UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DAP (DYNAMIC AUTHORIZATION PROTOCOL): UMA ABORDAGEM SEGURA OUT-OF-BAND PARA E-BANK COM UM SEGUNDO FATOR DE AUTENTICAÇÃO VISUAL

LAERTE PEOTTA DE MELO

ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JR.

COORIENTADOR: ANDERSON CLAYTON ALVES NASCIMENTO

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGEE.TD - 063 A/2012

BRASÍLIA/DF: 15 de AGOSTO de 2012.

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DAP (DYNAMIC AUTHORIZATION PROTOCOL): UMA ABORDAGEM SEGURA OUT-OF-BAND PARA E-BANK COM UM SEGUNDO FATOR DE AUTENTICAÇÃO VISUAL

LAERTE PEOTTA DE MELO

TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr., ENE/UNB
(ORIENTADOR)

GEORGES DANIEL AMVAME-NZE, Dr., FGA/UNB
(EXAMINADOR INTERNO)

FLÁVIO ELIAS GOMES DE DEUS, Dr., ENE/UNB
(EXAMINADOR INTERNO)

ROBSON DE OLIVEIRA ALBUQUERQUE, Dr., ABIN
(EXAMINADOR EXTERNO)

JOEL GUILHERME DA SILVA FILHO, Dr., IESB
(EXAMINADOR EXTERNO)

Brasília, 15 de agosto de 2012.

FICHA CATALOGRÁFICA

MELO, LAERTE PEOTTA DE
DAP (Dynamic Authorization Protocol): Uma Abordagem Segura Out-Of-Band Para E-Bank Com Um Segundo Fator De Autenticação Visual.
xv, 118 p., 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2012).
Tese de Doutorado - Universidade de Brasília.
Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Segurança 2. Autenticação
3. Autorização 4. Internet Banking

I. ENE/FT/UnB II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

Melo, L. P. (2012). DAP (Dynamic Authorization Protocol): Uma Abordagem Segura Out-Of-Band Para E-Bank Com Um Segundo Fator De Autenticação Visual. Tese de Doutorado em Engenharia Elétrica, Publicação PPGEE.TD - 063 A/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 118p.

CESSÃO DE DIREITOS

AUTOR: Laerte Peotta de Melo.

TÍTULO: DAP (Dynamic Authorization Protocol): Uma Abordagem Segura Out-Of-Band Para E-Bank Com Um Segundo Fator De Autenticação Visual.

GRAU / ANO: Doutor / 2012

Laerte Peotta de Melo
Sobradinho - 70.673-076 Brasília - DF - Brasil.

DEDICATÓRIA

À minha família

AGRADECIMENTOS

Gostaria de agradecer ao Prof. Rafael Timóteo, que acreditou em um projeto em que poucos se aventurariam, e soube, nos momentos certos, conduzir as pesquisas, direcionando para caminhos que pareciam incertos, mas que se mostraram as melhores escolhas: o meu sincero agradecimento.

Ao Prof. Anderson Nascimento, que com sua grande capacidade de transformar a informação em conhecimento, soube identificar as melhores respostas, sem as quais dificilmente chegaria neste resultado.

Ao verdadeiramente grande amigo Dino Amaral, um dos grandes incentivadores deste trabalho, que quando quase desisti, soube interceder e mostrar que poderia caminhar um pouco mais e chegar onde precisava chegar: meu eterno agradecimento.

A amiga Edna, que compartilhou os mesmos anseios e angústias de um doutorado, pela revisão inicial e sugestões.

Agradeço também aos colegas do Banco do Brasil, que tiveram o desafio de acreditar em mim, me ajudando a manter o entusiasmo e uma certa dose de obstinação.

Ao colega Josias (Joca), que com grande sabedoria conduziu discussões importantes, permitindo chegar a esse resultado.

Aos professores Flávio Elias, Joel Guilherme, Georges Amvame Nze e Robson Albuquerque pelas contribuições.

A todos do Departamento de Engenharia Elétrica, que me aturaram com meu "entusiasmo"em explicar minha pesquisa, e me ouviam simplesmente por acreditar e saber que isso era muito importante para mim.

Aos gestores da Diretoria de Gestão da Segurança do Banco do Brasil, instituição que apoiou, financiou e acreditou nesta pesquisa.

Laerte Peotta de Melo

RESUMO

DAP (DYNAMIC AUTHORIZATION PROTOCOL): UMA ABORDAGEM SEGURA OUT-OF-BAND PARA E-BANK COM UM SEGUNDO FATOR DE AUTENTICAÇÃO VISUAL

Autor: Laerte Peotta de Melo

Orientador: Rafael Timóteo de Sousa Jr.

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 15 de Agosto de 2012

A comodidade gerada pelos serviços eletrônicos torna seu uso cada vez mais demandado e sua segurança constantemente colocada à prova, seja pela disponibilidade das informações, da integridade dos documentos acessados e armazenados e da confidencialidade, fator preponderante na garantia do sigilo e privacidade. Com números e tendências de crescimento, os modelos de segurança não evoluíram juntamente com a tecnologia de acesso. O incremento da sofisticação das técnicas de ataques aos sistemas de seguranças, tem desafiado pesquisadores e organizações a produzir novas soluções, onde os usuários tenham liberdade de mobilidade e acesso, que essas soluções não protejam apenas alguns sistemas proprietários, ou que restrinjam o uso a determinados navegadores e suas versões, difíceis de proteger e de administrar, onde a identificação da origem da conexão ou modelos de hardwares e suas versões de softwares não sejam impedimentos ao acesso seguro e eficiente. Utilizando um telefone inteligente e confiável, é possível determinar um modelo seguro para a identificação e autenticação de um usuário. Associando um desafio de autorizar uma mensagem específica, através de um código único para cada mensagem. O usuário, pode então capturar essa mensagem e, através de um computador não confiável, responder ao desafio. A câmera do telefone do usuário é usada para decifrar o desafio através de um código visual (Ex. QR-Code). Neste trabalho é proposto um protocolo de autorização dinâmico que possua uma abordagem *out-of-band* e *offline* para acesso seguro a sistemas de *Internet Banking* com um segundo fator visual de autenticação.

ABSTRACT

DAP (DYNAMIC AUTHORIZATION PROTOCOL): SECURE APPROACH OUT-OF-BAND FOR E-BANK WITH A TWO FACTOR VISUAL AUTHENTICATION

Author: Laerte Peotta de Melo

Supervisor: Rafael Timóteo de Sousa Jr.

Programa de Pós-graduação em Engenharia Elétrica

Brasília, August 15, 2012

The convenience generated by electronic services its use becomes increasingly demanding and constantly put their security proof, is the availability of information on the integrity of documents stored and accessed and confidentiality, major factor in ensuring the confidentiality and privacy. With numbers and growth trends, the security models have not evolved along with technology access. The increasing sophistication of attack techniques to system security, has challenged researchers and organizations to produce new solutions, where users have the freedom of mobility and access, these solutions not only protecting some proprietary systems, or restricting the use of certain browsers and their versions, difficult to secure and manage, where the identification of the connection origin or hardware models and software versions are not impediments to access to safe and efficient. Using a true smartphone, you can determine a safe model for identifying and authenticating a user. Combining a challenge to authorize a specific message, using a unique code for each message. The user can then capture this message and, using a untrusted computer, answer the challenge. The user camera phone is used to decipher the challenge, capturing a visual code (QR-Code). The proposed authorization protocol having a dynamic approach *out-of-band* and *offline* to secure access to Internet Banking with a second factor visual authentication.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIAÇÕES

ABNT	Associação Brasileira de Normas Técnicas
AES	Advanced Encryption Standard
ATM	Automatic Teller Machine
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CNC	Computador Não Confiável
DNS	Domain Name System
HMAC	Hash-based Message Authentication Code
HSM	Hardware Security Modules
MAC	Message Authentication Code
MD5	Message-Digest algorithm 5
OTP	One-Time Password
QRCode	Quick Response
PIN	Personal Identification Number
SHA	Secure Hash Algorithm
TAN	Transaction Authentication Number

1 INTRODUÇÃO

A comodidade gerada pelos serviços eletrônicos torna seu uso cada vez mais demandado e exigido, seja pela disponibilidade constante das informações, da integridade dos documentos acessados e armazenados e da confidencialidade, fator preponderante na garantia do sigilo e privacidade.

Bancos *online*, comumente chamados de *Internet Banking*, ou mesmo *e-bank*, trazem a comodidade exigida pela sociedade da informação. Usuários dessa modalidade de serviço mudaram o conceito de localização, onde antigamente era necessário proximidade de uma entidade física, hoje essa necessidade é desconsiderada, pois um cliente de Banco pode efetuar suas transações financeiras em qualquer local, como se estivesse exatamente dentro da instituição financeira, não ficando restrito a horário ou local.

Os modelos de *Internet Banking* têm evoluído conjuntamente com a própria *Internet*, esse modelo surgiu na década de 1980, com diversas instituições financeiras na Europa e Estados Unidos criando experimentos do conceito de *home banking*, buscando facilitar o acesso a seus serviços utilizando equipamentos como fax e telefones, criando novas oportunidades que afetaram o comércio eletrônico e possivelmente alavancou o uso da própria Internet, antes restrita a um público bastante direcionado.

No Brasil o conceito *home banking* chegou praticamente junto com a *Internet* comercial. No início dos anos 90 já existiam Bancos que provinham alguns serviços através de redes dedicadas, mas que evoluiu rapidamente para o ambiente da web, no início apenas com transações de consulta.

Atualmente, 66,7% de todas as transações bancárias ocorrem por algum canal sem atendimento presencial, onde o *Internet Banking* responde por 34% do total de transações, isso corresponde a um incremento de 26,7% em relação ao ano de 2009 (BCB, 2010).

O número de Usuários vem crescendo nos últimos anos, entre 2009 e 2010 houve um acréscimo de 8%, perfazendo um total de 38 milhões de Usuários (FEBRABAN, 2007) (BCB, 2010), o que demonstra que o crescimento superou novas contas. Nesse mesmo cenário um destaque no crescimento aponta o uso de tecnologias móveis, demonstrando

a importância do segmento, com um crescimento de 72% em relação a 2009, passando de 2,2 milhões de Usuários.

Com números impressionantes de movimentação, em 2010 10,57 bilhões de transações foram executadas pelos sistemas de *Internet Banking*, perfazendo um total de R$ 9,8 bilhões movimentados, com uma média de 253 transações para cada Usuário por ano (BCB, 2010) (BCB, 2009).

Sistemas de serviços bancários, que utilizam o canal *Internet*, são considerados métodos convenientes da adaptação de um sistema dependente de infraestrutura de atendimento físico. Atualmente, é uma ferramenta que tem como objetivo aumentar a eficiência do sistema bancário, pois leva à casa dos clientes o próprio Banco, gerando economia de tempo, tanto do Banco como do cliente. O modelo permite encurtar as longas filas dos Bancos bem como agilizar muitas operações executadas.

Mesmo com números e tendências de crescimento de utilização desse canal, os modelos de segurança não evoluíram juntamente com a tecnologia de acesso. Neste trabalho são apresentados os modelos adotados atualmente, que mostram claramente, a utilização de processos para identificação de um Usuário e de seu equipamento de acesso. Esses modelos são fortemente associados à capacidade de seu Usuário em resistir em fornecer informações necessárias para que sejam identificados, informações essas que são utilizadas pelo atacante para se passar por um Usuário legítimo e evitar que seja detectado através de atividades não autorizadas. Infelizmente este cenário não é considerado o melhor, pois diversos ataques atuais utilizam-se desse fator para explorar uma vulnerabilidade. Da mesma forma, produzir modelos que permitam o acesso apenas de equipamentos conhecidos ou previamente cadastrados torna a manutenção do sistema difícil e restringe o acesso do Usuário legítimo. Portanto, fornecer uma informação a um Usuário e exigir como desafio que essa informação seja confirmada expõe todo o processo e possibilita uma série de vulnerabilidades intrínsecas ao próprio sistema de acesso, onde acaba por permitir que um adversário seja capaz de se passar por um Usuário legítimo, solicitando e confirmando mensagens de transação.

A segurança dos sistemas bancários deve ser considerada como parte integrante e de grande importância sob a visão da segurança nacional. Portanto, considera-se que para proteção da soberania de um país, o sistema financeiro, faz parte da infraestrutura considerada crítica, e que as fraudes nesses serviços podem afetar a economia direta do país. Criminosos se valem de diversas fragilidades dos modelos atuais, não só direta-

mente associados ao serviço, mas pela falta de uma legislação tipificada, que faça coibir a prática desses delitos.

Diversas instituições financeiras têm buscado alternativas para mitigar os riscos do canal, aumentando a produtividade e massificando o uso. Nessas estratégias incluem-se os desafios relacionados à autenticação do Usuário. Entretanto, a maneira como se enfrenta esses desafios é considerada um problema, pois praticamente todas as instituições possuem estratégias de detecção e prevenção de fraudes, que mostra a aparente pouca efetividade desses sistemas de autenticação. Modelos emergentes e promissores devem, necessariamente, ser imunes aos ataques atuais, o que seria um contrassenso se não o fossem. Entretanto, percebe-se que, ainda assim, tanto a indústria de segurança, quanto as instituições financeiras, continuam a utilizar modelos que atingem parcialmente as soluções desses problemas.

Um paradigma é necessário para que sejam criados novos modelos de acesso e que esses modelos sejam considerados seguros. Identificar e autenticar um Usuário pode simplesmente fragilizar o sistema, pois coloca todos os requisitos de segurança e autenticidade nesses processos, da mesma forma, o desafio de comprovação de identidade deve ser feito no momento que este se fizer necessário, permitindo que um fator de autenticação seja utilizado em conjunto com outros fatores, preparando os futuros modelos, onde a autenticação é uma premissa básica, necessária e está diretamente associada à identificação.

Considera-se que autenticação com fatores simples (*single-factor*), ou únicos, são inadequados e geram altos riscos, muitas vezes inaceitáveis, quando envolve o acesso e transações de Usuários. Essas técnicas de autenticação e identificação de clientes, possuem riscos associados aos produtos oferecidos. A fragilidade desse modelos frequentemente têm como resultado a exploração do acesso indevido, o que leva a fraudes e prejuízos, tanto financeiro quanto da imagem da instituição, que pode ser abalada caso se mostre negligente em proteger seus Usuários. Nos casos de utilização de múltiplos fatores (*two-factor*), é preciso identificar todas as transações e seus respectivos níveis de acesso associados a produtos e serviços. Identificando e avaliando os riscos é possível determinar qual a melhor metodologia de autenticação para cada nível de acesso.

É necessário que os riscos sejam avaliados e classificados, considerando-se as variáveis diretamente afetadas (FFIEC, 2005). Devem ser indicadas ações que busquem mitigar os riscos, incluindo-se métodos de autenticação fortes e resistentes a ataques atualmente

conhecidos ou que possam ser previstos. É preciso juntar às ações adotadas, conscientizar os Usuários, sem que sejam responsabilizados pela segurança. A estratégia de responsabilização atribuída ao Usuário pode ser desastrosa, possibilitando que o sistema caia em descrédito por outros Usuários, causando uma migração do canal para outros meios de acesso, que em alguns casos, podem não comportar essa movimentação.

O incremento da sofisticação das técnicas de ataques aos sistema de segurança, que são implementados pelas instituições financeiras, tem desafiado pesquisadores e organizações a produzir esses paradigmas de novas soluções. Nesse cenário surge, como grande desafio, que os Usuários tenham liberdade de mobilidade e de acesso, que esses sistemas não protejam apenas alguns sistemas proprietários, ou que restrinjam o uso a determinados navegadores e suas versões, difíceis de proteger e de controlar, onde a identificação da origem da conexão ou modelos de hardwares e suas versões de softwares não sejam bloqueadores para o acesso seguro e eficiente. É nesse cenário que os dispositivos móveis se fixaram e mostram-se promissores e com tendências reais e de curto prazo (TUBIN, 2011). Somente entre os anos 2009 e 2010 o incremento da utilização de dispositivos móveis, para acesso a *Internet Banking*, no mercado brasileiro cresceu 73% (BCB, 2010).

1.1 HIPÓTESE

Diante desse paradigma, é possível determinar um modelo seguro, que utilize não apenas a autenticação e identificação de um Usuário, mas o desafio de autenticar uma mensagem ou transação, de maneira *Out-of-Band* e visual, onde deve ser possível que o Usuário visualize o que está autorizando, e que essa visualização seja feita utilizando um segundo fator externo ao acesso, sem custo de comunicação. Neste modelo seguro é possível que, tanto a origem quando destino da informação, possam ser confirmados através de autenticação mútua, e que todas as mensagens sejam garantidas contra modificações e seja possível detectar falta de integridade nas mensagens recebidas e enviadas.

1.2 MOTIVAÇÃO E PROBLEMA

O crescente número de Usuários, bem como os elevados valores de perdas no canal de serviço *Internet Banking*, fazem perceber a necessidade de se elaborar um estudo profundo em busca de definir as melhores abordagens e práticas a fim de prover segurança neste meio. Os modelos atualmente em uso mostram-se ineficientes, pois se utilizam de

técnicas produzidas não diretamente pensadas para resolver o problema, que basicamente trata de encontrar uma maneira de identificar e autorizar Usuários e transações eletrônicas. Neste cenário, o desenvolvimento de solução segura e eficiente, é considerado um tema atual e motivacional para quebra de novos paradigmas em encontrar e propor inovação para o sistema bancário, o que não deve ser restringido apenas neste cenário, preparando novos pensamentos e soluções para identificação e autorização de mensagens.

1.3 OBJETIVO GERAL

Propor e avaliar um protocolo de autorização de transações em um modelo de acesso seguro *out-of-band* a serviços de *Internet Banking*, utilizando conceitos de primitivas criptográficas e fatores externos de autenticação forte.

1.4 OBJETIVOS ESPECÍFICOS

Diante do objetivo principal proposto, definiu-se objetivos específicos que foram atingidos juntos com esta pesquisa, os quais se apresentam:

- Minimizar efeitos de ataques de engenharia social, introduzindo informações necessárias para o acesso e autorização de mensagens, sem a necessidade de conhecimento por parte de um Usuário legítimo;
- Reduzir dependência de sistemas específicos para acesso;
- Reduzir efeitos de ameaças atualmente conhecidas;
- Elaborar prova de conceito em um estudo de caso real, permitindo medir a eficiência dos objetivos descritos.

1.5 JUSTIFICATIVAS

Modelos atuais utilizados pelas instituições financeiras geram grande trabalho de identificação de transações e mensagens não legítimas. Forçando seu Usuário a utilizar somente sistemas conhecidos e previamente identificados. Da mesma forma obriga

as instituições a manter um sistema de monitoração ininterruptamente, produzindo elevados custos de infraestrutura e pessoas.

Os desafios criados para identificar o Usuário legítimo podem ser capturados e utilizados em transações ou mensagens ilegítimas, tornando os modelos pouco eficiente aos ataques conhecidos, pois deposita a responsabilidade de utilização segura ao Usuário.

Com a adoção de um modelo que empregue um desafio externo ao canal utilizado para acesso, pode-se servir de um mecanismo mais fraco de autenticação, onde, a princípio, a informação fornecida é considerada confiável. Entretanto, no momento de emitir uma mensagem que necessite de um grau maior de segurança outro desafio deve ser lançado, de modo que independa do canal principal, e que seja possível provar a posse de um segredo associado diretamente à mensagem solicitada.

1.6 CONTRIBUIÇÕES

As contribuições relacionadas com este trabalho são as seguintes:

- Estudo e apresentação detalhada do estado da arte em sistemas de seguração para *Internet Banking*;
- Apresentação dos modelos de ameaças e ataques inseridos no ambiente das instituições financeiras e seus usuários;
- Proposta de um modelo analítico para medição de eficiência de um sistema de seguraça para *Internet Banking*;
- Apresentação de um novo protocolo *out-of-band* que busca resolver problemas conhecidos;
- Desenvolvimento de um modelo de teste implementado em um sistema real.

Ressalta-se a produção de artigo em revista em que se apresentou os modelos atuais e suas deficiências (PEOTTA et al., 2011a), sendo a base para o desenvolvimento do protocolo DAP (Dynamic Authorization Protocol): Uma Abordagem Segura Out-Of-Band Para e-Bank Utilizando Segundo Fator De Autenticação Visual, onde foi produzida patente de invenção devidamente registrada (PEOTTA et al., 2011b). Como complemento das pesquisas foi produzido um capítulo de livro (PEOTTA et al., 2011c)

em que se explorou análises de artefatos maliciosos e os ataques mais comuns utilizados para comprometimento dos Usuários de *Internet Banking*. Outros trabalhos produzidos foram focados para análise de segurança de dispositivos móveis (MORUM et al., 2011b) (MORUM et al., 2011a) e gerência de identidades seguras (FEITOSA; SANTIAGO; PEOTTA, 2011).

Este trabalho foi totalmente implementado no Banco do Brasil [1], e conta, até a data de 15/08/2012, com mais de 55 mil Usuários ativos na solução. Diversas reportagens foram produzidas, descrevendo a solução, dos quais pode-se citar algumas como: jornais impressos (OSWALD, 2012), (OLIVEIRA, 2012), revistas (VICENTE, 2012), mídia eletrônica (ADVILLAGE, 2012), (FERRER, 2012), (TI-INSIDE, 2012), (PAYAO, 2012) e um vídeo institucional (Banco do Brasil, 2012).

1.7 ESTRUTURA DA TESE

Esta tese está organizada da seguinte maneira: No capítulo 2 são abordados os principais conceitos e definições referentes ao tema desta pesquisa. Uma descrição formal de um sistema de *Internet Banking* é apresentada seguida de um estudo sobre as principais vulnerabilidades, classificação dos ataques e modelos de classificação de ameaças que são discutidos no capítulo 3, também são apresentados os modelos atuais em utilização pelos Bancos comerciais no Brasil, juntamente a um estudo comparativo de soluções e sistemas de autenticação fortes seguido pelo modelo adversarial, onde é apresentada uma matriz de avaliação desses modelos.

A proposta de desenvolvimento deste trabalho é apresentado no capítulo 4 que discute e descreve um novo modelo para ser utilizado em sistemas de *Internet Banking*. Este modelo é composto por um conjunto de protocolos criptográficos que busca garantir integridade, autenticidade e sigilo das transações que ocorrem por meio deste canal, utilizando, para isso, um fator externo e visual. Por fim, no capítulo 5 é feita a conclusão apresentando os principais desafios e resultados deste trabalho através da exploração dos principais fatores estudados e desenvolvidos, apresentando premissas para trabalhos esperados e futuros.

[1] `http://www.bb.com.br/bbcode`

2 CONCEITOS E FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos relacionados a este trabalho. São abordados os principais fatores de respostas a incidentes, grupos de respostas a incidentes de segurança (CSIRT), suas etapas de trabalho e metodologias de análise, bem como a condução de incidentes, composta por atividades de coleta, extração, tratamento e resposta.

2.1 SISTEMAS DE INTERNET BANKING

Sistemas de *Internet Banking* possuem diversos componentes interagindo entre si. De maneira geral, o modelo apresentado pela Figura 2.1, descreve um sistema típico.

Através do estudo das bibliografias, não se encontrou uma descrição formal para detalhar um sistema de *Internet Banking*, por razões muitas vezes discutidas sob o ponto de vista do interesse das instituições financeiras, que não encontram canais de discussão menos competitivas do que suas atividades.

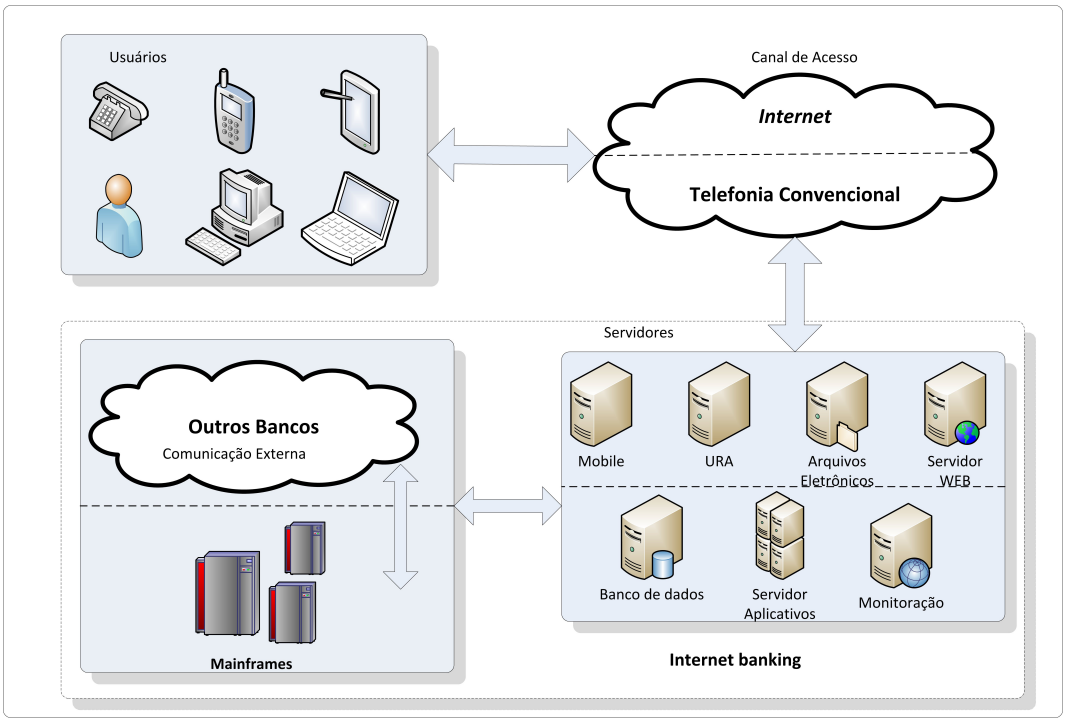

Figura 2.1: Diagrama descritivo de Internet Banking

Os Usuários devem ter a liberdade de escolher qual plataforma será utilizada para o acesso aos serviços, sendo que o canal pode ser determinado por essa escolha, comumente o acesso se dá pela rede de telefonia convencional, inclusive a rede celular. O serviço é determinado também pela plataforma de acesso, onde cada serviço está disponível e interage com o sistema como um todo.

Serviços móveis possuem as características de serem mais compactos e ter uma capacidade de transação menor que um sistema Web. O mesmo ocorre com o serviço Unidade de Resposta Audível (URA), onde o Usuário, utilizando o sistema de telefonia convencional, acessa suas informações e é capaz de emitir e receber mensagens através de suas interações diretas por voz e teclado de um aparelho telefônico comum ou mesmo um dispositivo celular.

Empresas e outras organizações necessitam, muitas vezes, de efetuar mais interações e transações que um Usuário comum, nesses casos a demanda e o modelo atual pode tornar o processo lento, pois o tempo para cada transação pode ultrapassar o limite considerado ideal. Nessa situação, o envio de lotes de transações através de arquivos eletrônicos é uma boa solução, torna o processo rápido e confiável.

Um serviço muito demandado é a monitoração, onde todas as transações e mensagens recebidas e enviadas, são verificadas a fim de se identificar eventuais inconsistências devido à utilização fraudulenta. Esse processo além de demorado é trabalhoso, pois a quantidade cada vez maior de transações dificulta as análises, tornando um modelo considerado fraco.

Uma instituição financeira deve ser capaz de se comunicar com outras, de modo a transmitir e receber mensagens em tempo real, possibilitando que Usuários possam enviar e receber transações independente de seu Banco, garantindo a integridade das informações trocadas, bem como a confidencialidade das mensagens, sendo assim um modelo integrado e completo, tanto de acesso às informações privadas de cada Usuário como seguro.

2.1.1 Canal de Comunicação

Para que seja possível traçar estratégias de combate à fraudes, é necessário identificar qual canal de comunicação é utilizado pelos Usuários dos sistemas de *Internet Banking* para acessar suas informações. A Figura 2.2 apresenta informações de utilização do canal de acesso.

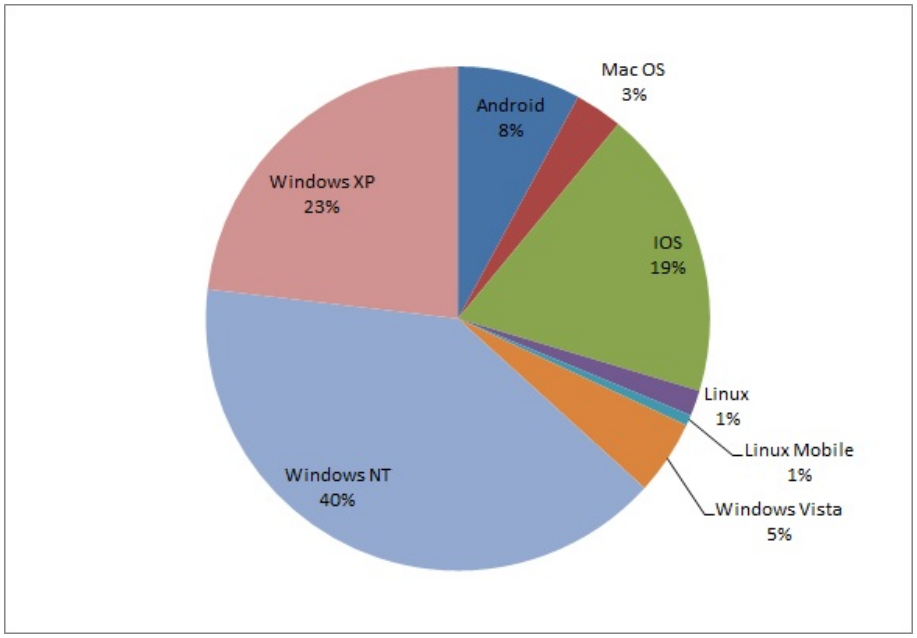

Figura 2.2: Análise de Utilização dos Sistemas Operacionais Como Meio de Acesso

Utilizando uma base privada com 1.3 milhão de eventos distintos obtidos na observação de um servidor real de um Banco nacional, foi possível identificar os tipos de sistemas operacionais mais utilizados pelos Usuários no acesso.

A grande utilização de sistemas proprietários, como Windows XP, Windows Vista e Win NT (que contempla qualquer sistema operacional que não seja XP ou Vista) demonstra que no caso de um sistema de proteção para os Usuários, estes devem ser priorizados, pois juntos representam 68% de todos os acessos.

Sistemas operacionais para dispositivos móveis, como *smartphones* e *tablets* representam 28% dos acessos, demonstrando que existe grande perspectiva de crescimento rápido neste segmento. Esse tipo de acesso tem crescido consideravelmente nos últimos anos. Uma série de hipóteses corroboram com esses dados, sejam elas incentivos fiscais, barateamento de tecnologias e maior acesso devido à população em geral.

Outros sistemas, apesar de possuírem menor expressão, não devem ser esquecidos, pois quantitativamente representam um número considerável de acessos, e que podem se tornar vetores caso sejam negligenciados pelas instituições financeiras, pois a taxonomia dos ataques indica que ao se proteger um segmento ocorre migração para ambientes menos protegidos.

A fim de poder delimitar a estratégia de defesa, através de proteção de navegadores,

observou-se nas amostras que existe uma grande quantidade de versões de navegadores utilizados pelos Usuários ao acessar o canal de *Internet Banking*. A Figura 2.3 esclarece através do relacionamento com o tipo de sistema operacional.

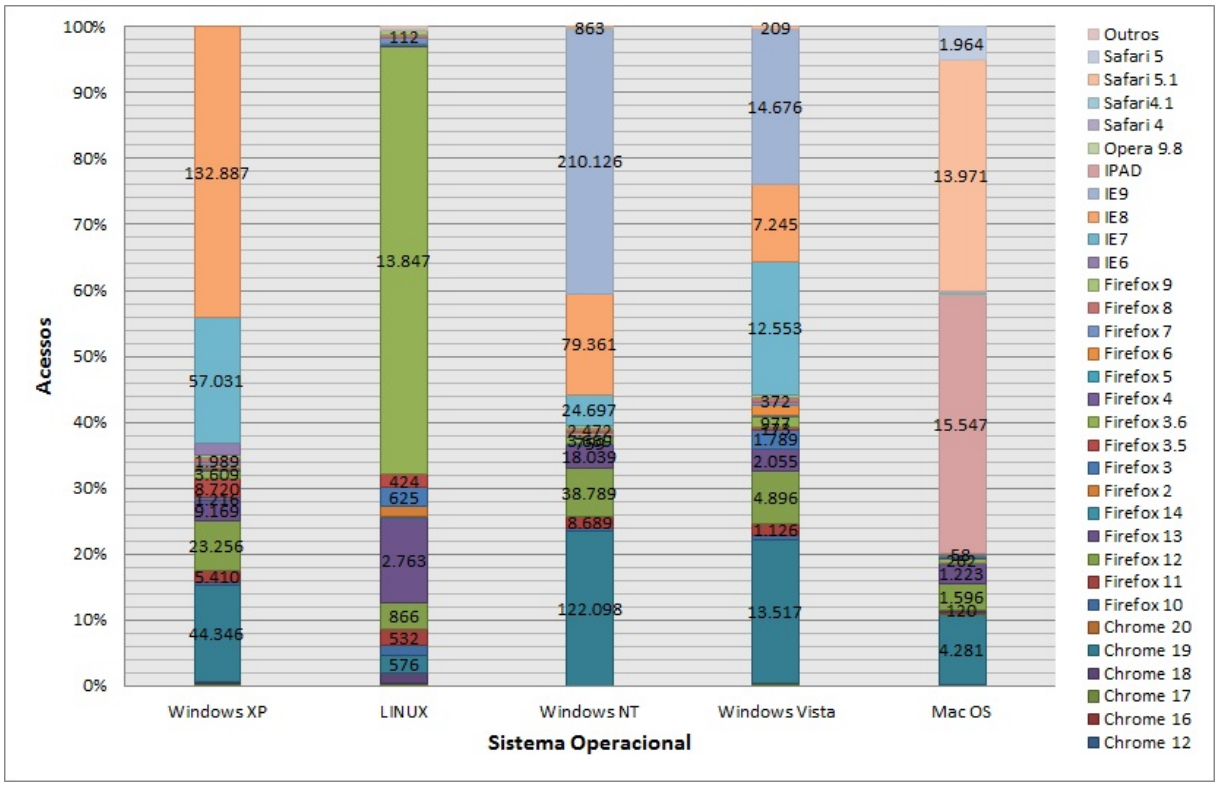

Figura 2.3: Tipos de navegadores utilizados - Análise por versão

Essa relação é necessária, pois as ferramentas de proteção desenvolvidas geralmente dependem mais do sistema operacional em que o acesso é feito, do que propriamente do tipo de navegador utilizado. Também é importante entender que mesmo os sistemas e navegadores mais utilizados, não necessariamente dizem respeito ao perfil mais adequado do *Usuário*. Isso é explicado pois alguns Bancos restringem os acessos somente através de sistemas homologados e protegidos pelas ferramentas de segurança desenvolvidas por essas empresas.

