UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

RENATO DUARTE ROZO FONSECA

SATES - Sistema de autenticação para transações eletrônicas seguras

São Paulo
2006

RENATO DUARTE ROZO FONSECA

SATES - Sistema de autenticação para transações eletrônicas seguras

Dissertação apresentada a Escola Politécnica da
Universidade de São Paulo para obtenção do título de
Mestre em Engenharia Elétrica

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. João Antônio Zuffo

São Paulo
2006

Fonseca, Renato Duarte Rozo
SATES – sistema de autenticação para transações eletrônicas seguras / R.D.R. Fonseca. -- São Paulo, 2006.
110 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Redes de computadores 2.Segurança de computadores 3.Codificação da informação 4.Protocolos de comunicação I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

Aos meus pais, Sérgio e Claudia, que sempre me incentivam em quaisquer dos objetivos almejados.

Ao Rodrigo e a minha família por seu apoio.

A Laudicéia, por seu carinho, compreensão, presença durante todo o período do mestrado.

Aos meus amigos, Artur, Fábio, Leonardo e Murilo, que dividem comigo este caminho em busca do conhecimento.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. João Antônio Zuffo, por seu conhecimento, pela oportunidade dada, paciência e incentivo.

Ao Dr. Adilson Eduardo Guelfi, pela experiência, amizade e confiança depositada.

Aos amigos do Grupo NSRAV, pela ajuda e apoio em todos os momentos.

À Escola Politécnica da Universidade de São Paulo e ao Laboratório de Sistemas Integráveis por colocar a disposição sua infra-estrutura.

RESUMO

FONSECA, R. D. R. **SATES – Sistema de Autenticação para Transações Eletrônicas Seguras.** 2006. 110 f. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo, 2006.

A maioria dos sistemas de autenticação atuais são baseados no fator de autenticação de conhecimento. Como alternativa com maior nível de segurança em relação à senha, pode-se citar a certificação digital que provê a autenticação baseada no fator de posse. Porém, a certificação digital não goza de uma ampla difusão nos ambientes computacionais se comparada com senhas. O cartão inteligente representa um dispositivo de armazenamento de certificados digitais e chaves assimétricas e pode ser considerado um dos instrumentos para a difusão da certificação digital. Porém, isto não acontece devido a menor quantidade de aplicações disponíveis e ao custo de implantação, visto que o cartão inteligente necessita de um dispositivo de leitura específico. Assim, este trabalho tem o objetivo de especificar um sistema de autenticação (entidade e autoria), o SATES, o qual agrega um fator adicional (posse) aos sistemas de autenticação usuais baseados em senhas. O fator de posse proposto é dado pelo uso de certificados digitais e por um dispositivo de armazenamento de baixo custo. O dispositivo de baixo custo fornece acesso a um repositório de chaves remoto e seguro, e permite que o usuário recupere as chaves armazenadas sob demanda. Neste trabalho, também são encontradas medidas de desempenho do sistema proposto de acordo com três modelos de implementação: A, B e C. Em termos de desempenho o modelo C apresentou aproximadamente o dobro do tempo requerido pela execução dos modelos A e B, isto ocorre porque o modelo C utiliza um servidor adicional (servidor de parâmetros) com relação aos modelos A e B, que também utiliza os protocolos SSL e SRP.

ABSTRACT

Nowadays most of the authentication systems are based on the knowledge authentication factor. As an alternative solution with higher security level than passwords, can mention digital certification which provides authentication based on the possession authentication factor. However, the digital certification does not have wide dissemination on computer's environment if compared to passwords. Smart card represents a storage device for digital certificates and asymmetric keys and could be considered one of the instruments to disseminate the digital certification. However, this does not happen deeply because the low amount of applications available and the implementation costs, since the smart card requires a specific reader. Thus, this work has the objective of specifying an authentication system (entity and authorship), SATES, which associates an additional authentication factor (possession) to the usual authentication systems based on passwords. The possession authentication factor proposed is given by the use of the digital certificates and by a low cost storage device. This low cost device provides access to a secure key repository, and permit the user to recover stored keys on demand. On this work, performance measures of the proposed system following three implementation models. By performance terms, the C model has presented approximately double of the time required to execute the models A and B, this occurs because the C model uses an additional server (parameters server) relative to A model and B model, which also uses SSL and SRP protocols.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE ABREVIATURAS

AES	*Advanced Encryption Standard*
API	*Application Program Interface*
CD-ROM	Disco compacto com memória somente de leitura, *Compact Disk Read Only Memory*
CMS	*Cryptographic Message Syntax*
CSR	*Certificate Signing Request*
DES	*Data Encryption Standard*
EKE	*Encryption Key Exchange*
HTTP	*HiperText Transfer Protocol*
LCR	Lista de Certificados Revogados
OATH	*Open AuTHentication Organization*
OTP	*One Time Password*
PIN	Número de identificação pessoal, *Personal Identification Number*
PKCS	*Public Key Cryptography Standard*
PKDF1	*Password Key Derivation Function 1*
SATES	Sistema de Autenticação para Transações Eletrônicas Seguras
SHA1	*Secure Hash Algorithm revision 1*
SPEKE	*Simple Password Exponential Key Exchange*
SRP	*Secure Remote Protocol*
SSH	*Secure Shell Protocol*
SSL	*Secure Sockets Layer*
TLS	*Transport Layer Secure*
USB	*Universal Serial Bus*

SUMÁRIO

1 Introdução

Este capítulo é dedicado a contextualizar os ambientes atuais de autenticação, determinar a motivação e o objetivo do trabalho, bem como descrever sua organização.

1.1 Contexto

Um serviço de segurança determina uma funcionalidade relacionada à segurança provida por um módulo de *software* ou *hardware*. Os principais serviços de segurança são:

- Serviço de Autenticação;
- Serviço de Sigilo;
- Serviço de Integridade;
- Serviço de Não repudio;
- Serviço de Disponibilidade;
- Serviço de Controle de Acesso;
- Serviço de Auditoria.

O serviço de autenticação tem como objetivo validar a identidade de uma dada entidade (por exemplo, pessoa, máquina, serviço, empresa, processo, etc.) e pode ser dividido em dois tipos: autenticação de entidade e autoria.

O serviço de autenticação de entidade representa um conjunto de algumas modalidades de autenticação, por exemplo: um usuário realizando a autenticação em um sistema operacional, um parceiro de comunicação estabelecendo um canal de dados. Assim, quando se refere a uma autenticação para obtenção de acesso ou durante uma comunicação,

pode-se referenciar a autenticação de entidade. Entretanto, o serviço de autoria provê a possibilidade de associação de uma ou mais entidades à concepção de dados.

A utilização de sistemas criptográficos pode prover diversos serviços de segurança, tais como, sigilo, autenticação, integridade e não repúdio. Especificamente para os dois tipos de serviços de autenticação (entidade e autoria), dentro do domínio dos sistemas criptográficos, algoritmos de criptografia simétrica e assimétrica podem ser empregados. No caso de algoritmos de criptografia simétrica, devido à necessidade de compartilhamento de um segredo, ou seja, a chave simétrica, tais algoritmos não alcançam um nível adequado de segurança e, portanto, são menos utilizados para esta finalidade.

Os algoritmos de chave assimétrica, também denominados de algoritmos de chave pública, utilizam duas chaves, uma pública e uma privada, as quais são complementares entre si nas operações de criptografia. A propriedade complementar das chaves possibilita que a entidade proprietária do par mantenha protegida a sua chave privada e possa divulgar a chave pública correspondente. Por exemplo, caso seja utilizada a chave privada em uma cifragem de dados, a decifragem é dada pelo uso da chave pública correspondente. Supondo que somente uma entidade tem o controle de uma determinada chave privada, a pública correspondente é capaz de identificar a origem de dados por ela decifrados. Assim, a criptografia de chave pública provê uma opção mais adequada para autenticação.

O algoritmo de chave pública também está relacionado à concepção de certificados digitais e, consequentemente, ao modelo infra-estrutura de chaves públicas (ICP). Um certificado digital representa um mecanismo para a distribuição confiável de chaves públicas, visto que já estão associadas informações e uma chave pública ao seu proprietário. Um certificado digital é sempre emitido por uma autoridade certificadora (AC) que desempenha o papel de uma entidade confiável (âncora de confiança). Uma ICP representa uma infra-estrutura necessária para o gerenciamento do ciclo de vida dos certificados digitais. Neste

caso, gerenciar o ciclo de vida de um certificado digital abrange atividades de emissão, renovação e revogação do certificado.

Como exemplo de autenticação de entidade utilizando a certificação digital, pode-se citar o protocolo *Secure Sockets Layer* (SSL) (FRIER; KARLTON; KOCHER, 1996), o qual tem o objetivo de estabelecer um canal seguro com sigilo e prove meios de autenticar as entidades em uma comunicação. A assinatura digital, que também faz uso de um certificado digital, é um exemplo do método que implanta o serviço de autoria.

A autenticação pode ser classificada baseando-se em três fatores (MENEZES et al., 1997), que são utilizados de forma independente ou conjugados:

- Biométrico (algo que o usuário é) – abrange sistemas biométricos, tais como, identificação de parâmetros pela impressão digital, íris, retina entre outros;
- Posse (algo que o usuário possui) – crachás, cartões inteligentes;
- Conhecimento (algo que o usuário sabe) – senhas, número de identificação pessoal (*personal identification number* – PIN).

O fator biométrico exige equipamentos especializados e o nível de segurança alcançado depende dos custos necessários, precisão dos equipamentos e algoritmos.

O fator de posse pode exigir ou não um dispositivo de interface com o objeto que o usuário possui. Por exemplo, a tecnologia de cartão inteligente que necessita de uma leitora para a sua comunicação com o computador. Outro exemplo que representa o fator de posse é o dispositivo *one-time password* (OTP) que fornece um visor, no qual é exibido um código que, normalmente, deve ser concatenado a uma senha de conhecimento do usuário. Portanto, um dispositivo OTP é um exemplo de fator de posse que não necessita de um dispositivo de interface específico.

O fator de conhecimento, na maioria dos casos, é representado por um mecanismo de autenticação de entidade baseado em dois parâmetros, usuário e senha. Este fator embora seja bastante utilizado apresenta algumas fragilidades que serão apontadas na próxima seção.

1.2 Justificativa e motivação

Hoje em dia, a autenticação de usuários por meio dos parâmetros de identificação do usuário e senha é bastante difundida na rede mundial, na qual se encontram meios de comunicação não confiáveis. Assim, ao longo do tempo, e com o desenvolvimento dos sistemas computacionais, a autenticação baseada em senha nos moldes comuns tem se mostrado frágil em relação a diversos aspectos, tais como:

- Senhas de fácil adivinhação – o usuário tende a associar uma senha a alguma coisa familiar, como por exemplo, números de telefone e datas. Portanto, técnicas como a coleta de determinadas informações pessoais ou engenharia social poderia orientar a realização de ataques de adivinhação de senhas;
- Tamanho da senha – a entropia[1] é favorecida conforme aumenta o tamanho da senha. O tamanho de uma senha é diretamente proporcional ao tempo gasto na realização de um ataque de força bruta, no qual a senha pode ser adivinhada por meio de tentativas que usam todas as possíveis combinações. Para o aumento da segurança, também se deve levar em consideração a variedade de caracteres empregados na formação da senha. Nos sistemas de autenticação atuais, usuários podem escolher senhas que contenham caracteres especiais[2], letras e números, inclusive a diferenciação sobre letras maiúsculas e minúsculas (*case sensitive*);

[1] Grau de desordem, impresivibilidade.
[2] Caracteres como (!, @, ?, $).

- Senhas equivalentes em diversos ambientes – a maioria dos sistemas computacionais acessados solicita diferentes autenticações ao usuário. Neste cenário, o usuário normalmente utiliza a mesma senha nos diversos ambientes em que atua, aumentando o risco de prejuízo caso uma pessoa maliciosa obtenha esta senha;
- Senhas trafegadas sem proteção – os dados do usuário enviados pela rede, como por exemplo, a identificação do nome do usuário e sua senha, podem trafegar de forma legível, ou seja, sem proteção de sigilo. Deste modo, um atacante poderia realizar uma captura[3] de pacotes na rede, e observar as informações do usuário sem maiores dificuldades.

Devido a estas fragilidades, a tendência é que senhas poderiam existir em menor escala, porém continuam ainda sendo comumente empregadas na autenticação de usuários. O uso de certificados digitais, cartões inteligentes e equipamentos biométricos para propiciar outros fatores de autenticação estão longe de substituir a utilização das senhas (SANDHU; BELLARE; GANESAN, 2002), pois estes dispositivos apresentam como desvantagens, tanto o custo de investimento em *hardware* específico, como também a menor quantidade de componentes necessários ao funcionamento destes dispositivos. Entende-se aqui por componentes bibliotecas de desenvolvimento padronizadas, documentação e exemplos.

Portanto, baseando-se nos aspectos até então discutidos, que determinam que senhas apresentam uma série de fragilidades, mas gozam de grande aceitação dos usuários e ampla utilização pelos sistemas, este trabalho visa fornecer uma técnica alternativa de menor custo em relação aos dispositivos como cartões inteligentes ou biométricos. Técnica esta que não deve atingir os mesmos níveis de segurança, porém será capaz de fornecer um fator adicional de autenticação ao modelo tradicional baseado em senha, que contempla apenas o fator

[3] S*niffing, Eavesdropping* – captação de pacotes na rede, independendo do destino, também conhecido como modo promíscuo de recepção de pacotes.

conhecimento. Além disso, este trabalho também faz uso da certificação digital possibilitando a autenticação mútua entre entidades.

1.3 Objetivo do trabalho

O objetivo deste trabalho é propor um sistema de autenticação para transações eletrônicas seguras, chamado de SATES. O SATES utiliza dois mecanismos de autenticação de entidade, um baseado na certificação digital e outro baseado na utilização dos parâmetros usuário e senha. Além disso, o SATES também acrescenta um fator de autenticação de posse em relação ao modelo tradicional que apenas utiliza o fator de autenticação de conhecimento, baseado em senha. Portanto, mantendo ainda a idéia de autenticação baseada em senha, o SATES proporciona um nível de autenticação melhor devido à utilização conjunta de dois mecanismos e fatores de autenticação. O fator de autenticação de conhecimento é representado pela senha e o fator de autenticação de posse é representado por dois elementos, uma chave privada e um dispositivo de baixo custo.

1.4 Estruturação do documento

Esta dissertação está organizada da seguinte forma: o capítulo 2 aborda conceitos de criptografia; o capítulo 3 descreve os conceitos de autenticação relacionados ao tema do trabalho; o capítulo 4 é dedicado a listar alguns trabalhos relacionados; o capítulo 5 expressa de forma mais detalhada os objetivos da proposta deste trabalho; o capítulo 6 mostra as etapas de desenvolvimento do projeto; o capítulo 7 descreve as conclusões obtidas no

desenvolvimento do trabalho; o capítulo 8 lista as referências bibliográficas citadas ao longo deste documento.

2 Sistemas Criptográficos

Os sistemas criptográficos fornecem os serviços de segurança de: sigilo, integridade, autenticação e não repudio. Um algoritmo de criptografia, seja simétrico ou assimétrico, deve ter como princípio ser de conhecimento público, ou seja, o algoritmo de criptografia deve ser aberto quanto aos seus métodos e modelagem matemática. Outro princípio da área de sistemas criptográficos define que a segurança ou robustez da criptografia nunca deve estar baseada na ocultação do algoritmo.

Assim, toda proteção e força proporcionadas pela criptografia devem estar baseadas em um algoritmo forte, e também no segredo e tamanho da chave criptográfica utilizada.

2.1 Criptografia simétrica

Os algoritmos de criptografia simétrica ou convencional utilizam uma única chave. Deste modo, a chave utilizada na cifragem é a mesma para a decifragem. Como se pode perceber na Figura 1, o texto plano (T) é submetido pela entidade A ao processo de cifragem com a chave simétrica (Ks) produzindo o texto cifrado (X). A entidade B, tendo em posse a chave simétrica, realiza a decifragem recompondo o texto plano original (T).

A criptografia simétrica, se comparada à criptografia assimétrica, possui um custo computacional reduzido, porém com algumas características negativas:

- Para que seja possível a troca de mensagens de forma sigilosa entre parceiros é necessário o compartilhamento do segredo da chave simétrica de criptografia.

Desta forma, não é contemplado o serviço de segurança de não repudio, pois existe o conhecimento da chave simétrica entre os parceiros;

- Para a distribuição segura da chave simétrica é necessária a escolha de um algoritmo de estabelecimento de chaves;
- Em um algoritmo simétrico, o nível de autenticação não é tão adequado, porque o nível está baseado no conhecimento e na posse da chave criptográfica que é compartilhada entre as entidades (no caso da Figura 1, A e B) que se comunicam.

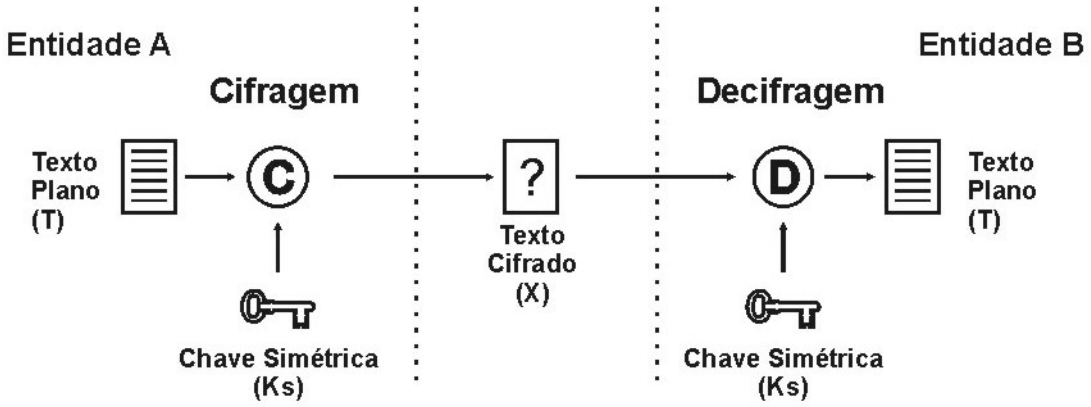

Figura 1 - Criptografia simétrica

Dentro do contexto deste trabalho o algoritmo de criptografia é utilizado no estabelecimento de um canal seguro e na proteção de informações do usuário.

A criptografia simétrica tem com principais algoritmos:

- DES – utiliza chaves com o tamanho de 56 bits;
- Triple DES – utiliza chaves com os tamanhos de 112 e 168 bits;
- RC4 – o tamanho mais comum de chave é igual a 128 bits;
- AES - utiliza chaves com os tamanhos de 128, 192 e 256 bits.

2.2 Criptografia assimétrica

Os algoritmos de criptografia de chave pública utilizam duas chaves criptográficas por cada entidade, como comentado na seção 1.1, sendo uma para cifrar e outra para decifrar. Neste caso, a regra geral do modelo de criptografia de chave pública estabelece que, se uma mensagem for cifrada com uma chave, a única chave que permite restaurar a mensagem original é a chave parceira.

Assim, existem duas chaves conforme mostra a Figura 2, uma pública e outra privada. A chave pública pode ser de conhecimento de todos, enquanto que a chave privada deve ser mantida de forma secreta pela entidade. A forma de utilização das chaves pode prover a idéia de diferentes serviços de segurança. Por exemplo, a cifragem de dados utilizando a chave privada pode ser usada como forma de identificar a origem de dados (autoria), pois se admite que a chave privada seja apenas de conhecimento e posse da entidade proprietária (no caso da Figura 2, a entidade (A)). Portanto, conforme mostra a Figura 2, somente a entidade (A) seria capaz de produzir o texto cifrado X.

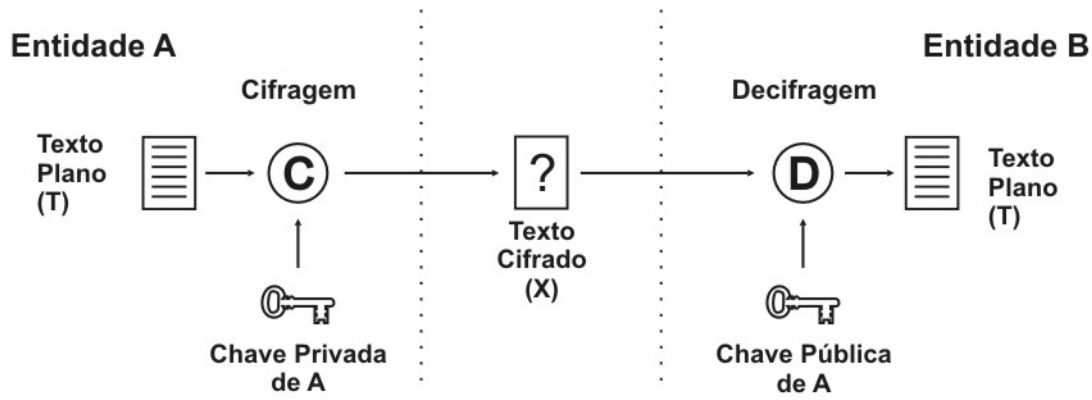

Figura 2 - Criptografia assimétrica - Autoria

Utilizando-se a chave pública de uma entidade parceira de comunicação consegue-se obter o serviço de sigilo. Conforme exibe a Figura 3, o sigilo é obtido por meio da utilização da chave pública de B sobre o texto plano (T), o resultado da operação é o texto cifrado (X). Assim, somente a entidade B pode decifrar o texto cifrado (X), já que, supostamente, somente esta entidade (B) tem a posse da chave privada correspondente.

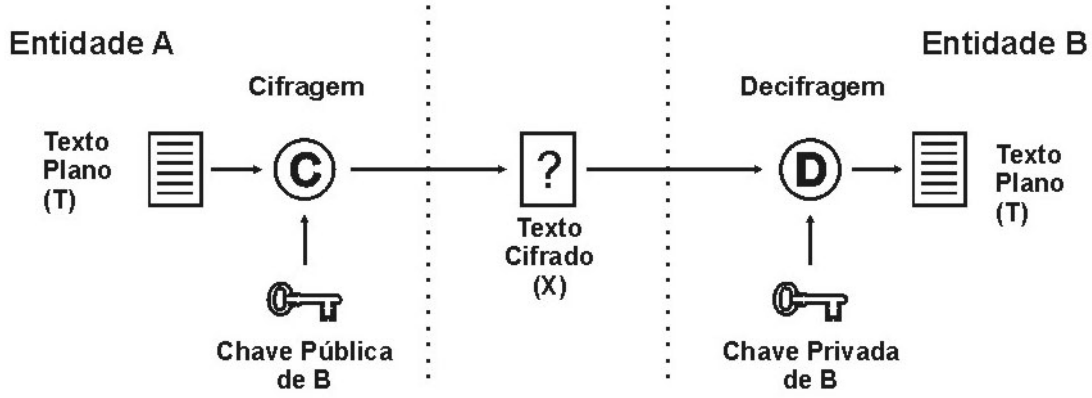

Figura 3 - Criptografia assimétrica - Sigilo

Entretanto, o algoritmo de chaves públicas possui também algumas desvantagens:

- Desempenho – a execução da criptográfica assimétrica é mais lenta do que a criptografia simétrica;
- Armazenamento da chave privada – devido à importância relacionada à posse da chave privada, o armazenamento de tal elemento deve ser feito de forma segura;
- Autenticidade da chave pública – o algoritmo de criptografia de chave pública não especifica como se deve associar uma entidade ao seu par de chaves.

Como proposta de solução do problema da autenticidade da chave pública pode-se citar a certificação digital.

Os principais algoritmos de criptografia de chave pública são:

- RSA – este algoritmo tem os valores de 1024 e 2048 bits como mais comuns para o tamanho de chave;

- Curvas elípticas – este algoritmo é baseado na teoria de curvas elípticas e tem como tamanhos comuns de chave os valores 160, 224 e 256 bits.

