ALEXANDRE OMOTO DE PAULA
FERNANDO FERREIRA DE SOUZA PINTO VERGUEIRO
MARIO DAZHAO NG

MICROPAG

SISTEMA DE MICROPAGAMENTOS COM ASSINATURA CEGA PARA APARELHOS CELULARES

São Paulo
2011

ALEXANDRE OMOTO DE PAULA
FERNANDO FERREIRA DE SOUZA PINTO VERGUEIRO
MARIO DAZHAO NG

MICROPAG

SISTEMA DE MICROPAGAMENTOS COM ASSINATURA CEGA PARA APARELHOS CELULARES

Trabalho apresentado à disciplina PCS2050 – Projeto de Formatura II do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo.

São Paulo
2011

ALEXANDRE OMOTO DE PAULA
FERNANDO FERREIRA DE SOUZA PINTO VERGUEIRO
MARIO DAZHAO NG

MICROPAG

SISTEMA DE MICROPAGAMENTOS COM ASSINATURA CEGA PARA APARELHOS CELULARES

Trabalho apresentado à disciplina PCS2050 – Projeto de Formatura II do Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo.

Orientador: Prof. Dr. Paulo Sérgio Licciardi Messeder Barreto
Co-orientador: Geovandro Carlos Crepaldi Firmino Pereira

São Paulo
2011

FICHA CATALOGRÁFICA

DEDICATÓRIA

Aos meus pais, João Bosco e Sueca, minha irmã Adriana e meu irmão Mauro Sérgio, que me ofereceram apoio e suporte durante toda minha vida.

Alexandre

Aos meus pais, José e Nadyr, à minha irmã Júlia, ao meu irmão Rafael e à minha namorada Karen por me proporcionarem momentos de discussão do mais alto nível de conhecimento.

Fernando

Aos meus pais e minha irmã Mônica. Para Marcelo (in memoriam), querido irmão e meu maior guia, ensinou-me a continuar em frente mesmo em sua ausência.

Mario

AGRADECIMENTOS

Ao Prof. Dr. Paulo Sérgio Licciardi Messeder Barreto, pela orientação e incentivo durante o projeto.

Ao Geovandro Carlos Crepaldi Firmino Pereira, pelo apoioteórico e técnico prestado.

À nossas famílias, pelo suporte constante durante nossas vidas, que nos permitiu alcançar nossos objetivos.

Aos nossos professores, que nos auxiliaram em nossa busca por conhecimento.

Aos nossos colegas e amigos, sem os quais jamais teríamos chegado onde estamos.

E à Escola Politécnica da Universidade de São Paulo, que muito nos ensinou nesse importante momento de nossas vidas.

"What is the use of living, if it be not
to strive for noble causes and to
make this muddled world a better
place for those who will live in it
after we are gone?"
Sir Winston Churchill

RESUMO

Micropagamentos são operações de baixo valor monetário, com custo financeiro operacional amortizado. É proposto um sistema de micropagamentos, envolvendo telefones celulares, no qual é utilizada moeda eletrônica para realizar transações de pequeno valor. É garantida a segurança da transação utilizando assinatura digital, e é utilizada assinatura cega para garantir a não rastreabilidade das moedas de um usuário. O baixo custo computacional oferecido pela criptografia baseada em Curvas Elípticas permite que o sistema seja livremente utilizado em aparelhos celulares, e a homogeneidade dos algoritmos garante a compatibilidade e facilidade de alteração dos mesmos, além de oferecer maior velocidade e economia de recursos.

Palavras-Chave: Micropagamento. Moeda Eletrônica. Assinatura Digital. Assinatura Cega. Curvas Elípticas.

ABSTRACT

Micropayments are a low value monetary operation, with amortized financial operational costs. We propose a micropayment system, involving cell phones, in which is used electronic coins to make small value transactions. We guarantee the security of the transaction by using digital signature, and it ensures the untraceability of the coins of the user. The low computational cost offered by Elliptic Curves based cryptography allows the system to be freely used in cell phones, and the homogeneity of the algorithms ensures the compatibility and easiness to alter them, while also offering greater speed end economy of resources.