Os navegadores variam praticamente em todas as versões conhecidas, fazendo com que seja necessário um grande esforço de padronização por parte das instituições financeiras, que obrigatoriamente necessitam desenvolver seus aplicativos com a maior compatibilidade possível, sem que os mesmos deixem de funcionar para outras plataformas.

De maneira a ficar claro qual o tipo de navegador mais utilizado, e portanto, mais

adequado a ser protegido. Na Figura 2.4 é apresentada as informações extraídas da mesma base descrita nesta seção.

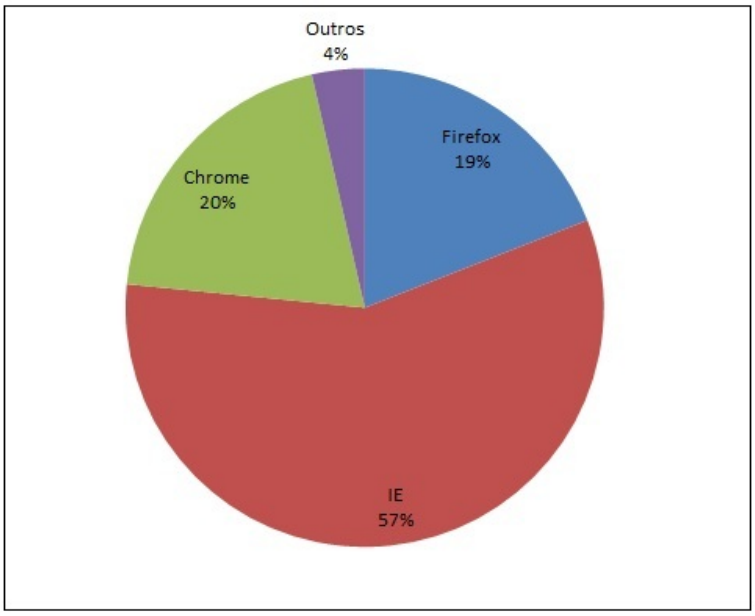

Figura 2.4: Tipos de Navegadores - Total

Navegadores como IE[2] são os mais utilizados para o acesso aos sistemas de *Internet Banking*, conforme apresentado na Figura 2.4. Uma série de fatores podem ser postulados a fim de se compreender essa vantagem. Bancos geralmente se dedicam a proteger sistemas mais populares, de forma que seus clientes estejam mais seguros. Entretanto, existe uma grande capacidade de crescimento para outros navegadores, de modo que colocar todo um modelo de segurança baseado em tipo de navegador, além de ser difícil, torna o processo bastante dependente, ao mesmo tempo em que obriga os *Usuários* a utilizarem apenas o que é compatível.

Os atacantes sabem disso, e procuram explorar essa característica, forçando com que o *Usuário* legítimo não consiga acessar as informações corretamente, direcionando a utilização de navegadores menos protegidos e consequentemente conseguindo burlar as técnicas de proteção adotadas pelas instituições financeiras.

2.2 AMEAÇAS, VULNERABILIDADES E ATAQUES

Nesta seção são apresentadas as definições de ameaças, vulnerabilidades e ataques, incluindo artefatos como vírus, *worm*, *trojan horse* e as principais ameaças, como engenharia social, *spam*, *honeynet*, *phishing* (STEDING-JESSEN, 2008), *botnets*, negação

[2]Microsoft Internet Explorer

de serviço (BINSALLEEH et al., 2009) e exploração de vulnerabilidades (ABBAS; SADDIK; MIRI, 2006).

2.2.1 Incidentes de Segurança

Um incidente de segurança é um evento indesejado ou inesperado, que pode ser único ou em série e que possui características de comprometimento do bom funcionamento de uma organização, resultante de uma ameaça à segurança da informação(ABNT-B, 2005). Especificamente, um incidente pode ser relacionado a eventos como invasões, acessos não autorizados, negação de serviço, contaminação de sistemas através de *malwares*, furto de informações ou qualquer atividade considerada ilegal ou ilegítima. Nesses casos, é necessário que os eventos sejam investigados por pessoas qualificadas a fim de resolverem o problema causado.

Programa malicioso, código malicioso, softwares malicioso, ou simplesmente *malware*, são expressões que se referem a um programa que é inserido em um sistema, normalmente de forma encoberta, com a intenção de comprometer a confidencialidade, integridade ou disponibilidade dos dados da vítima, aplicativos ou o próprio sistema operacional. A produção de *malwares* se tornou a ameaça mais significativa para a maioria dos sistemas, causando danos generalizados e perturbação, ocorrências que necessitam de esforços de recuperação extensa dentro da maioria das organizações.

As técnicas utilizadas pelos *malwares* são associados à violação de privacidade de um Usuário, valendo observar que o denominado *spyware* tornou-se uma grande preocupação para as organizações. Embora a violação da privacidade através desses programas não seja novidade, tornou-se muito mais difundida recentemente (MELL; KENT; NUSBAUM, 2005). A monitoração das atividades pessoais e a realização de fraudes financeiras são ocorrências cada vez mais comuns, assim como a ocorrência de sistemas invadidos. As organizações também enfrentam outras ameaças que são frequentemente associadas aos *malwares*. Uma das formas que tem se tornado comum é o *phishing*. Outras formas de artefatos, como vírus, *backdoors*, cavalos de tróia, *keyloggers*, *rootkits*, também se denominam pelo termo *malware* e serão abordadas nas seções seguintes.

2.2.2 Spam e Phishing

O termo *spam* é utilizado para referenciar o recebimento de uma mensagem não solicitada, que geralmente tem o caráter de fazer propaganda de algum produto ou assunto

não desejado. O termo em inglês utilizado para esse tipo de mensagem é *UCE (Unsolicited Commercial Email)*. A palavra *spammer* é o termo utilizado para definir quem envia esse tipo de mensagem, em cuja elaboração geralmente são utilizados softwares especiais automatizados para coletar bases de e-mails através de listas de discussão, caixas postais de grupos, ou exploração, através de listas adquiridas, por empresas especializadas nesse tipo de *marketing*. Existem diversas soluções no mercado que buscam conter o recebimento de *spams*, ferramentas estas que basicamente utilizam filtros contendo *blacklists* de *proxies* e *relays* abertos, que são utilizados para o envio das mensagens.

A Figura 2.5 confirma que é crescente o envio de mensagens não solicitadas [3].

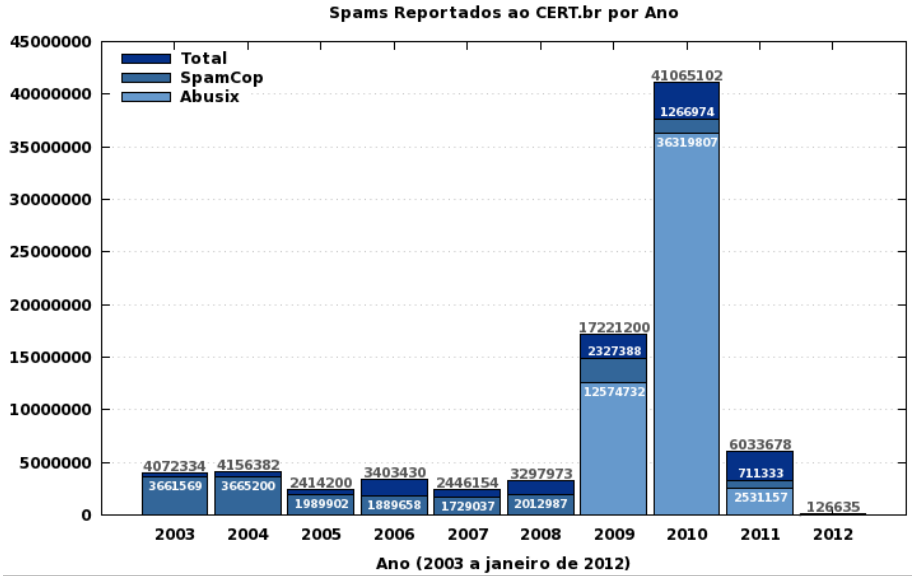

Figura 2.5: Valores acumulados: 2003 a janeiro de 2012(CERT, 2012))

Servidores de *proxies* são utilizados por organizações de maneira a tornar suas estruturas de comunicação mais seguras, pois permitem prover acesso de redes internas a redes externas, como a *Internet*, sendo capazes de impor controles de acesso a conteúdos definidos em uma política de segurança e gerar informações de auditoria. Os servidores *relay* são utilizados conjuntamente com o protocolo SMTP *(Simple Mail Transfer Protocol)* de transmissão e recebimento de mensagens, protocolo este que é considerado inseguro, pois, caso um adversário tenha acesso ao meio de comunicação e capture o tráfego gerado entre cliente e servidor, as credenciais de cada um não são protegidas por criptografia. Apesar de existir serviços SMTP que utilizam criptografia, a utilização é baixa. Por tais razões, servidores SMTP que estejam com o serviço de *relay*

[3] `http://www.cert.br/stats/spam/`

ativado podem permitir que os serviços sejam utilizados por *spammers* e que as mensagens sejam enviadas. Diante de tal situação, recomenda-se que os servidores sejam configurados utilizando técnicas de *hardening*, de modo a permitir que somente pessoas autenticadas e autorizadas possam enviar mensagens, restringindo o uso por terceiros.

O termo *phishing*, *phishing scam* ou *phishing/scam* ("fishing"significa pescaria), refere-se ao método pelo qual *"iscas"* (e-mails) são usadas para *"pescar"* informações de Usuários da Internet, constituindo-se um ataque que tem como objetivo efetuar algo ilícito através do envio de mensagem não solicitada. Essa técnica geralmente é associada à engenharia social, atividade em que o *spammer* explora algum tema de modo a induzir o Usuário final a executar alguma atividade ou ação que não faria normalmente. Essas mensagens são projetadas para roubar informações como: dados pessoais e credenciais de clientes de instituições financeiras.

A Figura 2.6 apresenta um exemplo desse tipo de mensagem. Todo o processo de envio e recebimento é passível de auditoria, permitindo que se compreenda a taxonomia do evento que um *spammer* utiliza.

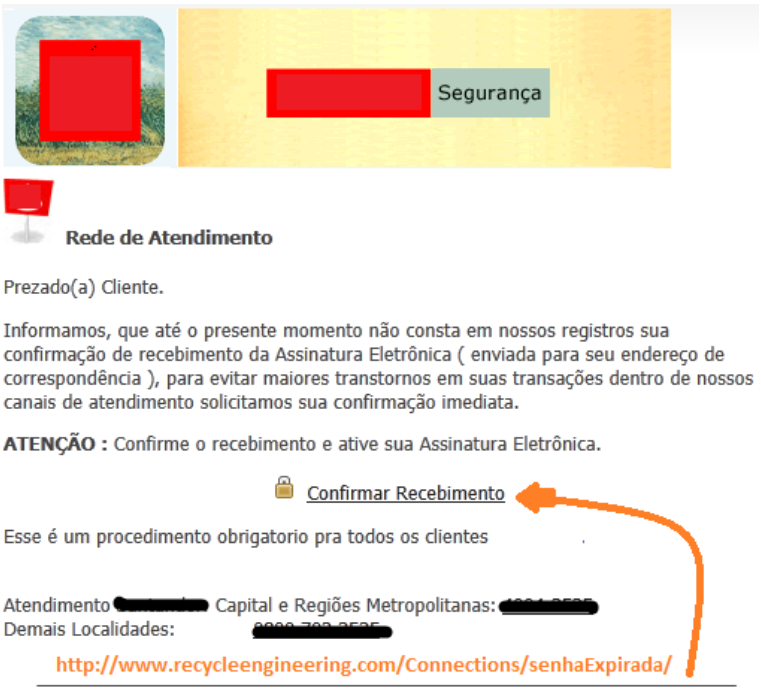

Figura 2.6: Exemplo de phishing

Em certas ocasiões, o *spammer* não se preocupa em ocultar sua origem, tornando o rastreamento rápido. Entretanto, na maioria dos casos a origem é ocultada ou alterada de maneira a indicar outro local de envio da mensagem.

O método de roubo pode ser associado a um formulário web, onde as informações são solicitadas à vítima, outro método é conhecido como cavalo de tróia, em que a vítima instala o artefato de maneira desavisada. Os cavalos de tróia são tratados na seção 2.2.5. Alguns exemplos de situações que envolvem esse tipo de ataque:

- Formulários recebidos por e-mails que solicitam informações confidenciais;
- Links recebidos que direcionam para a instalação de cavalos de tróia;
- Alteração através de malwares que direcionam o tráfego DNS para um *proxy* controlado pelo adversário;
- Páginas falsificadas de empresas de comércio eletrônico, sites de leilões, instituições financeiras, órgãos do governo.

Praticamente os ataques que, utilizam técnicas de engenharia social, tem como vetor de comprometimento o *phishing*. *Spammers* precisam convencer o Usuário a executar ações que tem como objetivo comprometer seu dispositivo, portanto, esse tipo de ataque é fortemente dependente da colaboração da vítima.

Ganhar a confiança do Usuário tem sido um desafio para os adversários, que buscam camuflar suas investidas tornando-se parecidos com sites de serviços mais comumente utilizados, como Bancos, redes sociais, servidores de e-mail, comércio eletrônico. Como exemplo pode-se observar a Figura 2.7, onde o adversário tenta reproduzir, de maneira fiel, o *site* de uma instituição financeira. Entretanto, nesse tipo de ataque o ponto a ser explorado é a distração da vítima, que não percebe as modificações, muitas vezes grosseiras e que solicita de maneira direta o que o adversário precisa para se passar por um cliente legítimo da instituição.

Os programas maliciosos podem ser divididos primordialmente através de sua dependência ou não do hospedeiro. Os tipos dependentes são por exemplo: vírus, bombas lógicas e *backdoors*, os não dependentes *worms* e zumbis. A replicação é outra maneira de se classificar, pois existem os programas que possuem a capacidade de se replicar, como vírus e *worms*, e os que não se replicam, como bombas lógicas, *backdoors*, zumbis, *rootkits* e *keylogger*. Entretanto, é bastante comum os *malwares* que possuem capacidades híbridas, sendo capazes de se replicar e ao mesmo tempo possuírem outras características inerente às necessidades. Para que a classificação de cada tipo seja

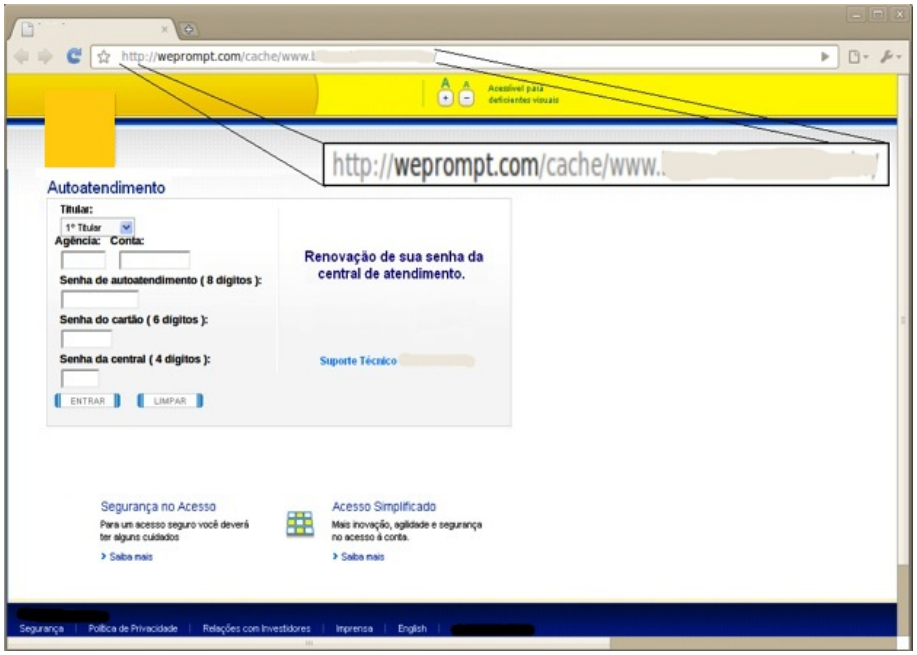

Figura 2.7: Exemplo de página falsa

feita, é necessário que os termos utilizados sejam de conhecimento do analista que irá conduzir a investigação, portanto, apresenta-se uma breve descrição de cada tipo mais comum de malware.

2.2.3 *Honeynets* e *Honeypots*

Honeypots são recursos e serviços computacionais em rede criados e monitorados com o objetivo de serem atacados e comprometidos, visando identificar novas ameaças e ferramentas utilizadas. *Honeynet* são conjuntos de sensores monitorados de maneira a identificar tráfego suspeito e malicioso.

Segundo (STEDING-JESSEN, 2008), um sistema de *honeypots* pode ser dividido em duas categorias:

- Alta Interatividade: Todos os sistemas e serviços disponibilizados são serviços reais sem inserções de vulnerabilidades propositadas. Nesse ambiente todo o tráfego é monitorado e controlado, de maneira que um comprometimento não seja utilizado para causar danos a outras redes;
- Baixa Interatividade: Os sistemas são reais, entretanto, os serviços disponibilizados são emulados através de ferramentas criadas exclusivamente para reproduzir

o comportamento esperado desses serviços. Oferecem um grau menor de risco de serem utilizados como base a ataques a outras redes.

O objetivo dos *honeypots*, independentemente de sua categoria, é coletar informações de ataques, identificando origens e destinos, bem como serviços mais procurados. A monitoração é feita geralmente utilizando-se sistemas de detecção de intrusão[4] e armazenamento e co-relacionamento de *logs*.

2.2.4 Vírus

Os vírus de computador foram introduzidos na década de 1980, com funções simples que ocasionalmente geravam inconvenientes ou apenas mostravam informações ao Usuário. Atualmente, esse tipo de código traz um risco significativo com potencial destrutivo e que demandam grande esforço das organizações para manterem seus sistemas a salvo. A designação de vírus é tratada de acordo com sua função. Para que um código malicioso seja considerado um vírus ele deve ter a capacidade de auto replicação, ou seja, fazer uma cópia de si mesmo e distribuir essa cópia para outros arquivos e programas do sistema infectado.

Cada vírus pode utilizar um mecanismo de infecção diferente, por exemplo: inserir seu código em programas executáveis sobrescrevendo parte do código de maneira a permitir que toda vez que o programa infectado seja executado o código viral também seja executado. Portanto, o principal objetivo de um vírus é replicar-se e contaminar o maior número possível de programas, de maneira a comprometer outros sistemas.

Técnicas de ofuscação podem ser utilizadas no código de vírus, particularmente para dificultar sua detecção. As técnicas mais comuns são:

- Criptografia: O código possui funções para cifrar e decifrar o executável do vírus, gerando porções de códigos aleatórios para cada infecção;
- Polimorfismo: Um modelo mais robusto que usa criptografia, com o código gerando um executável com tamanho e aparência diferentes para cada infecção, podem também habilitar funcionalidades diferentes na ação;

[4]IDS - Intrusion Detection System

- Metamorfismo: Assim como os polimórficos mudam o comportamento, podendo gerar códigos novos a cada iteração através de técnicas de códigos desnecessários, sendo recompilado e criando um novo executável.

Pode-se dividir os vírus em duas grandes categorias (MELL; KENT; NUSBAUM, 2005): vírus compilados e vírus interpretados.

2.2.4.1 Vírus Compilados

Esse tipo de código malicioso é composto de um programa fonte que é compilado com alguma linguagem e direcionado para um determinado sistema operacional, tipicamente esse tipo de vírus pode ser dividido em três categorias:

1. Infector de programas: Infectam programas executáveis, onde o programa infectado se propaga para infectar outros programas no sistema, bem como outros sistemas que usam um programa infectado compartilhado;
2. Setor de Boot: Infectam o setor *Master Boot Record* (MBR) do disco rígido ou mídia removível inicializável. Esse tipo de vírus foi muito utilizado nas décadas 1980 e 1990, atualmente está obsoleto;
3. Multipartite: Utiliza métodos de infecção múltipla, normalmente infectando arquivos e setores de boot. Assim, os vírus multipartite combinam as características dos vírus que infectam arquivos e setor de iniciação.

2.2.4.2 Vírus Interpretados

Esse tipo de vírus é composto de código-fonte que pode ser executado apenas por uma determinada aplicação ou serviço, ao contrário dos vírus compilado, que pode ser executada por um sistema operacional. Tornaram-se mais comuns, pois considerados mais fáceis de escrever e modificar do que outros tipos de vírus. Não é necessário desenvolver capacidades técnicas elevadas, pois um adversário pode adquirir um vírus interpretado e modificar seu código fonte. Muitas vezes há dezenas de variantes de um único vírus interpretado, apenas com alterações triviais a partir do original. Os dois principais tipos de vírus interpretados são: vírus de macro e vírus de script.

- Vírus de macro: Utilizam técnicas de propagação baseadas em anexo de documentos que executam macros, como planilhas eletrônicas e processadores de texto. Estes vírus tendem a se espalhar rapidamente, porque os Usuários frequentemente compartilham documentos com recursos de macro habilitados. Além disso, quando uma infecção ocorre, o vírus infecta o modelo que o programa usa para criar e abrir arquivos. Uma vez que um modelo é infectado, cada documento que é criado ou aberto com esse modelo também será infectado;

- Vírus de script: Nesse tipo de vírus a principal diferença, em relação ao de macro, é a linguagem utilizada. O código é escrito para um serviço e sistema operacional específico, como por exemplo, *VBScript*, que executa em um servidor com sistema operacional Windows.

2.2.5 Cavalos de Tróia

De acordo com a mitologia Grega, um cavalo de Tróia é um "presente"que esconde uma ação não esperada. São programas que não se replicam e que tem aparência inofensiva, escondendo sua proposta maliciosa. Esse tipo de malware pode tomar o lugar de um programa com intenções legítimas, executar todas as funções esperadas e então, conjuntamente, executar as ações para a qual foi programado, dentre essas ações pode-se citar:

- Coletar informações da máquina contaminada, como credenciais, fotos, arquivos;
- Esconder sua presença e ações de maneira a impedir ou dificultar sua detecção;
- Desabilitar softwares de segurança como *firewall*, antivírus, *anti-malwares*.

Atualmente, os cavalos de tróia utilizam diversas técnicas para que sua presença passe despercebida ou que sua detecção seja difícil. Esse tipo de malware é o mais comum quando se trata do cenário da Internet brasileira, pois é bastante utilizado por fraudadores que aproveitam sua ação para coletar todo o tipo de informação necessária para fraudes financeiras em lojas de comércio eletrônico, Bancos, financeiras, etc. Técnicas de ofuscação são utilizadas pelos adversários com o objetivo de dificultar sua identificação e análise.

2.2.6 Worms

Um programa que é capaz de se propagar automaticamente através de redes, enviando cópias de si mesmo de computador para computador, de sistema para sistema, essa é a definição de um *worm*. Diferentemente do vírus, o *worm* não tem a capacidade de infectar outros programas inserindo seu código em outros programas ou arquivos, também não depende de execução para se propagar em outros sistemas, pois é através da exploração de vulnerabilidades que esse tipo de malware contamina outros sistemas, aproveitando de falhas de configuração ou softwares vulneráveis.

A grande ameaça do *worm* reside em deteriorar o desempenho de um sistema, sobrecarregando serviços e gerando tráfego, que em alguns casos, pode provocar negação de serviço (*DoS - Deny of Service*). Por ser capaz de comprometer sistemas sem a necessidade de um Usuário instalar ou executar algum programa, esse tipo de ataque tem se popularizado entre os criminosos, pois permite que em um curto espaço de tempo um grande número de sistemas seja comprometido, diferente dos vírus, que dependem do hospedeiro e Usuário. Outra vantagem é que vulnerabilidades e configurações mal feitas são bastante comuns e diariamente surgem novas fraquezas em sistemas. Para efeito de classificação, existem dois tipos principais de *worms*, serviços em rede (*network service worms*) e envio em massa (*mass mailing worms*).

Os *worms* de serviços (*network service worms*) utilizam o processo de exploração de uma vulnerabilidade em um serviço de rede, geralmente associado a um determinado sistema operacional ou aplicativo. Depois que um sistema esteja infectado o código malicioso inicia a verificação de outros sistemas vulneráveis pela rede. Sua ação aumenta à medida que outras máquinas são comprometidas, o que justifica a queda de performance do sistema de comunicação. Como não há necessidade de interação humana, rapidamente uma rede vulnerável pode ser totalmente mapeada e comprometida.

2.2.7 Backdoors

O termo *backdoor* é geralmente associado a um programa malicioso que fica aguardando uma determinada ação para que permita uma conexão remota ou acesso não autorizado, e que não seja detectado pelos dispositivos de segurança. A instalação de um backdoor é inerente à invasão de um sistema, onde o adversário habilita um "serviço"que o permita ter acesso quando necessário ao sistema comprometido, geralmente possuem capacidades especiais:

- Zombies: O termo zumbi é decorrente da utilização de máquinas comprometidas através de *bots*, possuindo a capacidade de propagação como um *worm*. Essas máquinas são controladas por adversários que compõem um sistema maior. *Botnets* são redes compostas por milhares de computadores zumbis e que são utilizadas para ataques coordenados, como DDoS (*Distribution Deny of Service*), envio de *spams* e *phishing*;

- Remote Administration Tools - RAT: Permite a um adversário ter acesso a um sistema quando desejar e em alguns casos o sistema controlado coleta e envia informações como imagens de tela, teclas digitadas, arquivos acessados sendo possível ativar remotamente dispositivos como impressoras, webcams, microfones e autofalantes. Exemplos de RAT's: Back Orifice, Netbus e SubSeven.

Um invasor, que tenha controle sobre uma *botnet*, poderá utilizá-la com o objetivo de aumentar o poder em seus ataques, inclusive a outras botnets, de modo a incorporar as máquinas zumbis de outras redes. Vários ataques tem sido reportados utilizando esse tipo de rede, como por exemplo: Envio de *phishing* ou *spam*, ataques de negação de serviço, fraudes bancárias entre outros.

2.2.8 Keyloggers

Keylogger ou *keystroke logger*, são programas utilizados para monitorar e armazenar atividades do teclado de uma determinada máquina. Qualquer tipo de informação inserida, como conteúdo de documentos, credenciais de Bancos, número de cartão de crédito, senhas são capturadas e armazenadas localmente ou enviadas para servidores remotos utilizando protocolos como FTP *(File Transfer Protocol)*, HTTP/HTTPS *(HyperText Transfer Protocol Secure)* ou anexo de e-mails. Existem algumas variações desse tipo de *malware*, que monitora cliques do mouse (*mouseloggers*) ou efetuam cópias de imagens de tela (*screenloggers*). Apesar de ser classificado como um *malware*, esse tipo de software pode ter sua utilização legítima, quando uma organização, através de uma política de segurança que preveja seu uso, habilite a funcionalidade e monitore as atividades de uso de sistemas que compõem sua rede interna.

2.2.9 Rootkits

O termo *rootkit* provém do uso em sistemas Unix/Like, onde um adversário, após obter sucesso em comprometer um sistema, instala um conjunto de ferramentas, ou *toolkit* que tem como finalidade esconder a presença maliciosa dos artefatos. Sua utilização

também ocorre em sistemas baseado em outras plataformas como o Windows. Apesar do termo *root* ser proveniente de sistemas Unix, que neste caso, seria mais indicado ter a nomenclatura modificada para *adminkits*, entretanto, não é isso que acontece.

Os *rootkits* permitem que um adversário possa controlar remotamente um sistema comprometido, apagar todas os rastros de comprometimento, e esconder sua presença do administrador, de modo que alguns programas são modificados ou simplesmente substituídos por programas modificados que compõem o pacote de ferramentas de um *rootkit*. É tarefa de um *rootkit* capturar outras informações que possam ser utilizados pelo adversário, como senhas, credenciais, outros sistemas que possam ser comprometidos.

2.2.10 Kits de Ferramentas de Ataque

Um adversário possui um conjunto de ferramentas que o auxiliam a identificar sistemas que possam ser comprometidos, de modo a ter o controle de uma rede, total ou parcialmente. Dentre essas ferramentas pode-se listar:

- Sniffers: Utilizados para monitorar uma rede capturando todos os pacotes do tráfego gerado, tanto sem fio quanto redes cabeadas. Alguns tipos de *sniffers* tem a capacidade de interagir com uma rede enviando pacotes, são conhecidos como *sniffers* ativos. Outros são projetados exclusivamente para capturar dados sensíveis como senhas e credenciais, utilizando protocolos mais comuns, como FTP, HTTP, POP3, IMAP *(Internet Message Access Protocol)*, entre outros;
- Port Scanners: É um programa utilizado para varrer uma rede buscando serviços que esteja disponíveis através de portas TCP *(Transmission Control Protocol)* ou UDP *(User Datagram Protocol)*, potencialmente localizam um alvo e coletam informações como serviços e versões, permitindo que o adversário identifique serviços vulneráveis;
- Scanners de vulnerabilidades: Esse tipo de programa pode ter sua utilização licita, onde uma organização o utiliza para varrer seus sistemas em busca de vulnerabilidades conhecidas, permitindo que seja identificada e corrigida. Entretanto, um adversário pode utilizar de maneira a encontrar e explorar as vulnerabilidades antes que sejam descobertas pela organização;
- Password Crackers: Sistemas que se utilizam de senhas para proteger suas aplicações são suscetível a ataques de força bruta, onde são testadas um conjunto de senhas

aleatórias, ou ataques por dicionário, onde uma lista de palavras são testadas para encontrar qual é utilizada para proteger um segredo.

Um adversário sofisticado pode desenvolver suas próprias ferramentas e ter seu kit exclusivo, já um adversário que possua menos conhecimento, como *script kids*, utilizam-se de ferramentas prontas, de modo a agir com a intenção de invadir ou comprometer um sistema de maneira indiscriminada.

2.2.11 Man-in-the-middle

Ataques que possuem a capacidade de interceptar e alterar o conteúdo de pacotes que trafegam por uma conexão de dados são verdadeiros desafios para os pesquisadores. O ataque *man-in-the-middle* é amplamente utilizado em sistemas bancários, baseando seu sucesso na negligência de um Usuário legítimo.

Quando o adversário consegue se inserir em uma comunicação legítima entre Banco e Usuário, é possível controlar todas as informações que trafegam pelo meio, inclusive em comunicação criptografada como é o caso do protocolo HTTPS, amplamente utilizado. Nesse caso, o atacante consegue interceptar o momento em que o servidor envia o certificado com a chave pública, criando um *proxy* de comunicação, em que o atacante passa a responder a comunicação entre as entidades (CALLEGATI; CERRONI; RAMILLI, 2009). Esse tipo de ataque também é conhecido como *proxy trojan*.

Apesar deste ataque ser relativamente simples, é utilizado para capturar informações que são necessárias para que um atacante possa se apresentar como um Usuário legítimo, geralmente é associado a uma página falsa da instituição e direcionado a esta como sendo verdadeira, o Usuário não consegue identificar o ataque e acredita que está acessando a instituição financeira. Após a captura das informações o Usuário legítimo é então direcionado ao verdadeiro sítio.

Como todo o processo de identificação do Usuário e autorização de transações é feito pelo mesmo canal, torna-se difícil proteger e evitar esse ataque, mesmo em situações em que existam ferramentas de proteção.

Praticamente uma evolução do ataque *man-in-the-middle*, o *ataque man-in-the-browser* é definido basicamente pelo comprometimento e controle da estação que se comunica

com a instituição financeira. O navegador passa a ser controlado por um artefato malicioso, neste caso, um *trojan* que chega através de ataques de *phishing*.

O adversário deverá utilizar o navegador, para isso aproveita-se da possibilidade de instalar extensões de funcionalidades, que muitos navegadores atualmente permitem. Quando detectado uma conexão com a instituição financeira, o ataque é então iniciado e qualquer transação poderá ser alterado, em tempo real (GUHRING, 2007).

Um tipo peculiar desse tipo de ataque é conhecido como *boy-in-the-browser*, uma variante que simplesmente altera a configuração da estação direcionando o tráfego do navegador, e após a execução se apaga do sistema, dificultando a identificação de comprometimento.

Qualquer solução de segurança, que não consiga impedir o controle de máquina, passa a ser obsoleta, pois a partir do momento em que a autenticação de um Usuário é solicitada o adversário terá todas as informações, e poderá contar com a colaboração não voluntária do Usuário legítimo, que passa a responder os desafios exigidos, pois o conceito *WYSIWYG (What You See Is What You Get)* é plenamente aceito. O que o adversário quer que o Usuário visualize será apresentado.

2.3 SISTEMAS BASEADOS EM DETECÇÃO DE ANOMALIAS

A detecção de anomalias consiste em determinar se uma mensagem ou transação é legítima ou não. A utilização desse modelo é reativo, ou seja, é necessário que se analise a mensagem somente depois dessa ter sido enviada.

Modelos de detecção buscam atuar como sistemas de prevenção, analisando as diversas fases do processo, desde autenticação tradicional através do desafio de prova do que o Usuário sabe, até a comprovação de um dispositivo que possa ser utilizado para provar a identidade, como geradores dinâmicos de senhas ou mesmo uma estação conhecida pelo sistema.