2.3 Resumo criptográfico (Hash)

Uma função *hash* representa uma função matemática usada para prover o serviço de integridade de dados. A função *hash* envolve como entrada todos os bits da mensagem, gerando como saída um código denominado de código *hash*. Por envolver todos os bits da mensagem, uma função *hash* permite detectar erros, pois qualquer alteração (de um bit ou conjunto de bits) deve provocar como saída um código *hash* diferente do original. Conforme mostra a Figura 4, qualquer que seja o parâmetro de entrada representado por "Texto" obtém-se como resultado um conjunto de bits representado pelo "Resumo".

O código *hash* tem um tamanho fixo dependente do algoritmo utilizado. Os algoritmos mais utilizados atualmente são: o MD5 e o SHA-1, respectivamente, com tamanhos de código *hash* equivalentes a 128 e 160 bits.

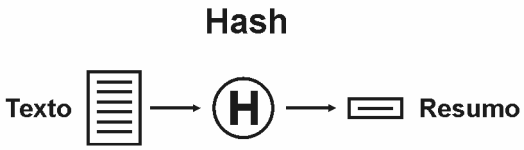

Figura 4 - Resumo criptográfico

A operação de *hash* faz parte do processo de assinatura digital, o qual é descrito na próxima subseção.

2.4 Assinatura digital

A assinatura digital é a combinação da criptografia assimétrica com a produção de um resumo criptográfico, isto é, para que seja produzida uma assinatura digital deve-se primeiro realizar a função de *hash* sobre os dados a serem assinados, e em seguida executar a cifragem *hash* gerado com a chave privada do autor da assinatura. A Figura 5 apresenta esquema do processo de assinatura digital. Em primeiro lugar, a função *hash* é aplicada sobre o valor "Texto", em seguida o código *hash* gerado é submetido à cifragem com a chave privada. Desta forma, por meio do valor "Texto" e do uso da chave privada é gerada uma assinatura digital.

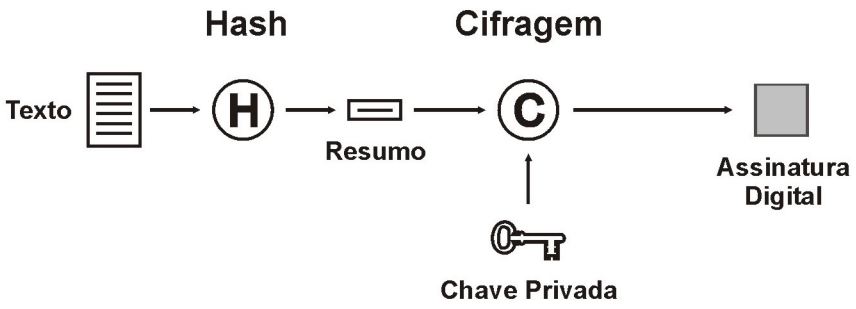

Figura 5 – Esquema de Assinatura Digital

A assinatura digital tem como desvantagens:

- Autenticidade da chave pública – como citado na seção 2.2 referente à associação do par de chaves;
- Determinar o instante de assinatura – pode não ser possível saber quando uma assinatura digital foi realizada sem a presença de estampilha de tempo confiável.

O formato mais comum utilizado é o *Public Key Cryptography Standard* #7 (PKCS#7) (KALISKI, 1998), que uma mensagem estruturada por meio da notação ASN.1

(ITU-T Recommendation X.690). O formato PKCS#7 tem uma proposta mais recente, o *Cryptographic Message Syntax* (CMS) (HOUSLEY, 2002).

Ambos os formatos, PKCS#7 ou CMS, podem ser dos tipos *attached* ou *detached*. O tipo *attached* representa que o documento assinado é inserido na mensagem. O termo documento refere-se aos dados assinados, assim como, o valor "Texto" ilustrado na Figura 5. Já o tipo *detached* na conta com o documento. Além do texto, podem ser inseridas informações sobre a assinatura e signatários.

2.5 Certificação digital

O certificado digital tem o seu formato especificado pelo padrão X.509 (ITU-T *Recommendation* X.509, 2000; ADAMS et al, 2005), o qual define que um certficado é constituído por uma série de informações, tais como: dados de identificação do usuário (por exemplo, nome, RG, CPF, etc), restrições, atributos, extensões, e outras informações que representem o proprietário. Assim, uma informação importante contida no certificado digital é a chave pública que pode ser relacionada à chave privada de posse do proprietário.

A certificação digital está baseada na relação de confiança mantida por uma entidade confiável (âncora de confiança) denominada de autoridade certificadora (AC). A autoridade certificadora raiz encontra-se no topo da cadeia de confiança. Além da raiz, a cadeia de confiança também é composta por certificados de autoridades certificadoras intermediárias e por certificados de entidades finais (por exemplo, pessoa física, pessoa jurídica, servidor, aplicação, etc).

Supondo três entidades A, B e C, sendo A uma AC raiz, B uma AC intermediária e C um certificado emitido para um usuário, para que C seja confiável devemos confiar em A, a

qual assinou o certificado de B, que emitiu o certificado de C. Desta forma, é estabelecida a relação de confiança, como mostra a Figura 6.

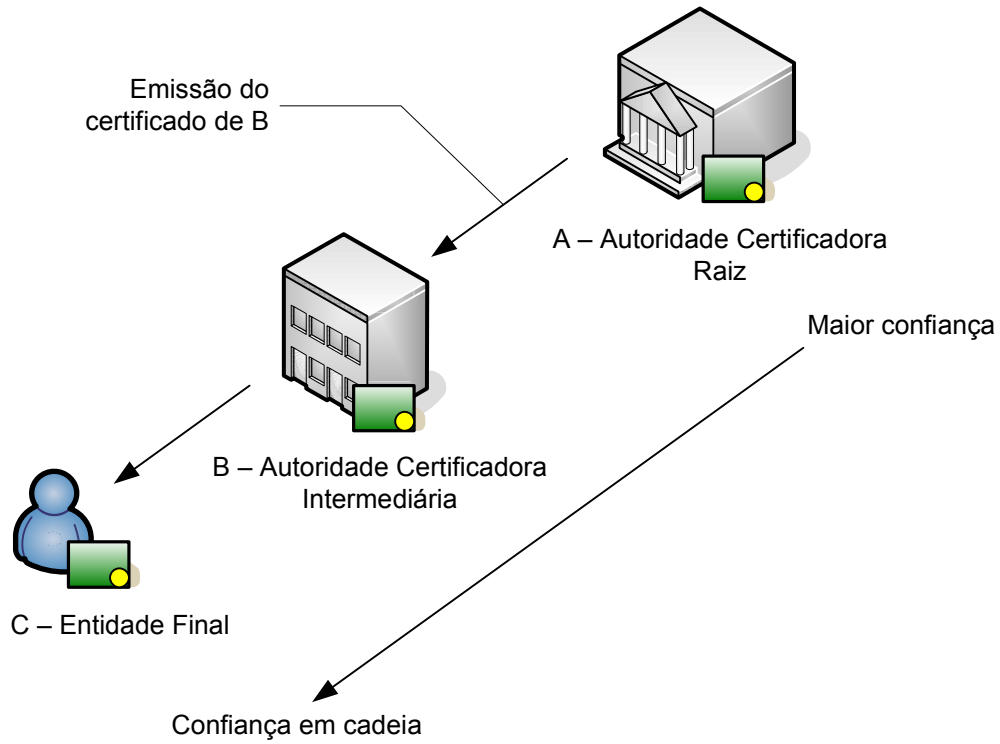

Figura 6 - Relação de confiança - certificação digital

O uso da certificação digital requer que um certificado seja verificado quanto a sua integridade, quanto ao seu período de validade e revogação.

Os certificados, geralmente, são emitidos com um período de validade de um ano, sendo parametrizadas as datas de início e fim.

A revogação de um certificado digital é o processo que determina o instante em que o certificado perde a sua validade, independentemente da data de expiração. A revogação é publicada em uma lista de certificados revogados, a qual tem a propriedade acumulativa mantendo o registro de todos os certificados revogados por uma determinada AC. O processo de revogação é importante para proteger a identidade do proprietário, por exemplo, em uma situação que a chave privada seja divulgada a uma entidade não autorizada.

Dentro da noção de certificação digital podemos identificar três componentes principais. São eles: a entidade final ou proprietário, a autoridade certificadora e a autoridade registradora.

A seguir apresenta-se uma descrição de algumas atividades relacionadas a cada um destes componentes.

2.5.1 Entidade Final

A entidade final que deseje obter um certificado digital deve reunir a documentação necessária para sua identificação e também fazer uma requisição de certificado digital. O processo de requisição de certificado digital gera uma informação chamada *Certificate Signing Request* (CSR) seguindo o padrão PKCS#10 (NYSTROM, 2000; SCHAAD, 2005).

A entidade final deve armazenar de forma segura a sua chave privada. Em caso de comprometimento, deve solicitar a revogação do certificado.

Para que a confiança seja estabelecida como exibido na Figura 6, a entidade final deve manter uma base de certificados de autoridades certificadoras raiz e intermediárias confiáveis para obter segurança durante a verificação de assinaturas digitais. Deste modo, percebe-se que a entidade final pode estabelecer a cadeia confiança de certificados desde AC raiz que representa a âncora de confiança da cadeia de certificados.

A entidade final tem a opção de não verificar um certificado quanto a sua revogação. Porém, dependendo da situação, ao não realizar a verificação, poderá propiciar riscos de segurança. Por outro lado, a verificação excessiva pode onerar o sistema que utiliza a certificação digital quanto ao desempenho, visto que é requerida uma recepção de dados da LCR, cujo tamanho tende a aumentar caso outras soluções não sejam adotadas.

2.5.2 Autoridade Certificadora

A autoridade certificadora tem como principais responsabilidades a realização das seguintes tarefas:

- Emitir, renovar e revogar certificados digitais;
- Emitir e publicar listas de certificados revogados, para que possam ser realizadas as verificações pelos usuários;
- Armazenar de forma segura a sua chave privada, devido à relação de confiança das chaves privadas emissoras de certificados.

2.5.3 Autoridade Registradora

As principais atividades que uma autoridade registradora pode desempenhar são:

- Validar a identidade do usuário;
- Encaminhar o pedido de emissão aprovado de certificado;
- Encaminhar o pedido de revogação de um certificado;

Portanto, a certificação digital é capaz de fornecer mecanismos para a sua utilização de forma correta, prevendo inclusive a revogação ocorrida durante a validade de um certificado.

2.6 Considerações finais sobre criptografia

A união dos conceitos de sistemas criptográficos discutidos neste capítulo fornece a base para o entendimento da proposta deste trabalho. O uso adequado da certificação digital e suas verificações podem proporcionar um bom nível de autenticação. A criptografia simétrica é responsável pelo sigilo. Os conceitos de *hash* e assinatura digital discutidos neste capítulo também estão inseridos na proposta deste trabalho, porém com menor ênfase.

O capítulo 3 vai discutir outros conceitos necessários para o entendimento deste trabalho.

3 Autenticação Digital

Este capítulo descreve outros conceitos de autenticação importantes desde os mais antigos mecanismos de autenticação até a arquitetura geral dos mecanismos de autenticação atuais.

Neste capítulo, o termo autenticação de servidor se enquadra na definição de autenticação de entidade. O termo texto ou senha em plano também é referenciado e tem a denotação de um valor legível sem qualquer proteção por um algoritmo de criptografia.

3.1 Autenticação Fraca

Atualmente, existe uma série de métodos para autenticação que são considerados fracos e obsoletos. Muitas destas alternativas ainda podem ser utilizadas, como por exemplo, mecanismos de autenticação de entidade que utilizam senhas em plano. Estes mecanismos podem ainda persistir devido a características relacionadas à implantação, como o baixo nível de dificuldade e baixo custo. Todos os mecanismos descritos nesta seção são baseados no fator de conhecimento.

3.1.1 Senha em plano

Este é o mecanismo de autenticação mais simples. A senha do usuário é armazenada, em uma base de dados, da mesma forma que foi digitada, ou seja, em plano sem nenhuma proteção.

Existem inúmeros problemas relacionados a este mecanismo, como por exemplo, um usuário administrador pode ler sem dificuldades as senhas dos usuários, cópias destas senhas podem ser obtidas a partir de cópias de segurança do sistema de autenticação e a forma de tráfego da senha.

Concluindo, o mecanismo de autenticação de senha em plano além de ser de baixa complexidade é obsoleto e inseguro.

3.1.2 "Resumo" Criptográfico de Senhas

Uma evolução do mecanismo apresentado na subseção 3.1.1 é o resumo criptográfico (*hash*) de senhas. Este mecanismo armazena em um servidor o resultado da operação de *hash* sobre a senha do usuário. Desta forma, não se torna trivial interpretar a informação armazenada.

A operação de *hash* provê de forma matemática a "inviabilidade computacional" de recomposição de um dado a partir do seu respectivo resumo. Portanto, este mecanismo não sofre de um ataque sobre o algoritmo, mas pode sofrer um ataque de comparação sobre o resultado do resumo.

Como exemplos de um ataque de comparação viável sobre as informações armazenadas no servidor, podem-se citar os ataques de tentativas de adivinhação ou força bruta, onde são testadas palavras já formadas (dicionário) ou inúmeras combinações de caracteres, visto que o custo computacional de uma operação de *hash* é baixo. Além disso, existem dicionários que contam com inúmeras combinações pré-calculadas (*Rainbow-Tables*[4]), facilitando ainda mais um ataque.

[4] http://www.rainbowtables.net/ . Este site comercializa dicionários de senhas pré-calculadas para algoritmos de hash.

Para aumentar a dificuldade de ataques sobre a base de resumos pode ser implementada a técnica de utilização do *salt*. O *salt* representa um valor que geralmente é concatenado à senha com o objetivo de aumentar a entropia e o tamanho do dado a ser calculado. Portanto, além da dificuldade de descoberta da combinação da senha deve-se somar a dificuldade proporcionada pelo uso do *salt*. Contudo, o armazenamento do *salt* também passa a ser um problema.

Além disso, normalmente, a senha é trafegada de forma desprotegida e somente no servidor é realizada a operação de *hash*. Portanto, o canal de comunicação também é um meio vulnerável.

Em geral, protocolos de autenticação baseados em senhas podem sofrer os ataques de adivinhação ou força bruta. As tentativas de adivinhação podem funcionar de acordo com dois modos:

- Conectado (*On-line*) – neste modo, um atacante seleciona uma senha do dicionário e tenta personificar um usuário do sistema de autenticação interagindo ativamente com o servidor. O processo se repete cada vez que uma tentativa falha. Este caso é mais fácil de detectar e prevenir;
- Desconectado (*Off-line*) – neste modo, o atacante não precisa interagir com entidades legítimas do sistema de autenticação. Pode ser executado com uma ou mais máquinas de forma que todas as possibilidades, ou valores de um dicionário, estejam divididas entre as máquinas envolvidas para otimização do ataque.

3.1.3 Desafio e Resposta

O mecanismo de desafio e resposta mais comum tem seu funcionamento relacionado à geração de um dado aleatório, que representa o desafio, pelo servidor. Na seqüência, o cliente

recebe o desafio para ser utilizado junto com a sua senha no processamento da resposta. Desta forma, a senha nunca é transmitida de forma legível. Portanto, este mecanismo atenta aos problemas de canais de comunicação inseguros.

Uma entidade não autorizada após a captura do tráfego de autenticação pode realizar um ataque de força bruta sobre a senha do cliente, pois tendo o conhecimento das mensagens do protocolo é possível identificar o valor do desafio no tráfego capturado. Além disso, também é conhecida a função de processamento, que tem como entradas os valores do desafio e a senha. Portanto, como acontece para o ataque sobre a base de resumos, podem ser selecionados valores de senhas para tentativas de descoberta.

3.2 Arquitetura geral de autenticação de entidades

Esta seção tem o objetivo de abordar os principais componentes existentes em sistemas de autenticação de entidades. Para esta seção, o termo autenticação apresentado no texto e em figuras refere-se à autenticação de entidades.

A Figura 7 é baseada no documento *Reference Architecture* do OATH (2006) e ilustra a arquitetura de autenticação e a ligação entre os componentes em destaque, são eles:

- Métodos de autenticação;
- Dispositivos de autenticação;
- Aplicativo cliente;
- Protocolos de autenticação;
- Aplicativo Servidor de autenticação;
- Aplicações.

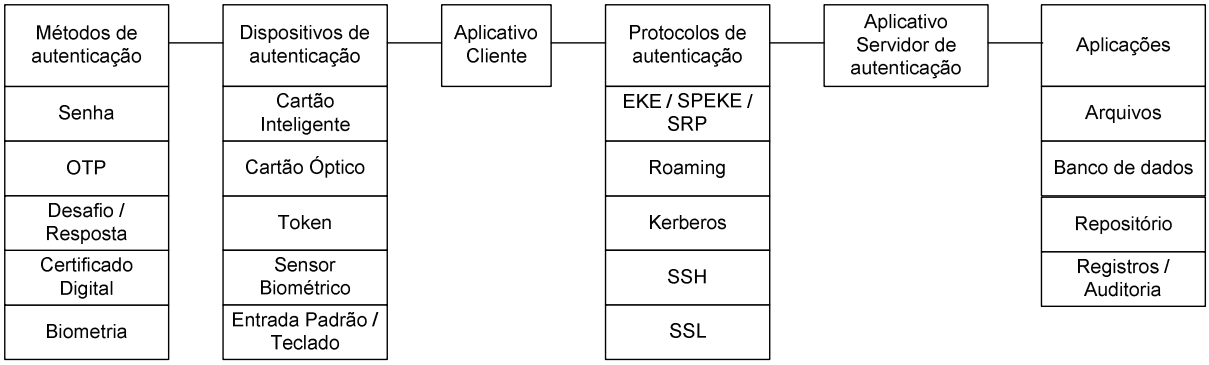

Figura 7 - Arquitetura Geral de Autenticação

Os sistemas de autenticação podem variar quanto à presença dos quatro primeiros componentes exibidos na Figura 7, inclusive podendo haver mais de um tipo do mesmo componente. No entanto, dois componentes, no mínimo, devem estar presentes em todos os sistemas: o método de autenticação, essência da autenticação; e o aplicativo cliente, responsável por implementar o método escolhido.

O aplicativo cliente, como dito, possui as rotinas necessárias para a execução de um método de autenticação. Além disso, o aplicativo cliente pode também realizar a interface com os protocolos e os dispositivos de autenticação.

Os outros componentes da arquitetura de autenticação, com exceção dos componentes servidores (Aplicativo Servidor de autenticação e Aplicações), e entidade são apresentados juntamente com seus principais exemplos nas próximas seções.

3.2.1 Métodos de autenticação

Os métodos de autenticação podem ser classificados pelos elementos que representam os fatores de autenticação. Um método de autenticação pode estar relacionado ou não a um dispositivo de autenticação.

3.2.1.1 Senha

O método de autenticação baseado em senha representa o fator de autenticação de conhecimento discutido nas seções 1.2 e 3.1.

3.2.1.2 Desafio / Resposta

O método de autenticação desafio/resposta representa o fator de autenticação de conhecimento e segue as características apresentadas na seção 3.1.3, onde se comentou sobre o mecanismo de autenticação baseada em senha.

De forma genérica, o método se resume na interação entre duas entidades, normalmente, um cliente e um servidor. A interação se inicia por meio da geração de um desafio (por exemplo, 10) pelo servidor, o qual é recebido e submetido a uma função (por exemplo, $f(x) = 2x$) pelo cliente. Em seguida, o resultado da função (seguindo o exemplo, 20) representa a resposta enviada ao servidor. Por fim, o servidor verifica se a resposta é o valor esperado, visto que o servidor conhece a função aplicada pelo cliente. Portanto, a autenticação é dada pela habilidade do cliente autorizado a executar uma função sobre o desafio e ter a resposta esperada pelo servidor.

3.2.1.3 *One Time Password* - OTP

O método de autenticação OTP (HALLER; METZ, 1996) permite que o usuário utilize uma senha diferente cada vez que seja necessária a autenticação, e compreende a utilização de, no mínimo, um servidor. Dispositivos ou cartões impressos podem melhorar o

nível de autenticação do método de autenticação OTP. O servidor é capaz de prever, de forma conectada ou não, a senha OTP a ser fornecida pelo usuário para efetuar a autenticação.

As senhas OTP podem ser geradas das seguintes formas:

- No início é gerada uma lista com todas as senhas OTP. Conforme são realizadas as interações com o servidor, as senhas vão sendo consumidas. Como exemplos, livros-código, cartões impressos de desafio utilizados por clientes de bancos. Esta forma de geração representa o fator de autenticação de posse;
- As senhas OTP podem ser derivadas de uma senha escolhida pelo usuário no momento de seu cadastro. A derivação é realizada por meio de um algoritmo que utiliza os parâmetros da senha do usuário e dados do servidor. O algoritmo de derivação normalmente recebe o nome de calculadora OTP. Como, por exemplo, programa Java OTP[5]. Esta forma de geração representa o fator de autenticação de conhecimento respectivo ao conhecimento do valor semente de geração e da senha;
- A derivação das senhas está vinculada ao tempo e a uma informação secreta armazenada na memória interna do dispositivo. Esta forma de geração de senhas está relacionada ao fator de autenticação de posse, como exemplo, os dispositivos de autenticação como o RSA SecurID[6].

A principal vantagem do OTP é dada pelo fato de uma senha poder ser utilizada apenas uma única vez, contemplando, assim, a idéia de senhas descartáveis, que, se comprometidas posteriormente, não propiciam vulnerabilidades, visto que as mesmas não são mais válidas.

[5] Exemplo disponível em http://www.cs.umd.edu/~harry/jotp/
[6] http://www.rsasecurity.com/

3.2.1.4 Certificação Digital

O método de autenticação baseado em certificação digital representa o fator de autenticação de posse. Embora, uma chave privada possa ser lida devido à quantidade de informações é quase inviável memorizar o conteúdo. Portanto, a chave privada necessita ser copiado ou armazenado caracterizando a posse.

Em seu funcionamento, normalmente, a entidade utiliza a sua chave privada para realizar uma assinatura digital de um valor gerado por um servidor. O servidor verifica a assinatura digital e utiliza as informações do certificado para a identificação do usuário.

O método de autenticação baseado em certificação digital, atualmente, é bastante relacionado ao protocolo de autenticação e sigilo SSL, o qual também é discutido na seção 3.2.3.2.

3.2.1.5 Biometria

Os métodos de autenticação baseados em biometria utilizam características físicas e de comportamento (JAIN, 2002). Estes métodos têm como grande vantagem não sofrer ameaças como a perda de um dispositivo ou divulgação de uma senha.

O investimento pode determinar o índice de segurança desejado em relação à precisão do equipamento. Também é importante a qualidade do algoritmo de funcionamento empregado no método.

Normalmente, os sistemas biométricos contam com uma base de dados que fornece os parâmetros de identificação. Para que um usuário seja identificado é realizada uma pesquisa dentro de um universo determinado pelos usuários cadastrados e, a partir disto, é eleito o registro armazenado com maior semelhança de parâmetros em relação aos coletados no

momento da autenticação. Portanto, quanto maior a precisão, menor a probabilidade de existirem usuários diferentes que estejam associadas ao mesmo registro. Na maioria das vezes, quanto maior a precisão também maior será o custo.

Independentemente dos custos, devido à falta de conscientização dos usuários os equipamentos biométricos ainda não gozam de uma aceitação total em ambientes computacionais, pois estes usuários têm receios de ataques[7] às suas partes físicas utilizadas na autenticação.

Para que uma característica possa ser adotada como um parâmetro biométrico, a mesma deve ser avaliada quanto aos seguintes critérios (JAIN, 2002):

- Universalidade – de forma que qualquer pessoa possua;
- Unicidade – de forma que somente uma pessoa tenha a determinada medida da característica;
- Estabilidade – de forma que a característica seja inalterada ao longo do tempo;
- Coletável – de forma que a característica possa ter uma quantidade relativa a uma determinada métrica.

Assim, existe uma diversidade de parâmetros biométricos que podem ser utilizados como método de autenticação, por exemplo (JAIN, 2002):

- Digital – além de ser o parâmetro mais comum, é muito utilizado nas técnicas de forense criminal. Coletado geralmente por meio de impressão com tinta ou escaneada diretamente. A leitura é realizada por meio da identificação de distâncias entre pontos determinados por arestas na digital;
- Íris – a identificação é realizada por meio de uma foto digital da íris para que sejam analisadas as texturas presentes na região que circunda a pupila;

[7] "Malaysia car thieves steal finger", 31 mar 2005, disponível em http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm.