Keywords: Micropayment. Electronic Coin. Digital Signature. Blind Signature. Elliptic Curves.

SUMÁRIO

1 INTRODUÇÃO 17
1.1 MOTIVAÇÃO 18
1.1.1 MERCADO *19*
1.1.2 TARIFAS *23*
1.1.3 NFC *24*
1.1.4 PRIVACIDADE *26*
1.2 OBJETIVOS 26
1.3 JUSTIFICATIVA 27
1.4 ESTRUTURA DESTE DOCUMENTO 28
2 LIGHTWEIGHT MICRO-CASH FOR THE INTERNET 29
2.1 ASSINATURA DIGITAL 29
2.1.1 LEGISLAÇÃO BRASILEIRA *31*
2.2 FUNÇÃO DE HASH 31
2.2.1 KECCAK *33*
2.3 ASSINATURA SCHNORR 34
2.4 PROTOCOLOS 35
2.4.1 PROTOCOLO DE RETIRADA *35*
2.4.2 PROTOCOLO DE PAGAMENTO *37*
2.4.3 PROTOCOLO DE DEPÓSITO *38*
2.4.4 DETECÇÃO DE FRAUDES *38*
3 MODIFICAÇÕES NO ARTIGO 43
3.1 CRIPTOGRAFIA DE CURVAS ELÍPTICAS 43
3.1.1 CURVAS ELÍPTICAS *44*
3.1.2 O PROBLEMA DO LOGARITMO DISCRETO *46*
3.2 ALGORITMOS MODIFICADOS 47
3.2.1 ASSINATURA SCHNORR *47*
3.2.2 ASSINATURA SCHNORR CEGA *48*
3.2.3 PROTOCOLO DE RETIRADA *48*
3.2.4 PROTOCOLO DE PAGAMENTO *49*
3.2.5 PROTOCOLO DE DEPÓSITO *49*
4 IMPLEMENTAÇÃO 50
4.1 METODOLOGIA 50

4.1.1 ESPECIFICAÇÃO DE REQUISITOS 50
4.1.2 MODELO DE CASOS DE USO 51
4.1.3 INTERFACE HOMEM MÁQUINA 51
4.1.4 MODELO DE CLASSES 51
4.1.5 MODELO DINÂMICO 51
4.1.6 DESCRIÇÃO DAS INTERFACES DE COMUNICAÇÃO 51
4.1.7 ESPECIFICAÇÃO DA ARQUITETURA 52
4.1.8 PLANO DE TESTES E ACEITAÇÃO 52
4.2 AMBIENTE DE DESENVOLVIMENTO 52
4.2.1 JAVA 53
4.2.2 ANDROID 54
4.2.3 BLUETOOTH 57
4.2.4 BNPAIRINGS 59
4.2.5 OUTRAS FERRAMENTAS 60
5 RESULTADOS E ANÁLISE 61
5.1 RESULTADOS 61
5.2 ANÁLISE DE VIABILIDADE 61
5.3 ANÁLISE DE DESEMPENHO 63
5.3.1 TEMPO DE COMUNICAÇÃO 63
5.3.2 TAMANHO DA MOEDA 65
5.4 ANÁLISE DE ALGUNS ATAQUES 66
5.5 CONTRIBUIÇÕES 67
5.6 FUTURAS VERSÕES 68
6 CONCLUSÃO 69
7 REFERÊNCIAS 70
APÊNDICE A – ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE 75
APÊNDICE B – MODELO DE CASOS DE USO 86
APÊNDICE C – INTERFACE HOMEM COMPUTADOR 91
APÊNDICE D – MODELO DE CLASSES 103
APÊNDICE E – MODELO DINÂMICO 106
APÊNDICE F – DESCRIÇÃO DAS INTERFACES DE COMUNICAÇÃO 112
APÊNDICE G – ESPECIFICAÇÂO DA ARQUITETURA 113
APÊNDICE H – PLANO DE TESTES E ACEITAÇÃO 120