Sempre que um novo método é implantado, os adversários buscam encontrar maneiras de contornar essa barreira, tornando-se um ciclo repetitivo e constante(KOVACH, 2011), o que muitas vezes pode desgastar a imagem da instituição que insiste nesse modelo.

Para que um modelo de detecção de anomalias seja implantado é necessário determinar um perfil de comportamento para cada Usuário legítimo. Como o comportamento pode variar de acordo com as necessidades de cada Usuário, deve-se definir o conjunto que será observado por um determinado tempo. As observações então irão compor um modelo estatístico probabilístico, onde será possível determinar o grau de normalidade de uma determinada interação e seu comportamento, classificando cada interação através de um escore.

2.3.1 Modelos Supervisionados

Um modelo supervisionado depende de uma base histórica de atividades do Usuário, essa base deve conter tanto os comportamentos legítimos como os considerados anômalos, de modo a possibilitar que um sistema consiga classificar as interações através de comparações com a base histórica. Dessa forma, é possível identificar interações ilegítimas apenas se essas forem conhecidas (KOVACH, 2011).

É possível tornar esse modelo em não supervisionado, que busca por padrões de comportamento diferentes do que foi definido como normal.

2.3.2 Modelos Estatísticos e Baseados em Regras

Nos modelos estatísticos, métricas são criadas a fim de identificar modificações no comportamento considerado normal para cada Usuário. Nesse tipo de abordagem, a análise das interações acontece de maneira absoluta ou diferencial.

A análise absoluta, a identificação de uma mensagem ou transação, utiliza variáveis com valores limiares definidos, de maneira que a variação de valores fora desses limiares gera um alerta de atividade suspeita.

A análise diferencial compara todas as interações recentes com novas interações, de modo a identificar atividades fora do padrão em um espaço de tempo reduzido (BURGE, 1997). A comparação é feita através de um escore gerado para cada interação observada, caso não seja identificado nenhuma anomalia, os valores observados passarão a compor o histórico do Usuário para comparações futuras.

De maneira complementar aos modelos estatísticos, pode-se utilizar métodos baseados em regras, que são funções supervisionadas de negócio que se valem de condições estáticas para tomada de decisão. A dificuldade nesse modelo é que deve estar em

constante atualização, de modo a identificar novos métodos de ataque que buscam subverter a regra estabelecida.

A fim de se identificar uma mensagem produzida que não seja de um Usuário legítimo, é necessário que se possua e conheça uma base histórica de utilização desse Usuário. Desse modo é atribuído uma certa probabilidade P de que a mensagem ou transação é legítima.

Diversos modelos são estudados e aplicados atualmente. É fato que o desafio desses modelos é que a resposta seja a mais correta possível, ocorrendo com menor índice de falsos positivos e que os alarmes de falso negativo sejam tratados de maneira rápida e que evite o desgaste da imagem da instituição. Claro que isso é considerado difícil, pois os atacantes buscam enganar esse tipo de abordagem executando ações o mais parecido possível de um Usuário legítimo. Também é necessário que esses sistemas sejam dinâmicos e atualizados em tempo real, que não impeçam que uma transação ou mensagem seja considerada legítima sem que isso seja verdade.

Para esses casos são utilizados diversos conceitos, todos baseados em probabilidade. Possuem três conceitos fundamentais (KOVACH; RUGGIERO, 2011):

1. Listas brancas: Onde a probabilidade de que uma mensagem seja considerada ilegítima é de $P = 0$;
2. Listas negras: Onde a probabilidade de que uma mensagem seja considerada ilegítima é de $P = 1$;
3. Listas de regras: Onde a probabilidade de que uma mensagem seja considerada ilegítima é dada através de uma série de regras definidas pela característica de utilização de um Usuário legítimo $0 < P \leq 1$.

2.4 SISTEMAS DE AUTENTICAÇÃO E IDENTIFICAÇÃO

A autenticação de Usuários têm sido um desafio há décadas (CONKLIN; DIETRICH; WALZ, 2004). Conceitos como autenticação baseado em informações de Usuário e senha são simples e de baixo custo. Esse modelo pode ser mais adequado quando se quer identificar uma entidade local, mas torna-se complexo quando essa entidade é remota e sua base de dados para autenticação deve estar disponível em uma rede. Proteger

esse sistema passa a ser um desafio, tanto para que não seja comprometido ou burlado, quanto para que seus Usuários sejam identificados corretamente (ECONOMIST; UNIT, 2007).

Os sistemas de acesso dos modelos bancários na Internet são diretamente associados a autenticação. Processo que estabelece um relacionamento de confiança na identificação de Usuários de um sistema, apresentando técnicas de desafio e resposta envolvendo autenticação remota de uma entidade que deseja utilizar algum serviço de rede. Essa autenticação pode ser de uma via, onde apenas uma parte deve ser desafiada e responder corretamente esse desafio, ou ambas, onde a autenticação mútua, onde todas as partes devem ser desafiadas a comprovar sua identidade.

Um protocolo de autenticação deve definir uma sequência de mensagens entre o solicitante e o verificador, de maneira que demonstre que o solicitante tenha as credenciais necessárias para que o mesmo seja identificado e a comunicação seja validada e considerada confiável.

O problema resultante dessa interação é associada a confiança diretamente intrínseca, ou seja, se o desafio for respondido de acordo com o esperado é associado total confiança. A entidade solicitante passa a fazer parte como identificada e autenticada.

Protocolos de identificação permitem a uma entidade provar sua identidade para outra entidade. Neste protocolo, uma parte chamada de *solicitante* interage com outra parte chamada de *verificador* , a fim de provar a sua identidade. Um esquema de identificação deve ser seguro contra ataques de representação, quando um adversário tenta fazer com que o verificador acredite que ele é um solicitante arbitrário.

Sistemas de autenticação são categorizados em números de fatores que os incorpora (SEUNGJAE; CUNNINGHAM; RYOO, 2008), portanto, neste trabalho serão considerados como modelo de autenticação as seguintes categorias:

Categoria 1 *O que se sabe*

Um segredo é compartilhado para que um Usuário saiba algo para que possa se autenticar. Esse segredo pode ser informações e elementos que somente é de conhecimento do Usuário e do servidor. Como exemplo: Senhas e códigos estáticos.

Categoria 2 *O que se tem*

Dispositivos são utilizados ao se provar uma identidade. Existem diversos tipos de dispositivos com essa finalidade, muitos popularizados e que utilizam interface *USB*[5] para que sejam inseridos no sistema. Cartões de identificação profissional, cartões de crédito, certificados digitais são exemplos desses dispositivos, assim como geradores de senhas dinâmicas baseado em hardware

Categoria 3 *O que se é*

Utilizados para identificar um Usuário através de suas características consideradas únicas. A biometria é considerada uma solução para os processos de segurança que necessitem identificar uma pessoa. Além disso, uma solução biométrica nem sempre é a melhor abordagem (GROUP, 2002), pois o sucesso ou fracasso de um sistema biométrico não depende da confiabilidade do produto biométrico sozinho, e isso não pode ser enfatizado. Há muitos outros fatores que contribuem para o sucesso ou a falha. Também é essencial entender que nenhuma tecnologia biométrica oferece uma solução para todos os requisitos de identificação e autenticação de um Usuário.

Qualquer sistema que incorpore os três fatores juntos são considerados sistemas seguros e fortes, no entanto, são modelos difíceis de implementar em sistemas que se utilizam do canal Internet, sendo necessário estrutura e hardware específicos para cada Usuário, tornando o modelo custoso e difícil de administrar.

Basicamente esses sistemas utilizam esquemas que possibilitam a comprovação de uma identidade, através de uma prova. Típicos cenários atuais para acesso a serviços remotos. Portanto, é necessário que as partes envolvidas em uma comunicação, possam comprovar suas identidades através de esquemas seguros e resistentes a ataques.

Um esquema de autenticação deve prever e resolver problemas como ataque por replay, onde um adversário, com acesso ao canal de comunicação entre as partes legítimas, captura os pacotes utilizados e posteriormente utiliza-os em parte ou totalmente, alterando por conveniência e necessidade, quando existe alterações o ataque é conhecido

[5]Universal Serial Bus

por passivo, de outra forma é um ataque ativo. Da mesma maneira, ataques de personificação são problemas a serem enfrentados, onde um adversário consegue tomar o lugar de um Usuário legítimo, sem que a outra parte consiga determinar diferenças. Também existe a figura de um mal-intencionado, onde o adversário tem a capacidade de ler e alterar a comunicação, de maneira que a outra parte não consiga detectar falha na integridade. Todos esses ataques são discutidos com mais detalhes no capítulo 3.

2.4.1 Modelos de Identificação Baseados em Senhas

Um processo de identificação é geralmente associado quando uma entidade deseja confirmar a identidade de uma parte requisitante, onde é feito um desafio para que o requisitante prove sua identidade através de alguma evidência previamente combinada entre as partes. O modelo mais comum utilizado para se identificar uma entidade é baseado em senhas e códigos estáticos (WIESMAIER et al., 2005). Número de identificação pessoal *(PIN - Personal Identification Number)* são fáceis de utilizar e de administrar. Todo acesso é baseado no desafio imposto a um Usuário de informar o código correspondente para que o mesmo seja autorizado e tenha permissão de utilizar os serviços a ele associados. O esquema de identificação estático é vulnerável a uma séria de ataques, pois quando o adversário obtiver a informação necessária, poderá utilizar o sistema como se fosse um Usuário legítimo. Por esse motivo alguns sistemas utilizam-se de senhas dinâmicas, que permitem através de um controle ou protocolo apropriado desafiar o requisitante sempre que necessário.

Existem propriamente duas maneiras convencionais para se utilizar códigos dinâmicos, implementando-se modelos de *OTP (One-Time Password)* e *TAN (Transaction Authorization Numbers)* (WIESMAIER et al., 2005).

A utilização de esquemas dinâmicos facilita o modelo de identificação, no entanto, pode prover insegurança quando utilizado de maneira a permitir acesso total. Esquemas dinâmicos valem-se de protocolos que se utilizam de um segredo que é associado ao tempo, geralmente gerando uma senha descartável e utilizável em janelas curtas de tempo. Entretanto, isso não evita diversos ataques conhecidos e que pode prover um falso sentimento de segurança.

Esquemas que se utilizam de códigos autorizadores são promissores, permitem associar diversas senhas para que um Usuário forneça-as sempre que for solicitada a comprovar sua identidade. São diretamente associadas às transações, entretanto, alguns modelos descritos em (JOHNSON, 2008) apresentam ataques conhecidos, como exemplo, o

roubo dos códigos e utilização indevida.

Um modelo baseado em senhas pode utilizar mais de um fator de autenticação, pois o fator de algo que um Usuário sabe, como uma credencial, pode ser associada a algo que se tenha, como um hardware gerador de senhas dinâmicas descartáveis, ou outro tipo de dispositivo que permita associar essas informações.

A utilização de apenas um fator é altamente perigoso e deve ser descartada a utilização dessa estratégia, pois o adversário precisa obter apenas uma informação para que tenha acesso a um sistema, entretanto, dois fatores não é, necessariamente, a solução mais adequada a ser utilizada, quando confrontada ao problema das instituições financeiras, pois não basta apenas autenticar, deve se autorizar uma determinada ação, por esse motivo será discutido na seção 2.4.2 os modelos de fatores múltiplos.

2.4.2 Modelos de Autenticação Baseados em Múltiplos Fatores

Sistemas de autenticação múltiplos tem como premissa que exista pelo menos duas instâncias das três categorias apresentadas na seção 2.4, tendo como perspectiva que as instâncias estejam separadas por tipo de ataque e seus vetores, que serão discutidos e apresentados com detalhes no capítulo 3. Instituições financeiras usam os modelos de fatores múltiplos para combater roubo de identidade de seus clientes ou Usuários de canais *online*, de modo a impedir que transações sejam feitas de maneira espúria (SEUNGJAE; CUNNINGHAM; RYOO, 2008).

É importante considerar que uma boa implementação depende de sua disponibilização em camadas, sendo que não é prudente utilizar todo o modelo de autenticação em uma mesma etapa, o que pode acarretar uma autenticação forte a princípio mas que demonstrada fraca a diversos tipos de ataques. O ideal é que a identidade do solicitante seja colocada a prova quando uma necessidade de autorização forte seja requerida. Outro ponto que deve ser considerado é o tipo de utilização, pois um modelo múltiplo pode ser bom para uma organização e não para outra.

Diversas organizações tem mostrado a grande importância em se utilizar mais de um fator de autenticação. Organizações financeiras tem como desafio produzir métodos de autenticação que evitem o roubo de credenciais, e consequentemente, protejam seus Usuários contra fraudes e utilização indevida. Entretanto alguns fatores afetam a adesão de novos processos de autenticação, como o custo elevado de aquisição de hardwares dedicados, no desenvolvimento e treinamento dos Usuários, o processo de

logística no envio ao Usuário bem como a gerência e manutenção de dispositivos e certificados, tanto software quanto hardware.

2.4.3 Códigos de Autenticação de Mensagens

Códigos de Autenticação de Mensagens *(MAC - Message authentication codes)* são utilizados para verificar a integridade e autenticidade das informações trocadas entre duas partes através de um canal potencialmente inseguro (BUCHMANN, 2001). Basicamente, uma parte precisa validar uma mensagem recebida da outra parte como sendo autêntica (ou não modificada). Isto geralmente é feito utilizando-se de algoritmos *MAC*, que toma como entrada a mensagem e um segredo compartilhado anteriormente entre as partes.

A parte *origem* envia uma mensagem a parte *destino*, computando *MAC* $h_k : k \in \kappa$ da mensagem m e o segredo k previamente trocados entre as partes, onde κ é um conjunto, acrescentando à mensagem um identificador *authentication tag* que é calculado utilizando-se de algoritmos *MAC* na forma $y = h_k(m)$. Quando a parte *destino* recebe a mensagem, recalcula a *authentication tag* da mensagem na forma $y' = h_k(m')$ e verifica se o valor obtido é igual a *tag* recebida e aceita m' se $y = y'$. Neste caso, a mensagem m é considerada autêntica e que não foi alterada no canal de comunicação entre *origem* e *destino*, pois sem o conhecimento de k é difícil calcular um par $(m', h_k(m'))$ partindo do par $(m, h_k(m))$ com $m \neq m'$.

Um algoritmo deve ser seguro contra falsificações, não deve permitir a um adversário computar um identificador *tag* de autenticação válida em mensagens arbitrárias.

É importante considerar a segurança desse algoritmo contra ataques de replay (JOHNSON, 2007), onde um adversário reenvia uma mensagem enviada anteriormente junto com sua *tag* de autenticação válida, fazendo acreditar que a mensagem é autêntica, ou seja, ela foi enviada pela outra parte.

A construção desses algoritmos podem utilizar diferentes primitivas, como cifras de blocos e resistente a colisões em funções de *hash*. O protocolo *HMAC(Hash based Message Authentication Code)*, proposto por Bellare (BELLARE; CANETTI; KRAWCYK, 1996) e aceito como padrão pela indústria, utiliza um esquema adaptativo e resistente a ataques de replay. A definição formal é dada por:

Teorema 1 *Dado H uma função de* hash $\sum^* \rightarrow \sum^n, n \in \mathbb{N}$*, resistência aceitável a colisão, que corresponde arbitrariamente a sequências longas e de comprimento fixo, assumido que se utilize as funções como* MD5 *(RIVEST, 1992) ou* SHA-1 *(JONES, 2001) uma função de tamanho de bloco b. A função* HMAC *recebe uma mensagem de entrada m de comprimento arbitrário e uma chave* $k \in \{0,1\}^b$*, sendo computada pela expressão 2.1.*

$$HMAC(k,m) = H((k \oplus \texttt{opad}) \mid H((k \oplus \texttt{ipad}) \mid m) \tag{2.1}$$

Onde `opad` e `ipad` são constantes de tamanho b que consiste em bytes hexadecimal $0x36$ e $0x5C$ repetido tantas vezes quanto necessário, respectivamente.

A fim de alcançar a segurança contra ataques de replay é utilizado *Nonces*, que são informações aleatórias anexadas na mensagem antes de se computar o *HMAC*. Para uma mensagem m de tamanho arbitrário, é computado a segurança contra ataques de replay, conforme a expressão 2.2.

$$HMAC_N(k,m) = HMAC(k, m \mid Nonce) \tag{2.2}$$

Onde $Nonce \in r\{0,1\}^n$ sendo n um parâmetro seguro. Para cada mensagem *HMAC* recebida, as partes armazenam o $Nonce$ e, se a mensagem contiver o mesmo $Nonce$ recebido posteriormente, então a mensagem é considerada inválida. Uma vez que é recebido o $Nonce$ e é escolhido aleatoriamente para cada *HMAC* calculado, um adversário que captura uma mensagem, enviada anteriormente, não poderá reproduzi-la sem ser detectado por uma das partes com alta probabilidade.

3 REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentado os principais modelos e medidas de segurança atualmente adotadas pelas instituições financeiras. Também é discutido as principais ameaças e vulnerabilidades desses modelos, finalizando com a exposição dos principais trabalhos relacionados e suas contribuições.

3.1 MEDIDAS ATUAIS DE SEGURANÇA

Os Bancos, que possuem atendimento Internet, estão buscando encontrar um modelo eficiente que permita identificar os Usuários e ao mesmo tempo autorizar as movimentações financeiras, evitando assim as fraudes eletrônicas. Entretanto, os modelos atuais estão fortemente inseridos em um cenário de identificação da fraude, o que deixa o modelo sempre reativo e não proativo, ou seja, somente quando a fraude ocorre é que algum processo de segurança é iniciado.

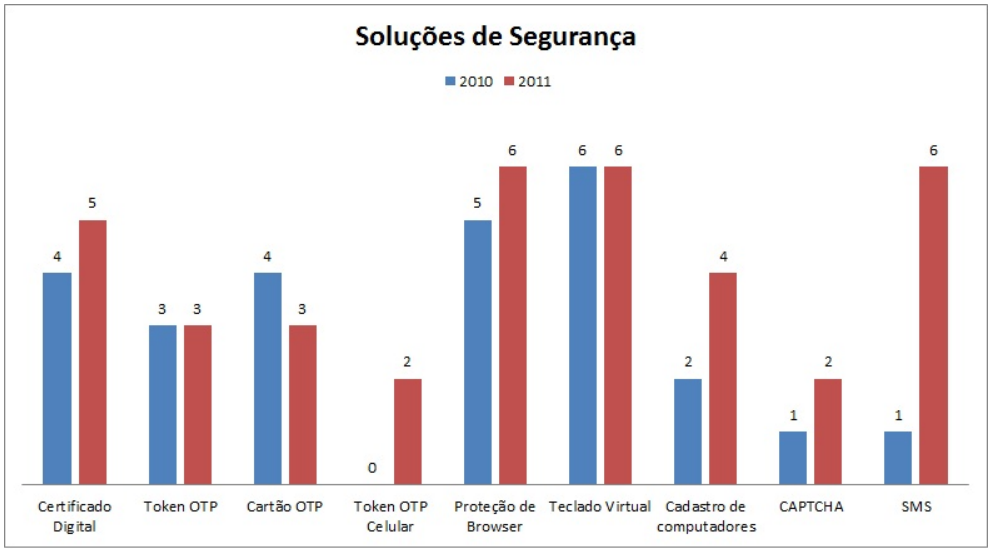

Figura 3.1: Avaliação dos Modelos de Segurança *Internet Banking*

Analisando os modelos de segurança implementados pelos sete maiores Bancos no Brasil entre 2010 e 2011, segundo o Banco Central (BCB, 2010), percebe-se a utilização de camadas de segurança e diversos métodos que operam em conjunto, conforme apresentado na Figura 3.1. Algumas soluções se sobressaem sobre outras como é o caso da solução baseado em teclado virtual, onde seis dos sete maiores Bancos do Brasil utilizam-na. Da mesma forma é possível determinar que ataques considerados antigos

como os baseados em artefatos maliciosos como *trojans* bancários continuam a operar, direcionando a segurança para os modelos de identificação da fraude, sendo assim reativo.

Todos os sistemas analisados usam, inicialmente, uma forma de identificação baseada em credenciais e senhas. Esse modelo de segurança baseado no que o Usuário sabe, é utilizado como outro fator de autenticação, geralmente aplicado no momento em que alguma transação, que envolva movimentação de recursos financeiros, é executada.

3.1.1 Certificação Digital

Os certificados digitais utilizados pelos Bancos são do tipo A1 e A3 padrão ICP-Brasil (BERTOL; SOUSA; PEOTTA, 2009). Em geral são utilizados para proverem a autenticação dos Usuários tanto pessoa física quando pessoa jurídica.

3.1.2 One-Time Password

Os dispositivos *One-Time Password* (HALLER; METZ; NESSER, 1998) são comumente utilizados como um segundo fator de autenticação forte, que pode ser exigido em um momento aleatório ou em uma situação específica. Busca evitar que os dados capturados sejam utilizados futuramente, pois a senha é descartável. O protocolo é aplicado em hardware dedicado ou softwares próprios, que executam em diversas plataformas e sistemas operacionais. Neste mesmo segmento, também são utilizados dispositivos chamados de *One-Time Password Card*, é uma maneira com custo menor de se implementar senhas dinâmicas, mas que em alguns Bancos são reutilizáveis, portanto, erroneamente chamados de *OTP*. Têm o mesmo princípio do *token OTP*, autenticar um Usuário em um segundo fator.

3.1.3 Proteção de Navegador

Neste modelo a atuação é a de proteger o Usuário através da utilização de um navegador de Internet, utilizado no momento do acesso ao serviço. Protege o Usuário contra ataques de malware conhecidos (PUENTE; GONZALEZ; SANDOVAL, 1999), monitorando a área de memória a procura desses artefatos e desabilitando a captura das informações das credenciais e outros dados importantes.

3.1.4 Teclado Virtual

Desenvolvido para dificultar a captura das credenciais do Usuário através da utilização de *keyloggers*, em geral são elaborados utilizando a linguagem de programação Java, com criptografia baseada em software. Atualmente, está sendo substituído pelos Bancos por outros métodos mais eficientes, que exijam menos consumo de hardware e de banda.

3.1.5 Registro e Identificação de Dispositivos

Restringe o acesso ao serviço apenas para os computadores conhecidos e previamente cadastrados. Utiliza-se de técnicas de identificadores único de hardware associados a identificação do Usuário através de suas credenciais. Este método é geralmente associado ao cadastramento de computadores, no entanto, alguns Bancos têm optado por retirar o cadastramento e utilizar apenas a identificação, buscando facilitar o acesso ao canal. Trata-se de um modelo de identificação baseado em características físicas do equipamento do Usuário o qual permite identificar sua origem e histórico.

3.1.6 Identificação Positiva

A identificação positiva é fortemente atrelada a monitoração do canal, mas de maneira automatizada, assim como o método de *SMS* é utilizado para confirmar a autenticação, a identificação positiva solicita, baseada em um contexto pré-estabelecido de segurança, informações que teoricamente apenas o Usuário legítimo teria como fornecer. *Completely Automated Public Turing test to tell Computers and Humans Apart*, é um método que foi recentemente adotado por apenas um Banco, basicamente busca inibir a atuação de robôs, que são *softwares* automatizados de ataque, ou *malwares* que se utilizam de sessões autenticadas e autorizadas para inserir transações fraudulentas.

3.1.7 Monitoração de Transação

Apesar deste modelo não estar listado nesta pesquisa é importante entender que todos os Bancos a utilizam, cada instituição à sua maneira e método particular. Sistemas inteligentes (FONSECA, 2006), histórico de transações, entre outros, pesquisando em todas as transações efetuadas a procura de possíveis fraudes e relatando como incidente.

3.2 COMPARATIVO DE SOLUÇÕES E VULNERABILIDADES

A Tabela 3.1 apresenta as vulnerabilidades para cada modelo apresentado neste trabalho.

Tabela 3.1: Vulnerabilidades nos modelos de Segurança de *Internet Banking*

Modelos	**Vulnerabilidades**
Certificado Digital	Certificados tipo A1 podem ser exportados e utilizados remotamente; Certificados tipo A3 podem ser utilizados enquanto o Usuário estiver utilizando.
Token OTP	A senha pode ser capturada e utilizada em tempo real; O Usuário pode ser enganado (Engenharia social) a informar a senha em transações que não são as que sejam reconhecidas.
OTP Card	Algum malware pode coletar as senhas ou solicitar que o Usuário as informe.
Proteção de navegador	Novos malwares continuam ativos até que sejam identificados pela solução; Páginas falsas que impedem que a proteção seja carregada, enganando o Usuário a informar seus dados em um ambiente não confiável.
Teclado Virtual	Ferramentas conhecidas como *Screenloggers* ou *mouseloggers* permitem a captura das informações; Técnicas de decifragem (GRANVILLE; TAROUCO; BARCELOS, 2009).
Cadastramento de Computadores	Possibilidade de reprodução de informações dos hardwares do Usuário; Reprodução da informação de cadastramento. O adversário pode utilizar técnicas de engenharia social onde o Usuário legítimo autorize um computador ilegítimo.
CAPTCHA	O método adotado é extremamente simples, permite que um software OCR leia a informação solicitada e a reproduza.
Short Message Service	O adversário pode alterar os dados do cadastro do Usuário informando outro número de celular.
Identificação de Máquinas	Possibilidade de reprodução de informações dos hardwares do Usuário.
Identificação Positiva	Informações sobre os Usuários estão cada vez mais disponíveis através de serviços de redes sociais, serviços de buscas na Internet e mesmo a engenharia social pode ser utilizada.
Frase Senha	Ferramentas conhecidas como Screenlogger, *keylogger* ou *mouselogger* permitem a captura das informações; Técnicas de criptoanálise.
Monitoração de Transações	Malwares estão aproveitando e criando perfis que permitem aproximar o tipo de utilização de cada Usuário. (BINSALLEEH et al., 2009).

Não existe a pretensão de encontrar a totalidade de vulnerabilidades, mas sim de apontar que os mesmos são atualmente vulneráveis a ataques. O que torna importante a correta identificação das ameaças (ABNT-A, 2006) pode prover um modelo de segurança mais eficiente, considerando essa informação na criação de um sistema seguro.

É possível identificar as tentativas dos Bancos Brasileiros em encontrar soluções seguras para que seus clientes acessem suas informações de maneira protegida, entretanto, é notório que os modelos atuais não atendem a esse objetivo, pois elevados valores de fraudes divulgados recentemente corroboram para essa afirmação. É necessário compreender que proteger um ambiente muito dinâmico, como é o caso do computador pessoal do Usuário, além de difícil, é algo custoso para os Bancos. Cada nova solução adotada deve considerar seu tempo de aceite e desgaste, ou seja, resolver um problema individual é agir em curto prazo, o que deixa esses modelos vulneráveis, ainda mais que o próprio Usuário legítimo poderá fragilizar a solução, pois um atacante pode convencer esse Usuário a agir de maneira não esperada, entregando informações sobre sua identidade ou confirmando os desafios de autenticação solicitados pelo Banco.

3.3 MODELO ADVERSARIAL

Os adversários tem como meta poder capturar informações sensíveis, como dados de autenticação, credenciais e senhas. Conseguir o controle total do computador pessoal do Usuário, infectando-o com algum tipo de *malware* ou outro código malicioso qualquer. Uma vez que a máquina do Usuário é geralmente vulnerável a vários ataques, não é difícil obter acesso e controle utilizando programas maliciosos. Essa estratégia é comumente observada em ataques contra sistemas de *Internet Banking*, através do qual os adversários obtém sucesso em roubar dados de autenticação e forjar transações arbitrariamente. Também é a estratégia mais barata e segura para um adversário, exigindo apenas a instalação de software malicioso e, geralmente, não fornece evidências forenses suficientes para identificar o infrator. Neste ambiente, o programa malicioso geralmente faz o download do código de um site remoto, faz a interação com os sistemas dos Bancos e carrega os dados recolhidos a partir da máquina infectada para outro local remoto.

Para ativar o acesso *Internet Banking*, muitas instituições financeiras requerem que o Usuário solicite tais serviços através de um terminal de auto atendimento (ATM). Por isso, é necessário considerar a segurança desses terminais e do tipo de informação que um adversário pode extrair ou inserir nesses dispositivos. Embora seja relativa-

mente fácil para um adversário monitorar a tela e teclado de um terminal de auto atendimento, não é necessário considerar o caso desse adversário ser capaz de, arbitrariamente, controlar esse terminal, uma vez que o adversário teria todas as credenciais necessárias para executar qualquer tipo de transação no próprio *ATM*. O acompanhamento da entrada e saída de informações dos caixas eletrônicos são obtidos com câmeras comuns, sensores e também com dispositivos falsos, como teclados e leitores de cartão, conhecidos como "*Skimmer*"(CASEY et al., 2011).

Adversários geralmente têm acesso limitado a um dispositivo móvel de um Usuário, uma vez que é difícil comprometer esses dispositivos, monitorar ou alterar a rede à qual eles estão conectados. O sistema operacional de um dispositivo móvel é feito sob medida e possui arquiteturaminimalista, com poucos serviços disponíveis para Usuários externos. Possuem uma probabilidade de comprometimento menor que os computadores pessoais. Por exemplo, redes de celulares *GSM*[6], na qual o acompanhamento, adulteração e inserção de informações sejam difíceis, e que exigem equipamentos sofisticados e de alto custo, e proximidade do dispositivo móvel do Usuário.

Para garantir um nível aceitável de segurança para o canal de comunicação entre Usuário e Banco, diversas soluções são testadas e usadas continuamente, com o intuito de contornar problemas de confidencialidade, integridade, autenticidade e não repúdio.

Como apresentado neste capítulo, os sistemas bancários atuais são afetados por diversas falhas, que permitem que os adversários executem operações arbitrárias como se fossem Usuários legítimos. No entanto, na literatura atual, não há nenhuma definição formal do ideal de segurança para sistemas de *Internet Banking* ou como os adversários podem atacá-los.

Nesta seção, será discutido como os adversários podem comprometer a vários dispositivos e canais envolvidos nesses sistemas, propondo um modelo adversarial, que capta os poderes dos adversários atuais em diferentes níveis de interação que é possível ao adversário executar em componentes existentes.

Nesta pesquisa é considerado um sistema composto por duas partes: *Banco* e *Usuário*, que se comunicam através de três diferentes dispositivos conectados a seus respectivos canais:

[6]Global System for Mobile Communications

1. *ATM (Automatic Teller Machine)* conectados diretamente à rede do *Banco*;
2. Dispositivo pessoal do *Usuário* conectado à Internet;
3. Dispositivo móvel do *Usuário*, conectado a uma rede celular privada.

Teorema 1 *O Modelo adversarial considera que o Adversário está disposto a realizar transações arbitrárias que tem as seguintes características:*

- *Adversário* tem o controle sobre tudo que o *Usuário* faz em seu dispositivo pessoal. *Adversário* é capaz de monitorar todas as mensagens trocadas pelas duas partes e arbitrariamente interferir no protocolo através da inserção de informações arbitrárias (o que é necessário para ser autenticado pelo *Usuário*);
- *Adversário* tem a capacidade de monitorar todas as informações provenientes de *Usuário*, como a entrada e o que sai como destino ao *Banco*. Entretanto, não é capaz de controlar o *ATM* e não é capaz de inserir ou modificar qualquer informação;
- *Adversário* não controla o dispositivo móvel do *Usuário* e não é capaz de obter ou modificar qualquer informação armazenada. Também não é capaz de, em segredo, monitorar as interações entre *Usuário* e *Banco*, ou adulterar qualquer tipo de informação trocada nessa interação, nem tão pouco transmitir informação como sendo um *Usuário* legítimo.

Idealmente, os sistemas de *Internet Banking* não devem permitir que os adversários possam realizar transações arbitrárias ou alterar o conteúdo das operações emitidas por um Usuário legítimo. Embora possa ser difícil garantir a privacidade em ambientes comprometidos (como o computador pessoal do Usuário), um sistema de *Internet Banking* deve impedir pessoas não autorizadas ou mal-intencionadas de executar transações que sejam consideradas legítimas.

Os protocolos utilizados para o processamento de transações em sistemas de *Internet Banking* são definidos como o que autoriza as transações individualmente na entrada de credenciais do *Usuário* e informações sobre a transação. Tais protocolos devem primeiro garantir que o Usuário é realmente quem diz ser (como um protocolo de

identificação) e depois somente depois autorizar uma transação, legitimamente emitidas pelo *Usuário*. Se a identidade do *Usuário* e da autenticidade da transação são confirmadas, a transação é autorizada.

A definição formal de protocolos de autorização de transação (KATZ; LINDELL, 2007) é apresentada pelo Teorema 2.

Teorema 2 *Considerando um processo de autenticação entre as partes Usuário e Banco, que se comunicam através de um ATM, um site e um dispositivo móvel, conectados em suas respectivas redes, o protocolo de autorização é considerado seguro se o adversário não for capaz de forjar uma transação de Usuário, personificando um Usuário legítimo.*

Observa-se que não é exigido do protocolo de autorização a confidencialidade das transações efetuadas. No entanto, espera-se ser possível impedir que o adversário possa personificar um *Usuário* legítimo e forjar transações. Esses protocolos estão sujeitos aos mesmos ataques que afetam os protocolos de identificação. Além disso, eles estão sujeitos a ataques onde um adversário tenta, arbitrariamente, alterar informações das transações enviadas pelo *Usuário* legítimo, a fim de forjar transações sem adulteração de identificação do *Usuário* legítimo.

A implementação natural de um protocolo de autorização é a utilização de um protocolo de identificação utilizado para verificar a identidade de um Usuário que tenta emitir uma transação ilegítima. No entanto, a verificação da identidade do Usuário não é suficiente, pois um adversário pode ter controle completo do dispositivo pessoal do Usuário, sendo capaz de executar transações arbitrárias depois que um Usuário legítimo se identifica com sucesso. Assim, cada execução do protocolo de identificação deve ser estritamente relacionada a uma transação específica permitindo verificar e garantir a sua autenticidade.

3.4 MODELAGEM DE AMEAÇAS

A modelagem de ameaças é importante para que se obtenha uma visão ampla das ameaças, utilizando o conceito de *attack tree* para análise de ameaças, permitirá entender a complexa situação do risco (SAINI; DUAN; PARUCHURI, 2008).