- Retina – o reconhecimento é realizado pelo mapeamento vascular da retina. Por meio da observação vascular também podem ser identificadas informações médicas (como por exemplo, hipertensão) o que deve representar baixa aceitação por parte dos usuários;
- Face – é o parâmetro coletado que mais se assemelha às percepções humanas, visto que no cotidiano identificamos uma pessoa principalmente pela face. O reconhecimento geralmente é realizado pela disposição geométrica dos olhos, nariz, boca, etc;
- Voz – é uma característica de comportamento podendo ser alterada devido às condições emocionais e de saúde do usuário. Além disso, apresenta baixa permanência, visto que ao longo do tempo sofre variações. O reconhecimento é realizado a partir da pronuncia de uma frase predeterminada;
- Assinatura / Caligrafia – também é uma característica de comportamento. Pode ser reconhecida de duas formas: estática e dinâmica. A estática identifica as formas geométricas produzidas após a escrita. Já a dinâmica além das formas geométricas, analisa informação de aceleração, velocidade, perfil de trajetória e pressão no momento da escrita. A segunda abordagem provê mais pontos de reconhecimento, porém requer um equipamento específico;
- Comportamento de digitação – baseado no intervalo de tempo entre a digitação de teclas. Também pode ser analisado o intervalo de tempo em que as teclas permanecem pressionadas. É uma característica que também apresenta baixa permanência, devido à influência emocional e também a evolução na habilidade de digitação.

Segundo Jain (2002) a difusão de sistemas biométricos está relacionada às seguintes premissas:

- O custo dos sensores continuarem a baixar;
- A tecnologia envolvida se tornar mais madura;
- Os usuários adquirirem o conhecimento sobre a força e os limites da tecnologia biométrica.

3.2.2 Dispositivos de autenticação

Os dispositivos de autenticação fornecem o meio para a execução método de autenticação. Como se pode perceber, alguns dispositivos são intrínsecos aos métodos específicos de autenticação, como exemplo, método de autenticação por biometria e o dispositivo sensor.

3.2.2.1 Sensores Biométricos

Os sensores biométricos são os dispositivos responsáveis por realizar a medida dos parâmetros biométricos do usuário.

São necessários dispositivos específicos para cada tipo de parâmetro. Conforme previamente listado na seção 3.2.1.5 (Digital, Íris, Face, Voz, etc).

3.2.2.2 Cartões

De forma geral, a Figura 8 ilustra as diversas tecnologias de cartões que combina conceitos dos trabalhos de Ratha e Bolle (2002) e Rank e Effing (2003).

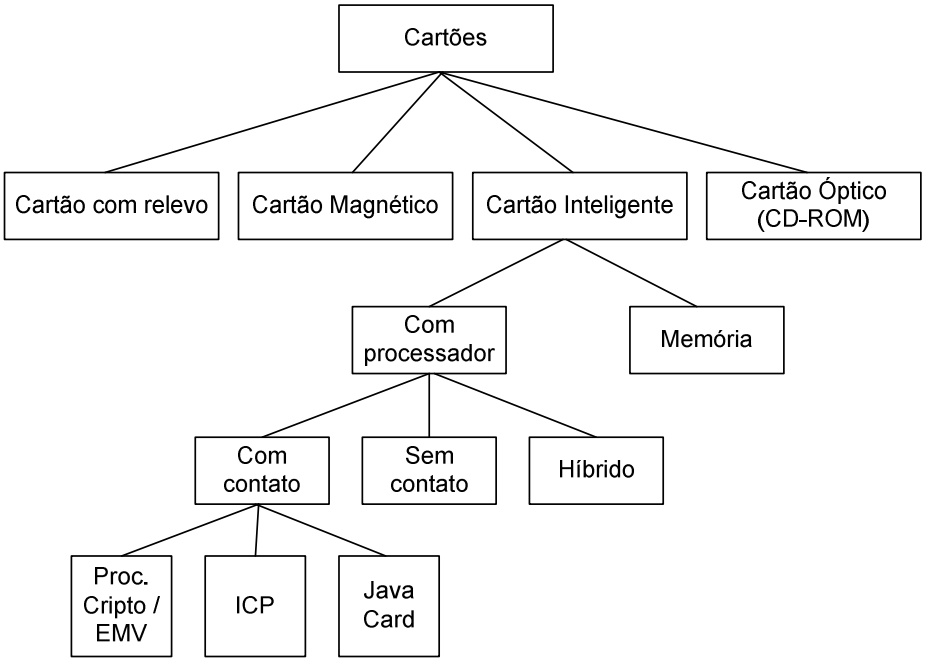

Figura 8 - Taxonomia de cartões

Os cartões em relevo são representados por cartões de crédito, os quais têm a finalidade de serem pessoais e intransferíveis, e armazenar a informação do número de série do cartão. Devido ao fato de não necessitarem de energia elétrica e de acesso a uma linha telefônica, são utilizados em escala mundial. Porém, a simplicidade do processo apresenta uma série de vulnerabilidades, como por exemplo, a técnica de reprodução não autorizada por meio da reutilização de papéis carbono. Conforme as técnicas de fraude foram sendo mais conhecidas a tecnologia foi se tornando obsoleta.

Os cartões magnéticos são mais uma etapa na evolução da tecnologia de cartões. Ao contrário da tecnologia de cartões em relevo, os cartões magnéticos armazenam uma quantidade maior de dados e necessitam de dispositivos específicos de leitura que utilizam alimentação elétrica e acesso telefônico. Desta forma, são agregados recursos de dispositivos de leitura conectados por rede a uma central de processamento que proporcionam maior controle sobre o processo de autenticação em relação aos cartões com relevo.

Um cartão magnético possui três “fitas” de dados, duas são somente de leitura e uma permite a leitura e escrita de dados. Assim, uma das principais desvantagens deste tipo de cartão é a possibilidade de alteração no seu conteúdo. O cartão magnético também sofre de ataques de clonagem.

Atualmente, o cartão inteligente vem se mostrando como o dispositivo mais robusto em relação aos tipos de cartões existentes, a próxima subseção apresenta mais detalhes sobre este tipo de cartão. Em seguida, uma outra prospecção também é descrita, o cartão óptico.

3.2.2.2.1 Cartão inteligente

Fundamentalmente, um cartão inteligente é um cartão plástico no formato semelhante a um cartão de crédito, que possui um circuito integrado acoplado a uma memória e, na maioria dos casos, a um processador.

Em relação à diversidade existente de tipos de cartões inteligentes podem ser enumeradas as seguintes modalidades (FINKENZELLER, 2003):

- Memória – fundamentalmente o cartão é constituído por um componente de memória EEPROM (*electrically erasable programmable read-only memory*). Existem em grande escala devido ao baixo custo;
- Processamento criptográfico – apresenta controle de acesso sobre seu conteúdo, muitas vezes por meio da inserção de um PIN que habilita a leitura de dados;
- EMV (Europay, MasterCard, Visa) – criado para atender as necessidades de controle de entidades do meio financeiro, como o próprio nome já indica. O EMV é capaz de realizar operações de criptografia simétrica sobre dados e também armazena um número limitado de chaves, que são utilizadas para autorizar transações bancarias;

- Cartão ICP – permite a execução de operações como a criptografia simétrica e assimétrica utilizando certificação digital. Tem como pontos negativos a baixa capacidade de armazenamento, provendo memória apenas para manter alguns certificados, e também a pouca versatilidade, visto que as operações fornecidas pelo cartão são fixas e implementadas no *hardware*;
- Java Card – abrange o conceito de cartões multi-aplicação, permitindo que programas/aplicações (*applets*) específicos codificados na linguagem Java possam ser carregados e executados no contexto da *Java Virtual Machine* (JVM) do cartão. Cartões multi-aplicação permitem diluir os custos de investimentos em cartões e em dispositivos específicos de leitura.

Outra modalidade emergente é a de cartões inteligentes sem contato (*contactless*) e híbrido (com e sem contato). Entretanto, esta modalidade não é muito relacionada à segurança, geralmente, é aplicada para a identificação de produtos, animais, e etc.

Comumente para todas as modalidades, uma desvantagem está relacionada à necessidade de difusão e o custo dos dispositivos de leitura, ou seja, a infra-estrutura de operação dos cartões pode não ser encontrada facilmente.

3.2.2.2.2 Cartão Óptico

Inicialmente, um cartão óptico é um cartão que possui uma mídia reflexiva que fornece o armazenamento de dados, ou seja, um CD-ROM. Porém, a padronização ISO/IEC 11693:2000 (RANKL; EFFING, 2002) normatiza como um chip inteligente de interface sem contato pode ser incorporado a um CD-ROM.

Como exemplo de aplicação, o conteúdo armazenado na parte reflexiva seria cifrado com uma chave armazenada no chip inteligente. Desta forma, seriam unidas as duas melhores

características de cada tecnologia, a capacidade de armazenamento do CD-ROM e a proteção de chaves do chip inteligente.

O cartão óptico embora possa ser facilmente lido em um leitor de CD-ROM, sofre a mesma desvantagem de um cartão inteligente, a qual descreve a necessidade de um dispositivo leitor específico de alto custo.

Para este documento a definição do cartão óptico que será utilizada refere-se apenas à mídia de armazenamento somente de leitura.

3.2.2.3 *Tokens*

Normalmente, os *tokens* de autenticação são dispositivos eletrônicos que essencialmente possuem uma memória interna. Em geral, os *tokens* têm o formato e o tamanho de chaveiros. A Figura 9 ilustra os principais tipos de *tokens* divididos por suas características.

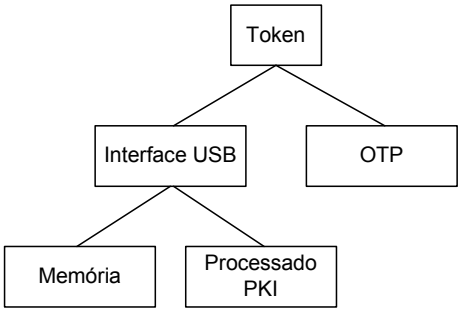

Figura 9 - Taxonomia de tokens

3.2.2.3.1 Memória

Os *tokens* de memória, também chamados de *Pen Drives* ou *Usb Mass Storage*, apresentam três vantagens principais:

- São mídias removíveis;
- Possuem capacidade de armazenamento superior, se comparados aos disquetes, cartões inteligentes e alguns tipos de CD-ROM;
- Utilizam a interface de comunicação padrão com o computador que é a *Universal Serial Bus* (USB).

Em desvantagem, os *tokens* de memória não possuem capacidade de processamento e controle de acesso nativo.

3.2.2.3.2 Processamento criptográfico

Os *tokens* com processamento criptográfico têm características semelhantes a um cartão inteligente ICP: operações com certificados digitais; criptografia simétrica e assimétrica; utilização de PIN para controle de acesso; memória de armazenamento reduzida.

Entretanto, como evolução ao cartão inteligente, o *token* criptográfico agrega a interface de comunicação do tipo USB eximindo a necessidade de um dispositivo de leitura específico.

3.2.2.3.3 OTP – One Time Password

Os *tokens* OTP representam os dispositivos utilizados no método de autenticação OTP discutido na seção 3.2.1.3.

3.2.3 Protocolos de autenticação

Um dos principais obstáculos enfrentados por sistemas de autenticação baseados em senha é gerado pelos usuários ao escolherem uma senha fraca. Os usuários tendem a escolher senhas simples e com tamanho pequeno, devido principalmente a dificuldade de se memorizar senhas extensas e complexas consideradas fortes.

Em ambientes de rede, protocolos de autenticação além de sofrerem com o obstáculo da escolha da senha, devem ser imunes a ataques tanto de escuta e captura de pacotes como também de repetição (*replay*). Um ataque de repetição é realizado a partir do armazenamento de mensagens que foram trocadas em comunicações anteriores para depois enviá-las novamente com o intuito de forjar uma autenticação válida (HALEVI; KRAWCZYK, 1999).

Os protocolos de autenticação baseados em senha atingem maiores níveis de segurança caso considerem os seguintes requisitos (HALEVI; KRAWCZYK, 1999):

- Autenticação Mútua – ambos os cliente e servidor devem ser autenticados. Este requisito adquire uma importância vital quando é preciso autenticar usuários remotos que estão conectados em redes completamente não confiáveis, como a Internet. Além disso, a autenticação mútua ajuda a evitar ataques de MITM (*man-in-the-middle*) e personificação dos servidores (*spoofing*);

- Troca de Chaves Autenticada – ao final da execução do protocolo, cliente e servidor devem compartilhar uma chave de sessão secreta. Tal chave será então usada para cifrar os dados da sessão corrente;
- Proteção da Identidade do Usuário – o protocolo de autenticação não deve divulgar a uma entidade não autorizada informações sobre a identidade do usuário.

Como exemplos de evolução, em relação aos protocolos descritos de autenticação fraca, podem ser listados: EKE (BELLOVIN; MERITT, 1992), SPEKE (JABLON, 1996), SRP (WU, 1998) e protocolos de deslocamento. Por causa do valor associado destes exemplos para com o Sistema SATES, os mesmos são discutidos na seção de trabalhos relacionados. Além destes protocolos, outros também importantes são:

- Kerberos;
- SSL;
- SSH (*Secure Shell*).

3.2.3.1 Kerberos

O protocolo Kerberos (KOHL; NEUMAN, 1993) é baseado na concepção de credenciais de acesso (*tickets*) fornecidas aos usuários por uma entidade terceira confiável.

O protocolo tem seu funcionamento conforme mostra a Figura 10. A primeira etapa é executada por meio de uma pré-autenticação realizada em um servidor chamado *Key Distribution Center* (KDC), o qual provê uma credencial de identificação chamada *Ticket Granting Ticket* (TGT). Na segunda etapa, o TGT é utilizado para obtenção de credenciais de acesso a serviços específicos junto a um outro servidor chamado *Ticket Granting Service* (TGS). Por fim, a etapa três caracteriza a utilização de um serviço localizado em um servidor de aplicação.

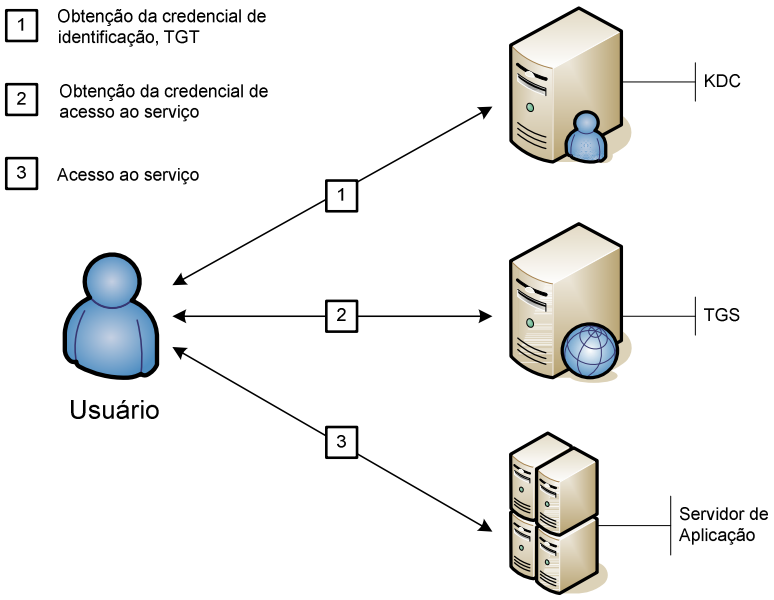

Figura 10 - Protocolo Kerberos

O protocolo Kerberos para a autenticação do usuário no KDC utiliza segredos compartilhados, os quais são representados pelas senhas dos usuários. As senhas são utilizadas como chaves simétricas para a cifragem do instante da autenticação. O servidor, sendo capaz de decifrar o instante, efetiva a autenticação, visto que este tem o conhecimento da senha do usuário.

3.2.3.2 SSL

O protocolo SSL (FRIER; KARLTON; KOCHER, 1996) tem o objetivo de fornecer um canal para o tráfego de informações de forma sigilosa. Além disso, também pode ser provida a autenticação de entidades por meio da utilização de certificados digitais ou Diffie-Hellman. A autenticação pode ser realizada de duas formas:

- Autenticação Simples (*Server-Side Authentication*) – somente o servidor possui certificado digital e é autenticado por meio do certificado digital.
- Autenticação mútua ou dupla (*Client-Side* ou *Both Sides Authentication*) – ambas as entidades possuem certificado digital.

O protocolo SSL é amplamente utilizado nos sistemas atuais, podendo ser encontrado em aplicações de Internet, como *sites* de comércio eletrônico, de bancos, etc. Estes exemplos demonstram o protocolo SSL fornecendo uma camada de segurança para o protocolo de mensagens *Hypertext Protocol* (HTTP) (FIELDING, 1999), o que resulta no protocolo *Secure Hypertext Protocol* (HTTPS) (FRIER; KARLTON; KOCHER, 1996).

O protocolo *Transport Layer Secure* (TLS) (DIERKS; ALLEN, 1999) é baseado no protocolo SSL e também pode ser associado a outros protocolos de serviço, como por exemplo, *Securing FTP with TLS* (FORD-HUTCHINSON, 2005) e o *Secure Simple Mail Transfer Protocol* (SMTPS) (HOFFMAN, 1999).

3.2.3.3 SSH

O *Secure Shell* (YLONEN; LONVICK, 2006) é um protocolo para acesso remoto seguro, que é baseado na utilização de chaves assimétricas do algoritmo RSA. O uso das chaves RSA durante o estabelecimento do canal de comunicação provê a troca de uma chave simétrica que é utilizada durante a sessão.

Na maioria das vezes somente o servidor possui o par de chaves, entretanto podem ser configuradas opções que contemplam autenticação mútua e certificação digital.

O usuário por meio do aplicativo cliente SSH acessa uma console de terminal remoto. O aplicativo cliente executa comandos, fornece transferências de arquivos e armazena as informações dos servidores acessados como o endereço de rede relacionado à chave pública

correspondente. Desta forma, o aplicativo pode auxiliar na autenticação do servidor alvo da conexão com base nas informações de sessões anteriores. Entretanto, os arquivos de chaves públicas armazenados podem propiciar vulnerabilidades, visto que, por meio de alterações nestes arquivos o usuário autorizado pode ser enganado de forma a se conectar a um servidor malicioso.

3.3 Considerações sobre autenticação digital

Os mecanismos de autenticação fraca apresentados possibilitam o entendimento de algumas fragilidades existentes nos sistemas antigos de autenticação que podem ainda persistir nos sistemas mais recentes.

Em seguida, a definição da arquitetura geral foi importante para situar os exemplos atuais em relação aos componentes destacados e também servir de referência para a seqüência deste trabalho.

O capítulo 4 vai abordar trabalhos que apresentam discussões, base teórica, vantagens e desvantagens em relação à proposta deste trabalho.

4 Trabalhos Relacionados

Este capítulo é destinado a mostrar trabalhos relacionados que influenciaram na concepção do SATES. Também apresenta algumas lacunas deixadas por estes trabalhos relacionados e aponta alguns elementos do SATES.

A linha de pesquisa abordada por este trabalho tangencia a iniciativa da especificação proposta pelo grupo de pesquisa IEEE 1363.2, a qual apresenta um trabalho em andamento referente ao uso de ICP habilitada por senha, provendo: especificações, embasamento teórico, e discussões sobre segurança e implementações. Deste modo, a leitura de trabalhos com citações associadas ao grupo foi valiosa para o desenvolvimento da pesquisa.

Os primeiros protocolos, *Encryption Key Exchange* (EKE) (BELLOVIN; MERITT, 1992) e *Simple Password Exponential Key Exchange* (SPEKE) (JABLON, 1996), motivaram uma série de novos trabalhos com propostas de melhorias em suas concepções apresentadas.

4.1 Protocolos baseados em senha

Os protocolos baseados em senha, em relação aos conceitos abordados na seção 3.2 sobre a arquitetura geral de autenticação, podem ser classificados como protocolos de autenticação. Além disso, todos os protocolos baseados em senha estão relacionados ao fator de autenticação de conhecimento.

A seguir, o detalhamento dos protocolos que influenciaram a especificação do sistema proposto por este trabalho, o Sistema SATES.

4.1.1 EKE e SPEKE

Os protocolos EKE (BELLOVIN; MERITT, 1992) e SPEKE (JABLON, 1996) se baseiam em duas entidades, um servidor e um cliente, as quais compartilham um mesmo segredo (senha) "fraco", e após a autenticação, tal segredo "fraco" possibilita a convergência em um segredo de maior proporção e robustez para que durante a sessão seja impraticável o ataque ao segredo "fraco" (PERLMAN; KAUFMAN, 1999). Ambos os protocolos são baseados no algoritmo de troca de chaves (DIFFIE; HELLMAN, 1976).

O protocolo EKE utiliza quatro mensagens, como ilustra a Figura 11, e os envolvidos na execução do protocolo conhecem um segredo fraco chamado de **W** [W = Hash(Senha)]. O cliente inicia a execução do protocolo ao escolher um valor aleatório e privado chamado de **A**. Em seguida, a mensagem **1** [M1 = (g^A mod p) cifrado com W] é calculada utilizando os parâmetros públicos do servidor **P** e **G**. O servidor gera dois valores, também aleatórios e privados, **B** e **C1,** e a partir de M1 calcula a chave de sessão **K** [K = g^{AB} mod p] . O servidor envia a mensagem **2** [M2 = (g^B mod p) cifrado com W + (C1) cifrado com K]. O cliente extrai o valor cifrado [g^B mod p] com a chave **W** e também chega ao valor de **K**. O cliente também gera um novo valor privado **C2** para a resposta e utiliza a chave **K** para recuperar o valor **C1**, o qual é concatenado com **C2**, e depois é enviado ao servidor sob a proteção da chave **K,** constituindo a mensagem **3** [M3 = (C1, C2) cifrado com K]. Por fim, o servidor envia a mensagem **4** contendo **C2**. Portanto, a mensagem **3** prova que o cliente conseguiu obter **C1** e a mensagem **4** [M4 = (C2) cifrado com K] prova que o servidor conseguiu obter **C2,** o que se conclui que ambos têm o conhecimento da chave **K**.

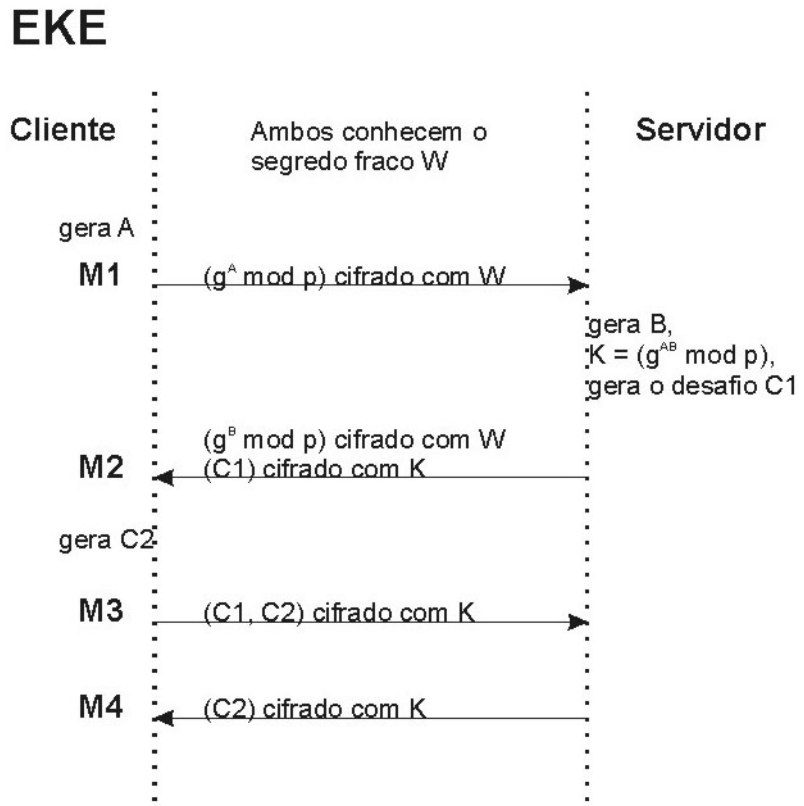

Figura 11 - Protocolo EKE

O protocolo SPEKE apresenta várias semelhanças se comparado com o protocolo EKE. A diferença pode ser vista na Figura 12 e representa o uso do segredo fraco **W** como gerador para os expoentes **A** e **B**, gerados aleatoriamente para a sessão. O termo gerador referencia-se ao valor utilizado como base para os expoentes.