LISTA DE ABREVIATURAS E SIGLAS

ABECS	Associação Brasileira de Empresas de Cartões de Crédito e Serviços
EC	*Elliptic Curve*
ECC	*Elliptic Curve Cryptography*
GSM	*Global System for Mobile Communications*
GPRS	*General Packet Radio Service*
MicroPag	Aplicativo de Micropagamentos para Android (Cliente)
PSG	*Public Signature Generator*
RSA	*Rivest Shamir Adleman*
Sistema Caixa	Aplicativo de Micropagamentos para o Comerciante
Sistema Banco	Aplicativo de Micropagamentos para o Banco
SSG	*Secret Signature Generator*

LISTA DE FIGURAS

Figura 1: Evolução da Posse de Meios Eletrônicos de Pagamento. 19
Figura 2: Evolução por meio de Pagamento Eletrônico.. 19
Figura 3: Evolução no Hábito de Uso de Pagamentos Eletrônicos. 20
Figura 4: Preferência do Consumidor por Forma de Pagamento. 20
Figura 5: Volume Pago por Forma de Pagamento. 21
Figura 6: Gastos por Faixa no Pagamento Eletrônico. 21
Figura 7: Preferência dos Comerciantes por Forma de Pagamento. 22
Figura 8: Pontos Fortes e Fracos dos Cartões segundo Comerciantes. 22
Figura 9: Tarifas no Cartão de Crédito 23
Figura 10: Tecnologia NFC. 25
Figura 11: Funcionamento do Google Wallet. 25
Figura 12: Esquema da Assinatura Digital 30
Figura 13: Regras Fundamentais dos Algoritmos de Hash. 32
Figura 14: Construção Esponja. 34
Figura 15: Fluxograma do Algoritmo de Detecção de Fraudes 41
Figura 16: Ciclo de Desenvolvimento Java.. 53
Figura 17: Java Virtual Machine para Diversas Plataformas. 54
Figura 18: Mercado de Smartphones. 55
Figura 19: Arquitetura do Sistema Android. 57
Figura 20: Android Bluetooth Stack. 59
Figura 21: Tela de Recarga 92
Figura 22: Tela de Pagamento 93
Figura 23: Tela de Saldo Insuficiente 93
Figura 24: Tela de Erro 94
Figura 25: Tela Inicial 94
Figura 26: Tela de Recarga 95
Figura 27: Tela Sucesso Recarga 95
Figura 28: Tela Recarga Rejeitada 95
Figura 29: Tela Falha Recarga 96
Figura 30: Tela Falha de Conexão 96
Figura 31: Tela Inicial 97
Figura 32: Tela de Cobrança 97

Figura 33:Tela de Falha na Cobrança 98
Figura 34: Tela de Depósito 98
Figura 35: Tela Falha no Depósito 98
Figura 36: Navegação do Aplicativo Android 99
Figura 37: Navegação do Sistema Banco 100
Figura 38: Navegação do Sistema Caixa 100

LISTA DE FÓRMULAS

Fórmula 1: Mensagem Original 36
Fórmula 2: Mensagem Assinada Cegamente 36
Fórmula 3: Moeda Eletrônica Original 36
Fórmula 4: Verificação da Validade do Pagamento 37
Fórmula 5: Chave Privada Descoberta com Moeda Duplicada 39
Fórmula 6: Chave Privada não é Descoberta sem Moeda Duplicada 39
Fórmula 7: Cálculo da Chave Privada Falho (Chaves Falsas) 40
Fórmula 8: Cálculo da Chave Privada Falho (PSGs Falsos) 40
Fórmula 9: Equação de Weierstrass de Uma Curva Elíptica 44
Fórmula 10: Equação de Weirstrass para Curva Elíptica em Zp. 44
Fórmula 11: Soma de Pontos em Curvas Elípticas em Zp. 45
Fórmula 12: Equação de Weirstrass para Curvas Elípticas sobre Corpo Finito de Característica 2. 45
Fórmula 13: Soma de Dois Pontos sobre Corpo Finito de Carac. 2(I). 45
Fórmula 14: Soma de Dois Pontos sobre Corpo Finito de Carac. 2(II). 45
Fórmula 15: Multiplicação em Curvas Elípticas. 46
Fórmula 16: Teorema de Hasse. 46