A meta do adversário é obter acesso que permita efetuar transações. A Figura 3.2 apresenta um modelo conceitual, proposto neste trabalho, que lista as pricipais ameaças

associadas diretamente à sua exploração, sendo condicionada a uma ação determinado pelo atacante e Usuário.

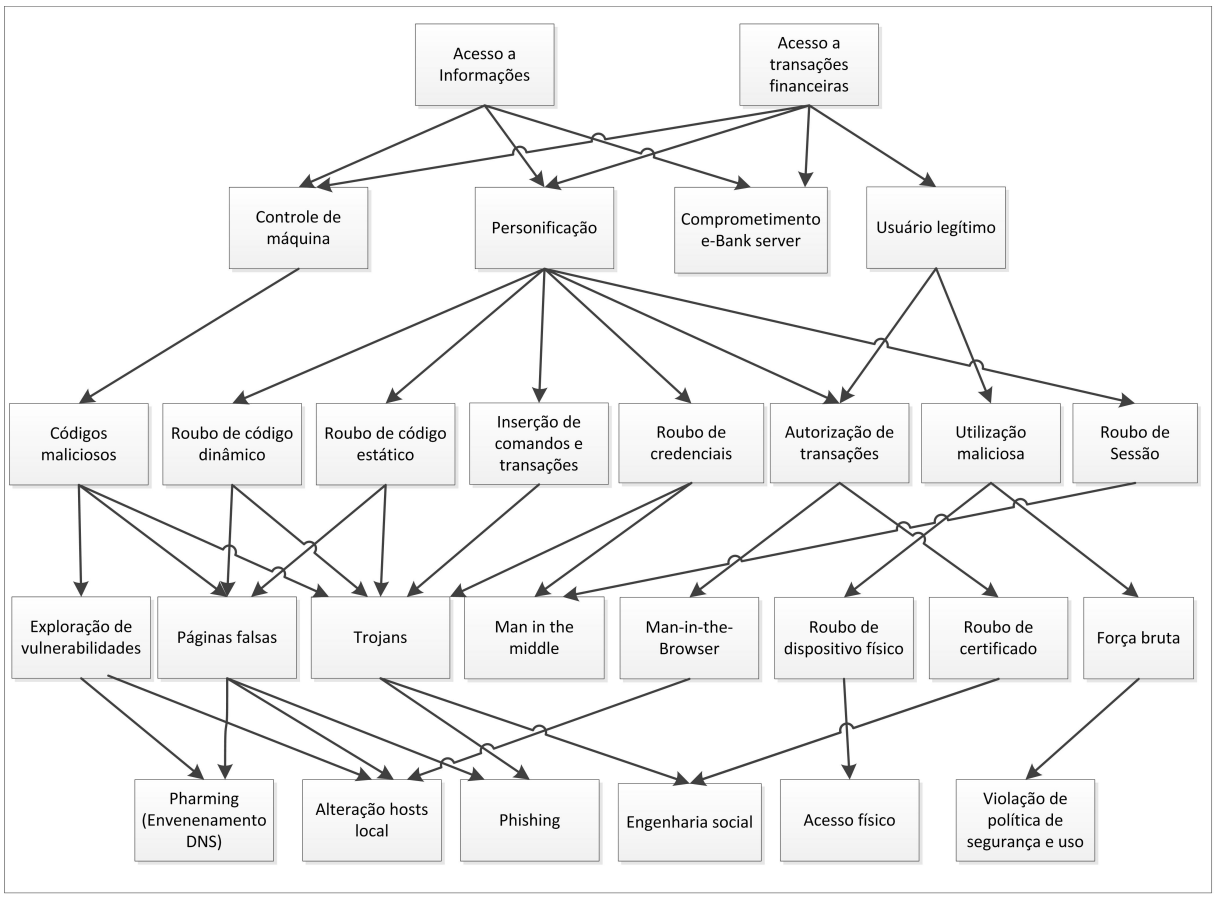

Figura 3.2: Árvore de Ameaças aos Sistemas de *Internet Banking*

O adversário utiliza-se de diversas técnicas de ataques para obter sucesso no quesito de controlar uma máquina pessoal de um Usuário legítimo, personificar um Usuário, enganar um Usuário legitimo ou mesmo, em casos extremos, mas não descartado, comprometer um servidor de um Banco.

É possível subdividir a classificação das ameaças em dois grupos de ataques(LTD, 2006): Ataques passivos e ataques ativos.

3.4.1 Ataques Passivos

Ataques passivos utilizam-se de modelos para captura de credenciais que são armazenadas e utilizadas posteriormente. Esse ataque gera duas possibilidades: Utilização *offline* ou *online*.

1. Ataque *Offline*: O adversário, seleciona um Usuário para que tenha acesso às informações exigidas pela instituição financeira que permita sua utilização. Esse tipo de ataque é dependente de instalação de algum artefato malicioso, como *keylogger* ou uma variante. Usuários podem ser vítimas de um ataque desse tipo simplesmente porque anotam suas senhas não criptografadas ou as armazenam em algum arquivo visível em um disco rígido local, remoto ou portátil.

2. Ataque *Online*: O adversário seleciona Usuário de maneira aleatória, buscando afetar o maior número possível de Usuários conectados à Internet. Utilizam-se de técnicas de exploração de vulnerabilidades remotas ou métodos de *phishing*, discutidos com mais detalhes neste capítulo.

3.4.2 Ataques Ativos

Ataques ativos são considerados mais sofisticados e exigem maior conhecimento do adversário. Um ataque desse tipo deve ser direcionado aos Usuários de um instituição que se deseja afetar, de maneira a convencer o Usuário legítimo a fornece suas credenciais ou executar alguma operação que de outra forma não faria. Traçar paralelos das ameaças e ataques, bem como definir vulnerabilidades que podem ser contornadas e que podem afetar de alguma maneira o modelo de segurança, diferentemente de transferir responsabilidades da segurança de um serviço a seu Usuário.

O vetor de ataque que se utiliza de técnicas de *phishing* (YU; NARGUNDKAR; TIRUTHANI, 2008) aproveitam-se de uma base de confiança para que tenham sucesso em sua exploração. Esse tipo de ataque basicamente tem como objetivo roubar dados sensíveis que permitam um adversário se passar por um Usuário legítimo.

O conteúdo de um e-mail *phishing* é baseado em informações enganosas, que através de técnicas de engenharia social (MARTINO; PERRAMON, 2008) se apresenta como algo confiável e legítimo, induzindo um Usuário legítimo a que siga as instruções contidas nessas mensagens, com isso, conta com a colaboração não voluntária ou descuidada desse Usuário, ou que decorrente de falta de informação acredita que seja verdade o que lhe é apresentado. É possível dividir esse ataque em dois módulos principais:

1. *Offline*: Onde as informações são roubadas para serem utilizadas sem a necessidade de colaboração do Usuário legítimo. Autenticação estática, onde as informações necessárias para acessar um sistema é sempre a mesma;

2. *Online*: Onde as informações devem ser roubadas ou solicitadas quando algum evento é necessário, exigindo a participação do Usuário legítimo. Autenticação dinâmica, onde as informações necessárias para acessar um sistema é diferente para cada acesso.

3.4.3 Vetores de Ameaças

A dependência dessa exploração é fortemente associada a diversos vetores de ataques, pois como o ataque de *phishing* está na base da árvore, conforme apresentado na Figura 3.2, é associado a outros vetores como inserção de *phishing* em mecanismos de busca, onde a vítima acessa um determinado site como se este fosse legítimo, ou quando uma injeção de links é feita em sites legítimos utilizando ataques baseado em cross-site-scripting, que direciona as vítimas a outros sites. Também os vetores de *spy-phishing* ou *phishing* baseado em códigos maliciosos (*malware*), dependentes de colaboração do Usuário para que este instale algum código malicioso, como cavalo de tróia, que permite a um adversário monitorar as ações da máquina infectada, e coletar as credenciais necessárias para acesso, ou mesmo controlar remotamente a máquina. Potencialmente é possível alterar configurações de acesso a rede de nomes (DNS[7]) e domínios locais, *pharming*, direcionando uma solicitação de conexão para outro host que não o legítimo.

Nos modelos que se utilizam de informações estáticas para identificar um Usuário o *phishing* é extremamente eficiente, entretanto, os ataques associados a técnica de monitorar, criar e modificar uma mensagem é relacionado aos ataques *man-in-the-middle* e *man-in-the-browser* (ANDERSON; BOND, 2009). Nenhum dos métodos recentes de autenticação listados no capítulo 2 e apresentados na Tabela 3.1 são fortes o suficiente para evitar o ataque. Portanto, considera não ser possível proteger um Usuário quanto a esse tipo de ataque, mesmo os que utilizam sistemas de autorização baseados em transações que são executadas em máquinas pessoais do Usuário. Esse tipo de ataque é dependente de um controle de máquina utilizando códigos maliciosos como cavalos de tróia.

Os modelos de segurança que utilizam *PIN*, *TAN*, Certificados digitais, *On-time-password tokens* (OTP), *smartcards*, ou qualquer outro dispositivo inserido na máquina pessoal de um Usuário considerado legítimo é suscetível a esse tipo de ataque. Basicamente, nos modelos considerados mais fortes, como *tokens* OTP ou certificados

[7]DNS (Domain Name System - Sistema de Nomes de Domínios)

digitais, podem ser utilizados sem o conhecimento do *Usuário*, ou no caso de assinaturas cegas, onde um *Usuário* legítimo autentica uma transação visualizada, que na verdade é executada outra, ou seja, o *Usuário* confirma a transação que visualiza, mas a transação real ocorre sem o conhecimento do *Usuário*. Da mesma forma é possível interceptar toda a comunicação entre *Usuário* e Banco, e em alguns casos solicitar que seja informado códigos *OTP*, ou que seja inserido um certificado digital, permitindo ao adversário utilizar todos os recursos esperados.

Os ataques de *man-in-the-middle* e *man-in-the-browser* são utilizados em todo o cenário financeiro (ANDERSON; BOND, 2009), inclusive ataques direcionados a caixas eletrônicos (*ATM*), ponto de atendimento que utilizam *smartcards* e consequentemente usam o protocolo *EMV (Europay, MasterCard e VISA)* padrão adotado nesses cartões para identificar seu Usuário através de um *PIN* e autorizar uma transação (CHOUDARY, 2010).

Os ataques que possuem como vetor o método físico, são considerados tradicionais, onde o roubo de credenciais ou dispositivos físicos ocorrem diretamente no ambiente do Usuário (PAGET, 2007). Portanto, o roubo do próprio dispositivo pessoal do Usuário, seus backups, acesso às informações armazenadas, desvio de correspondência, observância do adversário quando um Usuário acessa um sistema *"shoulder surfing"*, *ATM* disfarçados *"skimming"*, que se passam por um terminal legítimo e que tem como meta roubar senhas e tarjas de cartões magnéticos, recuperação de informações em busca de cestos de lixo *"dumpstar diving"* fazem parte deste vetor.

Existem alguns estudos que se propõem em mitigar o impacto dos ataques relacionados ao vetor *phishing*, como em (SCHMITZ, 2009), que utiliza de uma resposta baseada em *honeypots*, que com um *framework* permite detectar atividades suspeitas e automaticamente direcionar o tráfego para uma rede monitorada. Dessa maneira é possível coletar as tentativas de acesso utilizando credenciais roubadas, impedindo o acesso legítimo. Entretanto o modelo continua sendo reativo, dependendo da identificação de um acesso não legítimo, ainda assim permanece o problema de falsos positivos, que impede o acesso de Usuários legítimos ao sistema, ou mesmo nos falsos positivos, que permitem o acesso fraudulento aos sistemas legítimos.

Pesquisadores da Universidade de San Jose (YU; NARGUNDKAR; TIRUTHANI, 2008) analisaram as vulnerabilidades que esse tipo de ataque explora. Demonstrando claramente que a utilização tem como finalidade o roubo de dados pessoais, como cre-

denciais, números de cartões de créditos, identificação, informações financeiras, tendo como impacto direto em clientes que se utilizam desses serviços, e claro, o impacto financeiro que envolve os ataques.

Outro tipo de ataque associado ao *phishing* e *man-in-the-middle* é o ataque *Control-relaying (RT-MITM)*, que consiste em um adversário capturar as credenciais e utilizá-las de modo automatizado. Para isso a máquina deve estar totalmente comprometida e sob domínio de um malware. Em (LEUNG, 2009) é apresentado um modelo de proteção contra *phishing* utilizando *CAPTCHA*, consiste em desafiar um Usuário a provar que ele é um humano, evitando o ataque automatizado, comumente chamado de robô. Entretanto, um adversário pode ou tentar decifrar o código, se o modelo utilizado for simples o suficiente, ou solicitar que o *trojan* capture a imagem, envie para o adversário, e este por sua vez responda o desafio.

Na proposta de (LEUNG, 2009) uma variável é adicionada, inserindo um segundo fator de autenticação baseado em *OTP*. Ao receber o desafio o Usuário deverá fornecer uma identificação e seu código *OTP* associado, dificultando a ação de um adversário, mas não a impedindo totalmente, pois o Usuário pode ser levado a colaborar com o ataque, através de engenharia social, a fornecer o código *OTP*, ou então, o Usuário pode ser enganado através de uma assinatura cega e confirmar uma transação ilegítima.

3.5 CLASSIFICAÇÃO DE AMEAÇAS

Os ataques são resultados da exploração de uma entidade frágil, que possui algum tipo de vulnerabilidade, que pode ou não ser decorrente de cooperação. Um grande problema é que não existe um padrão que defina o tipo do ataque e qual seu grau de impacto, mesmo porque esse fator pode se diferente para cada contexto.

Os principais desafios são associados ao desenvolvimento seguro e o fator humano, que é diferente para cada contexto (BARNUM, 2007). Pode ser dividido em três níveis distintos:

1. Dificuldade em se construir códigos seguros;
2. Detrimento da segurança em relação a prazos e funcionalidades solicitadas pelo mercado;
3. Diferença entre o conhecimento do desenvolvedor e o adversário.

A construção de códigos seguros torna-se um desafio quando colocado à prova com a procura de vulnerabilidades, que são relativamente fáceis se comparado à produção de um código realmente seguro. Diante desse desafio deve-se analisar as possibilidades de comprometimento de um sistema (PEOTTA, 2008), baseado em probabilidade de ocorrência, ou risco.

O risco de comprometimento, nestes casos, é definido baseado em uma expressão probabilística da ocorrência(BYRES; FRANZ; MILLER, 2004), que é fortemente influenciada por um conjunto de indicadores (PEOTTA; GONDIM, 2011b):

- Dificuldade técnica;
- Custo do ataque;
- Probabilidade de sucesso do ataque;
- Condições esperadas;
- Contramedidas.

Os fatores influenciam diretamente na complexidade do ataque, podendo ser definido inversamente proporcional à sua dificuldade. Entretanto, isso se torna contraditório quando uma complexidade alta é exigida mas a um custo baixo, fazendo com que o ataque tenha uma maior probabilidade de sucesso.

A grande demanda gerada por novas tecnologias de acesso muitas vezes coloca a segurança como um modelo de segundo plano, que gera impacto direto na usabilidade, e que portanto, é considerado como algo impeditivo, mas que é fortemente aparente quando esse impacto torna-se financeiro, gerando perdas com a não observância de regras básicas. Os prazos curtos para que um determinado serviço seja disponibilizado é outro fator de risco, pois a falta de testes mais rígidos gera produtos pouco seguros.

Quando se fala em desenvolvimento de software uma série de requisitos devem ser considerados, o que pode não ser verdade quando o mesmo é colocado em produção. Devido a diversas alterações propositadas ou não durante a condução do projeto o desenvolvimento pode sofrer falhas, que geram vulnerabilidades. O adversário não precisa se preocupar com regras de desenvolvimento ou testes de performance, basta que o mesmo cumpra o requisito de explorar uma falha.

3.6 MODELO DE ANÁLISE DE AMEAÇAS

A importância em se identificar os problemas, que envolvem diretamente a segurança dos processos, auxilia no entendimento para que se possa propor novos modelos e melhorias nesses processos. Identificando os fatores de fragilidade como ameaças e vulnerabilidades é possível classificar cada evento utilizando modelos existentes.

Existem diversas tentativas de classificação de ameaças, basicamente utilizam da análise da taxonomia como discutido em (BARNUM, 2007) *Common Attack Pattern Enumeration and Classification (CAPEC)*, que através da divisão em método de ameaças propõe 11 níveis principais:

1. Abuso de Funcionalidade: Uma determinada aplicação pode ser explorada para que execute ações que normalmente não seria executada ou que não estivesse prevista;

2. Spoofing: Adversário constrói uma mensagem de tal maneira que é capaz de se passar como sendo uma mensagem autorizada, como resultado um servidor pode ser manipulado a responder a mensagem, esse tipo de ataque associa o conteúdo a uma identidade confiável, com isso uma mensagem pode ser alterada e ser considerada legítima;

3. Técnicas probabilísticas: Utiliza-se de modelos de ataques baseado em força bruta ou que utilizam dicionários para obter informações sobre credenciais;

4. Exploração de autenticação: Adversário explora sistemas que não realizam qualquer tipo de autenticação para uma funcionalidade que requer uma identidade de Usuário;

5. Esgotamento de recursos: Recursos são consumidos de maneira a que seja induzido a parada, permitindo que um adversário comprometa a disponibilidade da informação. O resultado de um ataque bem sucedido é geralmente a negação de um ou mais serviços;

6. Exploração de privilégio/confiança: Um sistema pode ser comprometido de maneira a que um adversário obtenha um privilégio mais alto ao que foi fornecido, esse tipo de ataque afeta a integridade das informações, pois se utiliza de um modelo de abuso de confiança;

7. Segurança física: Um adversário tem acesso físico ao local, permitindo que este comprometa de diversas maneiras a estrutura. Terrorismo, vandalismo, espionagem são exploradas;

8. Ataques a Estrutura de dados: Abuso na utilização de estrutura de dados permite que um adversário obtenha informações sem a necessidade de autenticação ou identificação;

9. Vazamento de dados: Adversário utiliza do conhecimento para obter informações e divulgá-las. A coleta de informações pode ser feita analisando mensagens de erros em aplicações, diretórios públicos, relatórios, documentos internos;

10. Manipulação de recursos: Um adversário manipula um ou mais recursos a fim de realizar um ataque, sendo capaz de mudar alguns aspectos do estado de um recurso e afetar o comportamento do sistema ou integridade das informações;

11. Ataques de tempo e estado: Um invasor explora as fraquezas associadas ao tempo e estado de uma aplicação, permitindo executar ações em processos que estejam em atividade. Um ataque de estado pode incluir a manipulação de informações de um aplicativo para alterar credenciais, permitindo acesso a informações que normalmente não seria permitido o acesso.

É importante considerar que um ataque tem certa probabilidade de ocorrer e de ter sucesso, por esse motivo é possível calcular um *score* ou pontuação para cada ataque. Assim como o risco, um ataque pode ser medido pelo seu impacto, que neste trabalho será considerado os pilares da confidencialidade, integridade, disponibilidade, autenticidade e não repúdio (ABNT-B, 2005).

Ataques podem ser associados a um grau de dificuldade para que sejam explorados com certa probabilidade de sucesso, sendo este o indicador mais crítico, pois os cibercriminosos, aparentemente, tem um custo baixo associado a suas ações. No entanto, faz-se necessário um modelo que permita classificar os diversos tipos de ataques de acordo com uma escala que pode ser medida entre 1 a 4, apresentada através da Tabela 3.2.

Utilizando o conceito apresentado na Tabela 3.2, é proposta uma metodologia que permita criticar os modelos atuais que os Bancos utilizam na Internet, sendo possível calcular o score de cada modelo, definindo qual é considerado mais ou menos seguro. A análise dos ataques será feita através de 3 vetores de comprometimento (ver Tabela 3.3).

Tabela 3.2: Escala de classificação de ataques

Escala	**Tipo**	**Descrição**	**Peso**
1	Trivial	Pouco conhecimento e baixa técnica utilizada	4
2	Moderado	Médio conhecimento e técnica necessária	3
3	Difícil	Alta exigência técnica e de conhecimento	2
4	Improvável	Conhecimento necessário está além da capacidade atual dos adversários	1

Tabela 3.3: Vetores de comprometimento

Vetor	**Tipo**	**Descrição**	**Peso**
1	Usuário final	Utilizador do serviço	5
2	Comunicação	Canal de comunicação entre servidor e Usuário final	3
3	Servidor	Provedor do serviço (Banco)	2

Os vetores são importantes como estratégia de defesa, pois permitem identificar pontos críticos de uma solução e apontar a criticidade do impacto na estrutura do modelo.

O *Usuário* final é considerado o elo mais fraco da cadeia de segurança (WONGTSCHOWSKI, 2005). Para esse fator, o peso definido é maior que os outros vetores, devido a maior dificuldade no controle de proteção, pois a confiança nas informações apresentadas é difícil de ser calculada com precisão, e o canal de comunicação, muitas vezes, é explorado com a colaboração do Usuário legítimo.

Tabela 3.4: Vetores de impacto

ID	**Tipo**	**Descrição**	**Peso**
1	Confidencialidade	Divulgação de informações	0,30
2	Integridade	Alteração das informações	0,50
3	Disponibilidade	Serviço disponível	0,20

Considerando a associação dos ataques e ameaças ao impacto gerado, apresentado na Tabela 3.4 é possível determinar um grau de risco através da equação 3.1:

$$(\sum Impacto)(Ataque + \sum comprometimento) \tag{3.1}$$

Onde:

O impacto é determinado pela soma dos pilares apresentados na Tabela 3.4 multiplicado pela soma do peso determinado para cada ataque, apresentado na Tabela 3.2, e a soma dos vetores de comprometimento, apresentado na Tabela 3.3.

Neste momento é possível relacionar as ameaças dos ataques e associar ao vetor de comprometimento de cada tipo de exploração. A Tabela 3.5 apresenta essa relação, com a classificação de cada ameaça sob o ponto de vista da dificuldade em se explorar.

Tabela 3.5: Classificação de Ameaças por Comprometimento em Relação ao Impacto

Tipo de Ameaça	**Classificação**	**Impacto**	**Usuário**	**Comunicação**	**Servidor**
Acesso físico	Trivial	C,I	•	•	•
Alteração host local (DNS)	Moderado	C,I	•		
Autorização de transação	Moderado	C,I	•		
Browser in the middle	Moderado	C,I	•		
Códigos maliciosos	Moderado	C,I	•		
Comprometimento servidor *Internet Banking*	Improvável	C,I			•
Engenharia social	Moderado	C,I	•		
Exploração de vulnerabilidades	Difícil	C,I,D	•	•	•
Forçaa bruta	Trivial	C			•
Injeção de transações	Difícil	I		•	
Man-in-the-middle	Moderado	C,I		•	
Monitoração	Moderado	C	•	•	
Pharming - envenenamento DNS	Difícil	C,I		•	
Phishing	Trivial	C,I	•		
Páginas falsas	Trivial	C,I	•	•	
Roubo de código dinâmico	Moderado	C,I	•		
Roubo de código estático	Trivial	C,I	•		
Roubo de dispositivo físico	Trivial	C,I	•		
Roubo de sessão	Difícil	C,I		•	
Roubo de *soft-tokens* (certificados, otp soft)	Moderado	C,I	•		
Sniffing (monitoração de rede)	Moderado	C		•	
Violação de política de segurança	Difícil	C,I,D			•

Legenda: C-Confidencialidade I-Integridade D-Disponibilidade

Considerando que cada ataque possui três possíveis eventos relacionado ao impacto, confidencialidade (C), disponibilidade (D) e integridade (I), que afetam diretamente a informação, tem-se um espaço amostral Ω. De modo que possam ser inter-relacionados como idempotentes $C \cap C = C$, $C \cup C = C$, comutativas $C \cap D = D \cap C$, $C \cup D = D \cup C$ associativas, onde $C \cap (D \cap I) = (C \cap D) \cap I$, $C \cup (D \cup I) = (C \cup D) \cup I$. Por esse motivo é possível determinar que os eventos de ataques (A) são relacionados aos eventos de impacto (I), conforme o espaço amostral Ω para os ataques $A_1, A_2..., A_n$ (equação 3.2).

$$A_i \neq \phi, i = 1, ..., n \tag{3.2}$$

A probabilidade de um determinado ataque (A) obter sucesso pode então ser definida pela equação 3.3, considerando um evento Φ seja um evento possível, caso contrário $P(\Phi) = 0$.

$$\sum_{i=1}^{n} P(A_i) = 1 \tag{3.3}$$

Para a probabilidade de um único evento ocorrer a equação 3.3 atende o requisito, entretanto, para mais de um evento simultâneo é necessário adicionar cada evento, como representado pela equação 3.4.

$$P\left(\bigcup_{i=1}^{n} A_i\right) = \sum_{i=1}^{n} P(A_i) - \sum_{i \neq j}^{n} P(A_i \cap A_j) \tag{3.4}$$

Nos casos dos eventos serem sequenciais é possível determinar sua ocorrência baseada em um modelo simplificado de multiplicação de probabilidade, portanto, para os eventos Ataque (A), Classificação (C), Impacto (I) e o vetor de comprometimento (V), têm-se a seguinte equação 3.5.

$$P(A)P(C|I)P(V) = 1 \tag{3.5}$$

Nos modelos apresentados no capítulo 2, é possível criticar as soluções adotadas, de modo a perceber o nível de segurança exigido onde, novas soluções propostas atendem um nível forte de proteção e o custo de implementação compense os gastos gerados com esses modelos.

3.7 DESCRIÇÃO E TAXONOMIA DOS ATAQUES

Nesta seção será apresentado, formalmente, a taxonomia de dois vetores de ataques que possibilita ao adversário ultrapassar a segurança dos modelos atuais. Também se apresenta um estudo detalhado do vetor de códigos maliciosos, considerando uma extensa base analisada e que permitirá analisar o cenário atual e prospectar novos modelos futuros.

Muitos problemas associados à segurança dos Usuários de um sistema de Internet, são dependentes de uma vasta quantidade de códigos maliciosos e ferramentas disponíveis. Vetores como *spam*, negação de serviço[8], *botnets*, *phishing*, *worms* e outros tipos de ataques são geralmente associados aos malwares (BAILEY et al., 2007). Nesta pesquisa foi identificado o vetor principal do ataque *trojan*, que é um código malicioso usado para obter acesso aos computadores de suas vítimas, possuindo outro vetor, a associação através de *phishing*, utilizado para infectar o computador do Usuário, geralmente explorando técnicas de engenharia social para obter sucesso.

Confrontando a hipótese de que o vetor trojan é a principal fonte dos problemas do mercado financeiro, sendo possível comprovar a veracidade do fato analisando este vetor. Uma base de 6.051 artefatos maliciosos, coletada entre janeiro e julho de 2011, com a análise foi possível chegar a conclusões importantes para o entendimento do cenário atual de segurança bancária.

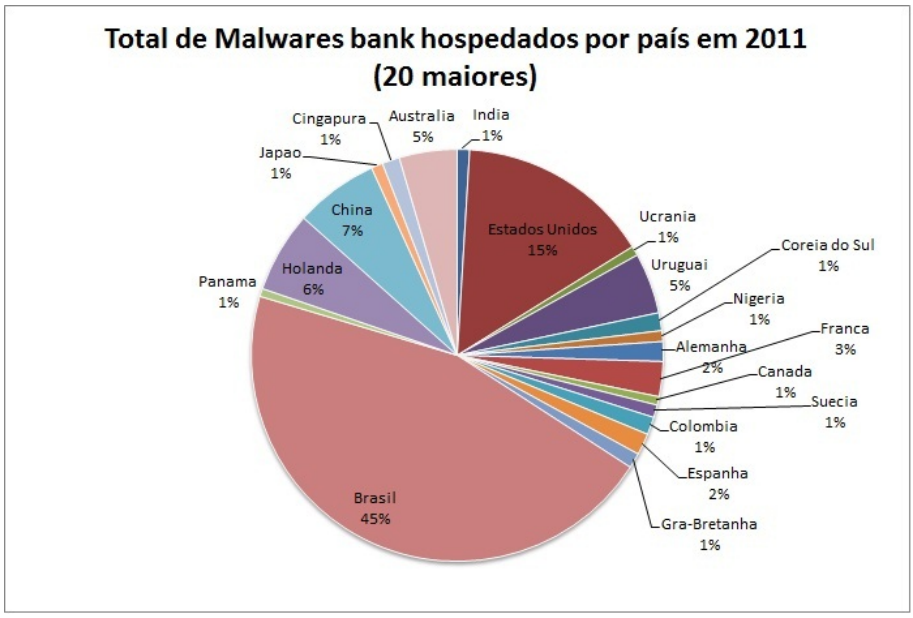

Figura 3.3: Malwares hospedado por país - Primeiro Semestre de 2011

Analisando os dados apresentados pela Figura 3.3 conclui-se que a maioria dos artefatos maliciosos, cerca de 45%, que buscam explorar os Usuários do sistema de *Internet Banking*, são hospedadas no Brasil. Algumas conjecturas podem ser formadas, por exemplo: dificuldade em se bloquear um link interno, falta de consciência de algumas empresas brasileiras que tem seus servidores invadidos, falta de leis específicas para se punir um invasor, entre outras.

[8]denial-of-service attack (DoS attack) ou distributed denial-of-service attack (DDoS attack)

Os artefatos, tem a capacidade de monitorar todas as atividades de sua vítima, utilizando funções para roubo de credenciais, certificados digitais, senhas, códigos estáticos e dinâmicos. Incrementando constantemente a dificuldade de detecção desses artefatos pelos softwares de proteção, como antivírus, antimalwares, proteção de navegadores, entre outros.

A dificuldade enfrentada pelos softwares de proteção estão associadas diretamente à detecção do código malicioso. É preciso, no menor tempo possível, detectar, analisar, corrigir, divulgar a correção e aguardar que o Usuário atualize a aplicação, isso se existir um software de proteção disponível e operacional, pois uma das ações do código malicioso é desabilitar as proteções, de maneira a impedir sua atualização ou ação direta.

Hipótese 1 *Comprovar a hipótese de que os módulos de proteção são incapazes de proteger um Usuário em um tempo suficientemente aceito.*

Cenário 1 *Utilizando uma amostra de 86 artefatos considerados maliciosos, disponível no Anexo A), coletados no mesmo dia de sua detecção por entidades de segurança e centros de resposta a incidentes, onde cada artefato, individualmente, foi submetido ao crivo de análise de 42 antivírus de mercado durante 10 dias contínuos.*

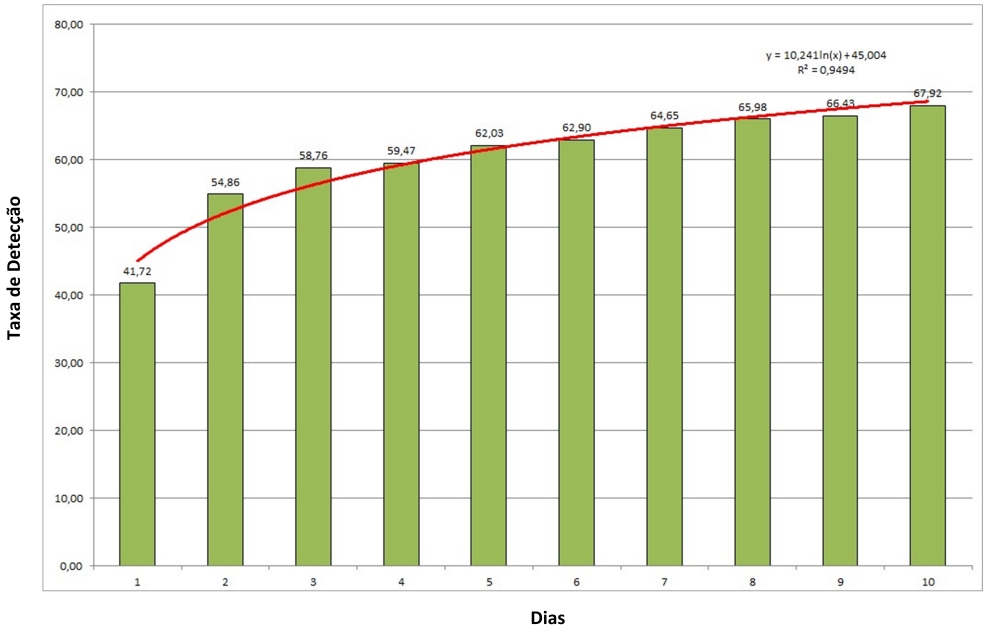

Figura 3.4: Comparação entre detecção e resposta baseado no tempo de análise da amostra

Como conclusão da hipótese 1 tem-se apresentado através da Figura 3.4.

Para cada dia, do total de 10 dias da amostra, foi analisado cada artefato individualmente, sendo coletada a resposta se o mesmo foi detectado ou não, por cada antivírus individualmente.

Neste total foi observado 1247 eventos, e traçado uma paralelo utilizando uma curva logarítmica da mediana do dia analisado e o valor da quantidade de detecção para cada artefato. A prova de conceito utilizada para gerar os eventos está disponível nos Anexos B.1 e B.2.

A média de detecção dos artefatos foi mínimo de 3 e máximo 40, observando os dados do dia 1 comprovou-se a menor média de detecção, 41,72% dos antivírus detectaram o artefato como sendo malicioso, e ao fim da análise da amostra, foi possível obter o valor médio de detecção dos artefatos, que foi de 60,47%.