A execução do protocolo SPEKE tem o seu início por meio da geração pelo cliente de um valor aleatório e privado chamado A, o qual é utilizado como expoente do valor **W** na composição da mensagem **1** [M1 = W^A mod p]. A mensagem **1** é enviada ao servidor, que ao recebê-la calcula a chave de sessão **K** e gera um também um valor aleatório e privado **B**. Para o envio da mensagem **2** [M2 = W^A mod p + (C1) cifrado com K], o servidor gera um valor aleatório e privado **C1**. O cliente tendo recebido a mensagem **2** também gera um valor aleatório e privado, chamado de **C2**. O cliente envia a mensagem **3** [M3 = (C1,C2) cifrado com K]. O servidor ao decifrar a mensagem **3** tem a prova de que o cliente conseguiu extrair o

valor **C1**, ou seja, o cliente prova o conhecimento de **K**. Por fim, o servidor também envia o valor extraído **C2** para que o cliente receba a prova de conhecimento de **K** por parte do servidor.

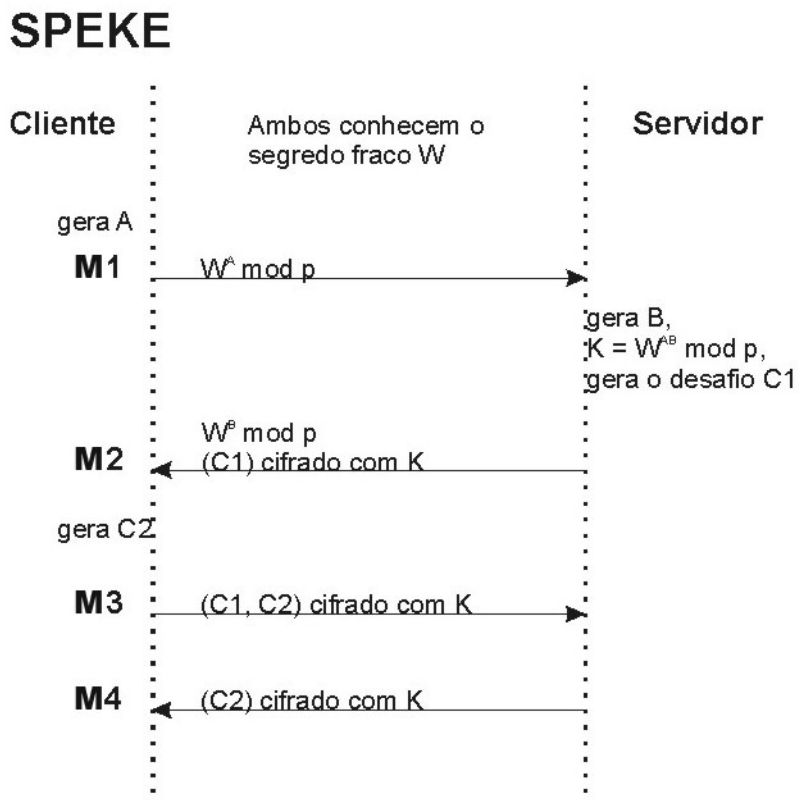

Figura 12 - Protocolo SPEKE

Ambos os protocolos apresentaram uma alternativa para o problema de autenticação encontrado no algoritmo de troca de chaves proposto por Diffie e Hellman (1976), o qual estabelece uma chave de sessão segura entre duas entidades, porém, não provê a autenticação das entidades parceiras. Deste modo, ambos os protocolos realizam a autenticação baseados no fator de conhecimento representado pelos valores W (segredo fraco) e K (segredo forte). Porém, estes protocolos têm como desvantagem o problema do repúdio, visto que ambos

realizam a autenticação baseada em um segredo compartilhado, a qual não fornece um nível de autenticação adequado, como discutido na seção 2.1.

4.1.2 SRP – *Secure Remote Protocol*

O protocolo SRP, assim como os protocolos EKE e SPEKE, tem o objetivo de autenticar a entidade parceira de comunicação e estabelecer uma chave sessão. Porém, este protocolo apresenta uma grande vantagem que é a utilização de um código verificador (v), ao invés de utilizar um segredo compartilhado.

O código verificador (v) é gerado por meio de uma função que pode ser visualizada na Figura 13.

g = gerador

n = número primo

Id = identificador do usuário

x = Hash(salt, Id, senha)

$v = g^x \bmod n$

Figura 13 - Função de geração do código verificador SRP

O protocolo SRP conta com a troca de seis mensagens, como indica a Figura 14, no processo de autenticação, o qual tem início com o cliente enviando seu identificador (Id) ao servidor o que define a mensagem **1**. O servidor baseando-se no valor Id busca os valores específicos do cliente, o *salt* e o verificador. A mensagem **2** é dada pela informação do *salt* e é enviada ao cliente.

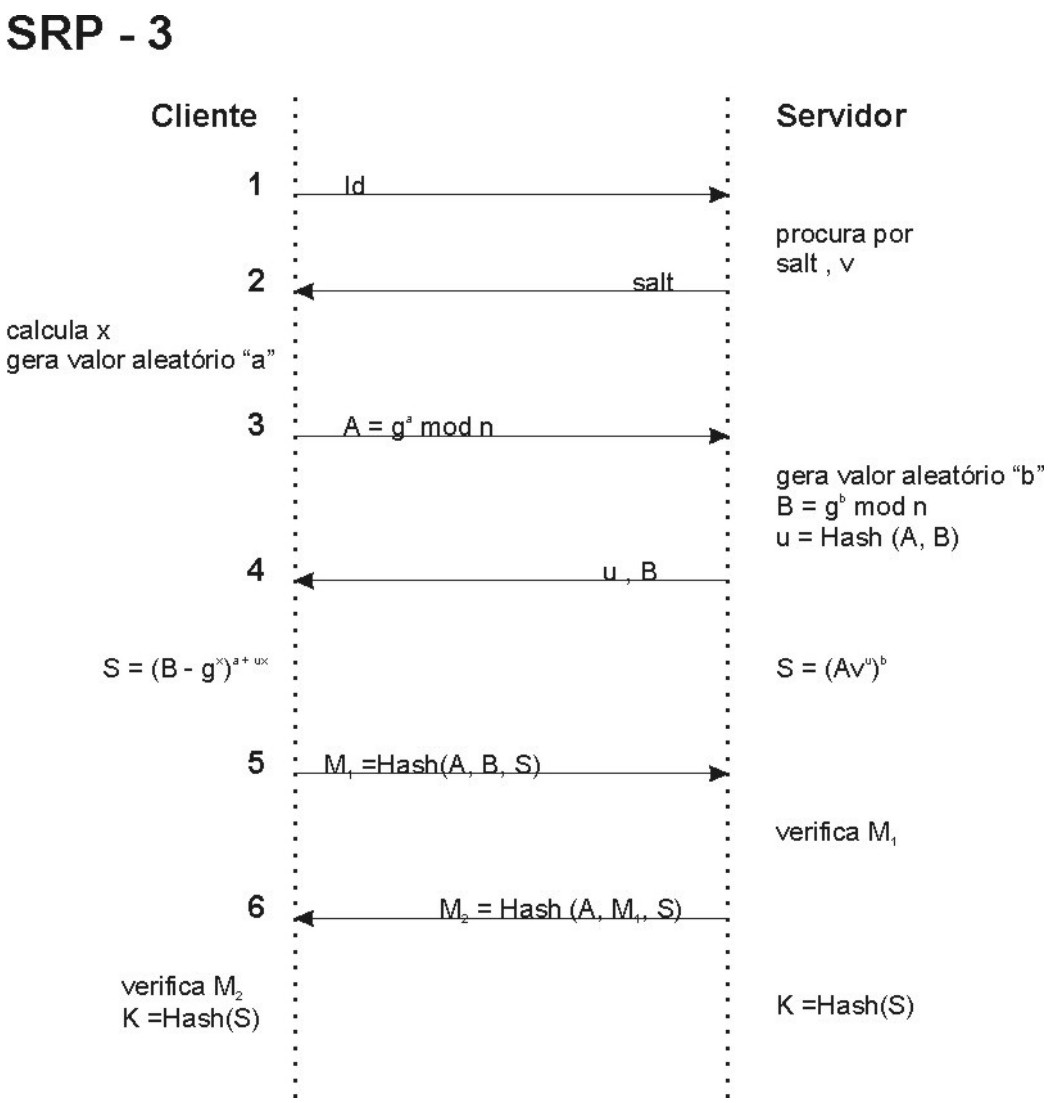

Figura 14 - Protocolo SRP - 3

No trabalho de Wu (1998) o SRP teve sua primeira versão, chamada de SRP-3. A Figura 14 mostra a troca de mensagens deste protocolo. Em Wu (2002) foi apresentada uma nova versão, SRP-6, a qual foi desenvolvida para a correção de dois problemas: o primeiro referente a um ataque de personificação do servidor que proporcionava o teste de duas senhas por cada tentativa de autenticação; e o segundo referente ao valor "u" enviado ao cliente sem que antes seja verificada a sua identidade, o que poderia possibilitar uma vulnerabilidade.

4.1.3 Protocolos de Deslocamento (Roaming)

O objetivo de um protocolo de deslocamento ou “roaming” é recuperar, sob demanda, as chaves criptográficas de um dado usuário e depois torná-las disponíveis para operação. Como ilustra a Figura 15, tais chaves podem ser obtidas de servidores remotos ou qualquer outro tipo de repositório considerado seguro mediante um processo de autenticação prévio. Dentro de um domínio privado, este tipo de sistema é particularmente importante para usuários de razoável mobilidade, para os quais o fato de armazenar chaves em discos rígidos de máquinas é um grande inconveniente. A mobilidade pode ainda ser estendida a domínios públicos, considerando usuários que utilizam terminais em quiosques públicos ou telecentros, e até usuários que utilizem dispositivos de bolso com suporte a rede sem fio (*wireless*), como por exemplo, o “*Pocket PC*” (JABLON, 2001).

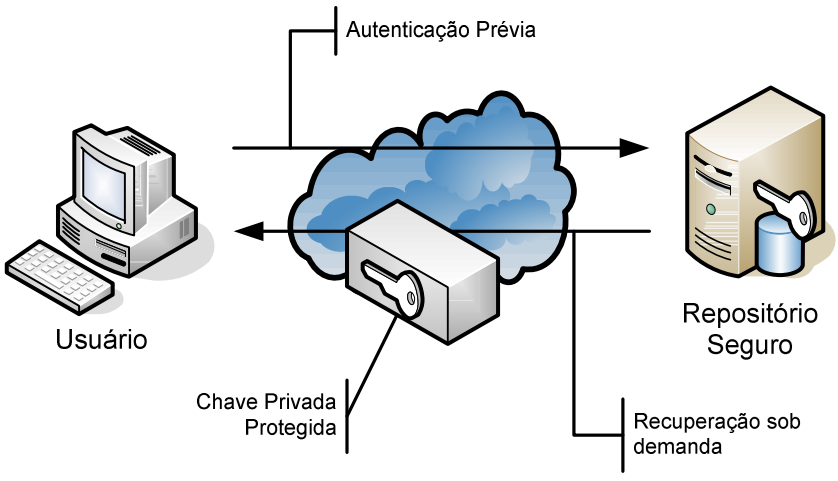

Figura 15 - Funcionamento do protocolo de Deslocamento

Ainda com referência à Figura 15, um exemplo bastante típico considera que o acesso de cada usuário à sua respectiva chave privada ocorra de maneira segura, uma vez que esta é protegida por criptografia simétrica antes de ser armazenada. Neste contexto, trabalhos como

Perlman e Kaufman (1999) e Ford e Kaliski (2000), entre outros, combinaram protocolos de deslocamento com protocolos baseados em senha para minimizar as oportunidades de um inimigo quanto ao roubo, divulgação ou quebra da senha e chaves. Desta forma, utilizando uma senha memorizada, um usuário pode obter sua chave privada para uso temporário em qualquer máquina cliente aceitável, lançando um ambiente seguro e sendo capaz de realizar uma ou mais transações, para então mais tarde apagar localmente esta chave do computador em uso.

Portanto, com o uso dos protocolos de deslocamento consegue-se prover uma proteção forte baseada em *software* quando o armazenamento das chaves em uma estação pelo usuário não é conveniente (JABLON, 2001).

Entretanto, os protocolos de deslocamento geram alguns itens de atenção que devem ser considerados, tais como:

- O repositório de chaves representa um ponto central de ataque, caso um indivíduo não autorizado ganhe acesso ao repositório seguro, os valores armazenados podem ser diretamente atacados (FORD; KALISKI, 2000);
- Como o repositório tende a suportar uma grande quantidade de chaves de diversos usuários (comunidade), um ataque pode resultar em prejuízos com largas proporções (FORD; KALISKI, 2000);
- Uma máquina pobremente gerenciada quanto à segurança pode criar oportunidades para a ocorrência de incidentes (JABLON, 2001);

Os trabalhos de Ford e Kaliski (2000), Jablon (2001) e Brainard et al. (2003) pretendem contornar estes itens de atenção usando múltiplos servidores colocados em diferentes localizações.

Devido à relevância com o tema deste trabalho, a seguir são descritos alguns protocolos de deslocamento.

4.1.3.1 Perlman e Kaufman (1999)

O trabalho discutido em Perlman e Kaufman (1999) apresenta variações dos protocolos EKE e SPEKE acrescentando novas contribuições, como por exemplo, a utilização dos parâmetros *salt* e *cookie*. Segundo o trabalho, *salt* é um valor individual do usuário gerado aleatoriamente enviado pelo servidor, o qual é utilizado como parâmetro adicional à senha. A autenticação é dada pelo resultado de uma função *hash* sobre os valores da senha e do *salt*, sendo tal resultado depois enviado ao servidor para que seja realizada uma comparação com um valor previamente cadastrado. O *cookie* é um número aleatório também enviado pelo servidor para que o cliente faça eco desta informação, com o propósito de detectar ataques que utilizam endereços de origem forjados, como por exemplo, ataques de negação de serviço (*Denial of Service* – DoS).

A Figura 16 ilustra o protocolo variante do SPEKE com seis mensagens proposto pelo trabalho de Perlman e Kaufman (1999). Este protocolo tem o objetivo de prover o deslocamento de uma chave privada.

Conforme a Figura 16, primeiro o cliente envia informação do nome do usuário, mensagem **1** [M1 = usuário]. O servidor com base no valor "Usuário" obtém, por meio de sua base de usuários, o valor do *salt*. O servidor junto com o *salt* envia ao cliente um valor aleatório chamado *cookie*, mensagem **2** [M2 = *salt, cookie*]. O cliente após o recebimento da mensagem **2**, utilizando o valor do *salt* e a senha calcula o valor **W** [W = Hash(Senha, *Salt*)] e gera um valor aleatório **A**. O cliente envia ao servidor a mensagem **3** [M3 = *cookie* + W^A mod p] utilizando os valores **A**, **W** e o ***cookie***. O servidor também gera um valor aleatório **B** e calcula a chave de sessão **K** [K = W^{AB} mod p]. O servidor envia ao cliente a mensagem **4** [M4 = W^B mod p]. O cliente também calcula a chave de sessão **K** e executa a função *hash* sobre o valor de **K** para que seja enviada para o servidor a mensagem **5** [M5 = Hash(K)]. Por fim, o

servidor cifra a **chave privada** do cliente com a chave de sessão **K**. O conteúdo cifrado representa a mensagem **6** [M6 = (chave privada) cifrada com K].

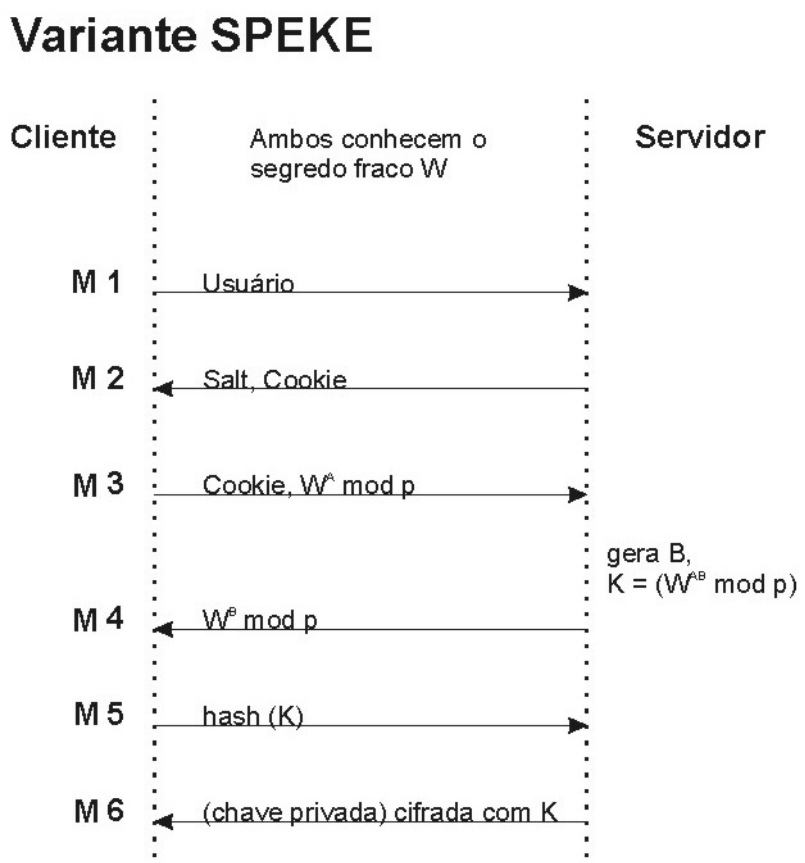

Figura 16 – Variante do SPEKE, 6 mensagens.

Além disso, o trabalho de Perlman e Kaufman (1999) ainda critica o protocolo *Netware* (LEE; ISRAEL, 1994), indicando como principal desvantagem a confiabilidade da transmissão da chave pública do servidor, uma vez que uma entidade não confiável poderia assumir a identidade do servidor, o que caracteriza um ataque de *spoofing*[8], todavia este protocolo também faz uso do *salt* destacando-se como uma de suas vantagens. A contribuição mais importante observada do protocolo *Netware* é relativa à iniciativa da criação de um protocolo de deslocamento de chave privada.

[8] Identificação falso-positiva de uma entidade não confiável.

A Figura 17 exibe de forma resumida a representação do protocolo de autenticação do *Netware*. Assim como o protocolo Variante SPEKE, a primeira mensagem [M1 = Id] é definida pela informação de um identificador de conta de usuário. Esta mensagem [M1] é enviada ao servidor. O servidor gera um valor aleatório **R** e obtém o valor *salt* correspondente ao usuário para compor a mensagem **2** [M2 = salt + R + KpubS] que também contém a sua chave pública (KpubS). O cliente calcula o valor **X** [Hash(senha, salt)] que é o resultado da função *hash* aplicada sobre os valores do *salt* e da senha. Além disso, o cliente também calcula o valor **Y** que é representado pelo resultado da função *hash* sobre os valores **X** e **R**. O cliente gera um valor aleatório **R2** e concebe a mensagem **3** [(Y, R2) cifrados com KpubS]. O servidor decifra os valores recebidos na mensagem **3**, obtém o valor **R** armazenado e compara o valor **Y**. O servidor executa a operação de ou-exclusivo (XOR) sobre os valores da chave privada do usuário e o valor **R2** e cifra o resultado da operação XOR com a chave **Y**. A mensagem **4** [M4 = (Z **XOR** R2) cifrado com Y] é composta pelo conteúdo cifrado por **Y**.

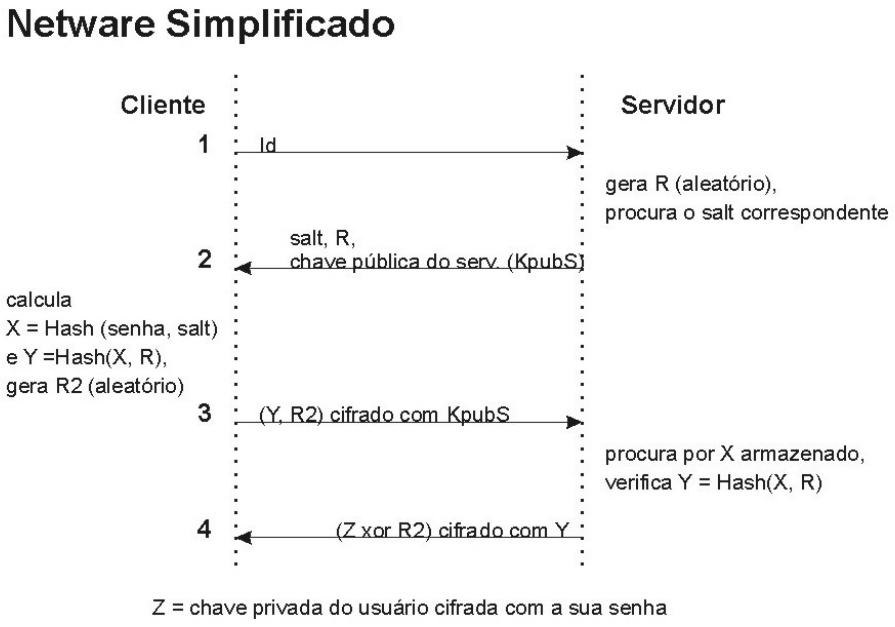

Figura 17 – Protocolo simplificado do Netware (PERLMAN; KAUFMAN, 1999).

O trabalho Perlman e Kaufman (1999) tem sua contribuição ao destacar os conceitos de:

- *Salt* que provê o aumento da entropia referente à senha;
- *Cookie* que representa um mecanismo contra ataques de *replay*.

Contudo, expõe variações aos protocolos da família EKE que utilizam segredos compartilhados, os quais têm como maior desvantagem o aspecto do repúdio em relação ao conhecimento da senha por parte do servidor.

4.1.3.2 Ford e Kaliski (2000)

Em Ford e Kaliski (2000) foi introduzido o modelo de múltiplos servidores, o qual tem a proposta de transformar um segredo fraco, após interações com "n" servidores, em um segredo forte utilizado como chave na proteção de informações sensíveis, como por exemplo, uma chave privada.

O processo de transformação do segredo fraco, uma senha, em forte, também é visto nos protocolos EKE e SPEKE. Entretanto, a idéia de múltiplos servidores atenta para um problema não previsto nos protocolos anteriores, o não repúdio em relação aos possíveis ataques realizados por administradores do sistema sobre as bases de senhas dos usuários.

O protocolo determina que, para o processo de autenticação, sejam utilizados no mínimo dois servidores, de modo que cada servidor seja gerenciado por um grupo de administradores independentes entre si. Assim, o trabalho de Ford e Kaliski (2000) tem o objetivo de descentralizar o controle sobre os "n" servidores responsáveis pela autenticação do usuário.

O gerenciamento segregado dos servidores tenta não oferecer o repúdio dos usuários, conforme comentado. Todavia, pode ser difícil e dispendiosa a prática de implantação e

manutenção de um sistema que exija “n” equipes para “n” servidores com o agravante de que cada servidor pode ser um ponto único de falha.

Em relação ao funcionamento, cada servidor (S_n) mantém um valor secreto (d_n) e um verificador (V_n) para cada usuário. A Figura 18 mostra a interação do usuário com dois servidores para a obtenção dos valores s1 e s2. Em seguida, como ilustra a Figura 19 estes valores são utilizados para recompor a chave mestra do usuário, a qual pode ser utilizada na cifragem de dados como a chave privada do usuário. Por fim, o protocolo prevê uma interação final com cada servidor para que o usuário prove a posse da chave calculada. Neste momento, o servidor utiliza o valor verificador (V) armazenado para a validação.

