signature scheme*. In: ADVANCES IN CRYPTOLOGY – ASIACRYPT'01. Proceedings. Gold Coast, Australia: Springer-Verlag, LNCS, v. 2248, 2002. p. 157-174.

FÄRNQVIST, T. *Number Theory Meets Cache Locality– Efficient Implementation of a Small Prime FFT for the GNU Multiple Precision Arithmetic Library***.** (Mestrado em Ciência da Computação). Stockholm University, 2005.

FEDERAL DEPOSIT INSURANCE CORPORATION - FDIC. *Putting an End to Account-Hijacking Identity Theft Study Supplement*, 2005. Disponível em: <http://www.fdic.gov/consumers/consumer/idtheftstudysupp/idtheftsupp.pdf>. Acesso em: 02 jun. 2006.

FEDERAL FINANCIAL INSTITUTIONS EXAMINATION COUNCIL - FFIEC. *Authentication in an Internet Banking Environment*, 2005. Disponível em: <http://www.ffiec.gov/pdf/authentication_guidance.pdf>. Acesso em: 02 jun. 2006.

GIOVANNINI, E.; BENJAMIN, E.; CONDON, C. *Why Half Of Europe's Net Users Don't Bank Online***.** Forrester Research**,** 2006**.**

GORMAN, L. *Comparing Passwords, Tokens, and Biometrics for User Authentication*. Proceedings of the IEEE, v. 91, n. 12, 2003. p. 2021-2040.

HILTGEN, A.; KRAMP, T.; WEIGOLD, T. *Secure Internet Banking Authentication*. Security & Privacy Magazine, IEEE, v. 4, n. 2, 2006. p. 21-29.

HOOVER, D.; KAUSIK, B. *Software smart cards via cryptographic camouflage*. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY. Proceedings. IEEE Computer Society, 1999. p. 208-215.

HOWGRAVE-GRAHAM, N. A.; SMART, N. P. *Lattice Attacks on Digital Signature Schemes*. Design, Codes and Cryptography, v. 23, n. 3, 2001. p. 283-290.

IEEE P1363 WORKING GROUP. *Standard Specifications for Public-Key Cryptography – IEEE*. [S.1.], IEEE Computer Society, 2000.

JIN, N.; CHENG, M.F. *Network Security Risks in Online Banking***.** In: INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING. Proceedings. 2005. p. 1229-1234.

KIRDA, E.; KRUEGEL, C. *Protecting Users Against Phishing Attacks with AntiPhish.* In: 29th ANNUAL INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE - COMPSAC'05. Proceedings. Washington, DC, USA: IEEE Computer Society, v. 1, 2005. p. 517-524.

KWON, T. *Robust Software Tokens: Towards Securing a Digital Identity*. Cryptology ePrint Archive, Report 2001/039, 2001. Disponível em: <http://eprint.iacr.org /2001/039>. Acesso em: 10 jan. 2006.

LAI, Y., P.; CHANG, C., C. *A simple forward secure blind signature scheme based on master keys and blind signatures*. In: 19th INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS - AINA'05. Proceedings. Taipei, Taiwan: IEEE Computer Society, v.2, 2005. p. 139-144.

MACLANE, S.; BIRKHOFF, G. *Algebra*. 3ª. ed. New York, USA: Chelsea Publishing Company, 1993.

MAO, W. *Modern Cryptography: Theory and Practice*. Hewlett-Packard Company. Prentice Hall PTR, 2003.

MENEZES, A.; OORSCHOT, P. C.; VANSTONE, S. A. *Handbook of Applied Cryptography*. CRC Press, 1996.

NACCACHE, D.; POINTCHEVAL, D.; STERN, J. *Twin Signatures: an Alternative to the Hash-and-Sign Paradigm***.** In: 8th ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY. Proceedings**.** New York, NY, USA: ACM Press, 2001. p. 20-27.