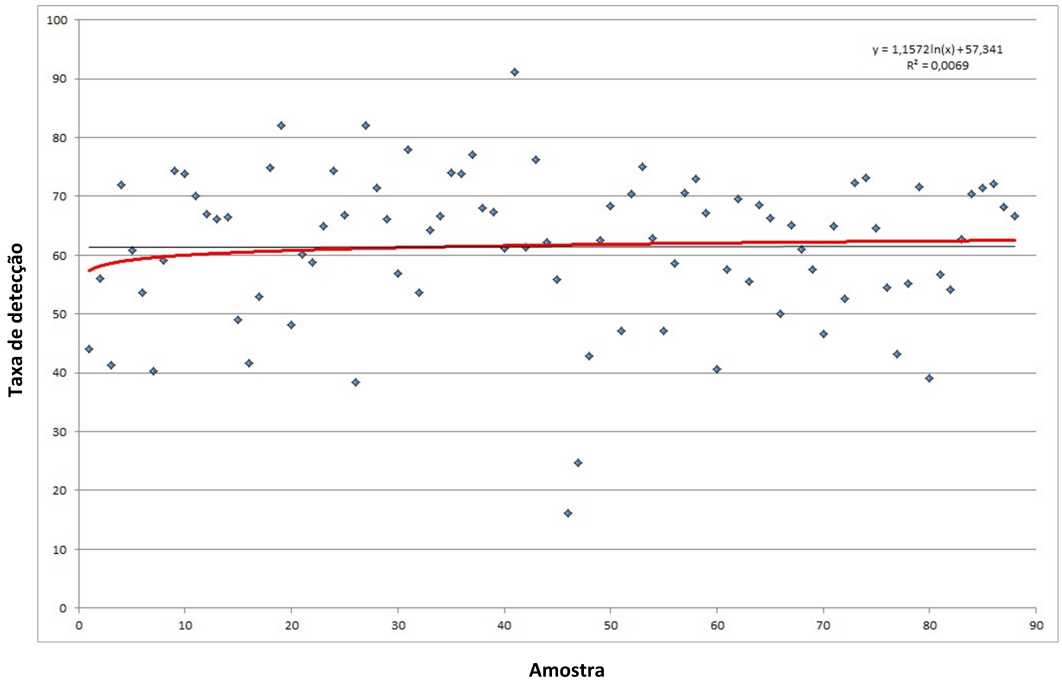

Figura 3.5: Comparação do total geral da amostra por detecção do artefato

A Figura 3.5 apresenta o resultado da análise utilizando como vetor o artefato, de maneira que se possa observar o total geral definido no espaço da amostra final. A curva de tendência apresentada, defini que, alguns artefatos analisados, ficaram abaixo da média de detecção, sendo que o esforço necessário para se proteger um Usuário acaba tornando-se ineficiente, pois ao final da análise o artefato, muitas vezes, não se encontrava mais disponível na rede. Portanto, o tempo de vida desse artefato ainda é

maior que o tempo de análise e correção, permitindo que o adversário, ao obter sucesso em explorar um Usuário, consiga em prazo menor, capturar as informações necessárias a sua utilização do sistema como sendo algo legítimo.

Através do gráfico da Figura 3.5, nota-se que a constelação apresenta informações importantes sobre o foco de uma análise, que ao invés de ser feita de modo geral, ou seja, todos os artefatos são analisados individualmente, de modo indiscriminado e sem qualquer método de qualificação de prioridade, desse modo seria possível identificar quais amostras podem ser analisados com um peso maior de prioridade, que passa a ser a baixa taxa de detecção de algumas amostras, que estão abaixo da faixa proposta da mediana.

Em casos recentes, foi possível identificar dois tipos de ataques usualmente utilizados com essa finalidade. Nas seções 3.7.1 e 3.7.2 é apresentado e discutido, tanto a taxonomia dos ataques e sua exploração, quando a associação aos modelos de segurança adotados e discutidos no capítulo 2.

3.7.1 Ataque de Personificação

Este tipo de ataque é o mais comum e relativamente simples de ser elaborado. Consiste basicamente em obter as credenciais e informações necessárias para que o adversário tenha a capacidade de personificar um Usuário legítimo, reproduzindo informações que somente um Usuário legítimo deveria possuir.

A personificação não sugere a utilização de esquemas robustos de identificação, onde existe um segredo aleatório de um protocolo de desafio e resposta, sendo necessário que o Usuário prove sua identidade através do que se sabe, ou no caso de alguns modelos bancários, o que se tem. Entretanto, associa-se aos modelos estudados um segredo compartilhado e de conhecimento de ambos, diferentemente de um sistema assimétrico, onde cada entidade conhece um segredo não compartilhado.

Provar a identidade através do emprego de uma prova *zero-knowledge* de conhecimento nulo, tem sido um paradigma aceito para os protocolos de identificação (BELLARE; FISCHLIN; MICALI, 2001). Entretanto esse paradigma não afeta os ataques, que se vale da exploração de um elo fraco, no caso um Usuário legítimo. A Figura 3.6 demonstra esse tipo de ataque, utilizando como vetor o *phishing*.

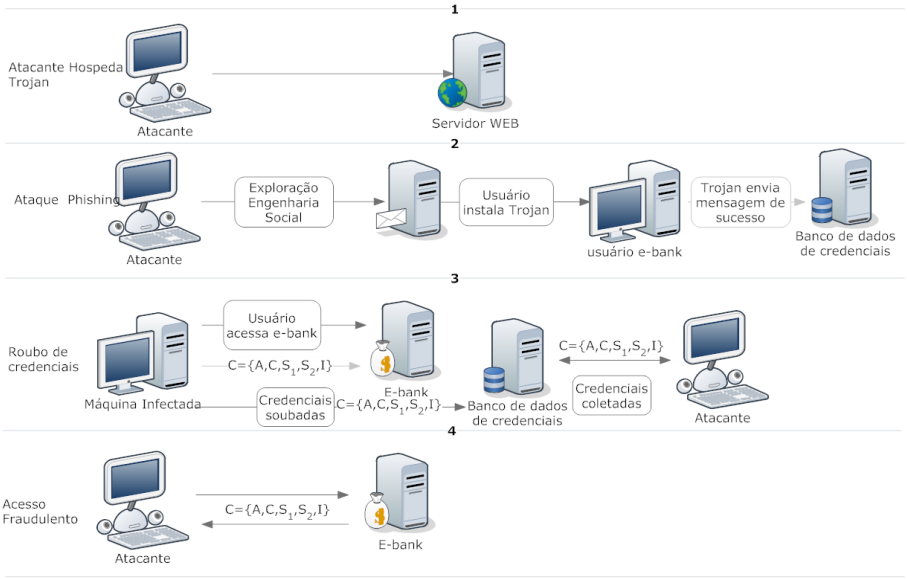

Figura 3.6: Personificação de Usuário legítimo

Teorema 3 *Dado o conjunto de credenciais* $C = \{A, C', S_1, S_2, I\}$*, tem-se* A *e* C' *o conjunto identificados da credencial,* S_1 *e* S_2 *segredo simétrico para autenticação de um Usuário,* I *equivale a qualquer código identificador de hardware que permite associar um determinado Usuário a um equipamento considerado único para acesso.*

O adversário será capaz de executar transações arbitrárias quando obter C de modo a reproduzir I quando necessário. Considera-se que o adversário possa tentar obter $S' = S_1, S_2 \in \{0,1\}^n$ em tempo polinomial escolhendo aleatoriamente $f(S')$ pode-se deduzir a probabilidade $P(S') = \frac{1}{2^n}$. Entretanto, como S_1 e S_2 são segredos estáticos e não protegidos, o adversário pode capturar essas informações e utilizar posteriormente.

A taxonomia desse ataque é caracterizada de maneira que um adversário consiga comprometer uma máquina de um Usuário legítimo, dividida em 4 etapas, até o acesso indevido:

Etapa 1: O adversário disponibiliza um artefato malicioso ou página falsa na Internet;

Etapa 2: Explorando técnicas de engenharia social ou vulnerabilidades em softwares como navegadores, a máquina do Usuário é infectada com algum tipo de malware. Ao acessar o serviço, o Usuário informa suas credenciais acreditando que esteja informando a um serviço legítimo, quando na verdade está em um ambiente controlado pelo adversário, o qual armazena essas informações em provedor externo;

Etapa 3: Após obter as informações necessárias para acessar o *Internet Banking*, o Usuário é direcionado ao serviço legítimo, geralmente sem apontar nenhum erro, mas exige que o Usuário refaça os procedimentos de autenticação. O adversário verifica quais informações foram coletadas;

Etapa 4: De posse de todas as informações necessárias, o adversário se apresenta ao serviço como se fosse o Usuário legítimo, personificando todas suas características, inclusive características físicas de identificação, que são utilizadas por alguns Bancos como uma maneira de se identificar unicamente cada Usuário.

Neste contexto, se o computador pessoal do Usuário, que acessa o site remoto, estiver comprometido, a inserção de qualquer solução, que dependa deste dispositivo, não adicionará segurança ao ambiente já comprometido.

Suponha que um adversário possa executar transações arbitrárias de maneira que o Banco confie nas mensagens como se as mesmas fossem enviadas pelo Usuário legítimo.

3.7.2 Ataque de Controle de Dispositivo

No ataque de controle de dispositivo, apresentado na Figura 3.7, nota-se que existe um nível de complexidade maior que o ataque de personificação, apresentado na Figura 3.6. Entretanto, assemelham-se bastante ao controle de dispositivo permite uma personificação de Usuário, sendo que as mensagens, além de serem autenticadas com informações que somente o Usuário legítimo sabe, também utilizam o que um Usuário tem, no caso um computador pessoal conhecido e identificado. Portanto, rastrear a fonte do ataque é considerado difícil, pois em momento algum o adversário conecta-se diretamente ao Banco, utilizando totalmente a máquina pessoal do Usuário para executar transações arbitrárias.

Esse tipo de ataque interfere na autenticidade das mensagens enviadas, afetando uma das premissas de segurança da informação, já que basicamente as instituições financeiras buscam encontrar uma maneira de identificar o Usuário e autorizar o acesso, tratando de dois pilares: o que se sabe (senhas e identificadores únicos) e o que se tem (algum tipo de hardware).

A taxonomia desse ataque é caracterizada de forma a permitir que um adversário comprometa uma máquina de um Usuário legítimo, e que o mesmo tenha a capacidade de

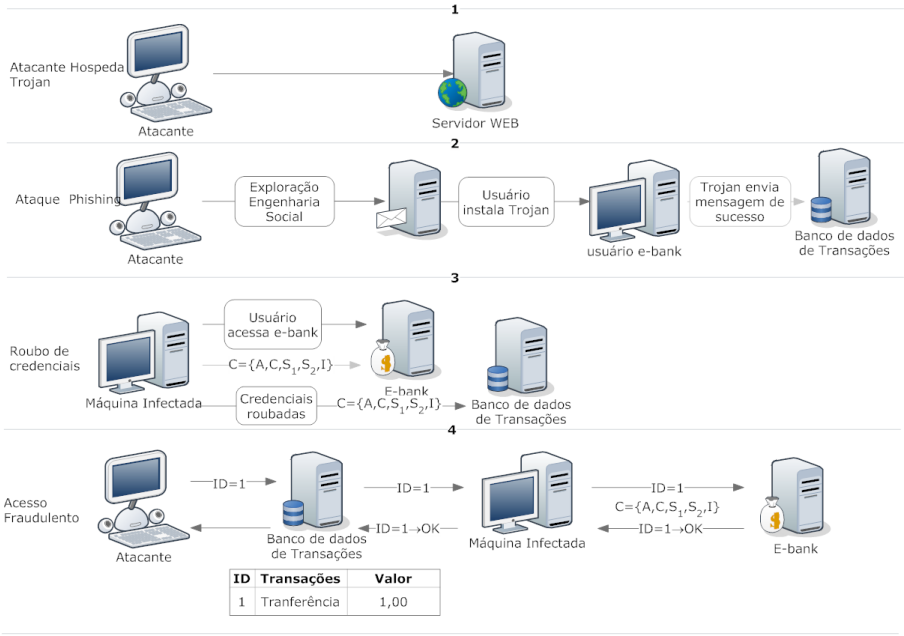

Figura 3.7: Controle de dispositivo

executar transações arbitrárias diretamente do equipamento. Pode-se dividir o modelo do ataque em 4 etapas:

Etapa 1: O adversário disponibiliza um artefato malicioso ou página falsa na Internet;

Etapa 2: Explorando técnicas de engenharia social ou vulnerabilidades em softwares como navegadores, a máquina do Usuário é infectada com algum tipo de malware, que aguarda o acesso ao serviço de *Internet Banking* e entra em ação coletando as informações das credenciais e tudo o mais que for necessário para efetuar a autenticação;

Etapa 3: De posse das informações necessárias para realizar as transações o malware (através da máquina do Usuário) verifica em um repositório na Internet, previamente alimentado pelo adversário, quais transações devem ser realizadas;

Etapa 4: Adversário utiliza malware que, de posse das credenciais do Usuário, autentica-se no Banco, executa as transações ilegítimas diretamente da máquina do Usuário, sem intervenção humana. O Usuário ao acessar o serviço, enquanto o *malware* permanecer ativo terá acesso às informações sobre as transações ilegítimas suprimidas, o que dificulta a detecção por parte do Usuário de qualquer anormalidade. O adversário atualiza a base do repositório de transações com outras transações a serem efetuadas.

3.8 TRABALHOS RELACIONADOS

No que abrange o uso de técnicas e modelos de autenticação com múltiplos fatores, as organizações financeiras estão na vanguarda da aplicação e utilização destas tecnologias. Nesta seção será apresentado os trabalhos mais relevantes que tem relacionamento com este estudo, o que pode aproximar mais da área de autenticação do que para uso em *Internet Banking* propriamente dito, isso se deve pelo fato de que muitas pesquisas desenvolvidas no âmbito das instituições financeiras podem não estar disponível para consulta, seja pelo motivo de se manter sigilosa a informação ou pelo fato de não ser conveniente sua publicação.

3.8.1 Métodos Para Autenticação Com Múltiplos Fatores

Autenticação em canais digitais é um processo que busca identificar um Usuário de modo a permitir um acesso a uma rede específica ou mesmo a um serviço. É recomendado que mais de um modelo de identificação seja utilizado (BURR et al., 2006), como discutido na seção 2.4.2 sobre a importância de um desafio forte.

Segundo o relatório sobre a influência de adoção de múltiplos fatores de autenticação (LIBICKI et al., 2011), a bastante tempo um fator somente não provê um modelo adequado. Os benefícios dos modelos de múltiplos fatores são que adversários externos e internos necessitam de maior esforço para quebrar simultaneamente duas ou mais barreiras.

Para sistemas de *Internet Banking* é preciso identificar, autenticar e autorizar pessoas e mensagens, portanto, deve ocorrer desafios fracos a fortes. Uma estratégia é identificar um Usuário através de algo que se saiba (desafio fraco[9]), com isso o sistema é simples de ser empregado e fácil de se comprovar a identidade, pelo menos o que se espera, mas mesmo assim não deve ser mandatário esse requisito. Ao se identificar o Usuário é possível lhe dar poder para acessar as informações, mas não de transferir qualquer informação a outro Usuário, isso deve ser feito com um desafio considerado forte.

A utilização de fatores externos ao canal de autenticação tem se mostrado promisor. Com a massificação de celulares inteligentes algumas propostas têm surgido, como um modelo que utiliza senhas dinâmicas (OTP) (YOUNG et al., 2010) e códigos bidimensionais baseado em resposta rápida (QRCode [10])(ISO/IEC 18004, 2006), que é

[9] Qualquer desafio que utilize um fator de autenticação

[10] Abreviação para Quick Response Code (Código de Resposta Rápida)

utilizado para transferir uma semente que é armazenada localmente e processada com a utilização de uma janela de tempo definida pelo provedor do serviço. A ideia dos autores é basicamente a mesma de possuir um PIN dinâmico.

(STARNBERGER; FROIHOFER; GOESCHKA, 2009) apresentam um trabalho baseado nos mesmos princípios do uso de códigos 2D intitulado QR-TAN, nesta pesquisa os autores incluíram um número de transação para cada mensagem, entretanto, o desafio da resposta é necessário que exista conectividade entre o dispositivo móvel e o servidor, para evitar ataques de replay é utilizado um valor aleatório nonce a fim de gerar mensagens diferentes. Os autores propõem o uso de um cartão inteligente (*smartcard*) para que sua proposta possa ser *offline*.

O sistema de autenticação baseado em desafio resposta para aplicações web chamado Snap2Pass (DODSON; DEBANGSU; BONEH, 2010), tem como proposta cifrar uma mensagem através de uma chave secreta pré compartilhada e apresentar o código através de uma imagem bidimensional, onde o Usuário utilizando um celular inteligente poderá decifrar a mensagem e enviar a resposta através do canal de dados da operadora de telefonia, autenticando-se a um serviço como uma solução de identificação aberta(RECORDON; REED, 2006).

Existem propostas que utilizam *tokens OTP* baseados no envio de mensagens curtas (SMS). Essa é a proposta do trabalho (JORSTAD; JONVIK, 2009), que desafia o Usuário a se autenticar através de resposta a uma mensagem (token) que a operadora de telefonia envia. O Usuário então responde a mensagem através de um dispositivo conhecido que pode ser comprovado através do identificador único que cada operadora possui para identificar seus Usuários.

3.8.2 Síntese dos Trabalhos Relacionados

Em todos os trabalhos é possível identificar que um dos problemas é a exigência de que o Usuário possua conectividade a uma rede de dados. Também é possível considerar que, abordagens que não permitam associar um segredo a uma mensagem fracassam em ataques conhecidos como *man-in-the-middle* ou ataques de *replay*, que permitem a utilização de um segredo para confirmar outras mensagens.

Modelos que se baseiam em autenticar uma transação podem falhar quando estas não estão associadas ao segredo, desse modo o termo mais correto é autorizar uma transação. Essa técnica necessita, como premissa fundamental, que o Usuário já esteja

autenticado e devidamente reconhecido pelo sistema. Neste caso, qualquer desafio resposta pode ser utilizado, inclusive os fatores considerados mais fracos, como o caso da utilização de desafios de prova do que se sabe (ex. Credencial e senha).

A seguir será apresentado o protocolo que se propõe a resolver o problema de autorização de mensagens através do uso de um segundo fator externo ao canal utilizado. Nesta proposta, não é exigido qualquer tipo de conexão na comunicação entre servidor e cliente, de modo que ataques conhecidos atualmente se tornem ineficientes.

4 PROTOCOLO DINÂMICO PARA AUTORIZAÇÃO SEGURA OUT-OF-BAND

Neste capítulo é apresentado o modelo para sistemas de *Internet Banking* seguro, que é composto por um conjunto de protocolos criptográficos que busca garantir integridade, autenticidade e sigilo das operações. A segurança dos protocolos é comprovada em hipótese padrão e o modelo adversarial, discutido na seção 3.3.

Os protocolos, aqui propostos, são considerados práticos, uma vez que podem ser implementados exigindo baixo poder computacional e mínima interação do Usuário. Apesar da solução utilizar como base primitivas criptográficas e protocolos de comunicação, é necessário considerar vários fatores, tais como interação com o Usuário baixa , além da largura de banda limitada e poder computacional. Estas restrições impõem limites claros sobre o tipo de primitivas e os protocolos utilizados, no sentido de que uma solução prática deve possuir criptografia forte e praticamente exequível para o Usuário.

Neste modelo, cada operação realizada através do canal *Internet Banking* é considera única e deve ser autorizada pelo Usuário legítimo, em vez de confiar nos dados recebidos durante uma sessão autenticada (e consequentemente realizar transações em um contexto como de sessão). Para cada transação realizada, o Usuário é desafiado a provar a sua identidade antes da sua autorização, além disso, cada prova de identidade fornecida pelo Usuário deve ser estritamente associado a uma transação específica, protegendo contra os ataques descritos no capítulo 3.

Isto é conseguido através de um protocolo de autorização de transação, que é um protocolo de identificação alterado. Se o Usuário for o legítimo proprietário de uma chave secreta trocada com o Banco, respondendo aos desafios estritamente associados às operações específicas. O Usuário também é capaz de verificar a autenticidade dos desafios enviados pelo sistema bancário através de um código de autenticação de mensagem, discutido na seção 2.4.3, e uma chave de autenticação previamente trocados com o Banco. Estes desafios contem informações sobre a transação para que seja autorizada e uma mensagem de confirmação única que o Usuário precisa para decifrar ou assinar com sua chave secreta, a fim de autorizar a operação com sucesso. Ao receber um desafio, o Usuário é capaz de verificar se ele foi realmente enviado pelo Banco e, em

seguida, verificar a operação que será autorizada. Se o desafio é autêntico, considera que a operação foi solicitada pelo Usuário, então ele pode optar por realizar a operação desejada (decifrar ou assinar) sobre a mensagem e enviá-lo para o Banco, que autoriza a transação. Cada mensagem única está associada a uma operação específica, permitindo ao Banco determinar exatamente qual a transação que o Usuário está autorizando.

Seria impraticável para que o Usuário memorizasse as duas chaves, interpretar os desafios enviados pelo Banco e computar as operações necessárias de criptografia na mensagem de confirmação. Assim, é necessário realizar essas ações em um dispositivo específico. Se o adversário controlar o dispositivo, ele será capaz de autorizar operações arbitrária, uma vez que teria acesso a ambas as chaves.

Considerando o modelo de adversarial apresentado na seção 3.3, as operações são emitidas através do computador pessoal do Usuário e autorizado através do dispositivo móvel do Usuário (possivelmente um celular *smartphone*).

Considerando que todas as operações de autorização de transações são realizadas por um aplicativo previamente instalado no aparelho celular do Usuário. Depois que uma transação é executada através do computador, o sistema de *Internet Banking* envia o desafio autorização correspondente para o dispositivo móvel (celular) por meio de um canal de comunicação, apresentando as informações através de códigos 2D. O aplicativo verifica se o desafio é autêntico e apresenta os dados da transação. O Usuário confirma as informações e pode escolher autorizar a transação na aplicação do *smartphone*, que realiza a operação necessária (decifrar) sobre a mensagem de verificação e apresenta o resultado. O Usuário insere a mensagem resultante no site de *Internet Banking*, que autoriza a transação.

Mesmo que o adversário seja capaz de realizar operações através das técnicas descritas no capítulo 3, ele não pode autorizá-la, pois não é capaz de realizar qualquer ação no *smartphone*. O adversário é então forçado a utilizar técnicas de engenharia social, e buscar enganar o Usuário (BYERS; SHAHMEHRI, 2010), de modo que autorize uma operação ilegítima, mesmo após a verificação da autenticidade dos desafios e os dados da transação. Esta abordagem impede todos os ataques comuns realizadas pelos adversários e descritos no capítulo 3, uma vez que o adversário não é capaz de autorizar a operação arbitrária. Mesmo que o adversário seja capaz de realizar uma transação ilegítima, não é capaz de alterar arbitrariamente os desafios enviados pelo Banco (o que poderia causar um erro de verificação de autenticidade). Uma vez que o adversário

não é capaz de modificar o desafio, o Usuário pode verificar os dados da transação e verificar se a operação não foi solicitada por ele mesmo e consequentemente abortar o processo.

Os sistemas de *Internet Banking* são complexos, com base em diferentes componentes que o compõem, como hardware, software e infraestruturas de comunicação, o que permite que os adversários monitorem e até modifiquem dados em diferentes pontos da comunicação. Até mesmo os modelos de ataques clássicos presentes na literatura atual não contemplam adequadamente as peculiaridades específicas dos sistemas de *Internet Banking*.

Nesta proposta, o autor considerou a aplicação de quatro conceitos fundamentais de sucesso, para que um modelo de segurança seja forte o suficiente para suportar um ataque, e que desestimule o adversário para que tenha que obter a colaboração do Usuário toda vez que submeter uma mensagem, a saber:

1. Todo equipamento de acesso está ou estará comprometido;
2. Todo e qualquer dispositivo inserido no ambiente comprometido passará a fazer parte do comprometimento;
3. Não é possível proteger a aplicação;
4. Não é possível proteger o Usuário.

Diante dos quatro conceitos fundamentais descritos aqui, é possível traçar um paralelo, onde praticamente a proteção baseado na reação torna-se difícil e ineficiente. A demora em se identificar um ataque é diretamente associado ao seu comprometimento. Mesmo nos modelos de identificação baseado no que o Usuário tem, pode ser utilizado a fim de se ganhar confiança nos sistemas e assim efetuar transações a partir do equipamento ou aplicação comprometida.

Conforme apresentado no capítulo 3, o tempo de resposta em se detectar um ataque e criar uma proteção para o Usuário não satisfaz o modelo de segurança. O mesmo acontece com as aplicações dos sistemas, que podem ser fragilizadas por uma ação intencional ou não por parte do Usuário, que mesmo não seguindo os requisitos de segurança em que os Bancos disponibilizam, não devem ser a eles imputado a responsabilidade de manter o sistema seguro.

Outra consideração é que sistemas comprometidos são difíceis de controlar, isso ocorre pelo fato de todas as mensagens trocadas entre as entidades poderem ter sido alteradas ou produzidas sem o conhecimento do Usuário legítimo. Novos modelos adotados pelos Bancos, e discutidos no capítulo 2, apresentam uma tendência de confiança irrestrita no Usuário. Os modelos de autenticação atuais sempre utilizam o canal solicitante para confirmar uma mensagem, permitindo que o adversário, que controla a origem, apresente-se entre a entidade Banco e Usuário legítimo, utilizando qualquer dispositivo que possa a ser inserido, alterando ou produzindo mensagens com ou sem participação do Usuário.

4.1 DEFINIÇÕES

Neste modelo formal, assume-se que o protocolo DAP possui os seguintes requisitos:

- $\kappa = \{0,1\}^{288}$ chave criptografica simétrica do Usuário dividida em k_1' e k_2'.
- $k_1' \in \{0,1\}^{160}$ utilizada para garantir a integridade da mensagem.
- $k_2' \in \{0,1\}^{128}$ utilizada para garantir a confidencialidade da mensagem.
- $k_1 = \{0,1\}^{288}$ sequência aleatória.
- $k_2 = \{0,1\}^{288}$ chave simétrica definida por $k \oplus k_1'$.
- $MK_{k_1}, ...MK_{k_n}$ conjunto de chaves mestre onde $MK \in \{0,1\}^{256}$ e é obtida através de $MK \leftarrow Gen(1^n)$.
- $k \in \kappa$ para cada Usuário é gerada $k \leftarrow Enc_{mk}(ID|timestamp)$, utilizando-se como premissas qualquer identificador único ID associado a uma data e hora $timestamp$ determinada pelo sistema bancário.
- $r \in \{0,1\}^{64}$ código autorizador aleatório associado a determinada mensagem m.
- m mensagem ou transação a ser transmitida.
- $m' = E_k(m)$ cifrar uma mensagem ou transação m a ser transmitida.
- $D_k(m')$ decifrar uma mensagem ou transação m' recebida.
- $m' = m$ identificação da mensagem ou transação.
- $h_k(m)$ hash de mensagem ou transação utilizando chave k.

- $r' \leftarrow Enc_{k_2'}(r)$ cifrar um código autorizador r com chave k_2'.
- $r \leftarrow Dec_{k_2'}(r')$ decifrar o código autorizador r' com chave k_2'.

A proposta do protocolo inclui o modelo de autorização baseado em uma mensagem associada a um código autorizador aleatório, onde um segredo é compartilhado entre as duas partes interessadas. Um sistema simétrico consistente de algoritmos de execução em tempo polinomial ($\texttt{Gen}, \texttt{Enc}, \texttt{Dec}$).

Assumindo que um Usuário obtém uma chave $k \in \kappa\{0,1\}^{288}$ dividida em duas partes $k_1' \in \{0,1\}^{160}$ e $k_2' \in \{0,1\}^{128}$ (Figura 4.1) , sendo a porção inicial de k_1' associada a integridade e autenticidade da mensagem e a segunda parte k_2' associada a confidencialidade do código autorizador $r \in \{0,1\}^{64}$ de uma determinada transação. Assume-se que $Gen(1^n)$ tem como saída k, satisfazendo o espaço amostral de $k = \{0,1\}^{288}$.

1 | 10	21 | 36
Chave de HMAC	Chave de AES ou 3des
160 bits (20 Bytes) – SHA1	128 bits (16 Bytes)

Figura 4.1: Formatação e divisão da mensagem de k em k_1' e k_2'

Um dispositivo externo é utilizado como um segundo fator para autenticação, neste caso, um dispositivo móvel (celular) capaz de obter uma imagem através de sua câmera e que possua a capacidade de executar uma aplicação dedicada e com um conjunto de processamento matemático mínimo. Este dispositivo, ao receber informações da transação realizada, averiguará a autenticidade e integridade dos dados e decifrará estes dados, com o intuito de emitir um código autorizador, a qual autorizará uma transação específica.

Um Banco que dispõe de serviços para atendimento a clientes na *Internet*, deve considerar o conceito de autenticação de transação, permitindo que a informação seja confirmada externamente ao dispositivo que esteja se comunicando, neste caso, um computador conectado. Dessa maneira é possível autorizar uma mensagem de forma que a mesma esteja integra e o que se esta vendo é o que realmente deve ser autorizado.

4.1.1 PREMISSAS NECESSÁRIAS AO DAP

Na proposta considera-se um sistema composto de duas partes, um servidor remoto para transação financeira e o Usuário, que podem interagir entre si através de três

diferentes dispositivos conectados a seus respectivos canais:

1. Terminal de auto atendimento (ATM);
2. O computador pessoal do Usuário ou outro dispositivo qualquer que está conectado à *Internet* (CNC - Computador não confiável);
3. Dispositivo móvel (celular *smartphone*) do Usuário.

Considera-se uma situação na qual um Usuário se conecta a um servidor remoto e necessita fornecer suas credenciais para se autenticar no mesmo e que um adversário, que está disposto a executar operações arbitrárias, tenha as seguintes características:

- O adversário tem controle completo sobre o computador pessoal do Usuário, através de um programa ou código malicioso que foi instalado e/ou executado no computador pessoal do Usuário. Portanto, o adversário é capaz de monitorar todas as mensagens trocadas, no canal *Internet*, pelas duas partes (Usuário e Banco) e arbitrariamente interferir na comunicação, podendo modificar, inserir, excluir qualquer informação.
- O adversário é capaz de monitorar e capturar todas as informações fornecidas pelo Usuário;
- O adversário não possui controle sobre o terminal de auto atendimento do Banco (*ATM*), mas o mesmo pode obter informações que são apresentadas na tela e dados fornecidos através interação direta do Usuário no terminal;
- O adversário não tem capacidade de obter qualquer informação diretamente do dispositivo móvel (celular *smartphone*) do Usuário;
- O adversário não tem como obter a posse do cartão de identificação do Usuário (*smartcard*).

Proposição 1 *Suponha que o Usuário deseja enviar ao Banco uma mensagem m utilizando o modelo proposto por esta tese, e que um adversário pode acessar a mensagem m.*

O segurança do modelo será garantida se o adversário não for capaz de descobrir as informações cifradas C com uma chave finita k de m, sendo $k \in \kappa$ e as funções de codificação definidas em E_κ e decifração definidas em D_κ. Onde a probabilidade da mensagem é dada por p seja $Pr_\rho(p)$ e a probabilidade de uma chave k é dada por $Pr_\kappa(k)$, ambas determinadas pela probabilidade no espaço. Então é possível definir a probabilidade de que uma mensagem esteja codificada pela chave k através da expressão 4.1.

$$Pr(p,k) = Pr_\rho(p)Pr_\kappa(k) \tag{4.1}$$

Para a distribuição de probabilidade de Pr no espaço amostral $\rho x \kappa$, considera p um texto normal denotando o evento $(p,k) : k \in \kappa$ de que seja codificado, tem-se na probabilidade da expressão 4.2.

$$Pr(p) = Pr_\rho(p) \tag{4.2}$$

Então para uma chave $k \in \kappa$, denota-se k o evento $(p,k) : p \in \rho$ de que k seja escolhido para a codificação, tem-se na expressão 4.3.

$$Pr(k) = Pr_\rho(k) \tag{4.3}$$

Portanto, os eventos p e k são independentes, pois para que um texto cifrado $c \in C$, denota-se c o evento $(p,k) : E_k(p) = c$ de que o resultado da codificação seja c.

Conclui-se que o adversário tem conhecimento de Pr_p do texto, pois não existe obscuridade no protocolo de comunicação entre origem e destino legítimo. Portanto o texto cifrado c é de conhecimento do adversário. Segundo a distribuição de probabilidade Pr_p menos provável, o adversário através de observação de c descobre algo, entretanto, o contrário, se a probabilidade de cada c permanecer inalterada então o adversário não descobre coisa alguma. De acordo com Shannon (SHANNON, 1948) isso motiva o conceito de sigilo perfeito.

4.2 DESCRIÇÃO GERAL DO MODELO

Uma vez que o computador pessoal do Usuário é considerado vulnerável a diversos ataques, é relativamente fácil o comprometimento utilizando-se de códigos e/ou programas maliciosos. A utilização destes códigos e/ou programas maliciosos é comumente observada em ataques contra computadores que fazem uso de sistemas bancários, sendo o modelo mais adotado pelos adversários a fim de roubar informações de autenticação e inserir os dados necessários para forjar transações financeiras.