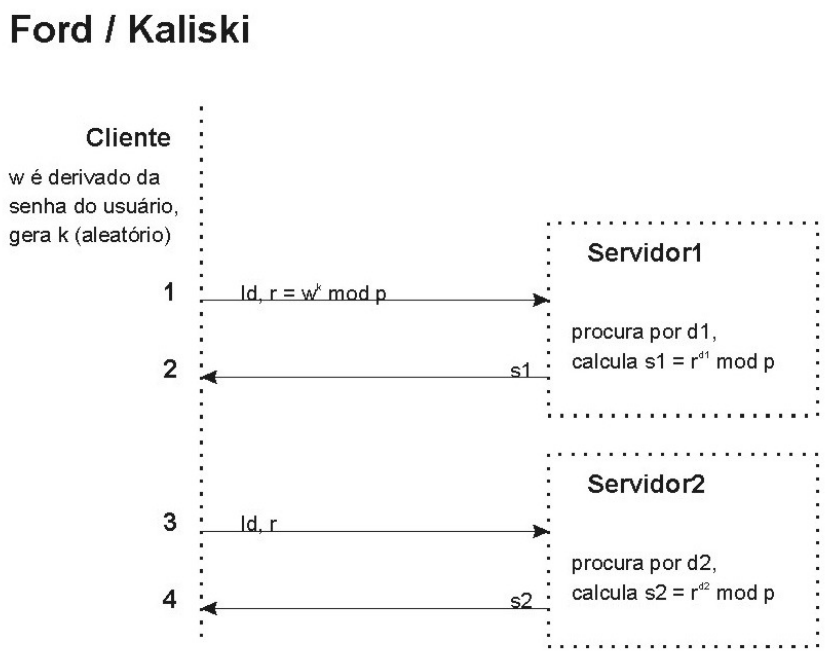

Figura 18 - Ford / Kaliski - Interação com servidores

$$R1 = s1^{1/k} \bmod p = w^{d1} \bmod p$$
$$R2 = s2^{1/k} \bmod p = w^{d2} \bmod p$$

Chave Mestra = Função(R1, R2)

Figura 19 - Ford / Kaliski - Composição da chave mestra

As vantagens destacadas por este trabalho são:

- Apresenta uma alternativa para o problema do repúdio em relação ao gerenciamento de servidores;
- A utilização do protocolo SSL como um mecanismo de pré-autenticação.

4.1.3.3 Jablon (2001)

A proposta encontrada em Jablon (2001) é uma extensão ao trabalho de Ford e Kaliski (2000). Além disso, o trabalho faz uma revisão dos conceitos e aponta algumas desvantagens quanto à utilização de uma pré-autenticação por meio do protocolo SSL e quanto ao número de operações custosas em relação ao poder computacional.

O trabalho desenvolve uma discussão em torno da utilização de expoentes menores dos que os utilizados na proposta de Ford e Kaliski (2000) com a finalidade de diminuir o custo computacional envolvido na operação. Ainda, é apresentada uma nova funcionalidade em relação aos protocolos anteriores que é a definição de um mecanismo de "perdão".

O mecanismo de "perdão" tem o objetivo de prover uma tolerância ao usuário autorizado, de modo que este possa falhar em uma tentativa de autenticação. O mecanismo é responsável por armazenar dados de tentativas inválidas, para que após seja realizada a autenticação com sucesso os dados armazenados sejam assinados digitalmente e enviados ao servidor, o qual será capaz de identificar a origem das tentativas inválidas.

Este trabalho tem sua contribuição por meio dos seguintes pontos:

- Desenvolve a proposta de protocolos de deslocamento;
- Apresenta base histórica de evolução dos protocolos de deslocamento;
- Discute questões de redução de custo computacional;
- Introduz o mecanismo de perdão;

- Comenta sobre a dependência dos sistemas de autenticação sobre a confiança nas máquinas dos usuários. De modo, que a ameaça pode se estender até aos mecanismos que utilizem cartões inteligentes.

4.1.3.4 Brainard et al. (2003)

Em Brainard et al. (2003) são realizadas críticas aos dois trabalhos de múltiplos servidores citados, destacando os custos computacionais e de comunicação devido à complexidade e às várias interações com os servidores. Além disso, propõe um protocolo em que o usuário precise apenas interagir com um servidor, o qual será responsável por interagir com outro através de um canal seguro com autenticação mútua.

O protocolo descrito conta com utilização de dois servidores, Blue e Red, os quais armazenam partes de um segredo do usuário. O segredo pode ser determinado por uma informação pessoal, número do cartão de crédito ou uma chave privada. A Figura 20 expressa a interação do cliente com os servidores.

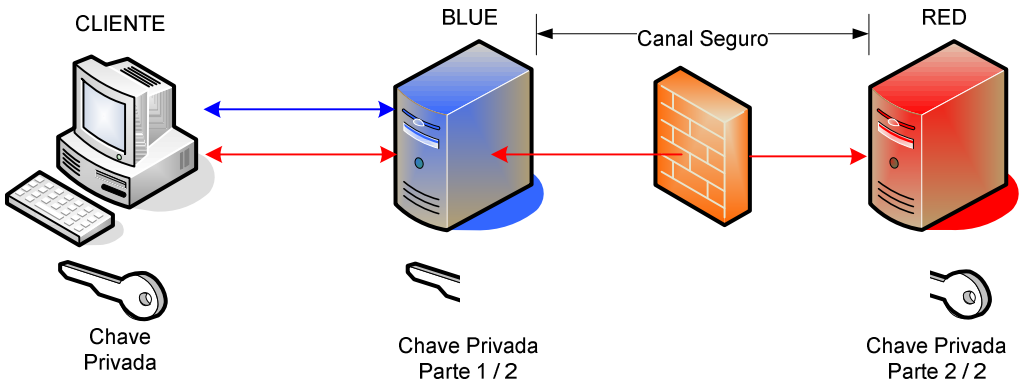

Figura 20 - Protocolo Brainard et al. simplificado

Uma questão relevante abordada pelo trabalho é em relação à necessidade de instalação de programas pelos usuários em *roaming*. A proposta feita por Brainard et al. (2003) é a instalação em tempo de execução (*on the fly*) para que os usuários possam usufruir do protocolo de deslocamento. Entretanto, o trabalho também diz que os usuários devem ter ferramentas capazes de realizar verificações sobre o programa carregado, como por exemplo, um programa navegador de páginas em relação a um *applet* Java ou ActiveX quanto às questões de verificação da assinatura digital destes programas.

Geralmente, os navegadores são programas grandes e complexos que já estão instalados nas estações. Portanto, para a instalação *on the fly* de um *applet* Java é necessária a prévia instalação de outros programas como o navegador ou a Java Virtual Machine[9].

O trabalho de Brainard et al. (2003) fornece uma contribuição para o sistema SATES quando o trabalho reforça a idéia de utilização de tecnologias já difundidas na concepção do protocolo, visto que o sistema SATES tem como alicerce o protocolo de comunicação segura SSL. Além disso, contribui também por meio da discussão sobre a confiabilidade dos programas instalados pelos usuários, pois esta questão também é relacionada ao sistema SATES.

4.2 Sandhu, Bellare e Ganesan (2002)

O trabalho de Sandhu, Bellare e Ganesan (2002) reforça a afirmação de que as senhas estão sendo largamente utilizadas atualmente nos ambientes computacionais e também comenta sobre o paradoxo em relação ao uso de senhas para habilitar o uso da chave privada.

Em sua proposta são desenvolvidas duas soluções: *Virtual Smartcard* e o *Virtual Soft Token*. De acordo com a arquitetura geral de autenticação apresentada na seção 3.2, as duas

[9] http://java.sun.com/

soluções podem ser referenciadas como dispositivos de autenticação. O *Virtual Smartcard*, como ilustra a Figura 21, emprega a senha como um componente da chave privada. Para isto, foi desenhado um modelo matemático que utiliza três chaves, duas do modelo tradicional RSA e a senha.

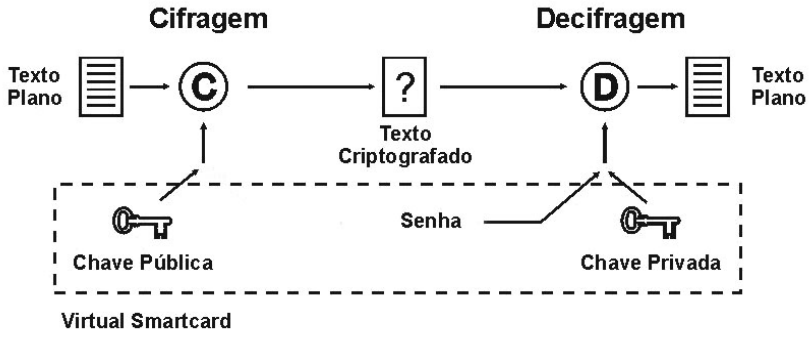

Figura 21 - Virtual Smartcard

O *Virtual Soft Token* define que a senha deverá ser utilizada na proteção da chave privada do usuário obtida a partir de um servidor remoto, Figura 22. Segundo o trabalho de Sandhu, Bellare e Ganesan (2002) esta proposta é passível de ataques de força bruta. Então, é sugerida a combinação com algum protocolo do tipo EKE ou SPEKE.

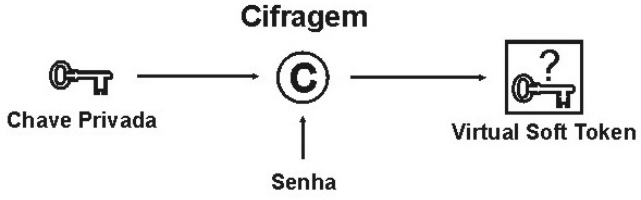

Figura 22 - Virtual Soft Token

O *Virtual Soft Token* apresenta três vantagens:

- A proposta de um modelo de ICP habilitado por senha;
- A utilização de software que implica em um sistema de baixo custo;
- A independência da mídia de armazenamento.

Entretanto, a falta da especificação de um armazenamento seguro pode criar oportunidades de comprometimento da segurança do modelo. Uma máquina pobremente gerenciada pode propiciar roubo e uma eventual divulgação de informações privativas.

O *Virtual Smartcard* é difícil de ser implantada devido a grande difusão de *toolkits*[10] criptográficos, os quais, provavelmente, não suportam o algoritmo de três chaves proposto.

O conceito do *Virtual Soft Token* faz parte da proposta do projeto SATES sob a forma do contexto de segurança do usuário.

4.3 Síntese dos trabalhos relacionados

A maioria dos trabalhos discutidos neste capítulo, com exceção do trabalho proposto por Sandhu, Bellare e Ganesan (2002), é baseada no fator conhecimento. Assim, percebe-se que o nível de segurança de protocolos baseados em senhas está evoluindo, devido aos inúmeros estudos que foram discutidos neste capítulo. Entretanto, uma senha de fácil adivinhação continuará a proporcionar deficiências e fraquezas para a segurança.

O trabalho proposto por Sandhu, Bellare e Ganesan (2002) é baseado na utilização de dois fatores de autenticação, conhecimento e posse, tanto para o *Virtual Smartcard* quanto para o *Virtual Soft Token*. O fator conhecimento é representado pela senha inserida no

[10] Conjunto de bibliotecas e APIs (*Application Program Interface*) que fornecem funções específicas, no caso, funções criptográficas.

processo de utilização da chave privada. O fator posse é referente à posse do conteúdo definido pelas duas soluções propostas (o *Virtual Smartcard* e o *Virtual Soft Token*).

Os trabalhos de Ford e Kaliski (2000) e de Brainard et al. (2003) realizam a identificação do servidor de autenticação por meio da certificação digital. Entretanto, os protocolos EKE, SPEKE, Perlman e Kaufman (1999) discutidos neste capítulo sofrem com o problema da autenticação fraca do servidor de autenticação, pois utilizam autenticação baseada em segredo compartilhado. O trabalho proposto por Jablon (2001) afirma que mesmo em uma situação que o cliente realize uma tentativa de autenticação com um servidor impostor, não é divulgada nenhuma informação sobre a senha do cliente. Em contrapartida, o sistema SATES utilizará o protocolo de autenticação mútua de entidades, o SSL, como alternativa de resolução deste problema.

Os protocolos que utilizam múltiplos servidores têm como desvantagem a possibilidade de apresentar um custo maior de implantação devido à exigência de "n" servidores, assim como também podem agregar uma carga adicional de gerenciamento e manutenção segura destas máquinas. Além disso, se não houver redundância dos servidores pode ser propiciada a negação do serviço do sistema de autenticação inteiro, caso apenas um dos servidores apresente uma falha.

Ainda em relação aos protocolos de múltiplos servidores, o problema do repúdio é em teoria resolvido devido à proposta da segregação do gerenciamento dos servidores. Entretanto, o controle de todos os servidores por um usuário malicioso pode propiciar um ataque sobre a base de usuários e senhas, um ataque semelhante a um sistema de autenticação que utilize apenas um servidor.

Os trabalhos como os de Perlman e Kaufman (1999), Ford e Kalisk (2000) e Jablon (2001), fazem propostas de protocolos complexos derivados dos protocolos SPEKE, os quais têm o objetivo de diminuir a possibilidade da divulgação de alguma informação privativa.

Contudo, o trabalho Brainard et al. (2003) propõe a utilização de um protocolo comum de comunicação por meio de um canal seguro como o SSL.

O escopo abordado pelos trabalhos discutidos neste capítulo tem o objetivo de especificar um protocolo seguro ou uma modelagem matemática aplicada por *software*, ou seja, componentes bem definidos que podem ser relacionados na arquitetura geral de autenticação (Figura 7). A Tabela 1 faz uma síntese dos trabalhos relacionados e do SATES com os quatro primeiros elementos da arquitetura de autenticação, os quais estão relacionados com a entidade cliente.

	Método de autenticação	Dispositivo de autenticação	Aplicativo Cliente	Protocolo de comunicação
SRP	X			X
Pearlman, Kaufman	X			X
Ford, Kaliski	X			X
Jablon	X			X
Brainard	X		X	X
Virtual Smartcard	X	X		
Virtual Soft Token	X	X		
Sistema SATES	X	X	X	X

Tabela 1 - Trabalhos Relacionados x Elementos da Arquitetura de autenticação

Com base na Tabela 1, pode-se perceber que somente o Sistema SATES possui todos os componentes da arquitetura de autenticação. O trabalho proposto por Brainard et al. (2003) é o que mais se aproxima de também ter todos os componentes da arquitetura, apenas com a exceção do dispositivo de autenticação. O trabalho Brainard et al. (2003) também se destaca em relação aos outros trabalhos relacionados, pois sua proposta descreve que um programa deve ser instalado sob demanda no computador do usuário, pelo fato de que os outros trabalhos apresentados não especificam detalhes sobre esta questão.

Os trabalhos relacionados até então discutidos apresentaram vantagens, desvantagens e características independentes em suas propostas. A Tabela 2 compara os trabalhos relacionados discutidos neste capítulo com o SATES de acordo os seguintes critérios:

- Autenticação mútua;
- Estabelecimento de chave simétrica;
- Fator multiplicador para redundância de máquinas;
- Quais trabalhos contemplam mais de um fator de autenticação;
- Instalação de Software para o funcionamento do protocolo;
- Escopo.

Trabalho relacionado / Critério	**SRP**	**Pearlman, Kaufman**	**Ford, Kaliski**	**Jablon**	**Brainard**	**Virtual Smartcard**	**Virtual Soft Token**	**Sistema SATES**
Autenticação Mútua	Não	Não	Não	Não	Não	-	-	Sim
Estabelecimento de chave simétrica (canal seguro)	Veri-ficador	Segredo Compar-tilhado	Veri-ficador e SSL	Veri-ficador	SSL	-	-	SSL e Veri-ficador
Fator multiplicador para a redundância de máquinas	1	1	n	N	2	-	-	1 ou 2
Contempla mais de um fator de autenticação	Não	Não	Não	Não	Não	Sim	Sim	Sim
Instalação de Software para o funcionamento do protocolo	Prévia	Prévia	Prévia	Prévia	*On the fly*	-	-	Não instala
Escopo	Protocolo				Sistema	Algoritmo / Software		Sistema

Tabela 2 - Síntese dos trabalhos discutidos

O sistema SATES, com base nos critérios apresentados na Tabela 2, é o trabalho que tem maior destaque, visto que ele provê mais elementos de segurança do que os trabalhos relacionados. Como exemplo dos elementos em destaque presentes no SATES, pode-se citar o critério de autenticação mútua, não encontrado em nenhum outro relacionado. Além disso, pode-se citar também o critério de possuir mais de um fator de autenticação, o qual não é encontrado em nenhum trabalho de protocolo de autenticação. A classificação dos trabalhos relacionados em protocolos de autenticação e componentes da arquitetura geral de autenticação é referente à Tabela 1.

No capítulo 3 foram identificados alguns dos principais componentes de autenticação presentes na arquitetura geral de autenticação. Cada componente atinge um nível de segurança. E, na maioria dos casos, a combinação destes componentes proporciona a soma dos níveis de segurança. Para exemplificar, a Tabela 3 mostra um comparativo dos componentes de autenticação destacando algumas características.

	Desafio-Resposta	EKE / SPEKE	SRP	Certificado Digital	Cartão Inteligente / Token Processamento Cript.	Cartão Óptico	Biometria Digital	Biometria Íris	Biometria Digital + Cartão Inteligente	Sistema SATES
Capacidade de armazenamento	-	-	-	-	Baixa	Alta	-	-	Baixa	Alta
Possui processador criptográfico	-	-	-	-	Sim	Não	-	-	Sim	Não
Proteção nativa da memória	-	-	-	-	Sim	Não	-	-	Sim	Não
Autenticação Mútua	Não	Não	Não	Sim	Sim	Não	Não	Não	Sim	Sim
Protocolo de comunicação	Sim	Sim	Sim	SSL	SSL	Não	Não	Não	SSL	SSL SRP
Fator Biométrico	Não	Não	Não	Não	Não	Não	Sim	Sim	Sim	Não
Estabelecimento de chave simétrica (canal seguro)	Segredo compartilhado		Verifi-cador	Cert. Digital	Cert. Digital	-	-	-	Cert. Digital	Cert. Digital e Verifi-cador
Dificuldade de personificação	Baixa	Baixa	Baixa	Média	Média	-	Alta	Alta	Alta	Média

Tabela 3 - Comparativo dos componentes de autenticação

Em relação à Tabela 3, no que diz respeito da dificuldade de personificação os níveis foram definidos de acordo os parâmetros utilizados na autenticação:

- Baixa – descoberta de uma senha;
- Média – posse de algum dispositivo e a descoberta da senha de acesso correspondente ao dispositivo;
- Alta – característica biométrica. Para a personificação deve-se ter a habilidade de enganar o sensor biométrico. Portanto, quanto maior a precisão do equipamento, maior a dificuldade de personificação por parte de uma entidade não autorizada.

Ainda com relação à Tabela 3, o cartão óptico, no critério de armazenamento, possui a maior capacidade do que os outros componentes apresentados. Analisando-se os componentes com nível médio no critério da dificuldade de personificação, pode-se perceber que todos os

componentes indicam a posse de um dispositivo e também de um certificado digital. Estas duas observações, relacionadas aos critérios de capacidade de armazenamento e da dificuldade de personificação, estão presentes no SATES.

O próximo capítulo tem o objetivo de definir a arquitetura proposta por este trabalho descrevendo os componentes, o funcionamento e o protocolo definido pelo sistema SATES. Além disso, são descritas também características de três modelos relacionados ao SATES.

5 Sistema SATES

Este capítulo descreve o objeto realizado deste projeto e aponta algumas questões que não foram envolvidas no desenvolvimento da proposta.

5.1 Visão Geral

O objetivo deste trabalho é propor um sistema de autenticação de usuários para transações eletrônicas seguras. Como previamente dito, o trabalho utilizará a junção de dois fatores autenticação, a posse e o conhecimento.

Um nível mais adequado de autenticação será obtido por meio do conjunto de vários elementos:

- Posse do cartão SATES;
- Autenticação SRP;
- O cartão possuir um certificado digital de lote, em termos de que o certificado do cartão é relacionado a um usuário do SATES;
- Canais seguros baseados em certificação digital.
- Uso da certificação digital.

Para isto, o sistema compreende a utilização de dois aplicativos, um cliente e outro servidor.

Ao final deste capítulo, com a variação dos elementos e de algumas características, são destacados três modelos para o sistema SATES.

5.2 Arquitetura Sates

Esta seção descreve os elementos que compõem a arquitetura do sistema SATES. A Figura 23 mostra a ligação entre estes elementos, que são:

- Cartão SATES;
- Aplicativo Cliente SATES;
- Protocolo SATES;
- Aplicativo Servidor SATES;
- Conteúdo remoto.

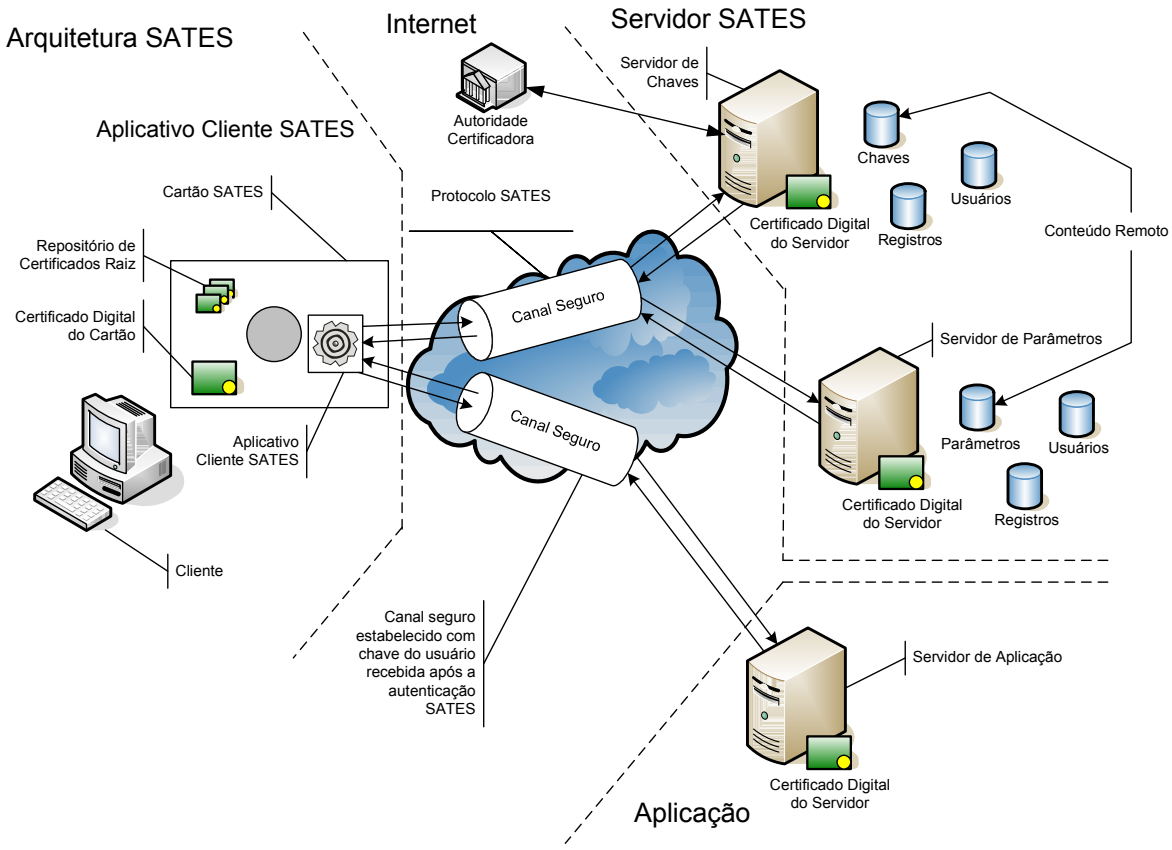

Figura 23 - Arquitetura SATES

A Figura 23 também destaca o servidor de parâmetros que pode não estar presente. Isto pode acontecer devido à escolha das opções de modelo de implementação do SATES que são descritos na seção 5.5.