NACCACHE, D.; STERN, J. *Signing on a Postcard***.** In: 4th INTERNATIONAL CONFERENCE ON FINANCIAL CRYPTOGRAPHY. Proceedings. London, UK: Springer-Verlag, LNCS, v. 1962, 2000. p. 121-135.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY – NIST. *Federal Information Processing Standards (FIPS 186-2) - Digital Signature Standard (DSS).* [S.l.], 2000.

_____. *Federal Information Processing Standards (FIPS 197) - Advanced Encryption Standard (AES).* [S.l.], 2001a.

_____. *Recommendation for Block Cipher Modes of Operation*. Special Publication 800-38A, [S.1.], 2001b.

_____. *Federal Information Processing Standards (FIPS 180-2) - Secure Hash Standard (SHS).* (+Change Notice to include SHA-224). [S.1.], 2002.

_____. *Electronic Authentication Guideline*. Special Publication 800-63, [S.1.], 2004.

NGUYEN, P. Q.; SHPARLINSKI, I. *The Insecurity of the Digital Signature Algorithm with Partially Known Nonces*. Journal of Cryptology, v. 15, n. 3, 2002. p. 151-176.

PIETRO, R.; ME, G.; STRANGIO, M. A. *A Two-Factor Mobile Authentication Scheme for Secure Financial Transactions*. In: 4th INTERNATIONAL CONFERENCE ON MOBILE BUSINESS – ICMB'05. Proceedings. IEEE Computer Society, 2005. p. 28- 34.

POHLIG, S. C.; HELLMAN, M. E. *An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*. IEEE Transactions on Information Theory, v. 24, 1978. p. 106-110.

PUBLIC KEY CRYPTOGRAPHY STANDARDS - PKCS. *Password-Based Cryptography Standard (PKCS #5).* RSA Laboratories, [S.1.], 1999.

PUENTE, F.; SANDOVAL, J.; HERNÁNDEZ, P. *Pocket Device for authentication and data integrity on Internet banking applications*. In: IEEE 37th ANNUAL INTERNATIONAL CARNAHAN CONFERENCE ON SECURITY TECHNOLOGY. Proceedings. 2003. p. 43-50.

RIVEST, R., L.; SHAMIR, A.; ADLEMAN, L. *A method for obtaining digital signatures and public-key cryptosystems.* Communications of the ACM, New York, NY, USA: ACM Press, v. 21, n. 2, 1978. p. 120-126.

SCHNEIER, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley Computer Publishing, John Wiley & Sons, Inc., 2ª ed., 1996.

SCHNEIER, B.; KELSEY, J. *Second Preimages on n-bit Hash Functions for Much Less than 2^n Work***.** In: ADVANCES IN CRYPTOLOGY - EUROCRYPT'2005. Proceedings. Aarhus, Denmark: Springer-Verlag, LNCS, v. 3494, 2005. p. 474-490.

STALLINGS, W. *Cryptography and Network Security. Principles and Practice*. 3ª ed., New Jersey, 2003.

STINSON, D. *Cryptography: Theory and Practice*. CRC Press, 2ª. ed., 1995.

ULUDAG, U. et al. *Biometric Cryptosystems: Issues and Challenges***.** Proceedings of the IEEE, v. 92, n. 6, 2004. p. 948-960.

VAUDENAY, S. *Hidden Collisions on DSS*. In: ADVANCES IN CRYPTOLOGY – CRYPTO'96. Proceedings. Santa Barbara, USA: Springer-Verlag, LNCS, v. 1109, 1996. p. 83-88.

XIAO, Q. *Security Issues in Biometric Authentication*. In: 6th ANNUAL IEEE SYSTEMS, MAN AND CYBERNETICS INFORMATION ASSURANCE WORKSHOP. Proceedings. West Point, NY, 2005. p. 8-13.

WANG, X., YAO, A., YAO, F. *New Collision search for SHA-1***.** Rump Session ADVANCES IN CRYPTOLOGY – CRYPTO 2005, 2005.

WANG, X., YIN, Y. L., YU, H. *Finding Collisions in the Full SHA-1*. In: ADVANCES IN CRYPTOLOGY – CRYPTO 2005. Proceedings. Santa Barbara, USA: Springer-Verlag, LNCS, v. 3621, 2005. p. 17-36.