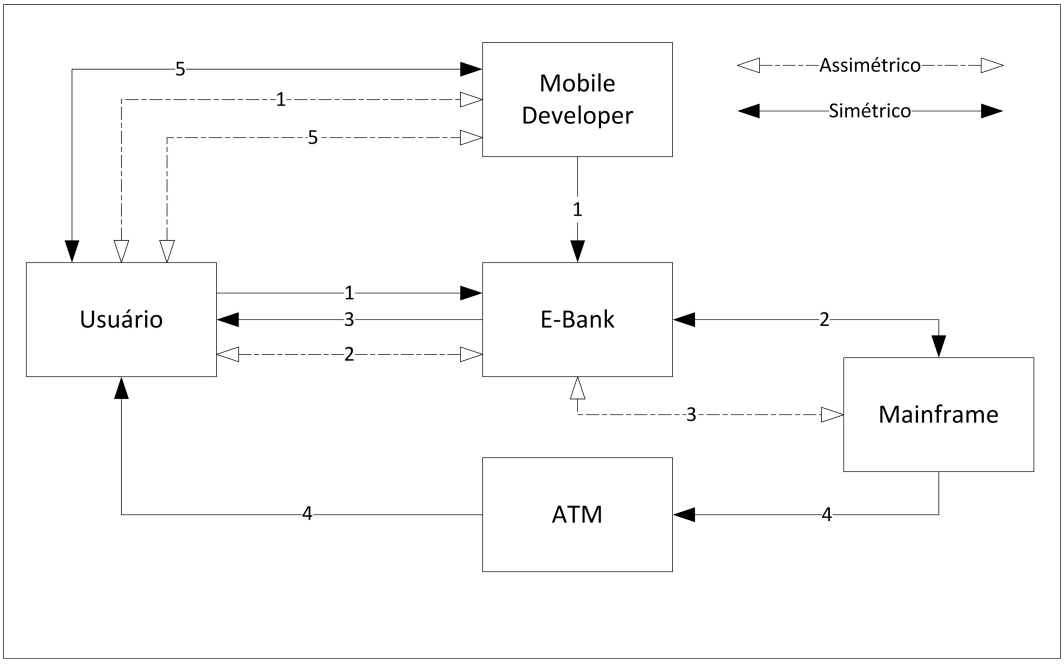

Figura 4.2: Diagrama Geral do Modelo

A Figura 4.2 apresenta um diagrama com o modelo geral de comunicação entre as partes Usuário e Banco, onde as etapas são listadas e detalhadas em:

Etapa 1: O Usuário se autentica através do canal *Internet* e solicita a autorização para uso do sistema de acesso com segundo fator de autenticação, faz o *download* e instala aplicação desenvolvida especificamente para seu modelo de dispositivo móvel (celular *smartphone*);

Etapa 2: O sistema mainframe gera e calcula as chaves criptográficas, utilizando como entrada *input* um identificador único para cada Usuário *id* e data e hora *timestamp* da informação constante no momento da solicitação. Juntando a isso o processamento do resultado gerado através das *masters key*. O sistema mainframe devolve ao sistema *Internet Banking*, onde o Usuário está autenticado,

que irá apresentar a informação da chave k_1, utilizando para isso um gerador de código visual, que apresentará as informações;

Etapa 3: Usuário utilizando um dispositivo móvel e aplicação específica, recupera a chave apresentada visualmente pelo sistema *Internet Banking* e armazena a informação, que em seguida é direcionado a um terminal de auto atendimento (ATM) para recuperar a segunda parte da chave;

Etapa 4: O Usuário se autentica em um ATM, de posse do cartão bancário e finaliza o procedimento coletando a chave k_2, armazenada no sistema mainframe;

Etapa 5: O dispositivo móvel processa e armazena a chave k.

4.3 MODELO SIMÉTRICO

Esquemas modernos de criptografia permitem que sejam considerados seguros, com uma probabilidade desprezível de que esses modelos sejam comprometidos(KATZ; LINDELL, 2007). Pode-se considerar que seja fácil a execução em tempo polinomial, e que a execução em tempo não polinomial tenha uma probabilidade insignificante.

Um adversário, para ter sucesso em sua investida, tem como probabilidade $\frac{1}{p^n}$ para alguns polinômios positivos p. Entretanto se a probabilidade de quebrar o esquema é assintoticamente menor que $\frac{1}{p^n}$ para cada polinômio p, então o esquema é considerado seguro, pois a possibilidade de um adversário ter sucesso é considerada muito pequena ou mesmo desprezível, conforme a seguinte Definição 1:

Definição 1 *Uma determinada função f é desprezível para cada polinômio $p(l)$, onde existe N para todo inteiro $n > N$, conforme a equação 4.4.*

$$f(n) < \frac{1}{p^n} \tag{4.4}$$

Para todas as constantes c existe um N para todo $n > N$ sustentado por $f(n) < n^{-c}$. Portanto, todo polinômio $p(l)$ e todos valores de n suficientemente grandes tal que $f(n) < \frac{1}{p^n}$.

A Figura 4.3 apresenta a comunicação entre os módulos criptográficos, onde a chave K é a chave associada a um Usuário considerado legítimo, K_1 é um dado aleatório do mesmo tamanho de K, sendo obtida conforme descrito na seção 4.3 para modelo simétrico e seção 4.4 para o modelo assimétrico. Os dados da transação são concatenados associando uma mensagem M com o código autorizador (TAN) $r\prime$ que é cifrado com a primeira parte de K onde K'_1 e de tamanho 2^{128}. A segunda parte da chave K definida como K'_2 permite calcular a integridade dos dados associados à mensagem com $HMAC - SHA1$ equivalente a 2^{160}, com isso é possível determinar que um adversário não tenha a capacidade de modificar qualquer informação sem que seja detectada sua tentativa.

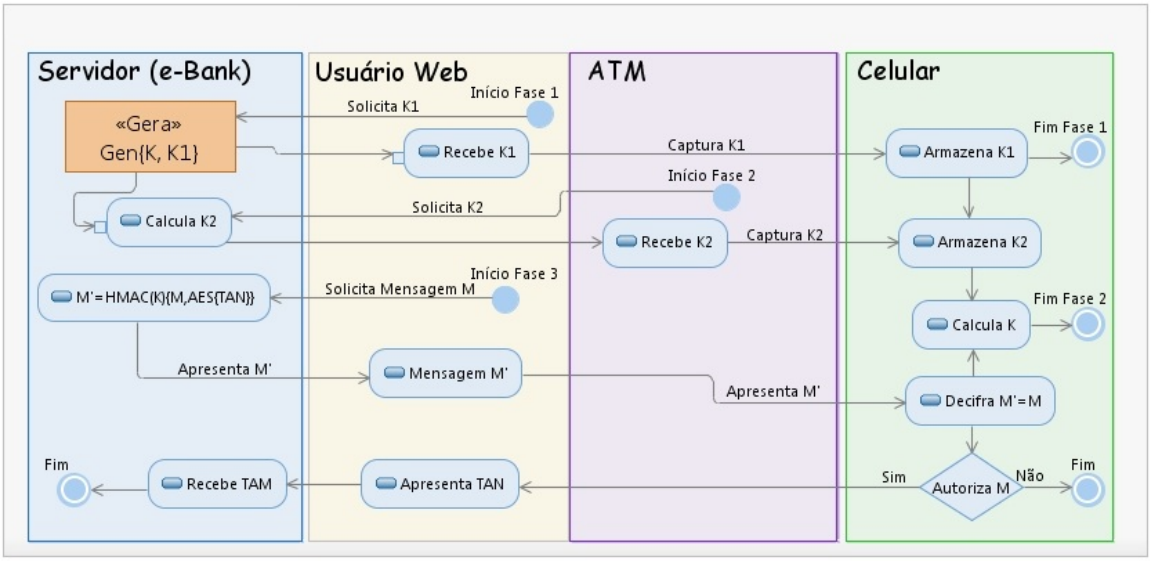

Figura 4.3: Esquema criptográfico do modelo

A mensagem M então será concatenada com o código autorizador $r\prime$ cifrado em AES e então gerado sua saída concatenando as informações geradas pelo $HMAC - SHA1$, tendo como saída $M\prime$, que é a mensagem final a ser entregue para o destino que possui a chave K utilizada nesse processo. Para cada código autorizador (TAN) $r\prime$ é gerado um aleatório $SALT$ a fim de que se gere criptogramas diferentes mesmo em código iguais, dessa maneira o adversário não será capaz de observar a comunicação e reutilizar uma mensagem anteriormente capturada e que possua o mesmo código autorizador.

4.3.1 Iniciação do Modelo Simétrico

A habilitação do processo necessita que o protocolo seja iniciado, onde a troca de chaves é necessária apenas uma vez, dessa maneira o dispositivo móvel (celular *smartphone*)

do Usuário estará apto a trocar mensagens de maneira segura.

A Figura 4.4 apresenta o diagrama de sequência para geração e troca das chaves k, k_1 e k_2, de maneira que o segredo, que deve ser de conhecimento entre Usuário e Banco, não trafegue pelo canal conhecidamente inseguro. Portanto o Usuário é considerado autenticado, utilizando de um segredo estático e associado a uma identificação única. Será considerado um identificador de identificação w. Ao ser desafiado a comprovar sua identidade, o Usuário fornece seu identificador e sua senha w, o sistema bancário então irá computar $f(w)$ e comparar o resultado com o valor armazenado, sendo igual o acesso é liberado.

Considera-se, a princípio, este modelo inseguro e com pouco custo, pois tanto w como as credenciais de acesso são estáticas, o que torna suscetível a ataques de roubo de credenciais, considerados triviais.

O mesmo ocorre com senhas descartáveis, a cada nova conexão de um Usuário é solicitada uma nova senha sendo $f(w_1), f(w_2), ..., f(w_n)$ o que torna o modelo aparentemente mais seguro. Entretanto, o tipo de desafio pode ser facilmente contornado quando um adversário solicita a colaboração não voluntária de um Usuário legítimo para obter essa senha. Igualmente ocorre com modelos baseados em uma chave pré compartilhada, onde o desafio é feito para se identificar um Usuário, baseado em $w_i = f^i(w), i \geq 0$.

Então, é possível determinar que o adversário obtém vantagem, quando um sistema utiliza-se de senhas estáticas ou mesmo dinâmicas para identificar um Usuário. Entretanto, quando se implementa um modelo baseado em chaves simétricas não utilizáveis para identificar, mas para autorizar uma determinada transação, o modelo torna-se mais robusto.

Um desafio é o envio de uma chave k utilizando dois canais, um conhecidamente inseguro. O módulo HSM *(Hardware Security Module)* Bank deve ser armazenado em ambiente protegido, conforme determinado nas melhores práticas de segurança (ABNT-A, 2006) um conjunto de chaves mestres $MK_{k_1}, MK_{k_2}, ...MK_{k_n}$ onde $MK \in \{0,1\}^{256}$ e é obtida com $MK \leftarrow Gen(1^n)$.

Seguindo o diagrama apresentado pela Figura 4.4 é gerada uma chave $k \in \kappa$ para o Usuário, onde $k \leftarrow Enc_{mk}(ID|timestamp)$, utilizando-se como premissas qualquer identificador único associado a uma data e hora determinada pelo sistema bancário no

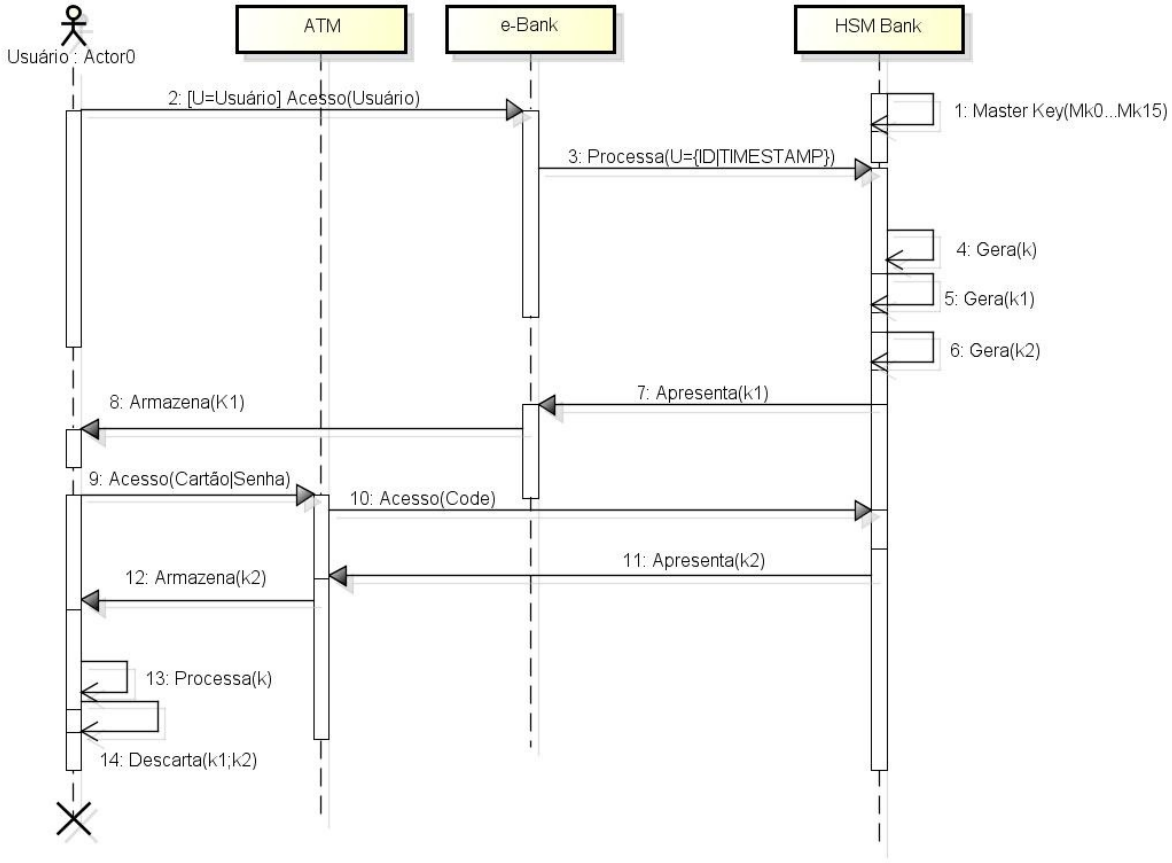

Figura 4.4: Geração de sementes *Master key*

momento da geração de $k \in \{0,1\}^{288}$. Então a chave k é armazenada em ambiente protegido.

O próximo passo é gerar k_1 e k_2, onde k_1 é uma sequência aleatória de tamanho $b = \{0,1\}^{288}$ e k_2 é definida por $k \oplus k_1$, o espaço da chave possui uma probabilidade exata de 2^{288}, gravada $k_1 \leftarrow Gen(1^n)$.

É enviado para o *Usuário* a chave k_1 pelo canal inseguro, onde o adversário pode ter acesso e armazenar a informação, no entanto, k_2 é enviada e resgatada pelo Usuário em um canal seguro. Portanto, a distribuição da chave é associada a um esquema de segredo perfeito, onde uma determinada mensagem M de tamanho fixo é arbitrariamente $m \in M$ e $c \in C$. Observando a chave k é dado pela expressão 4.5.

$$
\begin{aligned}
Pr[C = c|M = m] &= Pr[M \oplus k = c|M = m] \\
&= Pr[m \oplus k = c] = Pr[k = m \oplus k = c] = \frac{1}{2^{288}} \qquad (4.5)
\end{aligned}
$$

Portanto, após o Usuário receber as partes k_1 e k_2 processa $k = k_1 \oplus k_2$ e descarta k_1 e k_2, possuindo agora um segredo compartilhado entre Banco e Usuário.

A Figura 4.5 detalha todo o processo, tendo os atores envolvidos na comunicação: CNC (Computador não confiável) representa um dispositivo pessoa l de conexão a *Internet* que o Usuário utiliza na comunicação com o Banco, conforme discutido na seção 3.3, este dispositivo não é passível de proteção, portanto, considera-se não confiável. O dispositivo móvel do Usuário deve estar previamente com aplicação desenvolvida pelo Banco e que possua as características mínimas necessárias para processar e armazenar uma chave criptográfica. Por último o provedor do serviço de *Internet Banking* (e-Bank), considerado seguro.

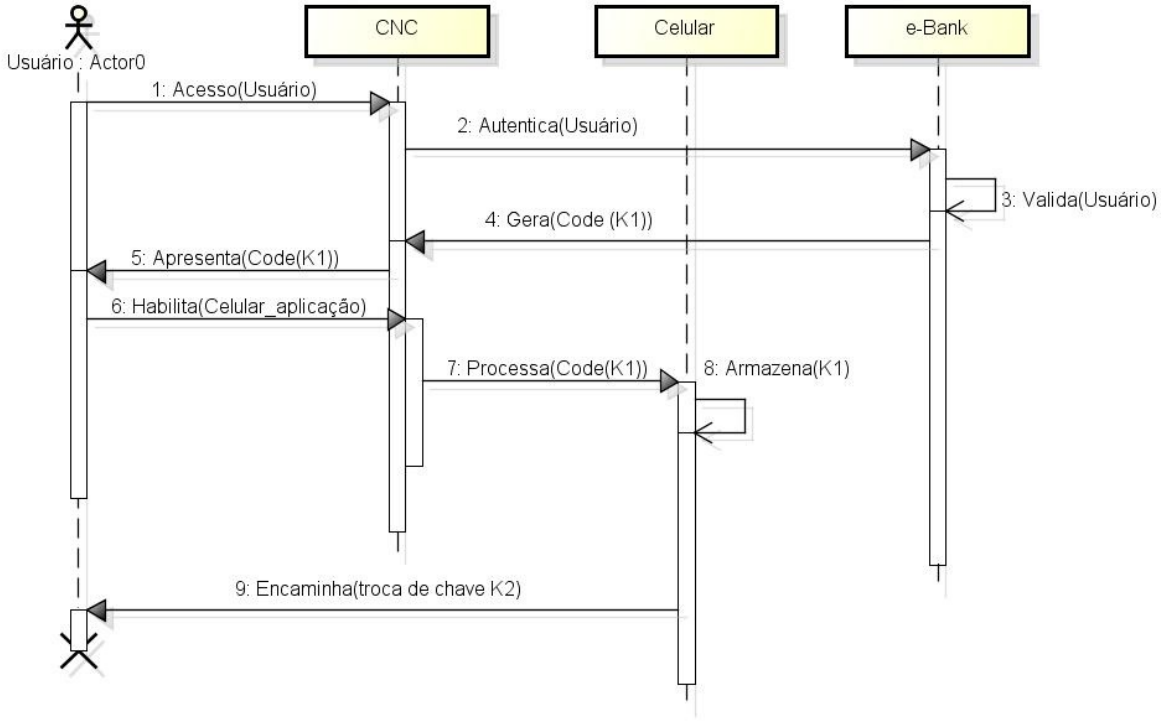

Figura 4.5: Troca de Chaves - k_1

O Usuário autentica-se com sua credencial estática w e solicita o compartilhamento do segredo k. O Banco, após validar o acesso como legítimo, envia k_1 apresentando

as informações da chave através de um código visual mostrado na tela do dispositivo pessoal.

O Usuário, utilizando seu dispositivo móvel (celular *smartphone*), considerado seguro, conforme discutido na seção 3.3 habilita a aplicação desenvolvida pelo Banco para processar e armazenar k_1. Como não existe conexão direta pela rede, ou seja, todo processo é executado utilizando apenas o poder de processamento do dispositivo móvel, ao coletar e armazenar k_1 o Usuário é informado a continuar o processo e coletar o restante da informação k_2, que comporá sua chave secreta k. Neste momento, os sistemas de *Internet Banking*, armazenam a informação de que k_1 foi entregue.

A Figura 4.6 representa o passo final na troca da chave k, e consequente habilitação do Usuário para operacionalização e uso do modelo.

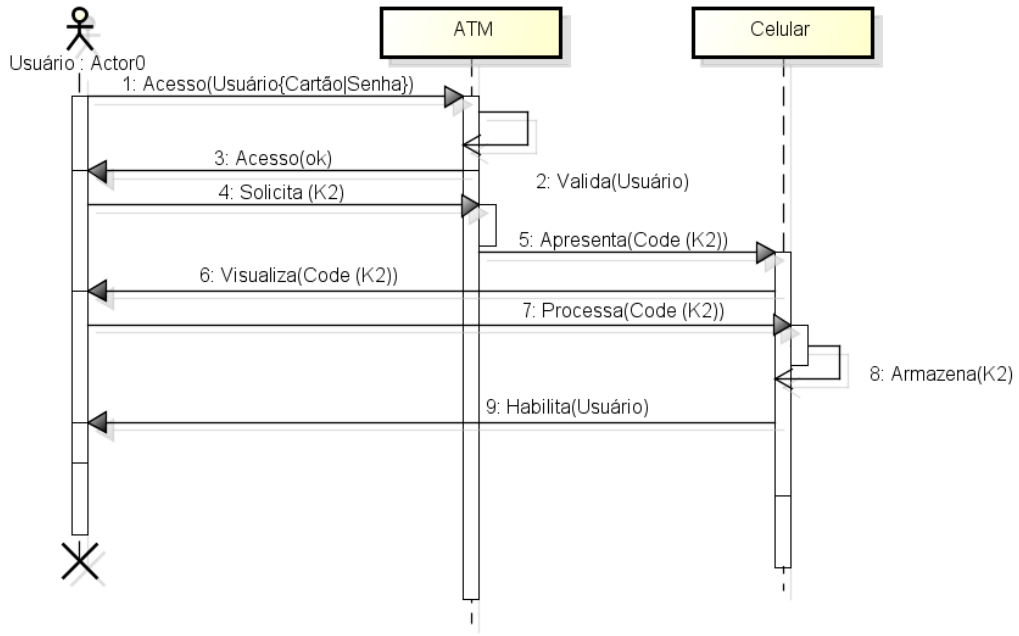

Figura 4.6: Troca de Chaves - k_2

Os atores envolvidos nesta etapa são: ATM[11] e dispositivo móvel pessoal do Usuário. Nesta etapa, o Usuário é desafiado a se identificar utilizando dois fatores, um cartão bancário e sua senha pessoal estática.

Após a identidade do Usuário ser confirmada e validada. É apresentado a segunda parte do segredo k_2, que deverá ser visualizado diretamente na tela do terminal *ATM*, para

[11] Automated/Automatic Teller Machine (Terminal de Auto Atendimento Bancário)

isso utilizando o mesmo processo apresentado na Figura 4.5. Um código 2D contendo informações de k_2 é apresentado para o Usuário, o qual utilizando seu dispositivo móvel receberá, processará e armazenará o segredo.

O dispositivo móvel ao receber a última parte do segredo irá processar $k = k_1 \oplus k_2$, armazenar k e descartar k_1 e k_2. Nesta etapa, o processo está finalizado, um segredo foi trocado entre Banco e Usuário.

De posse do segredo o Usuário agora pode ser desafiado a receber uma mensagem m e codificar $m' = E_k(m)$ de modo que ao executar $D_k(m')$ e se $m' = m$ aceitar a identificação da mensagem.

4.4 MODELO ASSIMÉTRICO

A complexidade do modelo simétrico, discutido na seção 4.3, trata principalmente a distribuição da chave k. A origem e destino deve, previamente, combinar uma chave secreta k. O armazenamento dessas chaves é custoso, quando a comunicação entre todas as partes é exigida a complexidade aumenta. Para um sistema que contenha n Usuários seria necessário $n(n-1)/2$ chaves secretas. Entretanto, a proposta desta tese é que a comunicação seja sempre entre solicitante e provedor, nunca entre solicitantes Portanto, mesmo em sistemas assimétricos, que o número de chaves é igual ao número de Usuários, não afetaria a quantidade de chaves, que permanece nos dois modelos em n.

Definição 2 *Um esquema de chaves públicas é uma tupla probabilística de um algoritmo de execução em tempo polinomial* (Gen, Enc, Dec) *que:*

1. O algoritmo Gen gera um par de chaves pk, sk como parâmetro de entrada 1^n. pk faz referência a chave pública e sk à chave privada. Assume-se que pk, sk tem tamanho n e que n é determinado por pk, sk;

2. O ciframento é dado por Enc, que tem como entrada pk e uma mensagem m, sendo m um texto em claro. Como saída tem-se um texto cifrado c, que pode ser expresso por $c \leftarrow Enc_{pk}(m)$;

3. A decifração é dada por Dec que tem como entrada a chave sk o texto cifrado c, e a mensagem de saída pode ser m ou $\emptyset$ que denota uma falha. Assume-se que Dec é determinístico, e que $m := Dec_{sk}(c)$.

É exigido que a saída seja desprezível ($desp$) dado qualquer n, para qualquer pk, sk e saída $Gen(1^n)$, e cada mensagem m para o espaço amostral do texto em claro seja expressa em 4.6.

$$Pr[Dec_{sk}(Enc_{pk}(m)) \neq m] \leq desp(n) \tag{4.6}$$

Para que esse esquema seja forte o suficiente, considera-se que um adversário não seja capaz de processar em tempo polinomial qualquer mensagem c e obter m, sem possuir sk. Portanto, define-se que é computacionalmente inviável obter k, mesmo conhecendo-se o algoritmo ou protocolo utilizado para cifrar m.

4.4.1 Iniciação do Modelo Assimétrico

A iniciação do protocolo do modelo assimétrico, parte do principio que o dispositivo móvel do Usuário seja capaz de gerar um par de chaves criptográficas através do hardware, baseado em protocolos reconhecidamente seguros.

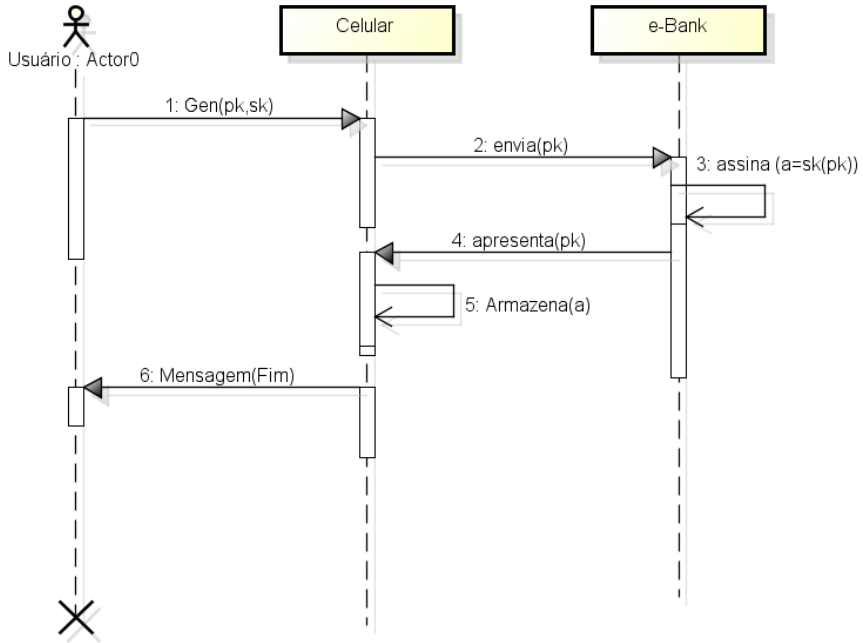

Figura 4.7: Modelo Assimétrico de chaves criptográficas

A Figura 4.7 apresenta o modelo que se baseia na geração do par de chaves em hardware, onde o dispositivo móvel seja capaz de executar todas as operações matemáticas de um modelo baseado em uma infraestruturas de chaves públicas (PKI[12]).

[12]Public-Key Infrastructure

Este modelo descreve o algoritmo de chaves públicas *RSA*, onde um Usuário executa $Gen(1^n)$, obtendo (N, e, d), sendo $sk = (N, d)$ e $pk = (N, e)$, seguindo a mesma proposta apresentada pela definição 2.

4.5 TROCA DE MENSAGENS

A premissa inicial é de que o Usuário conseguirá verificar, a autenticidade da mensagem m , ou seja, que foi realmente o emissor que gerou a informação e apresentou ao Usuário. Da mesma maneira é possível garantir a integridade das informações, na forma $h_k(m)$, onde se alguma informação for modificada computando $h_k m' = h_k(m)$ o protocolo impedirá sua continuação, será alertado e não apresentará a informação ao Usuário. A Figura 4.8 apresenta o formato das mensagens que será utilizado para envio e processamento das mensagens m.

1	21	37	158
Chave de HMAC	Chave de AES ou 3des	Transação	
160 bits (20 Bytes) – SHA1	128 bits (16 Bytes)	120 Bytes	

Figura 4.8: Formato da Mensagem m

O dispositivo móvel deverá ser capaz de executar as operações de verificação de integridade e confidencialidade. Para isso irá utilizar a chave k armazenada e de conhecimento do Banco e do Usuário, previamente compartilhada e apresentada na seção 4.3. A chave k é de tamanho único da ordem de 288 bits, onde $k \in \{0,1\}^{288}$. A aplicação deverá dividir $k \in \kappa\{0,1\}^{288}$ em duas partes $k'_1 \in \{0,1\}^{160}$ e $k'_2 \in \{0,1\}^{128}$ (Figura 4.1).

Com as chaves armazenadas no dispositivo móvel, o Usuário terá a capacidade de verificar a autenticidade e integridade das informações enviadas e mesmo garantir o sigilo. Caso se confirme a autenticidade e integridade das informações enviadas, o Usuário poderá confirmar ou não a mensagem.

Quando o Usuário, previamente autenticado no sistema da maneira convencional, utilizando um computador não confiável (CNC) e conectado a *Internet*, solicitar uma transação, convenientemente chamada aqui de mensagem m, será resgatada a chave k desse Usuário, a fim de iniciar o processo de autorização de m.

Na próxima etapa, a autorização de uma mensagem m é determinada através de um código autorizador r associado unicamente à mensagem m. É então gerado $r \in \mathbb{N}$ onde $\mathbb{N}$ é um conjunto de números inteiros positivos e que $\mathbb{N} \in \{x_1, ...x_{999999}\}$. Utilizando-se de modelos pseudoaleatórios fornecidos pelos sistemas computacionais com espaço

amostral $0 < r \leq 999999$, onde o conjunto de números está uniformemente distribuídos no espaço amostral conforme expressão 4.7, onde $n = 999999$.

$$r = \sum_{i=1}^{n} x_i 2^{n-i} \tag{4.7}$$

O código autorizador então é cifrado com k'_2 através de $r' \leftarrow Enc_{k'_2}(r)$, e o dispositivo móvel é capaz de computar $r' := Dec_{k'_2}(r)$ apresentando ao Usuário o código autorizador r'.

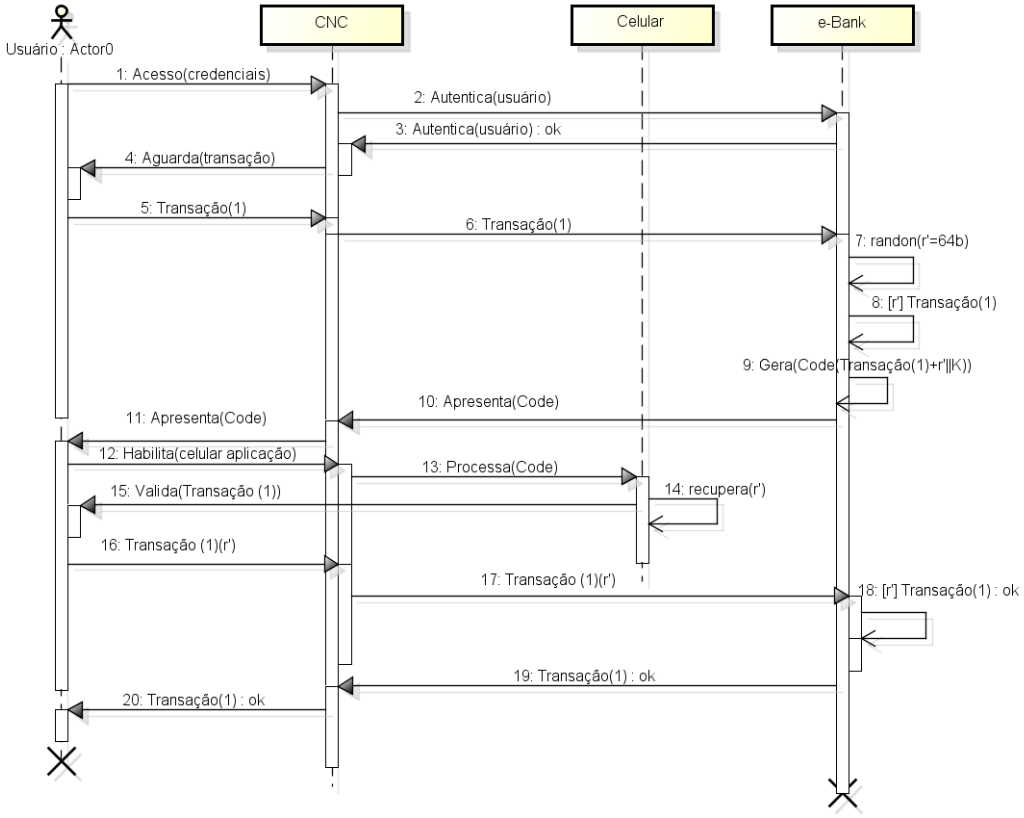

Figura 4.9: Fluxo de assinatura de transação com k

O diagrama apresentado na Figura 4.9 demonstra as sequências dos processos de utilização por parte do Usuário que utiliza um computador não confiável, um dispositivo móvel (*smartphone*) e o próprio servidor do Banco.

O servidor de *Internet Banking* ao receber e processar a mensagem m, de forma a obter o formato descrito através da Figura 4.8, irá computar a expressão $y = h_{k'_1}(m, r')$.

A mensagem m, será apresentada no canal solicitante, para que o Usuário, utilizando

o dispositivo móvel recupere visualmente $m_{\prime}$. O dispositivo móvel irá validar $m_{\prime}$ de forma a verificar a autenticidade e integridade da mensagem, computando a expressão $y' = h_{k'_1}(m, r')$.

Então se $y' = y$ a mensagem m é considerada autêntica e integra, pois sem o conhecimento de k'_1 é inexequível obter y, y'. Nesse momento, o Usuário pode recuperar a mensagem e esta é apresentada na forma da Figura 4.13 e o código de autorização é mostrado na (Figura 4.14), que será discutido na seção 4.6.

A última sequência de comunicação é enviar r' pelo canal que solicitou a transação. Então, r' será informado pelo Usuário no canal original da solicitação da mensagem m. Enviado ao emissor Banco ,este irá computar $r' = r$, se for igual, o desafio foi aceito e a mensagem confirmada. A mesma prova aqui é aplicada, sem o conhecimento de k'_2, um adversário não tem poder suficiente para calcular r, r'.

Para que um ataque seja eficiente, o adversário deveria possuir a capacidade de comprometer o dispositivo móvel do Usuário, que na descrição do modelo adversarial apresentado na seção 3.3 é considerado inexequível. Da mesma forma, a descrição do ataque baseado na colaboração do Usuário legítimo é possível e é considerada, ou seja, o adversário deve convencer o Usuário a executar operações arbitrárias. No entanto, esse tipo de ação, apesar de prevista, dificulta a maioria dos ataques atuais, que se valem da automatização e da personificação, e ainda necessita da colaboração constante para todas as transações solicitadas.

4.6 DEMONSTRAÇÃO DO PROTOCOLO

A simulação foi executada utilizando os dispositivos móveis modelos: Samsung Galaxy S, LG P500 e Iphone 4, contendo sistema operacional Android e IOS respectivamente.