5.2.1 Cartão Sates

Uma das propostas deste trabalho é utilizar um dispositivo de baixo custo, que será capaz de fornecer um segundo fator de autenticação. Neste caso, o fator agregado será a posse de um dispositivo de armazenamento de mídia CD-ROM. O dispositivo pode ser definido por duas características: física e lógica.

Para a característica física foi escolhido o formato da mídia de um *cd-card* ou *business-card*, que tem as mesmas propriedades de um CD-ROM comum, como previamente discutido na seção 3.2.2.2 sobre o cartão óptico. Desta forma, é preservada a familiaridade do usuário com cartões plásticos, como por exemplo: cartões de banco, academia, vídeo locadora, inclusive o próprio cartão inteligente.

O cartão sendo uma mídia somente de leitura, o conteúdo armazenado é protegido quanto a sua integridade, visto que não se podem alterar os dados gravados.

A característica lógica deste dispositivo é constituída por um conjunto de programas e bibliotecas dinâmicas necessários para a execução do protocolo. O conteúdo deve ter o tamanho menor que 50 megabytes, devido à capacidade de armazenamento média das mídias disponíveis atualmente sob o formato escolhido.

As características de desenvolvimento serão discutidas na próxima seção, aplicativo cliente SATES.

5.2.2 Aplicativo Cliente Sates

O aplicativo cliente SATES é armazenado no cartão SATES, o qual é constituído por uma série de funcionalidades:

- Interação com o servidor SATES – funcionalidade necessária para a execução do protocolo SATES que suporta uma interface de entrada de dados necessária para a inserção da senha de autenticação com os servidores envolvidos no processo;
- Interação com o servidor de aplicação;
- Repositório de certificados de autoridades certificadoras raiz confiáveis – durante a execução do protocolo serão necessárias validações dos parceiros de comunicação. Esta validação é possível por meio do uso de certificados digitais no estabelecimento do canal seguro. De acordo com o modelo de certificação digital, é necessária a existência de um repositório de certificados digitais confiáveis, que deve conter os certificados que representem as autoridades certificadoras para a determinação de uma cadeia de confiança;
- Identificação do cartão – a identificação do cartão ocorre por meio de um certificado digital gravado em seu conteúdo;
- Geração de chaves – segundo o modelo de certificação digital, o próprio usuário deve ser responsável pela geração da sua chave privada. Assim, o aplicativo tem a responsabilidade de auxiliar o usuário durante a etapa de geração de chaves. Além disso, o aplicativo deve possuir um algoritmo de geração de números pseudo-aleatórios para proporcionar maior segurança durante esta etapa. Após a geração é criada uma requisição de certificado para a associação do par de chaves ao usuário. A requisição deve ser enviada ao aplicativo servidor SATES para que o pedido possa ser encaminhado, resultando na emissão do certificado digital do usuário.

- Gerenciamento do contexto de segurança pessoal do usuário – tem o objetivo de proteger quaisquer informações sensíveis do usuário utilizadas pelo sistema SATES. A execução do protocolo SATES pode gerar resíduos no sistema operacional do usuário, os quais o aplicativo cliente SATES poderá limpar após o término da sessão do usuário. Como por exemplo, a utilização do sistema SATES no navegador Internet Explorer requer uma instalação da chave privada do usuário no sistema operacional, representando o resíduo. A chave privada será removida antes do encerramento do aplicativo cliente SATES;

O aplicativo cliente SATES tem a característica de prover uma interface que possibilite a um usuário operar o sistema de maneira simples e clara, de modo, que operações de um nível técnico elevado sejam feitas de forma transparente, como por exemplo, a geração de um par de chaves.

5.2.3 Protocolo Sates

O protocolo SATES está presente no aplicativo cliente e no aplicativo servidor SATES para que a interação entre eles seja feita de forma compatível. Este protocolo é composto por quatro elementos: canal seguro, um protocolo de autenticação, protocolo comunicação e formato de mensagens.

O canal seguro é provido por meio do protocolo SSL devido ao fato de que propicia autenticação mútua de entidades por certificado digital sendo utilizado o certificado gravado no “Cartão SATES”.

O protocolo de autenticação do usuário é o SRP, escolhido apenas com o intuito de prova de conceito. Entretanto, este elemento serve para restringir o deslocamento do contexto seguro do usuário a um usuário autorizado.

A comunicação é realizada por comandos do *Hyper Text Protocol* (HTTP), que possibilita o tráfego de mensagens de texto. As mensagens utilizam o formato *Extensible Markup Language* (XML). As mensagens compreendem a troca de parâmetros durante a execução do SRP, como também as mensagens necessárias para o deslocamento da chave.

O protocolo SATES, conforme mostra a Figura 24, tem seu funcionamento baseado nas seguintes interações:

1. O aplicativo cliente SATES inicialmente se conecta ao servidor e estabelece o canal SSL utilizando o certificado de lote do cartão SATES;
2. O aplicativo cliente SATES envia o nome do usuário com objetivo de realizar a autenticação SRP;
3. O servidor SATES compara a informação do nome do usuário recebido com o certificado de lote utilizado no estabelecimento do canal SSL;
4. O aplicativo cliente SATES realiza a autenticação SRP no servidor SATES;
5. O aplicativo cliente SATES envia o nome do usuário com objetivo de executar o protocolo de deslocamento;
6. O servidor SATES procura na base de dados pelas informações do estado de cadastro do usuário e do contexto de segurança armazenado do usuário;
7. O servidor SATES envia o estado de cadastro do usuário e o contexto de segurança armazenado do usuário.

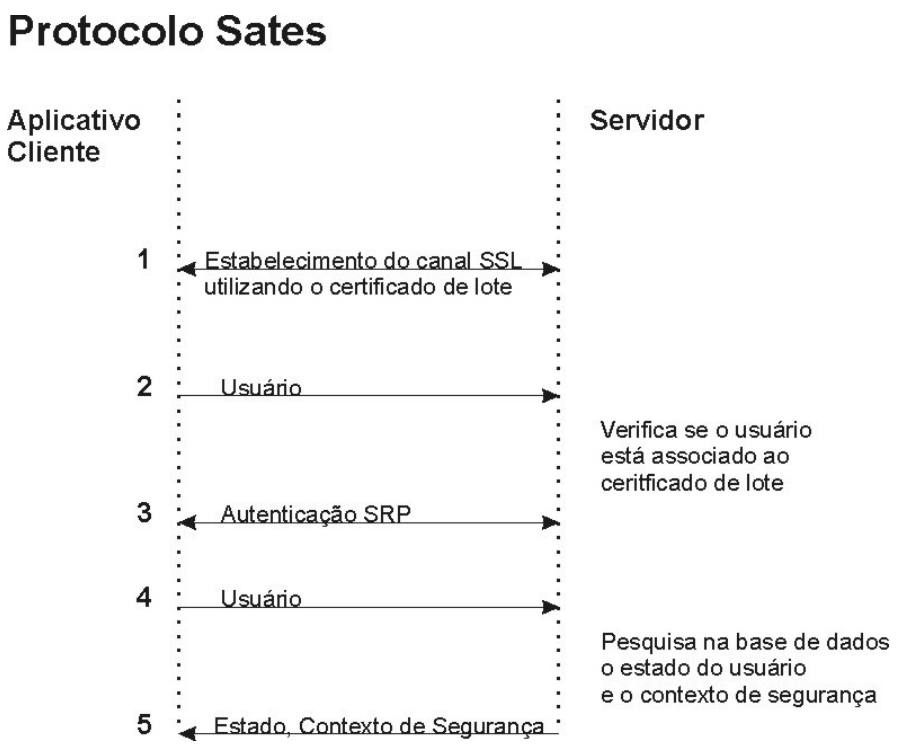

Figura 24 - Protocolo SATES

5.2.4 Conteúdo Remoto

O conteúdo remoto representa a informação do usuário armazenada nos servidores seguros, a qual também pode ser recuperada sob demanda de acordo com o protocolo de deslocamento.

Podem ser utilizados dois tipos de conteúdo armazenados remotamente no servidor de parâmetros e no servidor de chaves, respectivamente, o contexto de segurança e parâmetros de derivação da senha de proteção do contexto de segurança.

Os parâmetros de derivação da senha podem ser armazenados no conteúdo do cartão SATES. Assim, esta opção também é utilizada de forma a propor um novo modelo.

5.2.4.1 Contexto de Segurança

O contexto de segurança do usuário geralmente é definido pelo par de chaves assimétricos protegido por uma criptografia simétrica. Pode também ser composto por quaisquer outras informações privativas que deverão estar disponíveis ao usuário via acesso remoto.

5.2.4.2 Parâmetros de derivação

Os parâmetros de derivação da senha de proteção do contexto de segurança podem ser utilizados para aumentar o nível de segurança em relação a um sistema de autenticação qualquer que utilize como chave apenas a derivação direta da senha secreta do usuário. Sendo feita a opção pelo armazenamento remoto, assim como para o armazenamento do contexto de segurança, é utilizado um servidor de repositório responsável pelo armazenamento destes parâmetros.

Os parâmetros de derivação remotos são armazenados em memória volátil, ou seja, para cada tentativa de autenticação é necessário obtê-los novamente. Desta forma, além de aumentar a dificuldade de recomposição da chave de proteção, também se podem gerar registros de auditoria sobre os momentos de utilização da chave. Por outro lado, será necessário que o usuário tenha conexão com o servidor de senhas sempre que for utilizar a sua chave.

Como prova de conceito, os parâmetros (remotos ou não) são representados por dois valores, uma palavra e um número de iterações. A palavra durante a derivação é concatenada à senha. O número de iterações indica a quantidade de execuções da operação de *hash* sobre

as informações de senha e da palavra. Isto representa a função chamada *Password Key Derivation Function 1* (PKDF1) (KALISKI, 2000), a Figura 25 representa detalhes da função.

```
Valor parcial 0 = Hash (Senha + Palavra)
Valor parcial 1 = Hash (Valor parcial 0)
...
Chave derivada = Hash (Valor parcial (Iterações – 1))
```

Figura 25 - Função PKDF1

Concluindo, a utilização de parâmetros de derivação remotos apresenta duas questões: a disponibilidade do repositório e o custo de execução do processo de derivação. A segunda questão é relacionada também ao modelo de parâmetros de derivação armazenados no cartão SATES.

5.2.5 Servidor Sates

O servidor SATES é responsável por prover o serviço de autenticação para o usuário e tornar disponível o conteúdo armazenado dos usuários. O servidor SATES é composto por dois serviços:

- Servidor de chaves – que tem o objetivo de armazenar o contexto de segurança do usuário;
- Servidor de parâmetros – que tem o objetivo de armazenar os parâmetros de derivação da chave simétrica de proteção.

O usuário utiliza o aplicativo cliente SATES para estabelecer a comunicação com o Servidor SATES.

O servidor provê as seguintes funcionalidades:

- Base de dados das contas dos usuários;

- Repositório de conteúdos remotos;
- Verificação do parceiro de comunicação;
- Registro de atividades;
- Análise e Decisão;
- Interface com Autoridade de Certificadora;
- Gerenciamento.

A seguir estas funcionalidades são apresentadas com mais detalhes.

5.2.5.1 Base de dados das contas dos usuários

A base de dados das contas dos usuários é o principal elemento utilizado na autenticação. Cada conta do usuário pode ser relacionada a vários parâmetros, como:

- O nome do usuário;
- Os parâmetros de autenticação SRP, o *salt* e o valor verificador;
- O certificado digital armazenado em um cartão SATES;
- A requisição do certificado digital do usuário;
- O certificado emitido do usuário.

5.2.5.2 Gerenciamento

O servidor SATES provê uma interface de gerenciamento das contas dos usuários e dos parâmetros de configuração do sistema.

O gerenciamento relacionado à base das contas dos usuários fornece as operações de inserção, remoção e edição de usuários.

Os parâmetros de configuração representam as características relacionadas à execução do sistema SATES, como: a porta de escuta do serviço; quantas tentativas inválidas são necessárias para o travamento de uma conta de usuário; localização do certificado digital dos servidores de chave e de parâmetros e endereço de rede da autoridade certificadora.

5.2.5.3 Repositório de conteúdos remotos

Os aplicativos servidor de chave ou de parâmetros são responsáveis por armazenar de forma segura as informações protegidas dos usuários. Como discutido na seção 5.2.4, as informações podem ser o contexto de segurança ou parâmetros de derivação de senha.

5.2.5.4 Verificação do parceiro de comunicação

Durante o estabelecimento do canal seguro, ambos os servidores (chaves e parâmetros) por meio da comunicação SSL obtém o certificado digital parceiro contido no cartão SATES. O certificado do cartão SATES é verificado quanto a sua integridade e sua fidelidade. A verificação da lista de certificados revogados é opcional, visto que esta operação pode onerar o desempenho do sistema, como previamente discutido na seção 2.5 sobre certificação digital.

O servidor mantém o certificado de lote do cartão SATES dentro de uma sessão do usuário criada de tal forma que este certificado esteja relacionado ou compatível com o certificado já cadastrado na conta do usuário relacionado com informações da conta do usuário.

5.2.5.5 Registro de atividades

O servidor realiza o registro das atividades como: geração de chaves, tentativas de autenticação, origem da comunicação, entre outros parâmetros. O registro destes parâmetros tem o objetivo de fornecer informações sobre o funcionamento do sistema SATES e constituir uma base para a decisão sobre possíveis atividades que estejam sendo realizadas por usuários maliciosos.

Além disso, o registro das atividades provê o serviço de segurança de auditoria.

5.2.5.6 Análise e Decisão

Conforme a definição do protocolo SATES na seção 5.2.3, o servidor utiliza parâmetros armazenados referentes à conta do usuário e da sessão corrente. Estes parâmetros podem ser: o certificado do aplicativo cliente SATES, a autenticação SRP, tentativas inválidas de autenticação, entre outros. Assim, estes parâmetros são utilizados para a tomada de uma decisão quanto à legitimidade da autenticação.

A decisão pode também disparar três tipos de contramedidas: travamento temporário da conta do usuário, alerta ao usuário referente a comportamentos anômalos sobre sua conta e pedido de revogação do certificado digital do usuário.

5.2.5.7 Interface com Autoridade Certificadora

O servidor tem a funcionalidade de auxiliar o processo de geração do certificado digital do usuário de modo que ele seja a interface entre a AC e o usuário.

Durante a primeira utilização do cartão SATES, o usuário inicia o processo de pedido do certificado. O servidor de chaves recebe e encaminha a requisição de certificado (CSR) a uma autoridade certificadora configurada.

O resultado das interações entre o servidor e a autoridade certificadora pode ser o certificado emitido ou um aviso de pendência. Em ambos os casos, estas informações são armazenadas na conta do usuário.

5.3 Funcionamento

O sistema SATES possui dois fluxos de funcionamento principais:

- Cadastro do usuário – fluxo que caracteriza as atividades conseqüentes realizadas desde o recebimento do cartão SATES pelo usuário até a geração de sua respectiva identidade digital;
- Operação – fluxo que caracteriza as atividades de utilização do sistema SATES.

5.3.1 Cadastro do usuário

O fluxo de cadastro de usuário é dividido em três etapas: pré-cadastro, geração de identidade e atualização da identidade. A seguir estas fases são descritas detalhadamente.

5.3.1.1 Pré-cadastro – Fase 1

Primeiramente, o usuário encontra-se em uma estação administrativa e inicia seu cadastro ao definir o nome da conta e a respectiva senha de autenticação com o auxílio de um

administrador do sistema. Em seguida, o administrador escolhe ao acaso um cartão SATES e o entrega ao usuário.

5.3.1.2 Geração da identidade – Fase 2

Esta fase tem início depois da realização da fase 1.

O usuário, tendo a posse do cartão SATES, continua com o processo de cadastro. Agora, o usuário encontra-se na sua estação de trabalho e prossegue executando o aplicativo cliente SATES localizado no cartão.

O aplicativo cliente exibe uma interface solicitando a inserção dos parâmetros usuário e senha. Após isto, o aplicativo cliente inicia uma comunicação com o servidor para realizar a autenticação. Neste momento, o servidor identifica que o usuário em questão não possui o cadastro concluído e retorna ao aplicativo um comando para a seqüência do processo.

O aplicativo cliente exibe um assistente para esta etapa, a qual contempla a geração do par de chaves do usuário, a proteção par de chaves e o envio da requisição do certificado e do par de chaves protegido ao servidor. Para isto o usuário, no decorrer desta etapa, define dados do certificado e a senha de proteção do par de chaves. O par de chaves representa a identidade digital do usuário.

Ao término desta etapa no cliente, o servidor SATES fará uma tentativa de emissão imediata junto à autoridade certificadora. Caso obtenha sucesso, o cliente receberá uma mensagem de sucesso. Do contrário, o cliente receberá uma mensagem determinando que o pedido de certificado está pendente.

5.3.1.3 Atualização da identidade – Fase 3

Esta fase tem início depois da realização da fase 2, caso o servidor SATES tenha obtido sucesso na emissão imediata. Entretanto, esta fase também pode ser iniciada com a tentativa de autenticação do usuário após ter recebido a mensagem de pedido pedente.

O usuário informa a senha de proteção do par de chaves para que o aplicativo cliente realize a extração do mesmo para que ele possa ser atualizado com certificado correspondente emitido. Novamente, o aplicativo cliente protege o par de chaves, o qual se encontra atualizado, e o envia ao servidor.

Esta fase conclui o fluxo de cadastro do usuário. Desta forma, o usuário está habilitado a realizar a autenticação no sistema SATES.

5.3.2 Operação

O fluxo de operação representa as atividades que podem ser realizadas pelo usuário após autenticação efetuada de forma bem sucedida. Como prova de conceito foram escolhidas duas atividades:

- Acesso a uma página de comércio eletrônico – esta atividade envolve o uso da identidade digital do usuário no estabelecimento de um canal seguro (SSL) com autenticação mútua;
- Assinatura digital – esta atividade permite que o usuário utilize sua chave privada na geração de uma assinatura digital. A assinatura digital tem o formato PKCS#7.

5.4 Escopo

O sistema SATES limita-se à definição dos seguintes elementos:

- Um protocolo de autenticação de parceiros robusto que permita ao usuário receber uma informação sensível de forma segura através de uma rede não confiável;
- Introduz ao protocolo a posse de um dispositivo de armazenamento com propriedade de ser somente de leitura fornecendo um segundo fator de autenticação;
- Concentra-se nas proteções necessárias da parte cliente;
- Define e estabelece um paralelo entre os modelos propostos;
- Fornece uma especificação sobre os requisitos mínimos identificados para a execução dos protocolos de forma correta.

O trabalho está orientado a utilizar mecanismos e soluções já consolidadas, característica também observada no trabalho de Brainard et al. (2003). Desta forma, são utilizadas as soluções: canal seguro SSL e mídia CD-ROM. Dentro deste contexto, a contribuição é alcançada a partir da definição de elementos interconectados para que o conjunto forneça a plataforma de objetivo deste trabalho.

Por este trabalho não são abordadas as seguintes questões:

- Clonagem do CD-ROM;
- Detalhamento de contramedidas aos ataques de força bruta, *Distributed Denial of Service* (DDOS).
- Detalhamento extensivo de um repositório seguro.

Este trabalho não pretende atingir os níveis de segurança obtidos em sistemas que utilizem dispositivos de autenticação criptográficos, como cartões inteligentes ou *tokens*.

Contudo, conforme a Figura 26, o trabalho eleva o nível de segurança comum hoje utilizado em aplicações da Internet.

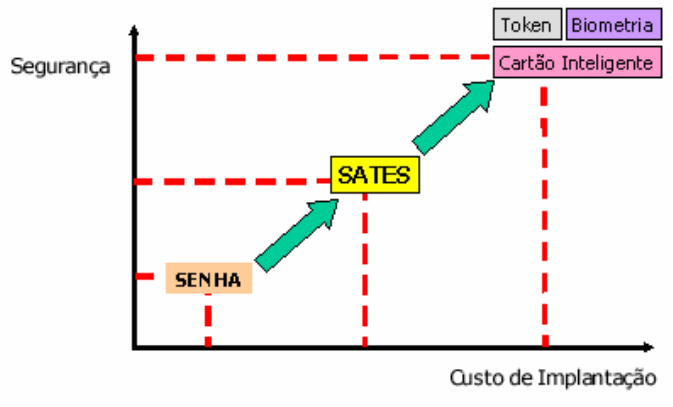

Figura 26 – Posicionamento do SATES no contexto de implantação da segurança dos sistemas de autenticação

5.5 Modelos do Sistema SATES

Foram consideradas neste trabalho três possíveis variações para o sistema SATES. Cada uma apresenta vantagens e desvantagens entre si. Primeiramente, é apresentado o modelo básico do sistema SATES e, em seguida, suas derivações.

5.5.1 Modelo base

O modelo base tem como referência a reunião dos elementos discutidos anteriormente neste capítulo.

O usuário deve memorizar:

- Uma senha para a proteção da chave privada. Esta senha é inserida na interface do aplicativo cliente SATES e não é trafegada pela rede;

- Uma senha para a autenticação SRP. Esta senha é também inserida na interface do aplicativo cliente SATES e não é trafegada pela rede, mas é utilizada na execução do protocolo SRP, o qual trafega valores derivados da senha.

O certificado digital presente no cartão SATES é utilizado pelo aplicativo cliente apenas no estabelecimento do canal seguro. Este certificado poderia ser utilizado também para proteger o contexto de segurança. Porém, a perda do cartão resultaria na perda do contexto de segurança, devido à dependência sugerida. Embora o nível de segurança pudesse ser aumentado, foi feita a opção de manter independentes as duas informações, o certificado e o contexto de segurança.

O repositório de certificados raiz, como previamente dito, é utilizado nas validações dos certificados parceiros nas comunicações.

O canal seguro estabelecido entre os parceiros de comunicação é provido pela utilização do protocolo SSL com autenticação mútua. Por este canal são trafegadas informações de:

- Operação – como o contexto de segurança e quais são as atividades que o aplicativo cliente deve executar, entre outras;
- Autenticação SRP – como o nome do usuário e parâmetros utilizados na execução do protocolo.

Em todos os modelos a proteção realizada sobre a chave privada é feita por meio de uma criptografia simétrica, resultado que define o contexto de segurança. Entretanto, são apresentadas variações para cada modelo quanto à forma de geração da chave simétrica a partir da senha.

5.5.2 Modelo A

Este modelo é baseado no artigo desenvolvido por Guelfi, Meylan e Fonseca (2003).

O modelo A apresenta a alternativa de menor custo em relação aos outros modelos, pois são definidas matrizes para o conteúdo do dispositivo, as quais recebem certificados digitais e outras informações. Como mostra a Figura 27, "n" usuários utilizam o mesmo conteúdo do cartão SATES, ou seja, o mesmo certificado de lote. Entretanto, cada usuário utiliza seu cartão individualmente. Assim, o custo é diminuído devido à replicação dos cartões SATES.

A proporção do número de cartões replicados em relação ao número de usuários do sistema e a lógica de distribuição de cartões vão determinar o nível de segurança alcançado. Estes fatores devem ser ajustados de modo que a probabilidade de usuários vizinhos terem o mesmo cartão seja baixa. Do contrário, um usuário pode fazer uma tentativa de acesso não autorizado por meio de um cartão correspondente, ou seja, usuários com o mesmo lote são vulneráveis entre si.

Em analogia aos sistemas de autenticação de *internet banking* atuais que utilizam informações como cartão de códigos numéricos ou senhas alfabéticas com tamanhos pequenos (três ou quatro caracteres) a replicação destas informações é comum, pois se supondo uma senha de três caracteres são constituídos lotes de mil usuários.

Este modelo para a geração da chave simétrica de proteção utiliza o algoritmo PKDF1 com parâmetros fixos por lote de cartões. A recomendação encontrada em Kaliski (2000) sugere que o valor de iterações seja maior que 1.000. Desta forma, o valor da quantidade de iterações é próximo a 1.000 com base na recomendação e é gerado aleatoriamente, assim como o valor do *salt*, no momento de geração do conteúdo do cartão SATES.

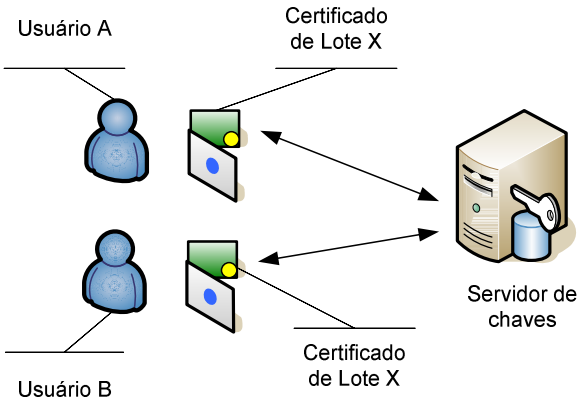

Figura 27 - SATES - Modelo A

5.5.3 Modelo B

Este modelo é uma variação do modelo A de modo que o número de usuários por lote de cartões seja reduzido a um, ou seja, cada cartão possui um certificado de lote diferente. Assim, são gerados cartões SATES com conteúdos diferentes para cada usuário, como exibe a Figura 28. Desta forma, um cartão pode ser utilizado apenas por um determinado usuário.

A questão da replicação não está presente neste modelo. Entretanto, a individualidade do cartão prevê um aumento de custo em relação ao modelo A.

Este modelo para a geração da chave simétrica de proteção utiliza a mesma definição apresentada pelo modelo A. Porém, os parâmetros de derivação são únicos por usuário. Assim, é aplicada também a concepção de cartões individuais.

O trabalho de Brainard et al. (2003) comenta que o protocolo de Ford e Kaliski (2000) tem o objetivo de deslocamento de uma credencial ou uma chave privada protegida. Este objetivo influenciou a definição do modelo B.

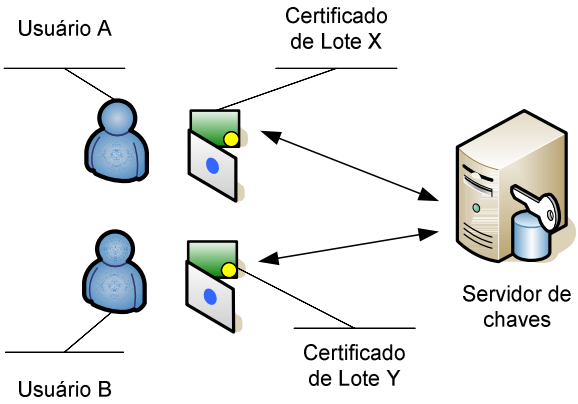

Figura 28 - SATES - Modelo B

5.5.4 Modelo C

O modelo C continua a evolução a partir do modelo B. Define a necessidade de uma comunicação adicional para a abertura do contexto de segurança além da comunicação usual necessária para obtê-lo, conforme a Figura 29.

Este modelo tem o objetivo de oferecer o não repúdio como visto nos modelos de múltiplos servidores. Além disso, tem o objetivo de fornecer a dependência da obtenção remota dos parâmetros utilizados na abertura do contexto de segurança. Desta forma, o sistema SATES pode ter mais um ponto de rastreamento da utilização do contexto de segurança do usuário.

Com a utilização do deslocamento de parâmetros de derivação, o modelo C requer a presença de mais um servidor, se comparado aos modelos anteriores. O servidor adicional é o repositório de parâmetros, o qual também requer um canal seguro e uma autenticação SRP. Assim, o usuário deve memorizar três senhas: senha de proteção da chave, senha de autenticação SRP com o repositório seguro, e senha de autenticação SRP com o repositório de parâmetros de derivação.

O modelo C no que diz respeito à forma de geração da chave simétrica de proteção segue as definições apresentadas pelos modelos A e B, o algoritmo utilizado é o PKDF1 e é provida a individualidade dos parâmetros. Porém, os parâmetros são armazenados em um servidor remoto, como previamente discutido na seção 5.2.4.2.

Novamente com base no trabalho de Brainard et al. (2003), é dito que o seu objetivo além do deslocamento de uma chave privada, também pode ser o deslocamento de uma informação sobre a senha de proteção utilizada sobre a chave privada. Portanto, esta proposta se assemelha à definição do modelo C.

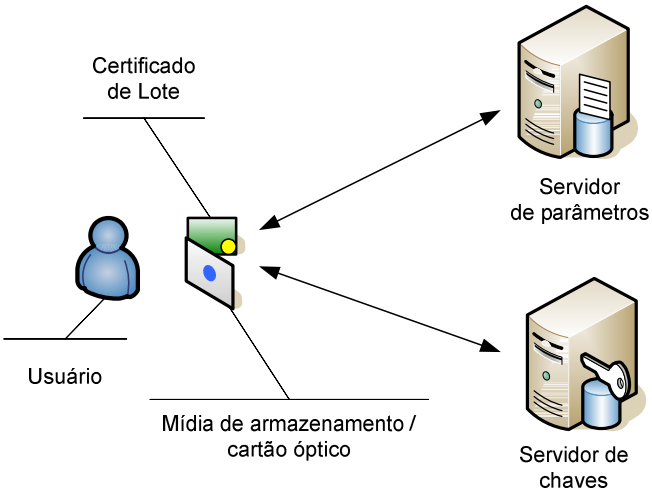

Figura 29 - SATES - Modelo C

Os modelos do sistema SATES foram concebidos considerando algumas características. O modelo A foi concebido com o objetivo de baixar o custo operacional da gravação dos cartões SATES. O modelo B foi concebido com o objetivo de manter a individualidade do dispositivo. O modelo C foi concebido com o objetivo de controle sobre a abertura da chave privada do usuário.

O próximo capítulo apresenta como foram encaminhadas as tarefas executadas no desenvolvimento do projeto, como a preparação de ambientes de teste, implementação dos protótipos e coleta de resultados.

6 Avaliação do Sistema SATES

Este capítulo descreve as avaliações que foram executadas para o sistema SATES. Além do estudo realizado sobre as tecnologias relacionadas e a definição do sistema SATES, este capítulo abrange as seguintes etapas:

- Implementação dos protótipos avaliados;
- Especificação do ambiente de testes;
- Experimentação, testes e análise de resultados de desempenho;
- Conclusão da avaliação do SATES.

As próximas subseções vão discorrer de forma mais detalhada sobre cada uma das etapas de avaliação do sistema SATES.

6.1 Implementação

A etapa de implementação foi definida pela criação de dois protótipos, aplicativo cliente e o aplicativo servidor, que têm o objetivo de fornecer exemplos práticos dos modelos propostos pelo sistema SATES e coletar parâmetros para a etapa de testes. Além disso, foram também implementados mais dois aplicativos, o aplicativo gerador de medidas e o aplicativo servidor de medidas, com objetivo focado nos testes.

O sistema operacional utilizado para a execução dos aplicativos nesta etapa consta da plataforma Microsoft Windows 2000 Advanced Server com os serviços *Certificate Services* e o de banco de dados MySQL instalados.

A ferramenta de desenvolvimento dos aplicativos do protótipo foi o Borland C++Builder, logo, a linguagem de programação escolhida foi o C++. Além disso, as bibliotecas OpenSSL[11] e Microsoft CriptoAPI[12] forneceram o suporte para efetuar as operações criptográficas necessárias ao funcionamento do protótipo. A biblioteca LibSRP (WU, 2005) também foi utilizada para prover as rotinas de execução do protocolo de autenticação SRP. A proteção da chave privada do usuário utilizou

6.1.1 Aplicativo Servidor

O aplicativo servidor tem o objetivo de prover tanto o servidor de chaves quanto o servidor de parâmetros. Para isto, existem valores de configuração que orientam o mesmo aplicativo servidor a atuar de duas formas.

O servidor faz interface com o banco de dados MySQL para o armazenamento de valores divididos nas seguintes tabelas:

- Usuários – tem o objetivo de prover a base dos usuários cadastrados do sistema e armazena os valores: código identificador do usuário, nome do usuário, *salt*, verificador, estado, código identificador do lote, parâmetros de derivação de senha do usuário;
- Lotes – tem o objetivo de prover a base dos lotes de cartão SATES representados por certificados digitais e armazena os valores: código identificador do lote, nome do lote; a chave pública do certificado digital do lote, o certificado digital do lote codificado em Base 64;

[11] Padrão aberto da implementação dos mecanismos de criptografia, hospedado no site http://www.openssl.org.
[12] Biblioteca de serviços de certificação desenvolvida pela Microsoft contida nas plataformas Windows 2000.

- Conteúdos - tem o objetivo de prover o repositório de conteúdos dos usuários, como por exemplo, o certificado digital, o pedido de certificado (CSR) e a chave privada. Esta tabela armazena os valores: código identificador do conteúdo, código identificador do usuário proprietário, código identificador do tipo do conteúdo, o conteúdo codificado em Base 64.

Como se pode perceber as tabelas possuem relacionamentos, como por exemplo, a tabela Usuários está relacionada à tabela de Lotes e a tabela de Conteúdo está relacionada à tabela de Usuários.

O aplicativo servidor utiliza uma API específica para se comunicar com o serviço *Certificate Services* da plataforma Windows 2000, o qual é o exemplo de autoridade certificadora utilizado na prova de conceito.

6.1.2 Aplicativo cliente

O aplicativo cliente SATES é responsável por realizar as atividades necessárias para a autenticação do usuário. Ainda, o aplicativo cliente tem a função de fornecer a coleta dos valores de tempo durante a sua execução.

A coleta dos valores de tempo é feita da seguinte forma: são registrados os instantes das operações executadas durante a autenticação, como mostra a reprodução ilustrada na Figura 30.

```
Tempo em milisegundos ; Tempo formato padrão – Atividade
33536812 ; 09:18:56:812 – [cliente] Início da sessão
33536859 ; 09:18:56:859 – [cliente] Socket Ativo
33536859 ; 09:18:56:859 – [cliente] Configuração SSL
33537156 ; 09:18:57:156 – [cliente] Conexão SSL estabelecida
33537156 ; 09:18:57:156 – [cliente] Início da autenticação SRP com o servidor de chaves
```

```
33537156 ; 09:18:57:156 - [cliente] inicio da sessão
33537156 ; 09:18:57:156 - [cliente] enviando nome do usuário 'Renato'
33537187 ; 09:18:57:187 - [cliente] parâmetros carregados
33537187 ; 09:18:57:187 - [cliente] geração do valor público A_PARAM
33537187 ; 09:18:57:187 - [cliente] realizada a geração do valor público A_PARAM
33537187 ; 09:18:57:187 - [cliente] enviando A_PARAM
33537250 ; 09:18:57:250 - [cliente] A_PARAM enviado
33537250 ; 09:18:57:250 - [cliente] calculando chave
33537250 ; 09:18:57:250 - [cliente] chave calculada
33537250 ; 09:18:57:250 - [cliente] enviando PROOF
33537281 ; 09:18:57:281 - [cliente] PROOF enviado
33537281 ; 09:18:57:281 - [cliente] SERVER_PROOF recebido
33537281 ; 09:18:57:281 - [cliente] verificando SERVER_PROOF
33537281 ; 09:18:57:281 - [cliente] SERVER_PROOF verificado
33537281 ; 09:18:57:281 - [cliente] SRP liberação de variáveis
33537281 ; 09:18:57:281 - [cliente] SRP fim
33537281 ; 09:18:57:281 - [cliente] Fim da autenticação SRP com o servidor de chaves
```

Figura 30 - Reprodução do registro de tempo das atividades do aplicativo cliente

O aplicativo cliente SATES é armazenado no cartão SATES, o qual é equipado com as bibliotecas e arquivos que são dependências para que o aplicativo cliente seja executado sem a necessidade de acesso ao disco rígido. Porém, durante os testes, descritos na seção 6.2, o aplicativo foi executado do disco rígido.

6.1.3 Aplicativo Gerador de Medidas

O aplicativo gerador de medidas tem o objetivo de coletar parâmetros de tempo gasto em funções específicas relacionadas à autenticação do sistema SATES. As funções específicas foram escolhidas dentre as atividades definidas no sistema SATES e algumas variações não previstas, baseando-se nos resultados obtidos na análise dos valores coletados pelo aplicativo cliente SATES.

6.1.4 Aplicativo Servidor de Medidas

O aplicativo servidor de medidas tem o objetivo de fornecer o suporte necessário para a execução do aplicativo gerador de medidas, visto que o aplicativo servidor SATES não atende as variações de implementação presentes no gerador de medidas.

6.2 Experimentação e Análise de resultados

A etapa de experimentação e testes tem objetivo de gerar o contexto necessário para a evolução deste trabalho. Para isto, foram utilizados três ambientes de trabalho diferentes, como mostra a Figura 31.

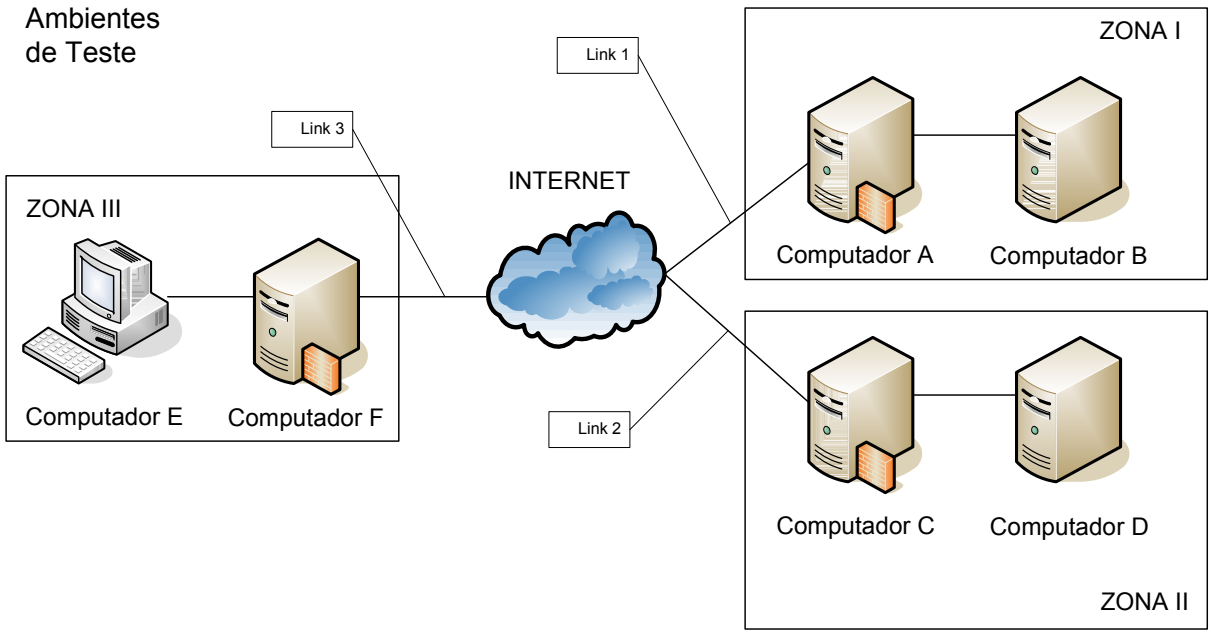

Figura 31 - Ambientes de teste

O primeiro ambiente de trabalho, chamado Zona I, encontra-se no Laboratório de Sistemas Integráveis (LSI), que conta com o Link 1 (1Gbps) e dois computadores envolvidos nos testes:

- Computador A – representa a Zona I por meio de um endereço de rede fixo na Internet. Além disso, têm as funcionalidades de *Firewall* e de *Network Address Translation* (NAT). O NAT é configurado para que seja possível acessar o Computador B pela Internet. O computador A conta com a seguinte especificação técnica: Processador Pentium 2, 300 MHz; 128 MB RAM; 20 GB de disco rígido; Sistema Operacional Linux Red Hat 9;
- Computador B – tem a funcionalidade de servidor e nele estão presentes: o aplicativo servidor; o aplicativo servidor de medidas e o servidor de banco de dados MySQL. O computador D conta com a seguinte especificação técnica: Processador AMD Athlon XP 1700, 1.46 GHz; 512 MB de memória RAM; 20 GB de disco rígido, Sistema Operacional Windows XP.

O ambiente chamado de Zona II também possui o Link 2 com o valor de 512 Kbps e conta com dois computadores para os testes:

- Computador C – assim como o computador A, representa a sua respectiva zona, Zona I, por meio um endereço de rede fixo na Internet e tem também as funcionalidades de *Firewall* e NAT. Porém, o computador apenas provê o acesso do Computador D à Internet. Processador. Intel Celeron, 450MHz, 128 MB de RAM, 100GB de disco rígido, Sistema Operacional Linux Fedora.
- Computador D – representa uma característica de cliente, de acordo com a configuração do Computador C. Estão presentes os aplicativos cliente e o gerador de medidas. O computador D conta com a seguinte especificação técnica:

Processador Intel Pentium 4, 1.8 GHz; 512 MB de memória RAM; 40 GB de disco rígido, Sistema Operacional Windows 2000 Advanced Server.

O ambiente chamado de Zona III é conectado à Internet por meio do Link 3 com valor de 1Mbps e também conta com dois computadores para os testes:

- Computador E – não possui um endereço fixo na Internet, porém conta com a funcionalidade de NAT. Assim, o Computador E apenas provê o acesso do Computador F à Internet. O computador E conta com a seguinte especificação técnica: Processador Intel Pentium 3, 450 MHz; 256 MB de memória RAM; 20 GB de disco rígido, Sistema Operacional Windows XP.
- Computador F – pode ser considerado um cliente comum, devido às suas características de acesso à Internet. Estão presentes os aplicativos cliente e o gerador de medidas. O computador F conta com a seguinte especificação técnica: Processador AMD Semprom 2600, 1.6GHz; 512 MB de memória RAM; 80 GB de disco rígido, Sistema Operacional Windows 2000 Advanced Server.

Com base no ambiente de testes ilustrado pela Figura 31, o qual conta com as três zonas discutidas, foram executados os seguintes testes:

- Teste de desempenho do Sistema SATES, como descrito na seção 5.3.2;
- Teste de desempenho do protocolo SRP em plano e com SSL;
- Teste de desempenho do *download* de LCRs.

6.2.1 Sistema SATES

O teste com o sistema SATES tem o objetivo de identificar o desempenho de funções executadas durante o funcionamento. Para este teste foram utilizados os aplicativos cliente SATES e servidor SATES executados a partir do disco rígido.

O teste também compreende a variação dos três modelos propostos, A, B e C. Entretanto, os modelos A e B para os testes apresentam a mesma característica de funcionamento, visto que a diferença entre eles é dada apenas pelo cadastro no banco de dados. O banco de dados está modelado para que um certificado de lote possa ser associado a um ou vários usuários.

Assim, para esta avaliação do desempenho dos modelos A e B foram coletadas 20.000 amostras para cada tipo de teste realizado. Foram realizados dois tipos de testes:

- Teste 1 – este teste utiliza o Computador F para a execução do aplicativo cliente SATES e o Computador B como servidor de chaves;
- Teste 2 - este teste utiliza o Computador D para a execução do aplicativo cliente SATES e o Computador B como servidor de chaves.

Etapas avaliadas	Teste 1		Teste 2		Média dos Testes	
	(ms)	(%)	(ms)	(%)	(ms)	(%)
Tempo médio gasto no estabelecimento do canal seguro SSL	195,5	44,69%	265,5	39,54%	230,5	41,57%
Tempo médio gasto na autenticação SRP	171,5	39,20%	187	27,85%	179,25	32,33%
Tempo médio gasto na abertura da chave	15,5	3,54%	15	2,23%	15,25	2,75%
Tempo gasto no protocolo de deslocamento	23,5	5,37%	149	22,19%	86,25	15,55%
Tempo gasto nas demais operações (transmissão de dados, leitura e decoficação de informações, etc)	31,5	7,20%	55	8,19%	43,25	7,80%
Tempo médio para o total dos testes	437,5	100,00%	671,5	100,00%	554,5	100,00%

Tabela 4 - Síntese dos valores dos testes realizados para os Modelos A e B

De acordo com a Tabela 4, nota-se que durante a execução do protocolo SATES a maioria do tempo gasto (mais de 70% do tempo total) é dada pela execução dos protocolos SSL e o SRP.

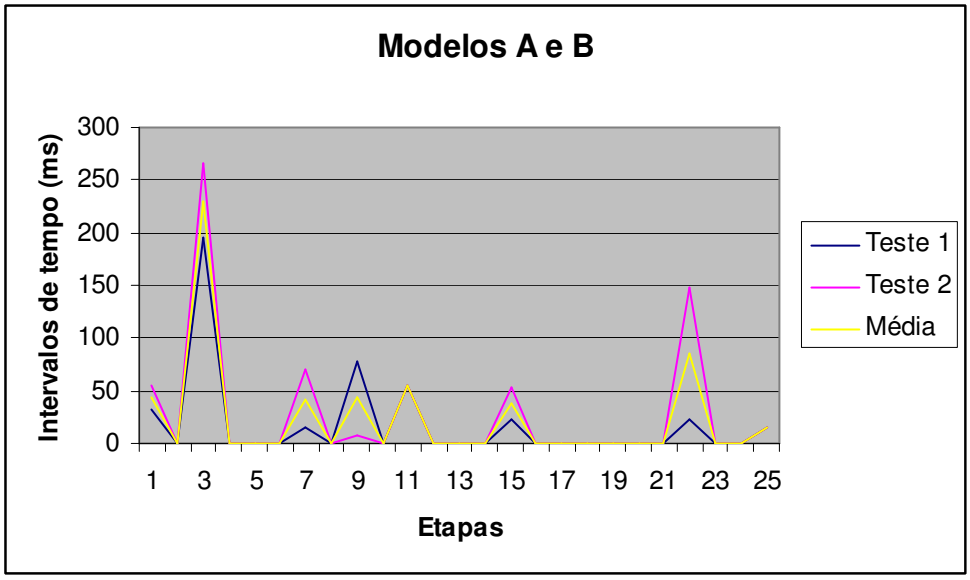

Gráfico 1 - Intervalos de tempo x Etapas da execução dos modelos A e B

O Gráfico 1 ilustra os intervalos de tempo coletados na execução do protocolo SATES. As etapas fazem referência às funções executadas pelo aplicativo cliente SATES, por exemplo, como mostra a Figura 30, a qual descreve os registros das funções.

Como se pode perceber, com base no Gráfico 1, existe um pico na execução da etapa 3 o qual representa o tempo gasto no estabelecimento da conexão SSL. Além disso, também existe outro pico na etapa 22 que possui grande variação entre o Teste 1 e o Teste 2. A etapa 22 é representada pela execução do protocolo de deslocamento.

Após a finalização dos testes de desempenho com os modelos A e B, foram realizados testes com o modelo C.

Para avaliação do modelo C foram coletadas 10.000 amostras para cada tipo de teste realizado. Foram realizados dois tipos de testes:

- Teste 3 – este teste tem como característica a utilização do Computador D como servidor de chaves, o Computador B como servidor de parâmetros e o Computador F como aplicativo cliente SATES;

- Teste 4 – este teste utiliza o esquema de servidores inverso ao Teste 3. Assim, novamente o Computador F é referente ao aplicativo cliente SATES, o Computador B como servidor de chaves, o Computador D como servidor de parâmetros.

Etapas avaliadas	Teste 3		Teste 4		Média dos Testes	
	(ms)	(%)	(ms)	(%)	(ms)	(%)
Tempo médio gasto no estabelecimento do canal seguro com o Servidor de parâmetros	277,4464	22,25%	402,7952	33,99%	340,1208	27,97%
Tempo gasto na autenticação SRP com o Servidor de parâmetros	137,9805	11,06%	229,0377	19,33%	183,5091	15,09%
Tempo gasto no estabelecimento do canal seguro com o Servidor de chaves	337,9935	27,10%	222,157	18,75%	280,07525	23,03%
Tempo gasto na autenticação SRP com o Servidor de chaves	241,3386	19,35%	136,0296	11,48%	188,6841	15,52%
Tempo gasto da derivação de parâmetros	6,1505	0,49%	6,0188	0,51%	6,08465	0,50%
Tempo gasto na abertura da chave	9,8927	0,79%	10,0788	0,85%	9,98575	0,82%
Tempo gasto no protocolo de deslocamento	111,4755	8,94%	62,8026	5,30%	87,13905	7,17%
Tempo gasto nas demais operações (transmissão de dados, leitura e decoficação de informações, etc)	124,7744	10,01%	115,9681	9,79%	120,37125	9,90%
Tempo de total	1247,052	100,00%	1184,888	100,00%	1215,97	100,00%

Tabela 5 - Síntese dos valores dos testes realizados para o Modelo C

Nota-se na Tabela 5, assim como para o teste com os modelos A e B, que a execução dos protocolos SSL e SRP representa também a maioria do tempo gasto (por volta de 80%).