A geração das chaves K_1 e K_2 seguiu as informações apresentadas pela Tabela 4.1:

Tabela 4.1: Criação de mensagens

N	Campo	Valor
1	ID	000127299
2	TIMESTAMP	20111021145605839866
3	K	98F39D76846C8049882FE6F09136B1D8F2746EE0CBE3BE92B5DA2F605DE3137BCC1E6909
4	K_1	E6113483125213B5F7FCDD5AC8C4DF0DA435C11C94FB405251F9A680A7A70EC73E45667C
5	K_2	7EE2A9F5963E93FC7FD33BAA59F26ED55641AFFC5F18FEC0E42389E0FA441DBCF25B0F75
6	MK_9	7279E2195760A904E6A5FA93F98910085452414E53464552454E43494120454E5452452D
7	HMAC-SHA1	596D7DA6ADDECE8A658F1995EF382819992B5D7F
8	TAN-AES	5947F311D472D240C31CEA5BE1022757
9	TRANSAÇÃO	818;24/11/2011;13:30:05;1.400,00;PEDRO ALVARES CABRAL;7988-X;170.159-2;
10	AUTORIZADOR	008062

Por conveniência utilizou-se a conversão das chaves K_1 e K_2 de binário para hexadecimal, o que melhora a captura das informações através de um *smartphone*. A codificação das chaves está representada pelas Figuras 4.10 e 4.11. A Figura 4.12 representa a mensagem a ser enviada e capturada pelo dispositivo móvel.

O dispositivo móvel do Usuário irá capturar a chave K_1 através do navegador de acesso convencional, e será informado a capturar a segunda parte K_2 em canal seguro, que neste caso, é um terminal de autoatendimento *ATM*, o qual o desafiará a apresentar suas credenciais, um cartão bancário e sua senha de acesso convencional. Este processo é executado apenas uma vez de modo que todas as futuras interações utilizarão a chave armazenada na fase de iniciação.

Figura 4.10: Chave K_1

Figura 4.11: Chave K_2

Figura 4.12: Mensagem M

Após a fase de iniciação do protocolo, o Usuário estará apto a trocar mensagens com o Banco de forma segura. Todo o acesso permanece inalterado, a modificação é feita no momento em que alguma transação é solicitada, então o Banco desafia seu Usuário a provar sua identidade, autorizando uma mensagem codificada através do protocolo *DAP*.

A mensagem final, que é apresentada ao Usuário, está representada pela Tabela 4.2. Neste caso, foi utilizado codificação Base64 (JOSEFSSON, 2006), por se tratar de informações binárias, esse tipo de codificação auxilia na transmissão de dados em canais analógicos.

Tabela 4.2: Mensagem Codificada

WW19pq3ezopljxmV7zgoGZkrXX9ZR/MR1HLSQMM
c6lvhAidXODE4OzI0LzExLzIwMTE7MTM6MzA6MD
U7MS40MDAsMDA7UEVEUk8gQUxWQVJFUyBDQUJSQ
Uw7Nzk4OC1YOzE3MC4xNTktMjs=

O dispositivo móvel irá então decodificar a mensagem apresentada na forma da Figura 4.12, seu conteúdo é apresentando através da Tabela 4.2, confirmando as informações

o Usuário poderá visualizar o código autorizador e assim informá-lo através do canal *Internet Banking*.

Nesse momento, o Usuário pode recuperar a mensagem que é apresentada na forma da Figura 4.13 e o código de autorização é mostrado pela Figura 4.14. Como o código é associado à mensagem, mesmo que o atacante consiga interceptar e capturar a informação, não será útil para outra função a não ser a própria autorização da mensagem, o que diferencia de outras soluções, onde o código TAN pode ser utilizado para outras transações, por esse motivo é utilizado o termo autorização e não autenticação.

Figura 4.13: Verificação de informações da transação

Figura 4.14: Visualização do código autorizador

Todo o processo é executado sem a necessidade de que o dispositivo móvel tenha qualquer tipo de conectividade. Uma vez que a chave esteja armazenada no dispositivo, todos os desafios são executados de modo que apenas o detentor desse segredo possa responder a esses desafios. Mesmo que um ataque seja bem sucedido em obter esse segredo, o atacante deverá ser capaz de associar a chave a uma identificação, ou seja, sincronizar o comprometimento do computador pessoal do Usuário ao dispositivo móvel.

4.6.1 Análise de Execução

Códigos 2D possuem limitação de tamanho de mensagens que podem ser apresentadas (ISO/IEC 18004, 2006), podendo chegar a apresentar até 4.296 bytes alfanuméricos com nível baixo de redundância de 7%, que é a capacidade de recuperar a informação, e 1.852 bytes para nível alto, com 30% de redundância. Mesmo para mensagens que

exploram a capacidade total do canal de transmissão, ainda existe a limitação do dispositivo que é usado para capturar e processar essa mensagem.

Para um total de 4.296 bytes de informação existe 2.956 bytes de *codewords*, que são símbolos representativos utilizados pela matriz do *QR Code*. Para substituir um erro é necessário dois *codewords*. Portanto, para um total de 2.956 bytes é preciso de 750 bytes de informações que resultam em aproximadamente 7% de taxa de correção de erros em uma matriz 177 x 177 módulos, que consome aproximadamente 10% da capacidade de dados. Quanto mais módulos forem usados para formar a matriz, mais complexo é sua leitura e decodificação.

A Tabela 4.3 apresenta 5 capturas, todas relacionados ao tamanho em *bytes* da mensagem m e o tempo em $milissegundos(ms)$ relacionado a cada captura ($t1$ $t2$ $t3$ $t4$ e $t5$). É possível notar que mesmo para mensagens maiores o tempo não varia conforme seu tamanho, ficando com média geral de $70ms$ para a captura e processamento da mensagem.

Tabela 4.3: Tempo de processamento e decodificação de mensagens

m	t1	t2	t3	t4	t5	Mínimo	Máximo	Média
140	60	87	78	55	79	55	87	71,8
152	55	53	50	61	57	50	61	55,2
188	99	60	50	80	76	50	99	73
200	64	85	66	84	71	64	85	74
204	66	70	96	55	59	55	96	69,2
208	54	53	69	64	72	53	72	62,4
224	57	83	75	81	80	57	83	75,2
228	54	70	51	97	74	51	97	69,2
232	63	55	53	62	64	53	64	59,4
248	152	65	54	79	84	54	152	86,8
252	61	85	76	71	73	61	85	73,2
336	103	80	73	70	82	70	103	81,6
528	89	76	103	87	94	76	103	89,8

As capturas foram executadas através de exposição direta da imagem a um dispositivo móvel aguardando a informação. Pode ocorrer variações de captura e processamento, dependendo do hardware utilizado e a qualidade da emissão e apresentação do código 2D. Acredita-se que essa variação não afeta o desempenho, pois a tendência é que novos hardwares estejam acessíveis e com capacidades de processamento cada vez maiores.

Na apresentação da mensagem m, utilizando código 2D, existe acréscimo de 15% de informações, devido à redundância nível médio pelo uso de código corretor de erro, podendo chegar, neste nível, a 3.391 *bytes* alfanuméricos. Esse acréscimo foi considerado eficiente, pois não acarretou nenhum erro para as leituras executadas. Caso seja necessário inserir maior redundância, é possível aumentar para até 30%, o que não necessariamente torna a leitura mais rápida, entretanto, permite que erros sejam corrigidos em tempo de execução.

Nas análises foi determinado que em média, as mensagens m possuem 242 bytes, sem considerar a redundância, que para este caso específico é necessário incluir mais 15% de informações, aumentando em 36 bytes a mensagem a ser processada. Também foi possível definir que o tempo médio de captura e processamento foi de $72ms$, sendo que no melhor caso ocorreu o tempo de $58ms$ e pior caso $91ms$. Este tempo é considerado aceitável para esse tipo de uso, portanto, se mostrando promissor.

5 CONCLUSÕES

Sistemas de *Internet Banking* têm se mostrado uma ferramenta preciosa para o estilo de vida atual. Além de permitirem facilidade de acesso, também permitem uma maior integração de serviços e negócios entre as instituições financeiras e seus clientes.

É importante considerar que o crescimento desses sistemas, tanto pelo seu uso por cada vez mais pessoas, quanto pela abrangência, não houve um acompanhamento conjunto com os modelos de segurança. Muitas instituições financeiras permanecem utilizando modelos considerados ultrapassados e, que em alguns casos, são adaptações de soluções desenvolvidas para outros tipos de sistemas, sendo necessário então, abordar o tema de maneira não convencional ao usual, e que o tratamento do problema vai além de se identificar um Usuário utilizando premissas que podem ser reproduzidas, deve-se consider um novo paradigma de acesso seguro.

Como hipótese discutida neste trabalho, a proposta de desenvolver um novo modelo de acesso a serviços de *Internet Banking*, utilizando conceitos de primitivas criptográficas e fatores de autenticação fortes, se mostrou promissora. A compreensão dos problemas relacionados à segurança bem como os tipos e sistemas de autenticação que essas instituições utilizam foram fundamentais para a identificação de pontos vulneráveis, permitindo identificar qualidades presentes nos ambientes atuais.

O objetivo deste trabalho foi desenvolver um protocolo de comunicação seguro, utilizando modelos simétrico e assimétrico, buscando garantir a integridade, autenticidade, não repúdio e também a confidencialidade de transações *online*. Através de uma prova de conceito de aplicabilidade de solução integrada a um protótipo, em um ambiente real de testes e uso em um Banco de varejo nacional foi possível identificar vantagens em relação aos modelos atualmente utilizados. O fato de ser possível utilizar qualquer tipo de sistema operacional, independente de existir solução que garanta a segurança nesses sistemas, foi um grande avanço frente à diversidade atual. A não necessidade de qualquer tipo de conectividade do autorizador é outra vantagem, pois provavelmente os Usuários desse sistema não estariam dispostos a arcar com custo de conectividade, portanto, seria um problema para que o sistema fosse largamente adotado. Também há de se expor que, cada vez mais pessoas aderem à utilização de sistemas móveis,

portanto, não geraria custos de distribuição de hardwares específicos, bem como não exige controles inerentes ao uso, bastando que a instituição financeira disponibilize a solução e incentive seu uso através de práticas de negócio.

Problemas relatados quanto à insegurança desses dispositivos móveis são relevantes e devem ser tratados com cuidado, no entanto, sistemas móveis possuem características bem definidas e são menos permissivos do que os sistemas operacionais produzidos para os computadores pessoais. As aplicações produzidas para uso em sistemas móveis não possuem permissões, em suas versões originais, para acessar informações de outras aplicações, dessa forma toda e qualquer informação armazenada no dispositivo só poderá ser acessado pela aplicação que a armazenou. Os sistemas que utilizam envio de mensagens podem ser interceptados, pois esse tipo de ação é permitido, inerente ao desejo do Usuário, o que não é verdade para o caso anterior, onde mesmo que o Usuário desejasse não seria possível autorizar que aplicações acessem informações de outras aplicações.

Durante a fase de pesquisa, foram identificadas dificuldades técnicas no desenvolvimento da prova de conceito, requerendo ajustes do protocolo. Neste caso específico, foi necessário a diminuição da segurança do protocolo criptográfico simétrico, que passou de 256 bits para 128 bits, visto que o tamanho reduzido da mensagem e sua transferência através do código 2D é um fator preponderante para maior adoção do modelo por parte das instituições financeiras e seus clientes, pois dessa maneira é possível atingir um número maior de dispositivos celulares, sendo que alguns possuem uma capacidade reduzida de leitura através de uma câmera fotográfica e menor processamento. A redução da segurança não necessariamente deixa o protocolo mais fraco, pois ainda assim o modelo é considerado robusto.

5.1 VANTAGENS DO PROTOCOLO DAP

O conceito de autenticação *out-of-band* tem sido largamente recomendado por órgãos regulamentadores do sistema financeiro de diversos países. O protocolo desenvolvido atende essa premissa, permitindo ir além, pois introduz conjuntamente um modelo *offline*. Com a troca de chaves, torna-se possível então, que o usuário autentica transações por um canal fora e banda sem conectividade, tornando o modelo um segundo fator real de autenticação.

A utilização de criptografia reconhecidamente forte e com aceitação por diversos órgãos

e agências de segurança mundiais, faz com que a quebra das cifras utilizadas estejam a um nível muito elevado para os ataques e atacantes atuais. Mitigar esses ataques é um desafio do ponto de vista da instituição financeira e seu usuário, pois a colaboração de um usuário legítimo torna qualquer modelo atual frágil, o que não acontece facilmente com o DAP, que utiliza uma chave criptográfica armazenada que não pode se digitada ou informada espontaneamente, dessa maneira ataques como engenharia social são mitigados, fazendo com que para toda autorização o usuário deve colaborar com o atacante.

A independência de sistemas de segurança, produzidos para proteger o canal de comunicação entre usuário e banco, é uma vantagem tratada pela não associação de hardwares ou softwares que buscam identificar a origem do acesso. Permitindo assim que um usuário possa acessar suas informações de qualquer sistema operacional, navegador ou mesmo computador. Com isso pode-se concluir que a segurança deixa o acesso livre e é transferida para o dispositivo móvel, que não deve possuir nenhum tipo de associação com a chave armazenada e seu usuário. Dessa forma, mesmo que o atacante produza técnicas de comprometimento desse dispositivo, não terá informações suficientes para relacionar a qual usuário pertence essa chave.

Problemas conhecidos como *man-in-the-middle* e *browser-in-the-middle* passam a ser ineficientes, pois o modelo de segurança utilizado é inerente à segurança do canal de comunicação, e apenas uma transação visualizada pelo usuário será autorizada. O atacante pode capturar os códigos utilizados para autorização dessas transações, entretanto, eles são válidos apenas para essas transações.

Outros segmentos de acesso vêm sendo estudados pelas instituições financeiras. Acessos provenientes de TV digital, pagamentos eletrônicos, comércio eletrônico e mesmo certificados digitais, podem utilizar o protocolo DAP, de modo a transformar esses modelos *out-of-band*.

5.2 TRABALHOS FUTUROS

O uso do protocolo DAP possui algumas limitações. Somente usuários que possuam um dispositivo capaz de capturar um código 2D e processar esse código estão aptos a usar o modelo. Diversos estudos têm mostrado um crescimento desse tipo de tecnologia, entretanto, não é possível atingir toda a base e usuários de *Internet Banking*.

Como trabalhos futuros, é importante que sejam estudadas técnicas de proteção contra comprometimento de dispositivos móveis sejam criadas. Mesmo não sendo possível associar informações capturadas nesses dispositivos, é necessário que se desenvolva maneiras de proteger a chave criptográfica armazenada. Neste sentido recomenda-se o uso de modelos de associação de *hardware* e *software*e, de modo que se algum comprometimento acontecer, o uso da chave seja restringida a apenas o dispositivo que originalmente capturou a informação, e que não seja possível a importação dessa chave por um dispositivo considerado espúrio.

O modelo assimétrico, proposto neste trabalho, é um modelo de referência. Depende de criação de mecanismos de geração de chaves públicas utilizando o próprio dispositivo, de maneira que as chaves sejam criadas conforme as melhores práticas adotadas pelos mecanismos de infraestrutura de chaves públicas. Recomenda-se que esse processo seja executado a partir do *chip* inteligente que cada dispositivo possui, e que as chaves criadas possuam a capacidade de não serem exportáveis, seguindo as premissas de segurança adotadas pelas principais autoridades certificadoras no mundo.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBAS, A.; SADDIK, A.; MIRI, A. A comprehensive approach to designing internet security taxonomy. *2006 Canadian Conference on Electrical and Computer Engineering*, IEEE, n. May, p. 1316–1319, 2006.

ABNT-A. *ISO/IEC 27002 Tecnologia da Informação - Técnicas de Segurança - Código de Prática para a Gestão da Segurança da Informação*. Rio de Janeiro - Brasil: ABNT - Associação Brasileira de Normas Técnicas, 2006. 1–144 p.

ABNT-B. *ISO/IEC 27001 - Tecnologia da informação - Técnicas de segurança - Sistemas de gestão de segurança da informação - Requisitos*. Rio de Janeiro - Brasil: Associação Brasileira de Normas Técnicas, 2005. 1–31 p.

ADVILLAGE. BB lança nova tecnologia de segurança para internet banking, por meio do smartphone. *Cidade BIZ - Edição online*, 04 de Junho 2012. Disponível em: <http://goo.gl/93A0B>. Acesso em: 02/07/2012.

ANDERSON, R.; BOND, M. The man-in-the-middle defence. *Computer Laboratory, University of Cambridge*, p. 2–5, 2009.

BAILEY, M.; OBERHEIDE, J.; ANDERSEN, J.; MAO, Z. M.; JAHANIAN, F.; NAZARIO, J. Automated classification and analysis of internet malware. *Proceedings of the 10th international conference on Recent advances in intrusion detection*, p. 178–197, 2007.

Banco do Brasil. *Video institucional TV BB: Conheça o BB Code, nova ferramenta de segurança para as transações no site do Banco na Internet*. Junho 2012. TVBB. Disponível em: <http://goo.gl/u3aU6>. Acesso em: 03/07/2012.

BARNUM, S. *Attack Patterns as a Knowledge Resource for Building Secure Software*. [S.l.], 2007. 1–31 p.

BCB. *Diagnóstico do Sistema de Pagamentos de Varejo do Brasil Adendo estatístico*. Banco Central do Brasil - Brasília - Brasil, 2009.

BCB. *Diagnóstico do Sistema de Pagamentos de Varejo do Brasil Adendo estatístico 2010*. Banco Central do Brasil - Brasília - DF, 2010. 30 p.

BELLARE, M.; CANETTI, R.; KRAWCYK, H. Message authentication using hash functions - the hmac construction. *Apperas in RSA Laboratories CryptoBytes (Spring 96)*, v. 2, n. 1, p. 1–5, 1996.

BELLARE, M.; FISCHLIN, M.; MICALI, S. Identification protocols secure against reset attacks. *EUROCRYPT 2001, Lecture Notes in Computer Science*, v. 2045, p. 1–35, 2001.

BERTOL, V.; SOUSA, R. T.; PEOTTA, L. Um modelo para as normas sobre certificação digital no brasil. *Procedings of the Four International Conference on Forensic Computer Science Investigation (ICoFCS 2009)*, 2009. Disponível em: <http://dx.doi-.org/10.5769/C2009006>. Acesso em: 01/08/2012.

BINSALLEEH, H.; ORMEROD, T.; BOUKHTOUTA, A.; SINHA, P.; A. On the analysis of the zeus botnet crimeware toolkit. *Proceedings of the Eighth Annual Conference on Privacy, Security and Trust (PST'2010)*, IEEE Press, Ottawa, ON, Canada, 2009.

BUCHMANN, J. *Introduction to Cryptography*. 2a.. ed. Darmstadt - Alemanha: Springer-Verlag Berlin Heidelberg, 2001. 312 p. ISBN 0-387-20756-2.

BURGE, P. Fraud detection and management in mobile telecommunications networks. In: *European Conference on Security and Detection - ECOS97 Incorporating the One Day Symposium on Technology Used for Combatting Fraud*. IEE, 1997. v. 1997, p. 91–96. ISBN 0 85296 683 0. Disponível em: <http://link.aip.org/link/IEECPS/v1997-/iCP437/p91/s1\&Agg=doi>. Acesso em: 01/08/2012.

BURR, W. E.; DODSON, D. F.; PERLNER, R. A.; POLK, W. T. *Electronic Authentication Guideline - Recommendations of the National Institute of Standards and Technology (Nist 800-63-1)*. Washington - DC: [s.n.], 2006. 108 p. ISSN 1668-3501.

BYERS, D.; SHAHMEHRI, N. Unified modeling of attacks, vulnerabilities and security activities. *Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems*, ACM Press, p. 36–42, 2010.

BYRES, E. J.; FRANZ, M.; MILLER, D. The use of attack trees in assessing vulnerabilities in scada systems. *International Infrastructure Survivability Workshop (IISW'04)*, IEEE Computer Society, Lisboa - Portugal, p. 1–9, 2004.

CALLEGATI, F.; CERRONI, W.; RAMILLI, M. Man-in-the-middle attack to the https protocol. *IEEE Security & Privacy Magazine*, v. 7, n. 1, p. 78–81, jan. 2009. ISSN 1540-7993.

CASEY, B.; MERRILL, D.; THAMM, J.; PIERCE, J. C.; WALLIS, M. E.; STONE, R. Mid-year trend and risk report. *IBM X-Force 2011*, n. September, 2011.

CHOUDARY, O. S. *The Smart Card Detective: a hand-held EMV interceptor*. 57 p. Dissertação (Mestrado) — University of Cambridge, 2010.

CONKLIN, A.; DIETRICH, G.; WALZ, D. Password-based authentication: a system perspective. *37th Annual Hawaii International Conference on System Sciences*, IEEE Comput. Soc. Press, Hawaii, v. 00, n. C, p. 170–179, 2004.

DODSON, B.; DEBANGSU, S.; BONEH, D. Secure, consumer-friendly web authentication and payments with a phone. In: *In Conference on Mobile Computing, Applications, and Services (MobiCASE 10)*. [S.l.: s.n.], 2010.

ECONOMIST, A.; UNIT, I. *Digital identity authentication in e-commerce*. London, 2007. 18 p.

FEBRABAN. CIAB Febraban 2009 - Bancarização Coletiva - O Setor Bancário em Números. *Congresso e Exposição de Tecnologia da Informação das Instituições Financeiras (CIAB 2009)*, Federação Brasileira de Bancos (Febraban), São Paulo, 2007.

FEITOSA, Y. N.; SANTIAGO, F. P. d. A.; PEOTTA, L. M. Avaliação de um sistema de gestão de identidade e acesso em uma organização pública federal. In: *SBSeg 2011*. Brasília - DF: SBC, 2011. Volume D.

FERRER, R. BB muda a maneira de certificar as transações na web. *INFO Online - Edição online*, 04 de Junho 2012. Disponível em: <http://goo.gl/0PcIm>. Acesso em: 02/07/2012.

FFIEC. Federal Financial Instituions Examinaton Council: Authentication in an Internet Banking Environment. IEEE, Arlington - VA, v. 1, n. 703, p. 23, 2005.

FONSECA, D. R. R. *SATES - Sistema de autenticação para transações eletrônicas seguras*. Dissertação (Mestrado) — Universidade de São Paulo, 2006.

GRANVILLE, L.; TAROUCO, L.; BARCELOS, R. R. Taxonomia de malwares: Uma avaliação dos malwares automaticamente propagados na rede. *IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (Sbseg 2009)*, Campinas - São Paulo, p. 43–56, 2009.

GROUP, U. B. W. *Use of Biometrics for Identification and Authentication Advice on Product Selection*. United Kingdom, 2002. 1–36 p.

GUHRING, P. *Concepts against Man-in-the-Browser Attacks*. [S.l.], 2007. 1–15 p.

HALLER, N.; METZ, C.; NESSER, P. *RFC 2289 (A One-Time Password System)*. IETF, 1998. 1–26 p. Disponível em: <http://tools.ietf.org/html/rfc2289>. Acesso em: 01/08/2012.

ISO/IEC 18004. ISO/IEC 18004 - Information Technology - Automatic Identification And Data Capture Techniques - Bar Code Symbology - QR Code. p. 122, 2006.

JOHNSON, M. *A New Approach to E-Banking - Second Year Research Report (PhD Thesis)*. [S.l.], 2007.

JOHNSON, M. *A new approach to Internet banking (Final Report PhD Thesis)*. 113 p. Tese (PhD) — University Cambridge, 2008.

JONES, P. *Secure Hash Algorithm 1 (SHA1) RFC 3174*. Internet Engineering Task Force, 2001. 1–22 p. Disponível em: <http://www.ietf.org/rfc/rfc3174.txt>. Acesso em: 01/08/2012.

JORSTAD, I.; JONVIK, T. Strong authentication with mobile phone as security token. *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, IEEE Computer Society, Macau, n. 1, p. 777–782, out. 2009.

JOSEFSSON, S. *The Base16, Base32, and Base64 Data Encodings*. [S.l.]: IETF, out. 2006. RFC 4648 (Proposed Standard). (Request for Comments, 4648).

KATZ, J.; LINDELL, Y. *Introduction to Modern Cryptography (Chapman Hall/Crc Cryptography and Network Security Series)*. Boca Raton - Flórida: Chapman Hall/-CRC, 2007. 534 p. ISBN 1584885513.

KOVACH, S. *Detecção De Fraudes Em Transações Financeiras Via Internet Em Tempo Real*. 134 p. Tese (Doutorado) — Universidade de São Paulo, 2011.

KOVACH, S.; RUGGIERO, W. V. Online banking fraud detection based on local and global behavior. *The Fifth International Conference on Digital Society (ICDS 2011)*, Gosier, Guadeloupe, France, p. 166–171, 2011.

LEUNG, C.-M. Visual security is feeble for anti-phishing. *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, IEEE Computer Society, Hong Kong, China, p. 118–123, ago. 2009.

LIBICKI, M. C.; BALKOVICH, E.; JACKSON, B. A.; RUDAVSKY, R. *Influences on the Adoption of Multifactor Authentication*. Arlington, VA, 2011. 63 p.

LTD, G. P. Two-factor authentication: An essential guide in the fight against internet fraud (whitepaper). Australia, p. 18, 2006.

MARTINO, A. S.; PERRAMON, X. Defending e-banking services: Antiphishing approach. *Second International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '08)*, IEEE Computer Society, Cap Esterel, p. 93–98, ago. 2008.

MELL, P.; KENT, K.; NUSBAUM, J. Guide to malware incident prevention and handling. *Computer Security Division Information Technology Laboratory National Institute of Standards and Technology*, Department of Homeland Security, Gaithersburg, v. 800-83, p. 101, 2005.

MORUM, A. S.; SíCOLI, F.; PEOTTA, L. M.; DEUS, F. G.; SOUSA, R. T. J. Acquisition and Analysis of Digital Evidence in Android Smartphones. *The International Journal of Forensic Computer Science*, v. 6, n. 1, p. 28–43, dez. 2011a. ISSN 18099807. Disponível em: <http://www.ijofcs.org/abstract-v06n1-pp02.html>. Acesso em: 01/08/2012.

MORUM, A. S.; SíCOLI, F. C.; PEOTTA, L. M.; DEUS, F. E. de; SOUSA, R. T. J. Aquisição de Evidências Digitais em Smartphones Android. In: (ED.)., D. d. P. F. (Ed.). *Procedings of the Sixth International Conference on Forensic Computer Science Investigation (ICoFCS 2011)*. Santa Catarina - Florianópolis - Brasil: [s.n.], 2011b. p. 92–99. ISBN 978-85-65069-07-6.

OLIVEIRA, C. Bancos devem preservar sigilo das transações. *Diário de Pernanbuco - Versão Impressa*, 29 de Junho 2012.

OSWALD, V. BB usará celulares contra hackers. *Jornal O Globo - Versão Impressa*, 01 de Junho 2012.

PAGET, F. *Roubo de identidade*. McAfee Avert Labs - Santa Clara, CA, 2007. 15 p.

PAYAO, F. Banco do Brasil lança QR-Code para segurança em Internet Banking. *PC Magazine - Edição online*, 04 de Junho 2012. Disponível em: <http://goo.gl/FB6vb>. Acesso em: 02/07/2012.

PEOTTA, L. M. *Proposta de metodologia de gestão de risco em ambientes corporativos na área de TI*. 181 p. Dissertação (mestrado) — Universidade de Brasília, 2008.

PEOTTA, L. M.; AMARAL, D. M.; DAVID, B. M.; CLAYTON, A. N.; SOUSA, R. T. J. *Sistema de Acesso com Segundo Fator de Autenticação para Transações Finan-*

ceiras e Comércio Eletrônico. 2011b. BR n. PI 1101459-8. Publicação: 01 abr. 2011, 20 dez. 2011.

PEOTTA, L. M.; AMARAL, D. M.; SAKAKIBARA, F.; ALMEIDA, A.; SOUSA, R. T. J.; CLAYTON, A. N. Análise de Malware: Investigação de Códigos Maliciosos Através de uma Abordagem Prática. In: SBC (Ed.). *Minicursos - Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2011)*. Brasília - DF: [s.n.], 2011c. p. 1–46. ISBN 978-85-7669-259-1.

PEOTTA, L. M.; HOLTZ, M. D.; DAVID, B. M.; DEUS, F. G.; SOUSA, R. T. J. A formal classification of internet banking attacks and vulnerabilities. *Journal of Computer Science and Information Technology (IJCSIT)*, v. 3, n. 1, p. 186–197, 2011a.

PEOTTA, L. M. P.; GONDIM, P. *Information Assurance and Security Technologies for Risk Assessment and Threat Management: Advances*. December. USA: IGI Global, 2011b. ISBN 9781613505076.

PUENTE, F. de L.; GONZALEZ, S.; SANDOVAL, J. Virus attack to the pc bank. *Proceedings IEEE 33rd Annual International Carnahan Conference on Security Technology*, IEEE Comput. Soc. Press, Madri, p. 304–310, 1999.

RECORDON, D.; REED, D. Openid 2.0: a platform for user-centric identity management. In: *Proceedings of the second ACM workshop on Digital identity management*. New York, NY, USA: ACM, 2006. (DIM '06), p. 11–16. ISBN 1-59593-547-9. Disponível em: <http://doi.acm.org/10.1145/1179529.1179532>. Acesso em: 01/08/2012.

RIVEST, R. *The MD5 message-digest algorithm (RFC1321)*. Internet Engineering Task Force, 1992. 1–22 p. Disponível em: <http://www.ietf.org/rfc/rfc1321.txt>. Acesso em: 01/08/2012.

SAINI, V.; DUAN, Q.; PARUCHURI, V. Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, Consortium for Computing Sciences in Colleges, v. 23, n. 4, p. 124–131, 2008.

SCHMITZ, R. A novel anti-phishing framework based on honeypots. *eCrime Researchers Summit (eCRIME '09)*, IEEE Computer Society, Tacoma, WA, p. 1–13, out. 2009.

SEUNGJAE, S.; CUNNINGHAM, J.; RYOO, J. A study of two-factor authentication against on-line identity theft. *Proceedings of the 39th Annual Meeting of the Decision Sciences Institute*, Baltimore - Maryland, n. 601, p. 11001–11006, 2008.

SHANNON, C. E. The mathematical theory of communication. *The Bell system technical journal*, v. 27, n. 4, p. 379–423, 1948. ISSN 0724-6811.

STARNBERGER, G.; FROIHOFER, L.; GOESCHKA, K. M. Qr-tan: Secure mobile transaction authentication. *2009 International Conference on Availability, Reliability and Security*, IEEE, p. 578–583, mar. 2009.

STEDING-JESSEN, K. *Uso de Honeypots para o estudo de Spam e Phishing*. 204 p. Tese (Doutorado) — INPE - Instituto Nacional de Pesquisas Espaciais, 2008.

TI-INSIDE. Clientes do BB ganham mais segurança para acessar internet banking. *TI Inside Online - Edição online*, 04 de Junho 2012. Disponível em: <http://goo.gl-/eIn9C>. Acesso em: 02/07/2012.

TUBIN, G. Using mobile-based security to combat new fraud threats. *Online*, n. May, p. 1–11, 2011.

VICENTE, J. P. O que eu criei para você: Senha Única. *Revista Darcy*, v. 11, p. 18–19, Junho e Julho 2012.

WIESMAIER, A.; FISCHER, M.; LIPPERT, M.; BUCHMANN, J. Outflanking and securely using the pin/tan-system. *Proceedings of the International Conference on Security and Management (SAM'05)*, Las Vegas, Nevada, p. 7, 2005.

WONGTSCHOWSKI, A. *Segurança Em Aplicações Transacionais Na Internet: O Elo Mais Fraco*. 108 p. Dissertação (Mestrado) — Universidade de São Paulo, 2005.

YOUNG, S. L.; NACK, H. K.; HYOTAEK, L.; JO, H. Online banking authentication system using mobile-OTP with QR-code. In: *International Conference on Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th*. Seoul, Korea: [s.n.], 2010. p. 644–648. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all-.jsp?arnumber=5711134>. Acesso em: 01/08/2012.

YU, W. D.; NARGUNDKAR, S.; TIRUTHANI, N. A phishing vulnerability analysis of web based systems. *IEEE Symposium on Computers and Communications (ISCC 2008)*, IEEE Computer Society, Marrakech, p. 326–331, jul. 2008.