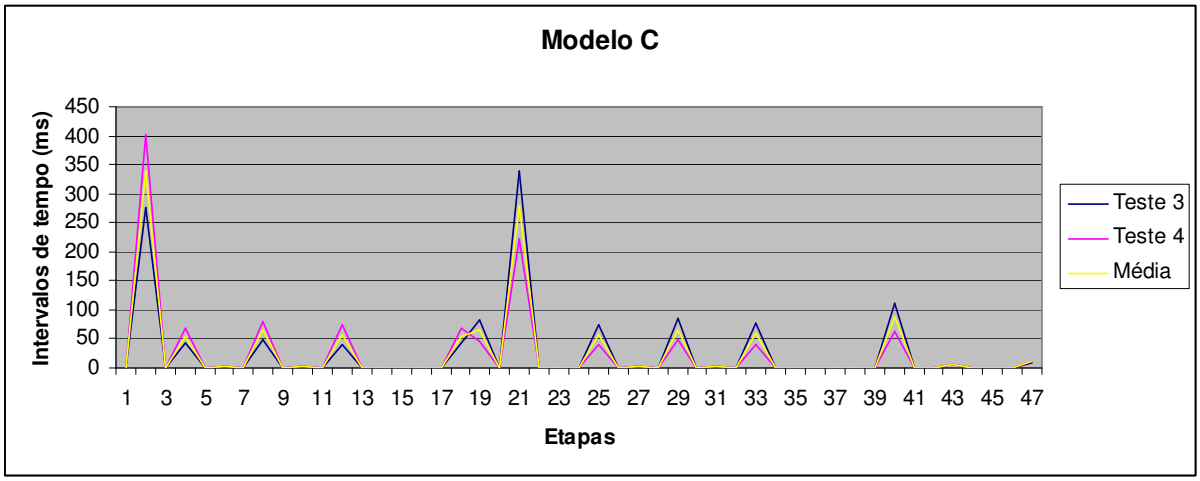

Gráfico 2 - Intervalos de tempo x Etapas da execução do modelo C

Com base no Gráfico 2, podem-se perceber dois picos, etapas 2 e 21, que representam o tempo gasto no estabelecimento do canal SSL.

Sendo relacionados os valores médios do tempo total da execução dos modelos A, B e C, com base na Tabela 4 e na Tabela 5, o modelo C apresenta um acréscimo de tempo se comparado aos modelos A e B, de aproximadamente 45%. Isto devido à quantidade de servidores utilizados, que para os modelos A e B é igual a um servidor e para o modelo C igual a dois.

Com base nos testes apresentados nesta seção, se pode notar que o estabelecimento do canal SSL é o processo mais dispendioso da execução de todos os modelos do sistema SATES.

6.2.2 Protocolo SRP

O teste com o protocolo SRP foi realizado com o intuito de avaliar uma condição que não está presente no Sistema SATES. A condição é dada pela execução em plano do protocolo SRP, visto que tanto o servidor de chaves, como o servidor de parâmetros, exigem conexão segura por meio do protocolo SSL.

O teste consiste na repetição de um número fixo de vezes em que o aplicativo gerador de medidas efetua a autenticação SRP. O instante antes do teste é armazenado, após as repetições o instante também é registrado, e o resultado é dado pelo intevalo de tempo dividido pelo número de repetições.

Para o teste do SRP em plano foram realizadas 5.000 amostras de execução do protocolo.

Zona Origem	Tempo médio gasto na execução do SRP em Plano (ms)
Zona 2	171,4123
Zona 3	168,4831
Média	169,9477

Tabela 6 - Síntese dos valores coletados no teste do SRP em plano com destino à Zona 1

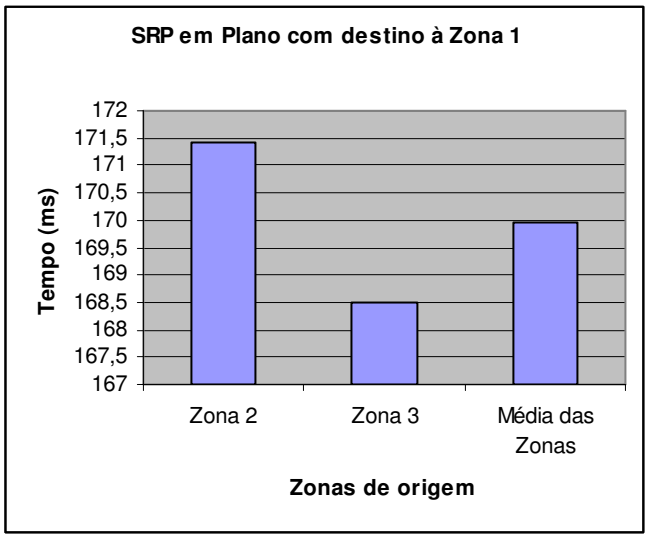

Gráfico 3 - SRP em plano com destino a Zona 1

Com base no Gráfico 3 e na Tabela 6, pode-se perceber que a zona de origem tem uma pequena influência o tempo gasto na execução. O aspecto da zona pode ser associado ao Link utilizado. Assim, pode-se supor que a Zona 3 teve um desempenho ligeiramente melhor devido à variações no trafego de pacotes na rede ou talvez por diferenças no poder de processamento dos computadores D e F.

Baseando-se na conclusão de que a zona de origem tem uma pequena influência no desempenho da execução do SRP, foram utilizados os valores coletados nos testes 1 e 2, os quais avaliaram do desempenho da execução do protocolo SATES. Assim, conforme demonstra a Tabela 4, foram somados os valores do tempo gasto no estabelecimento do canal seguro e da autenticação SRP, e o resultado foi relacionado, no Gráfico 4, com a média das zonas no teste do SRP plano.

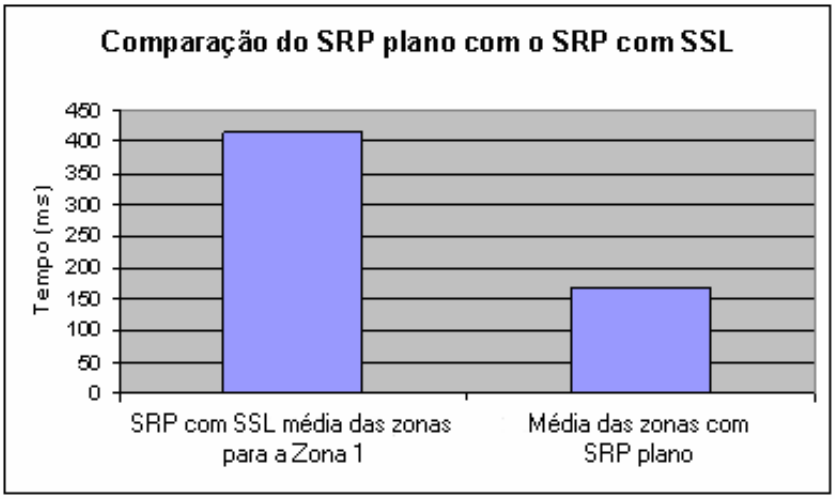

Gráfico 4 - Comparação do SRP plano com o SRP com SSL

Assim como identificado no teste do sistema SATES apresentado na seção 6.2.1, e com base no Gráfico 4, o protocolo SSL mostrou-se também dispendioso.

6.2.3 *Download* de LCR

Os testes de *download* de listas de certificados revogados também tiveram o objetivo de avaliar uma condição não encontrada no teste do Sistema SATES, visto que os protótipos

implementados não contam com a funcionalidade de verificação da revogação de um certificado digital.

Para os testes foram escolhidas LCRs ao acaso considerando uma escala de tamanho. A Tabela 7 identifica as LCRs utilizadas pelo endereço de publicação das ACs.

O teste consiste na repetição de um número fixo de vezes do *download* de cada LCR apontada na Tabela 7. O instante antes do teste é armazenado, após as repetições o instante também é registrado, e o resultado é dado pelo intevalo de tempo dividido pelo número de repetições.

URL	Tama-nho em bytes	Tempo de *download* em (ms)		
		Zona 1	Zona 2	Zona 3
http://www.receita.fazenda.gov.br/acsrf/acsrf.crl	484	57,2562	48,443	61,46367
http://cert.oab.org.br/oab.crl	753	570,6446	61,427	70,54667
http://www.certificadodigital.com.br/repositorio/crl/SerasaSRF.crl	1.113	1101,548	102,354	116,1887
http://icp-brasil.certisign.com.br/repositorio/lcr/ACCertiSignSRF/LatestCRL.crl	53.999	554,8178	865,5627	546,507
http://crl.verisign.com/ThawteServerCA.crl	137.768	1301,676	3113,463	2023,118

Tabela 7 - Lista de LCRs para os testes

Com base na Tabela 7, pode-se perceber que LCRs com o tamanho maior que 54 KB gastam por volta de 600 milisegundos no processo de *download*. Assim, se comparado ao resultado obtido nos testes com os modelos A e B, o tempo de *download* de uma LCR é igual ao tempo de execução do protocolo SATES.

6.3 Conclusão dos testes

Conforme os testes realizados, pode-se perceber que o desempenho tempo gasto na autenticação no modelo C representa aproximadamente dobro do tempo coletado para os modelos A e B.

A comparação dos testes com o SRP plano e o SRP com SSL e os testes com os modelos de implementação do SATES serviram para identificar o *overhead* provocado pelo protocolo SSL.

Os testes de *download* de LCR dão a idéia do possível problema de desempenho causado pela verificação *online* da revogação de certificados digitais.

A partir dos testes com o modelo C de implementação do SATES foi possível identificar que operações de criptografia simétrica e de derivação de senhas não oferecem grandes impactos no desempenho do SATES. A derivação de senhas é representada pela execução da função PKDF1 descrita na seção 5.2.4.2.

O protocolo SATES se mostrou com um desempenho razoável, visto que o tempo gasto na execução dos modelos A e B (uma interação com servidor) é menor do que três vezes a execução do SSL. O tempo gasto na execução do protocolo SSL serve de parâmetro porque os *sites* seguros requerem este consumo de tempo. Assim em proporção, o protocolo SATES requer por volta de 2 / 3 de tempo a mais do que os *sites* seguros comuns.

7 Conclusão

Este capítulo é dedicado a descrever algumas contribuições obtidas, algumas formas de extensão do trabalho e considerações finais.

7.1 Contribuição

Este trabalho abordou a especificação de um sistema de autenticação de entidades e autoria. Durante o desenvolvimento deste trabalho, foram discutidos os sistemas criptográficos, a arquitetura geral de autenticação e trabalhos relacionados. Entre os trabalhos relacionados, o trabalho proposto por Brainard et al. (2003) foi o que mais se destacou, devido à especificação de um sistema que contempla três dos quatro componentes da arquitetura de autenticação.

Ao final deste trabalho foram obtidas algumas contribuições:

a) Especificação de um sistema de autenticação composto pelos seguintes elementos: protocolo de autenticação baseado em senha; a utilização de um segundo fator de autenticação (posse); e posterior autenticação de parceiro por meio de certificação digital. A soma destes elementos foi capaz de descrever um sistema de autenticação de usuários mais robusto se comparado aos sistemas atuais que consideram apenas “usuário e senha”.

b) O sistema SATES apresentou uma discussão referente ao problema da mobilidade de usuários com certificados digitais que muitas vezes não estão em uma estação específica para realizar uma transação eletrônica na Internet. Para isto, foi utilizada a concepção de um contexto de segurança dos usuários, o qual representa o objeto do deslocamento (*roaming*).

c) O sistema propôs a utilização de uma técnica mais robusta de verificação da identidade do servidor parceiro de comunicação. Além disso, foi inserida a posse de um conjunto de certificados digitais raiz, cuja integridade é passível de aferir por meio da mídia somente de leitura, que validem a chave do servidor. Deste modo, foi aumentada a confiabilidade no parceiro das transações eletrônicas se comparado aos sistemas de *internet banking* atuais.

Como conseqüência, contribuições secundárias também foram estabelecidas, tais como os enumerados nos itens a seguir:

- Houve uma redução sobre o risco do fator humano diminuindo o número de senhas para os usuários, através do uso da certificação digital;
- Foi alcançada a definição de uma interface de alto nível ocultando detalhes peculiares do protocolo por meio do aplicativo cliente SATES;
- Foi fornecido o acesso remoto de informações confidenciais do usuário por meio da execução do protocolo de deslocamento;
- Foi provido um repositório seguro removível de certificados raiz confiáveis por meio da posse do cartão;
- Foi proposta uma alternativa versátil e com maior abrangência do ambiente usado por um determinado cliente se comparado ao ambientes comuns, visto que o usuário não necessita instalar ou ter instalado programas em sua estação de trabalho;
- Foi especificado um segundo fator de autenticação de baixo custo que se utiliza uma infra-estrutura hoje bastante difundida, a presença de leitores de CD-ROM;
- Foram apresentadas três alternativas de implementação, as quais apresentaram vantagens e desvantagens de acordo com o ambiente alvo de implantação.

7.2 Considerações finais

O sistema SATES reuniu conceitos atuais de autenticação de entidades e autoria em sua concepção. De acordo com o desenvolvimento deste trabalho, o sistema SATES mostrou-se uma proposta viável. Foram identificados pontos positivos e limitações.

Como principais limitações pode-se citar:

- O Modelo C requer maior manutenção do sistema, devido ao número de máquinas envolvidas;
- Em geral, o Sistema SATES requer no mínimo duas senhas, uma para a autenticação SRP no servidor de chaves e outra para abertura da chave. Porém, o Modelo C requer uma senha adicional em relação aos outros modelos, a senha para autenticação SRP no servidor de parâmetros;
- Execução de um *software*;
- Não foi definida uma forma de proteção contra a clonagem dos cartões.

E como principais pontos positivos pode-se citar:

- Propõe um sistema de autenticação que utiliza dois fatores de autenticação, posse e conhecimento;
- Propõe a difusão da certificação digital;
- O sistema SATES não trafega pela rede a senha de abertura da chave privada;
- Propõe a utilização de um dispositivo de baixo custo que representa um fator de autenticação adicional aos moldes atuais;
- Desenvolve a idéia de usuários móveis, utilizam o sistema independentemente da estação de trabalho utilizada, por meio do protocolo de deslocamento.

A análise dos resultados identificou que o sistema apresenta um desempenho aceitável dentro do ambiente testado e fornece uma base de conhecimento para a identificação de quais podem ser os elementos problemáticos em uma implantação.

Finalmente, o sistema SATES conseguiu ser concluído atendendo a expectativa inicial do projeto.

7.3 Trabalhos Futuros

Ao fim deste trabalho, ficaram algumas questões não resolvidas neste trabalho e também possíveis variações evolutivas. Assim, alguns trabalhos futuros podem ser listados:

- Estudar uma proposta que minimize os efeitos da clonagem do cartão;
- Expandir a definição do repositório seguro;
- Implementar e avaliar uma alternativa com suporte ao protocolo *Online Certificate Status Protocol* (OCSP). Alternativa apontada na etapa de testes de desempenho do sistema.
- Analisar e discutir a possibilidade de uma variação do sistema, na qual o contexto de segurança é representado por um perfil de senhas. Deste modo, o sistema poderia ser inserido em um ambiente com autenticação *Single Sign-On*[13].
- Avaliar a viabilidade e os impactos da utilização do cartão SATES criado como um *LiveCD*. O *LiveCD* é definido por uma mídia CD-ROM que contém um sistema operacional específico executável sem a necessidade de acesso ao disco rígido, isto é, diretamente a partir do CD. O sistema operacional contempla a execução de funcionalidades de sistemas comuns como configuração de rede e

[13] Single Sign-On é um mecânismo que provê a realização de apenas uma inserção de senha por parte do usuário para que o mesmo obtenha acesso a varios recursos que originalmente solicitariam senhas. O mecânismo é capaz de fornecer as senhas armazenadas em um perfil diretamente ao recurso sem a intervenção do usuário.

interface gráfica. Na maiorias dos casos o sistema operacional é o Linux. Desta forma, esta proposta estende a questão de programas confiáveis armazenados de forma auto-suficiente no cartão;

- Adaptar um modelo que utilize as tecnologias de cartão óptico e a *Identity Base Encryption*[14] (IBE). O IBE é um sistema de criptografia de chave assimétrica, no qual a chave pública é dada por uma frase ou palavra (string). A concepção do IBE prevê chaves privadas com períodos de validade pequenos, como diários ou semanais. Assim, estas chaves poderiam ser armazenadas em cartões ópticos de forma descartável;
- Adaptar um modelo que utilize como dispositivo de armazenamento o cartão *Java*, devido à propriedade de armazenamento de *applets* fornecida por este cartão.

[14] Site do projeto IBE – http://crypto.stanford.edu/ibe/

8 Referências Bibliográficas

ADAMS, C. et al, RFC 4210 – Internet X.509 Public Key Infrastructure Certificate Management Protocol, Setembro 2005, disponível em http://www.ietf.org/rfc/rfc2510.txt.

BELLOVIN, S. M.; MERRITT, M., Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In: PROCEEDINGS OF THE I.E.E.E. SYMPOSIUM ON RESEARCH IN SECURITY AND PRIVACY, 1992, Oakland.

BRAINARD, J.; JUELS A.; KALISKI, B.; SZYDLO M., A New Two-Server Approach for Authentication with Short Secrets, In: USENIX SECURITY '03, 2003.

CARRIÓN, D. de S. D., Avaliação de Protocolos de Autenticação em Redes sem Fio, dissertação de mestrado – Universidade Federal do Rio de Janeiro - COPPE, 2005.

DIERKS, T.; ALLEN, C., RFC 2246 - The TLS Protocol, 1999, disponível no site: http://www.ietf.org/rfc/rfc2246.txt.

DIFFIE, W.; HELLMAN, M. E., New Directions in Cryptography, In: IEEE Transactions on Information Theory, 22, 1976, p. 644-654.

FIELDING, R. et al., RFC 2616 - Hypertext Transfer Protocol – HTTP /1.1, 1999.

FINKENZELLER, K.; **RFID Handbook Fundamentals and Applications in Contactless Smart Cards and Identification**, Second.Edition, John Wiley & Sons, 2003.

FORD-HUTCHINSON, P., RFC 4217 - Securing FTP with TLS, 2005, disponível no site: http://www.ietf.org/rfc/rfc4217.txt.

FORD, W.; KALISKI, B., Server-Assisted Generation of a Strong Secret from a Password, In: PROCEEDINGS OF THE IEEE 9TH INTERNATIONAL WORKSHOPS ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, 2000, Gaithersburg.

FRIER, A.; KARLTON, P.; KOCHER, P., The SSL 3.0 Protocol, Netscape Communications Corp., 1996, disponível em http://wp.netscape.com/eng/ssl3/.

GUELFI, A. E.; MEYLAN, F.; FONSECA, R. D. R., Um Sistema de Autenticação Remota Baseado em Senha para Transações Eletrônicas Seguras, In: FORUM DE PRIVACIDADE, SEGURANÇA E CERTIFICAÇÃO DIGITAL, 2003, disponível em http://www.iti.br/twiki/bin/view/Forum/ArtigoE202.

HALLER, N.; METZ, C., RFC 1938 - A One-Time Password System, 1996, disponível em: http://www.ietf.org/rfc/rfc1938.txt.

HALEVI, S.; KRAWCZYK, H., Public-key Cryptography and Password Protocols, In: ACM TRANSACTIONS ON INFORMATION AND SYSTEM SECURITY (TISSEC), vol 2, no. 3, 1999, pp 230-268.

HOFFMAN, P., RFC 2487 - SMTP Service Extension for Secure SMTP over TLS, 1999.

HOUSLEY, R., RFC 3369 – Cryptographic Message Syntax (CMS), 2002, disponível em: http://www.ietf.org/rfc/rfc3369.txt.

ITU-T Recommendation X.690, Information Technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 2002.

ITU-T Recommendation X.509, Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks, 2000.

JABLON, D., Password Authentication Using Multiple Servers, In: LNCS 2020: TOPICS IN CRYPTOLOGY -- CT-RSA, 2001, pp. 344-360.

JABLON, D., Strong Password-Only Authenticated Key Exchange, Computer Communication Review, ACM SIGCOMM, vol. 26, no. 5, 1996, pp. 5-26.

JAIN, A.; BOLLE, R.; PAKANTI, S., Introduction to Biometrics, **BIOMETRICS Personal Identification in Networked Society**, Kluwer Academic Publishers, 2002, pp. 1-42.

KALISKI, B., RFC 2315 – PKCS #7: Cryptographic Message Syntax Version 1.5, 1998, disponível em: http://www.ietf.org/rfc/rfc2315.txt.

KALISKI, B., RFC 2898 – PKCS #5: Password-Based Cryptography Specification Version 2.0, 2000, disponível em: http://www.ietf.org/rfc/rfc2898.txt.

KOHL, J.; NEUMAN, C., RFC 1510 - The Kerberos Network Authentication Service (V5), 1993, disponível em: http://www.ietf.org/rfc/rfc1510.txt.

LEE, R.; ISRAEL, J., Understanding the Role of Identification and Authentication in NetWare 4, Novell Application Notes, 1994.

MENEZES, A.; VAN OORSCHOT, P.; VANSTONE, S., **Handbook of Applied Cryptography**, CRC Press, 1997.

MYERS, M. et al., RFC 2560 – X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, 1999, disponível em: http://www.ietf.org/rfc/rfc2560.txt.

NYSTROM, M.; KALISKI, B., RFC 2986 – PKCS #10: Certification Request Syntax Specification Version 1.7, 2000, disponível em: http://www.ietf.org/rfc/rfc2986.txt.

OATH - OPEN AUTHENTICATION, Open Authentication Reference Architecture, disponível em: http://www.openauthentication.org, consulta em: Janeiro 2006

PERLMAN, R.; KAUFMAN, C., Secure Password-Based Protocol for Downloading a Private Key, In: PROCEEDINGS OF THE 1999 NETWORK AND DISTRIBUTED SYSTEM SECURITY, 1999.

RANKL, W.; EFFING, W., **Smart Card Handbook**, 3rd Edition, John Wiley & Sons, 2003.

RATHA, N. K.; BOLLE, R., Smartcard Based Authentication, **BIOMETRICS Personal Identification in Networked Society**, Kluwer Academic Publishers, 2002, pp. 369-384.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. M., A method for obtaining digital signatures and public-key cryptosystems. In: COMMUNICATIONS OF THE ACM, 21(2), 1978, pp. 120-126.

SANDHU, R.; BELLARE, M.; GANESAN, R., Virtual Smartcards versus Virtual Soft Tokens, INFS 767: SECURE ELECTRONIC COMMERCE, 2002.

SCHAAD, J., RFC 4211 – Internet X.509 Public Key Infrastructure Certificate Request Message Format, 2005, disponível em: http://www.ietf.org/rfc/rfc4211.txt.

SMITH, R. E., **Authentication**: From Passwords to Public Keys, Boston: Addison-Wesley, 2002.

STEINER, M.; TSUDIK, G.; WAIDNER, M., Refinement and Extension of Encrypted Key Exchange, ACM Operating Systems Review 29 (3), 1995.

WU, T., The Secure Remote Password Protocol, In: PROCEEDINGS INTERNET SOCIETY NETWORK AND DISTRIBUTED SYSTEM SECURITY SYMPOSIUM, 1998.

WU, T., SRP-6: Improvements and Refinements to Secure Remote Password Protocol, Documento não publicado, 2002.

WU, T., The Stanford SRP Authentication Project, LibSRP 2.1.2 beta, 2005, disponível em: http://srp.stanford.edu/.

YLONEN, T.; LONVICK, C., RFC 4252 - The Secure Shell (SSH) Authentication Protocol, 2006, disponível em: http://www.ietf.org/rfc/rfc4252.txt.