ANEXOS

A LISTA DE ARTEFATOS ANALISADOS

Tabela A.1: Lista dos Artefatos Analisados

Assinatura MD5	1	2	3	4	5	6	7	8	9	10
0ac14a4f1efa58cd2d1a75e9d6d3b2d57c68001b1616060a4b3fdef29ddf4f10		19	21	22	24	23	24	25	26	28
0d985f70d5644acc7379ad384bdf31a85fa052b159690d61e81afc2bd0f0c81e			14	17	18	18	18	18	18	18
1ecb902a13230ade8322b04e6c11d1ce5771d8a4be8352d2b64455a786ac26fd		24	27	28	30	30	31	32	35	35
1ef6a850146db1b64d783f434081d49526bb674c65355f33575a966415147909		20		23	25	26	28	28		29
2bb688f9a438229b703b9a97c1d78798764836ddcd80a3008d73539be701ba78	11	19	22	23	24	24	25	25	25	27
2d3f397ce79b288048be08f45efed1593a478fae18250c28c25e2015db0d0d0f	9	16	17	16,5	18	18,5	18	19	19	18,5
3b67fd387779a5576c5c042d7af6f3c047ab06c6ed8aa053292f271d667c578e	16	21	23	23	26	24	28	28	29	30
5c538cd37ed83a575d3a1bb6d1cc1963ab8f56c5a658a54949654d1d0c04eb22		31	32	32	32	32	26	32	32	32
7d0a0a6bc6843543958c6cfc23cdfb08869b578eb28b1c03f9d5a367ca55945b	28	30	28	28	31	32	32	33	34	34
7f82d2dd7c63298c9938d375564123cd9050f7f07b3287a90c38c1eff48fb832	22	28	28	30	30	31	31	32	31	31
8aabc1e4c9ce2a83598f191c744a2beae75a88029904f875d61bf5702510c28e		25	27	27	28	29	29	30		30
8bc19b043b857d3bbd5ad92fa535e31724888ebe5a59ce47b4ef562df7a0417f	15	21	25,5	28	30,5	30,5	31	32	32	32
8bf35a2861f2da3f55b26deabae1a3b2540bfb49553ffa1bfcafba7451decdc3	17	22,5	28	27,5	28,5	28	31	32	33	32
8c90dfed49cc28a17765e5db0149c10a54cb3269ba5b35fe348c114199567eb1		14	18	21	22	22	22	22	22	22
8d040644ab1278cd5a9380f279b78882d66f0bbfd904acc5b068cd556847a358	3	13	16	17	20	21	21	21	21	22
8de37d0c293258f85e26083238a98cd39ca016eba26b0820fbc70b3316e3e22c	7		20	21	24	24	26	26	26	26
8de99ef34279e389b933dd69bb6aa76b31b6ff5ca8960918150e18c19b11958e		29	32	32	31	31	31	32	32	33
8f8527ba79b045a61919f58fd21686ab6d3b76462d139f8dff60c17e99555b5c		34	34	34	34	35	34	35	35	35
04ed56e068c8a6abcfec1c82a3a8a251cea59dfaad9e8e9271308f9850e03072			15	19	20	21	22	22	21	22
11bc078e63c06f4496fdabc04a121be43ffd4f40ec449add385bce41849b44c1			23	24	23	25	26	27	28	26
18c42bf10ed2e5023be22e0dd62083c660114fc7fa1cd82e7d6ab3cc16fc733d	18	24	25	25	26	26	25	26	26	26
18e983c18272acd7484104c91c87b52f71d188e20f72fd1ecb62ddc48acfe460	22	26	26	27	27	27	29	29	30	30
22b81672cbdc44f7b2d4a15cdfa24b97f934a6fad9aeaeeb75620b7265b3c79f		27	30	30	32	32	32	33		34
37e8432f580d07472d7bcc9b5d1887ac8ee0be78e3f15f194223cfb224fd8eff	18	23	26	29	29	29	30	32	32	33
49f8f12c0e0ec3532a8ba624335eabeaa783e547d7abcfecd09150720e841dd7	3	13	16	18	18	18	18	19	18	20
50d61f5ea67265c359797f04a68700a9250eb0e1cd807985afb3879efca86358		34	34	34	34	34	35	35	35	35
52f4915481ed4ae536238a42fa56cecaefd8853d3b66763302a904d47bce2573		28		28	29	30	31	31	31	32
56a45af2f96348ccb65c61b724db096b04be652250ad739b4477f53fa9c2b4be	18	24	28	27	28	29	31	31	31	31
61c27a71b3b47e0e78366dd2fef8f2b3a8988c0b08901d0418dc0775e1c1005a		20	23	22	22	23	24	27	27	27
73bde3648628d313da77e0e9188baf000556ed22a175c1208a5cb0ae72f62fd0		30	30	30	32	33	35	35	35	35
75b98e4b8605a9edb9c71638902bf0ac4f39dfeffedfdb5eeba4fd60cac207e9	17	20	21	22	22	23	24	25	25	26
81d8e329709a232b2face5dcf89d8f140f3ed2114d5fb1f86ba165f7f871d001		24	25	24	26	27	28	29	30	30
86a35c857e15efff54219b112fbda3c056b99089315f67a7a93c092e002d9249	19	21	26	28	29	28	31	32	32	34
069de6667136ed0d3c54fa8a64927a9aece3bad982dc3c4f92cb4713da8a34a8	25	28		30	30	31	32	34	35	35
090de62f9bb1dd7b90e7087f16e0675c4deb1dd4b5b2b1e89668683c7540bce3	24		29	29	30	31	32	34	35	35
179abec8a9415bbdd4769220cccdf1d49d87c95f21c1797b754f3c7fead0ee9b	26	30	33	33	33	32	34	34	34	35
259ece9fb5695f8b2fd9abb23feb3e489fcddef711c976746cf2fb711751467d		23		26	28	29	30	31		33
391e13728681bf6fbab9a61dc0296d70f91476e68d8eb9a0e25dd96ed97ca366	14	26	29	29	30	30	30	30	32	33
441d888c4701b9929425c922bb80ccb23a30398fa21e9eb57b487585eed9eb88		25	25	25	25	26	25	27	26	27
448dccc2f83fa648cd7376120f064100c5b0d0ae2ebaa339d2c68df448efa3ca	38	38	38	39	38	39	39	39	37	38
486cb3c32da9744e9577694ef4566916fd2a3504d9b6e117c4a96b00e2a24852	17	23	25	25	27	27	27	29	29	29
493ba693edc0b46001c4051c8a850006c9a1c446aeacf0bbb51addca324a5e3d		29	30	31	32	32	34	32	33	35
591c37837dc85bc7c1b8cfa281163784bb5d12e890ff64b08e16f0351d236a1b		22		22	23	25	29	28	30	30
712e665968958952c543ff7572685aa0f542ecad4d65b9019bf43418dad3c226	15	20	22	22,5	22	23,5	26	28	27,5	28
752a9508c484f194f4ea57f439b13a3a01864ab576306d67b0d1c1c119a5e92d		6	7	7	7	7	6	7	7	7
766d8826667203fd0905286f898f591a0a0746cbfef81d050300822d1f1830b1		4		11	11	11	12	12	11	11
788c814181b588f56b064bdfcb644936eeb01a2f9a1ffa981d60d314a10d3b2d	12	18		19	18	19	19	19		20
855e0dc721e4a01d6a5a2daf775f15fc2d97cfc8123e84ee11ced24fc477accc	13,5		25	24,5	28,5	29	29	29	29	29
964e07605da3eba368dd35b11eca34c540bc0d720fb19e8988c98cbcab567d12		22		30	29	29	30	30	30	30
3576c11d2eb206d9ca2b518856ac5e9e6ebbcfb482b3d47d4143569a7927c381	9			20				22	23	25
3976e02adac57e4e6d950183cc3f808a0a38378468be5cd3a0f5b7d91c8fc8ec		26	28	29	29	30	32	30	31	31
5212b779122bc14af406460c60bcd4a5c4198e5625ff22f15e49a070089cd4bf		31	31	32	31	32	31	32		32
7076e3d07d3d7497a742154255418f534ebb001ba8e0de2d1c4e21bee024657b		20	24	25	26	28	27	28	30	30
8399e858d30e1b5de4538bad725fdc9110ce0aae9396177bf5282b13463e9de0		14	17	19	21	21	21	21	22	22
35352fe3c60f599066152474dae4ed28c6f26b8622908165679925c4be035fde		21	23	22		25	26	25	27	28
57203ed828e335e905d22868294b7f4db7a2a0a99960a47dc7e511f6cbf4e074	30	29	29	29	30	30	30	30		30
83094acda21bb7efa0f04c9843c989b0ea353fcacf9e32cf9e8e2151cd31c66d		29	29,5	29,5	31	31	31	31,5	31	32,5
94786f0187862b69a93d1b0148538f3572385c1a3dc1ec1f20734a0e649c832a	22	24	27	27	28	29	31	30	32	32
954231b80e1c8501d258ef63cbb8979c26e205bfb267afce8ff78073610400b4	13	15	17	17	18	18	17	18	19	19
8297980ab2df9ce244f874b8bc3dd4895000c9f3efc48a83f603e8dc4fd081a7	17	21	24	23	24	24	26	27	28	28
a4ad3c259d4faff87294d874d0a596fbd19710c174070b9661690ac6e830d044	21	27	28	29	30	31	31	32	31	32
a439b150c57f38b24e75210704d25f23fd3e2e0e9bab27a05872c769639f3052		22	23	23	24	24	24	24	23	23
af41a7d6c9c806f1c4f010f03a96ebbfce223a6e8850ab87f99130f97106c409		25	26	26	29	30	29	31	31	32

continua na próxima página

Tabela A.1: Lista dos Artefatos Analisados (continuação)

Assinatura MD5	1	2	3	4	5	6	7	8	9	10
b7fbe2d96ccdd41170143ddb1beb362dacebf47b4e9b5a33a151884c850ccdf3	19		27	27				30	31	33
b50dd9808dcc81fc08fcf0a7ccaf60b22467c64c4d49495b33c4533a764cf21a	7	18	21	22	23	23	24	24	25	23
b1607e4119faaf6c7f03a70248e6052949cbbfa82035df596685e16482368667		24		25	27	27	29	29	29	29
c17b11e95dfe466b0b78748eef29a61348b29eb732085dd824325ea190c346ce			25	25				26	26	26
c42e8461941b83b3839051b68913c9b2371cd9ecb38cb7ca4196634606cd3ca7		22	24	23	24	24	25	25	25	26
c109f7d8ff5581de7c19c1c8367f30d1b0fe109dbb34512ea74bb252850654ee	6		16	19	21	22	22	23	23	24
c10460c3833144f67bee7c998d7460a1d5729e62bc0ea5b7bfb72abacb61237e		25,5	26	27,5	28,5	29	27	27	25,5	29,5
cb1d94fedc0e537af281125297b1a4d7d1de5dbf5b528e2768556407d0f8e691		15	20	19	23	23	24	24	25	26
cc743b6d9330d04a96a0dab666199407ea6a0f1d46ab319370f6a6072cf7189a	23	26	29	29	30	30	31	34	36	36
d1f76d5808165c6c826037b83ba22dfbf298959c46c684b43f1f2eb06fd43a41	27	29	29	28	30	32	33	33	33	33
d2f5daa0edd4d8842f44da118f0f1d2db4196f0ebf73660d107a241796284745		23	24	24	27	27	28	29	31	31
d2918a159bb13e472f040f1a39b36997e6e4e6f24a775078d9bd2f1ba39c40cc		15	22	22	23	23	24	25	25	27
d5661d74a57e68a3cc423339b57616e779afb9e15e976b62953515cbccae7692			16	17	18	18	19	19	18	20
e9bf81c14a429e8908b1b0121fdb149b0d093b769d10f9219f0a07a6acef7d47	14	21	23	23	23	23	25	26	27	27
e41cf3d8c68cd3349f7ad690863bd1e5b70be0f6ebc6b75a21489012ab6b2aaa	27	29	30	29	30	30	31	31	32	32
e79d6ee8a29d0305f70bdc612a400511abb00a67930a0f8475268b7a87a61fc6	15	17	16	15	16	17	17	17	17	17
e330770d4e681178efde920c80c0679116527fc3355fea28172aff45b247b94f	11	22	23	24	25	25	26	27	27	28
ea4916b540eb9062715bb9154c01cfa00bbae61c11a3e2cd88efa4d2c0baff93		22	22	21	21	21	23	24	25	26
ed62cdd2a832fe5fc411cd79a1bec2f021d1fd8a7eed8b5fac5964dbd3afa2dd	15	21	27	27	27,5	29	29	29	29	30
f1c98aada95c64d52d97c5b3a99e832d7bd980c83622278f1dde36c7bc25a333		25	27	28	29	29	31	32	32	33
fb3d44c7f2b7aed88345944c5545711673469242e8cc325a6b45a6968b2858da	27	30	30	29	30	29	31	31	31	32
fb6e116594f4d4ab789188d72d3944bc1e2c10cb229d1f95bd68a18a4c868cc0		28	28	30	31	30	31	31	32	32
fca129038121d505265914d7d4ebce7327ee2111d3413148fb735283defd65b9	23		27	28	29	28	31	31	29	32
fe08c43fd02eb894ac77642b319e5111bde880778f05940c8041bfafe752d391	21	25	27	29	29	30	30	31	30	28

B ANÁLISE DAS AMOSTRAS PARA DETECÇÃO DE ARTEFATOS - CÓDIGOS FONTES

Código Fonte B.1: Módulo para envio dos artefatos para análise

```
1  #!\bin\bash
2  #envia trojans para o virustotal
3
4  data=date +'%Y-%m-%d'
5  dia=01
6  hora=date +'%H:%M'
7  casa=/root/trojans
8
9
10 inicia os contadores
11 contador=0
12 contador1=1
13
14 #pega o únmero total de arquivos a enviar
15 contador2=ls casa/trojans_dia | wc -l
16
17
18 #Cria os ódiretrios para armazenar saidas de envio e resposta
19 mkdir -p casa/dia/reports
20 mkdir -p casa/dia/envio
21
22
23 for i in ls casa/trojans_dia; do
24
25    echo enviando: i contador1 de contador2
26
27          virustotal.py --mode scan --type file --output casa/dia/envio/
               saida-i-data casa/trojans_dia/i
28    let contador=contador+1;
29          let contador1=contador1+1;
30    echo arquivo enviado: i
31
32 #espera 5 minutos devido a limitacao de analise em 20 arquivos
33    if [ contador -gt 20 ]; then
34          echo "esperando! ja enviei 20 arquivos"
35          sleep 300
```

```
36              contador=0
37              fi
38  done
```

Código Fonte B.2: Módulo para coleta de relatórios das análises dos artefatos enviados

```
#!\bin\bash
#pega relatorio virustotal

data=date +'%Y-%m-%d'
dia=01
hora=date +'%H:%M'
casa=/root/trojans


#inicia os contadores
contador=0
contador1=1

#pega o únmero total de arquivos a enviar
contador2= ls casa/trojans_dia | wc -l


for i in ls casa/trojans_dia; do

    echo Relatorio: i contador1 de contador2
    virustotal.py --mode report --type file --output casa/dia/reports/
        report-i-data casa/dia/envio/saida-i-data
    let contador=contador+1;
    let contador1=contador1+1;
    echo Relatorio baixado: i

    if [ contador -gt 20 ]; then
          echo "esperando! ja peguei 20 relatorios. Estou em contador1 de
              contador2"
          sleep 300
          contador=0
          fi
done
```

C PROVA DE CONCEITO - FONTES

Código Fonte C.1: Módulo de autenticação e verificação das chaves

```
1  import java.io.File;
2  import java.io.FileInputStream;
3
4  public class Auth {
5     public static void main(String[] args) throws Exception {
6          File chaves = new File("chaves.txt");
7          if (!chaves.exists()){
8            Passo1 server1 =  new Passo1();
9            server1.run();
10           Passo2 phone1 = new Passo2(server1.qrcode, server1.sms);
11           phone1.run();
12         }
13         byte[] k1 = new byte[Fase.HMAC];
14         byte[] k2 = new byte[Fase.AES];
15        FileInputStream fis = new FileInputStream(chaves);
16        fis.read(k1);
17        fis.read(k2);
18        fis.close();
19        Passo3 server2 = new Passo3(k1, k2, args[0]);
20        server2.run();
21        Passo4 phone2 = new Passo4(server2.message, k1, k2);
22        phone2.run();
23     }
24 }
```

Código Fonte C.2: Passo1: Gerar chaves no servidor

```
1  import java.io.FileOutputStream;
2  import java.util.Random;
3  import javax.crypto.*;
4
5  public class Passo1 extends Fase{
6      String qrcode = "";
7      String sms = "";
8
9      @Override
10      public void run() throws Exception{
11        System.out.println("*** Passo1: gerar chave no servidor ***");
12        Random randomGenerator = new Random();
13        byte[] v = new byte[V];
14          byte[] cv = new byte[V];
15        for (int i = 0; i < v.length; i++) {
16           v[i] = (byte) randomGenerator.nextInt(10);
17           cv[i] = (byte) randomGenerator.nextInt(255);
18        }
19
20          KeyGenerator kg;
21          kg = KeyGenerator.getInstance("HmacSHA1");
22          kg.init(HMAC*8);
23          SecretKey sK1 = kg.generateKey();
24          kg = KeyGenerator.getInstance("AES");
25          kg.init(AES*8);
26          SecretKey sK2 = kg.generateKey();
27
28          kg = KeyGenerator.getInstance("HmacSHA1");
29          kg.init(HMAC*8);
30          SecretKey cK1 = kg.generateKey();
31          kg = KeyGenerator.getInstance("AES");
32          kg.init(AES*8);
33          SecretKey cK2 = kg.generateKey();
34
35          byte[] cvl = xor(v, cv);
36        byte[] ck1l = xor(sK1.getEncoded(),cK1.getEncoded());
37        byte[] ck2l = xor(sK2.getEncoded(),cK2.getEncoded());
38
39 //         System.out.print("Codigo de verificacao: ");
40          for (byte b : v) {
41           System.out.printf("%d",b);
42          }
43 //         System.out.println("");
```

```
44
45          k1 = sK1.getEncoded();
46          printBytes("Chave gerada do HMAC: ", k1);
47          k2 = sK2.getEncoded();
48          printBytes("Chave gerada do AES: ", k2);
49          System.out.println("");
50
51          FileOutputStream fos = new FileOutputStream("chaves.txt");
52          String chave = "\nHMAC = ";
53            for (byte b : k1) {
54             chave += String.format("%02X", b);
55          }
56          fos.write(k1);
57          chave += "\nAES = ";
58            for (byte b : k2) {
59             chave += String.format("%02X", b);
60          }
61          fos.write(k2);
62          fos.write(chave.getBytes());
63          fos.close();
64
65  //          System.out.print("Codigo de verificacao CV: ");
66            for (byte b : cv) {
67             System.out.printf("%02X",b);
68             sms += String.format("%02X",b);
69            }
70  //          System.out.println("");
71
72          System.out.print("Chave gerada do HMAC CK1: ");
73            for (byte b : cK1.getEncoded()) {
74             System.out.printf("%02X",b);
75             sms += String.format("%02X",b);
76            }
77          System.out.println("");
78
79          System.out.print("Chave gerada do AES CK2: ");
80            for (byte b : cK2.getEncoded()) {
81             System.out.printf("%02X",b);
82             sms += String.format("%02X",b);
83            }
84          System.out.println('\n');
85
86          System.out.print("Codigo de verificacao CV\': ");
87            for (byte b : cvl) {
88             System.out.printf("%02X",b);
```

```
89              qrcode += String.format("%02X",b);
90            }
91          System.out.println("");
92
93          System.out.print("Chave gerada do HMAC CK1\': ");
94            for (byte b : ck1l) {
95             System.out.printf("%02X",b);
96             qrcode += String.format("%02X",b);
97            }
98          System.out.println("");
99
100         System.out.print("Chave gerada do AES CK2\': ");
101           for (byte b : ck2l) {
102            System.out.printf("%02X",b);
103            qrcode += String.format("%02X",b);
104           }
105         System.out.println('\n');
106
107         System.out.println("QRCODE: " + qrcode);
108         System.out.println("SMS: " + sms);
109         System.out.println("");
110     }
111 }
```

Código Fonte C.3: Passo2: Gravar chave k no dispositivo móvel

```
1  public class Passo2 extends Fase{
2      String qrcode = "";
3      String sms = "";
4
5      public Passo2(String qrcode, String sms ) {
6          this.qrcode = qrcode;
7          this.sms = sms;
8      }
9
10     @Override
11      public void run() throws Exception{
12         System.out.print("*** Passo2: gravar chave no celular ***");
13         byte[] cvl = getBytes(qrcode.substring(0,V*2));
14         byte[] ck1l = getBytes(qrcode.substring(V*2,V*2+HMAC*2));
15         byte[] ck2l = getBytes(qrcode.substring(V*2+HMAC*2,V*2+HMAC*2+AES
               *2));
16         byte[] cv = getBytes(sms.substring(0,V*2));
17         byte[] ck1 = getBytes(sms.substring(V*2,V*2+HMAC*2));
18         byte[] ck2 = getBytes(sms.substring(V*2+HMAC*2,V*2+HMAC*2+AES*2));
19         byte[] v = xor(cvl, cv);
20         k1 = xor(ck1l, ck1);
21         k2 = xor(ck2l, ck2);
22         System.out.println("");
23
24         System.out.print("Codigo de verificacao: ");
25           for (byte b : v) {
26            System.out.printf("%d",b);
27           }
28         System.out.println("");
29         printBytes("Chave k1: ", k1);
30         printBytes("Chave k2: ", k2);
31         System.out.println("");
32      }
33
34
35 }
```

Código Fonte C.4: Passo3: Gerar código autorizador para transação no servidor

```
1  import java.io.File;
2  import java.io.FileInputStream;
3  import java.util.LinkedList;
4  import java.util.Random;
5
6  import javax.crypto.*;
7  import javax.crypto.spec.SecretKeySpec;
8
9  public class Passo3 extends Fase{
10     String message = "";
11     LinkedList<Byte> header = new LinkedList<Byte>();
12     File arqTran;
13
14     Passo3(byte[] k1, byte k2[], String args){
15        this.k1 = k1;
16        this.k2 = k2;
17        arqTran = new File(args);
18     }
19
20      public void run() throws Exception{
21        System.out.println("*** Passo3: gerar senhas no servidor ***");
22        int TRANSACTION = (int) arqTran.length();
23        byte[] transaction = new byte[TRANSACTION];
24        for (int i = 0; i < transaction.length; i++) {
25           transaction[i] = ' ';
26        }
27        FileInputStream fis = new FileInputStream(arqTran);
28        fis.read(transaction);
29        fis.close();
30        System.out.println("Mensagem:\n" + new String(transaction));
31
32        Random randomGenerator = new Random();
33          byte[] passwd = new byte[PASSWD];
34          byte[] nonce = new byte[NONCE];
35        for (int i = 0; i < passwd.length; i++) {
36           passwd[i] = (byte) randomGenerator.nextInt(10);
37        }
38        for (int i = 0; i < nonce.length; i++) {
39           nonce[i] = (byte) randomGenerator.nextInt(255);
40           message += String.format("%02X", nonce[i]);
41           header.add(nonce[i]);
42        }
43
```

```
44          SecretKey sK2 = new SecretKeySpec(k2, "AES");
45          Cipher cipher = Cipher.getInstance("AES");
46          cipher.init(Cipher.ENCRYPT_MODE, sK2);
47          byte[] encrypted = cipher.doFinal(passwd);
48        System.out.print("Senha: ");
49          for (byte b : passwd) {
50           System.out.printf("%d",b);
51        }
52        System.out.println("");
53        System.out.print("Senha encriptada: ");
54          for (byte b : encrypted) {
55           System.out.printf("%02X",b);
56           message += String.format("%02X", b);
57           header.add(b);
58        }
59        System.out.println("");
60
61        for (byte b : transaction) {
62           header.add(b);
63           message += String.format("%c", b);
64        }
65        byte[] messageHmac = new byte[header.size()];
66        for (int i = 0; i < header.size(); i++) {
67           messageHmac[i] = header.get(i);
68        }
69
70        SecretKey sK1 = new SecretKeySpec(k1, "HmacSHA1");
71          Mac mac = Mac.getInstance("HmacSHA1");
72          mac.init(sK1);
73          byte[] result = mac.doFinal(messageHmac);
74        System.out.print("HMAC: ");
75        String hmacStr = "";
76          for (byte b : result) {
77           System.out.printf("%02X",b);
78           hmacStr += String.format("%02X", b);
79        }
80          message = message.substring(0,NONCE*2+AES*2) + hmacStr + message.
                substring(NONCE*2+AES*2, message.length());
81        System.out.println("\n");
82        System.out.println("Mensagem a ser enviada:\n" + message);
83        System.out.println("");
84      }
85  }
```

Código Fonte C.5: Passo4: Decifrar código autorizador no dispositivo móvel

```
1  import java.util.LinkedList;
2
3  import javax.crypto.Cipher;
4  import javax.crypto.Mac;
5  import javax.crypto.SecretKey;
6  import javax.crypto.spec.SecretKeySpec;
7
8
9  public class Passo4 extends Fase {
10    String message = "";
11
12    Passo4(String message, byte[] k1, byte[] k2){
13       this.message = message;
14       this.k1 = k1;
15       this.k2 = k2;
16    }
17
18     public void run() throws Exception{
19       System.out.println("*** Passo4: decriptar senha no celular ***");
20       LinkedList<Byte> buffer = new LinkedList<Byte>();
21       for (Byte i: getBytes(message.substring(0,NONCE*2+AES*2))) {
22          buffer.add(i);
23       }
24       for (Byte i : message.substring(NONCE*2+AES*2+HMAC*2,message.length
             ()).getBytes()) {
25          buffer.add(i);
26       }
27       byte[] messageHmac = new byte[buffer.size()];
28       for (int i = 0; i < buffer.size(); i++) {
29          messageHmac[i] = buffer.get(i);
30       }
31
32          SecretKey sK1 = new SecretKeySpec(k1, "HmacSHA1");
33         Mac mac = Mac.getInstance("HmacSHA1");
34         mac.init(sK1);
35         byte[] result = mac.doFinal(messageHmac);
36       System.out.print("HMAC: ");
37       String HMACclient = "";
38         for (byte b : result) {
39          System.out.printf("%02X",b);
40          HMACclient += String.format("%02X", b);
41       }
42       System.out.print("\n");
```

```
43          String HMACserver = message.substring(NONCE*2+AES*2, NONCE*2+AES*2+
                HMAC*2);
44          if (HMACclient.contains(HMACserver)){
45              SecretKeySpec sK2 = new SecretKeySpec(k2, "AES");
46                Cipher cipher = Cipher.getInstance("AES");
47              cipher.init(Cipher.DECRYPT_MODE, sK2);
48                byte[] dencrypted = cipher.doFinal(getBytes(message.substring(
                      NONCE*2, NONCE*2+AES*2)));
49              System.out.print("Senha: ");
50                for (byte b : dencrypted) {
51                  System.out.printf("%d",b);
52              }
53          }else{
54              System.out.println("HMAC Ã£no coincide");
55          }
56          System.out.println();
57      }
58
59  }
```

Código Fonte C.6: Prova de conceito geral

```
1
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5  import java.io.*;
6  import java.net.*;
7  import java.security.NoSuchAlgorithmException;
8
9  import javax.crypto.*;
10 import javax.imageio.ImageIO;
11 import javax.swing.*;
12
13
14 public class poc {
15    static JFrame frame = new JFrame();
16    static JLabel labelqr = new JLabel();
17     static JLabel label = new JLabel();
18     static Image image = null;
19     static String[] ret = new String[2];
20     static  JTextArea jtAreaOutput = new JTextArea(5, 20);
21     static transaction keyExchange = new transaction(1);
22
23    //Esquema bruto pra converter byte[] pra string de hexa!!!!!!
24    static final String HEXES = "0123456789ABCDEF";
25      public static String getHex( byte [] raw ) {
26        if ( raw == null ) {
27          return null;
28        }
29        final StringBuilder hex = new StringBuilder( 2 * raw.length );
30        for ( final byte b : raw ) {
31          hex.append(HEXES.charAt((b & 0xF0) >> 4))
32              .append(HEXES.charAt((b & 0x0F)));
33        }
34        return hex.toString();
35      }
36
37    public static String[] genkeys() {
38       String message = "v, k1, k2, cv, ck1, ck2";
39       System.out.println("Gerando chaves: " + '"' + message + '"');
40
41         // Generate secret key for HMAC-MD5
42       try {
43         KeyGenerator kg = KeyGenerator.getInstance("HmacSHA1");
```

```
44          kg.init(64);
45          SecretKey v = kg.generateKey();
46          kg.init(160);
47          SecretKey k1 = kg.generateKey();
48          kg.init(256);
49          SecretKey k2 = kg.generateKey();
50
51          kg.init(64);
52          SecretKey cv = kg.generateKey();
53          kg.init(160);
54          SecretKey ck1 = kg.generateKey();
55          kg.init(256);
56          SecretKey ck2 = kg.generateKey();
57
58  //        SecretKey qrv = v ^ cv;
59  //        SecretKey qrk1 = k1 ^ ck1;
60  //      SecretKey qrk2 = k2 ^ ck2;
61  //      qrv + qrk1 + qrk2; qrcode
62  //      cv + ck1 + ck2; SMS
63      System.out.print("Chave gerada v: ");
64          for (byte b : v.getEncoded()) {
65              System.out.printf("%02X",b);
66          }
67      System.out.println("");
68
69
70      System.out.print("Chave gerada k1: ");
71
72          for (byte b : k1.getEncoded()) {
73              System.out.printf("%02X",b);
74          }
75      System.out.println("");
76
77      System.out.print("Chave gerada k2: ");
78          for (byte b : k2.getEncoded()) {
79              System.out.printf("%02X",b);
80          }
81      System.out.println("");
82
83      System.out.print("Chave gerada cv: ");
84          for (byte b : cv.getEncoded()) {
85              System.out.printf("%02X",b);
86          }
87      System.out.println("");
88
```

```
89      System.out.print("Chave gerada ck1: ");
90          for (byte b : ck1.getEncoded()) {
91              System.out.printf("%02X",b);
92          }
93      System.out.println("");
94
95      System.out.print("Chave gerada ck2: ");
96          for (byte b : ck2.getEncoded()) {
97              System.out.printf("%02X",b);
98          }
99
100
101         byte[] vbyte = v.getEncoded();
102         byte[] cvbyte = cv.getEncoded();
103         byte[] vprime = new byte[vbyte.length];
104          for (int i=0; i < vbyte.length; i++) {
105
106             vprime[i] = (byte) (vbyte[i] ^ cvbyte[i]);
107
108         }
109
110
111
112         System.out.print("\nv': ");
113
114         for (byte b : vprime) {
115             System.out.printf("%02X",b);
116         }
117
118         byte[] k1byte = k1.getEncoded();
119         byte[] ck1byte = ck1.getEncoded();
120         byte[] k1prime = new byte[k1byte.length];
121          for (int i=0; i < k1byte.length; i++) {
122
123             k1prime[i] = (byte) (k1byte[i] ^ ck1byte[i]);
124         }
125
126         System.out.print("\nk1': ");
127         for (byte b : k1prime) {
128             System.out.printf("%02X",b);
129         }
130
131         byte[] k2byte = k2.getEncoded();
132         byte[] ck2byte = ck2.getEncoded();
133         byte[] k2prime = new byte[k2byte.length];
```

```
134         for (int i=0; i < k2byte.length; i++) {
135
136             k2prime[i] = (byte) (k2byte[i] ^ ck2byte[i]);
137         }
138
139         System.out.print("\nk2': ");
140         for (byte b : k2prime) {
141             System.out.printf("%02X",b);
142         }
143
144         String vpstring = getHex(vprime);
145         String k1pstring = getHex(k1prime);
146         String k2pstring = getHex(k2prime);
147         keyExchange.setField(6, vpstring);
148         keyExchange.setField(7, k1pstring);
149         keyExchange.setField(8, k2pstring);
150         //String qrcode = vpstring+k1pstring+k2pstring;
151         String qrcode = keyExchange.getContent();
152
153         System.out.println("\nQR-Code: "+qrcode);
154
155         String cvstring = getHex(cvbyte);
156         String ck1string = getHex(ck1byte);
157         String ck2string = getHex(ck2byte);
158         String sms = cvstring+ck1string+ck2string;
159
160         System.out.println("    SMS: "+cvstring+ck1string+ck2string);
161
162     System.out.println("");
163
164     String[] ret = {qrcode,sms};
165
166     return ret;
167
168         } catch (NoSuchAlgorithmException e){
169             System.out.println("System crashed with exception: "+ e.
                    getMessage());
170         }
171
172
173         String[] ret = {"error"};
174         return ret;
175
176     }
177
```

```
178     static public void update() {
179        ret = genkeys();
180
181          try {
182              // Read from a URL
183              //URL url = new URL("http://qrcode.kaywa.com/code/"+ret[0]);
184           URL url = new URL("http://chart.apis.google.com/chart?cht=qr&chs
                 =350x350&chl="+ret[0]);
185              image = ImageIO.read(url);
186          } catch (IOException e) {
187          }
188
189          label.setIcon(new ImageIcon(image));
190
191          jtAreaOutput.setText(ret[1]);
192
193     }
194
195     public static void drawui() {
196         // Use a label to display the image
197          frame.getContentPane().setLayout(new GridBagLayout());
198          frame.setSize(500, 570);
199          frame.setLocation(100, 100);
200          frame.setTitle("ATM: Key Exchange");
201
202          GridBagConstraints c = new GridBagConstraints();
203
204        labelqr.setText("QR-CODE");
205        c.weightx = 0.0;
206        c.fill = GridBagConstraints.HORIZONTAL;
207        c.gridx = 0;
208        c.gridy = 0;
209        frame.getContentPane().add(labelqr, c);
210
211
212          //label.setIcon(new ImageIcon(image));
213        c.fill = GridBagConstraints.CENTER;
214        c.gridx = 0;
215        c.gridy = 1;
216          frame.getContentPane().add(label, c);
217
218
219          JLabel labelsms = new JLabel("SMS content: ");
220          c.weightx = 1;
221          c.weighty = 0.5;
```

```
222         c.fill = GridBagConstraints.HORIZONTAL;
223         c.gridx = 0;
224         c.gridy = 2;
225         frame.getContentPane().add(labelsms, c);
226 
227 
228         jtAreaOutput.setCaretPosition(jtAreaOutput.getDocument()
229                 .getLength());
230         jtAreaOutput.setEditable(false);
231         jtAreaOutput.setLineWrap(true);
232         jtAreaOutput.setText(ret[1]);
233         jtAreaOutput.getCaret().setVisible(true);
234         JScrollPane scrollPane = new JScrollPane(jtAreaOutput,
235                 JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
236                 JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
237 
238 
239         c.weightx = 0.5;
240         c.fill = GridBagConstraints.HORIZONTAL;
241         c.gridx = 0;
242         c.gridy = 3;
243         frame.getContentPane().add(scrollPane, c);
244 
245         JButton button1 = new JButton("Generate Keys");
246 
247         ActionListener actionListener = new ActionListener() {
248               public void actionPerformed(ActionEvent actionEvent) {
249                 update();
250               }
251             };
252          button1.addActionListener(actionListener);
253 
254          c.fill = GridBagConstraints.CENTER;
255         c.gridx = 0;
256         c.gridy = 4;
257           frame.getContentPane().add(button1, c);
258 
259           //frame.pack();
260          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
261           frame.setVisible(true);
262     }
263 
264 
265      public static void main(String[] args) throws Exception {
266         update();
```

```
267             drawui();
268
269
270
271
272
273         }
274
275
276
277     }
```