LISTA DE TABELAS

Tabela 1: Velocidade de Transferência por Versão do Bluetooth 58
Tabela 2: Potência e Alcance por Versão do Bluetooth 58
Tabela 3: Tempo Médio de Operação - Recarga 64
Tabela 4: Tempo Médio de Operação - Pagamento 64
Tabela 5: Funcionalidades por Usuário 91
Tabela 6: Mensagens de Erro 101

1 INTRODUÇÃO

O uso de meios eletrônicos de pagamento é um hábito comum em nosso cotidiano. Pesquisa da Associação Brasileira de Empresas de Cartões de Crédito e Serviços (ABECS) em parceria com a Datafolha, setembro/2011, indica que 72,4% da população já possui alguma forma de pagamento eletrônico (cartão crédito, débito ou de loja). De toda a população, 60% possuem cartão de débito e 53% cartão de crédito.

A mesma pesquisa confirma o aumento do uso de meios eletrônicos de 68,4% para 72,4% entre 2008 e 2011, com tendência de alta. O hábito de se utilizar meios eletrônicos também aumentou, mostrando a consolidação deste mercado.

Porém, os grandes problemas apontados estão relacionados com as tarifas cobradas pelas administradoras de cartão de crédito, o que leva a 76% dos comerciantes preferirem pagamentos não eletrônicos (ABECS, 2011).

Os meios eletrônicos são geralmente usados para pagamentos acima dos R$50,00. Apenas 7% dos que possuem cartão de débito e 2% dos que possuem cartão de crédito fazem pagamentos de até R$ 10,00 (ABECS, 2011). Para este nicho, é proposto um sistema de micropagamentos com complexidade computacional baixa, que tenha segurança robusta e garanta a anonimidade dos gastos. Esse sistema usará dois equipamentos amplamente disponíveis no mercado: smartphones com sistema operacional Android e um PC com suporte a Bluetooth.

O sistema é uma prova de conceito baseado no artigo Lightweight Micro-Cash for the Internet (MAO,1996). Algumas adaptações foram feitas para este projeto, como o uso de curvas elípticas e a substituição de Assinatura Cega RSA usando fator de Chaum (CHAUM, 1982) para Assinatura Cega Schnorr (POINTCHEVAL, 2000). O uso de curvas elípticas permite o uso de chaves com menos bits em relação à assinatura RSA (GUPTA, 2002), oferecendo o mesmo nível de segurança. Além disso, como todo o protocolo utiliza Assinatura de Schnorr, a implementação fica uniforme, diminuindo a chance de erros resultantes da integração entre os algoritmos de Assinatura Cega RSA e Assinatura de Schnorr.

A anonimidade é um requisito fundamental do sistema, visando garantir a privacidade do usuário do sistema. Porém há mecanismos que permitem a identificação de fraudadores, casos elas ocorram.

1.1 MOTIVAÇÃO

O uso de meios eletrônicos de pagamentos é comum em nossa sociedade, seja utilizando cartões de créditos, débitos ou de lojas. Este "dinheiro de plástico" traz enormes facilidades para os usuários, porém pesquisa da ABECS - Associação Brasileira de Empresas de Cartões de Crédito e Serviços - mostra que o comerciante ainda prefere receber em dinheiro, e por um simples motivo: altas tarifas cobradas pelas administradoras de cartões de créditos e também o aluguel dos equipamentos de POS - *Point of Sale*.

Devido a este alto custo, muitos comerciantes não aceitam receber pagamentos em cartão para quantias abaixo de R$10,00. Este projeto visa atender a este nicho de mercado, criando um sistema eletrônico de micropagamentos utilizando dois equipamentos amplamente disponíveis no mercado: smartphones com sistema operacional Android e PCs com suporte a Bluetooth, padrão IEEE 802.15.1. Utilizando criptografia e com complexidade computacional baixa, seus custos de operação resumem-se à aquisição destes equipamentos e a cobrança do banco para retirar / depositar moedas eletrônicas. Não há a necessidade da participação de intermediários, como uma administradora de cartões, uma vez que a validação da moeda é feita pelo próprio sistema utilizando Assinaturas Digitais. O uso de Bluetooth elimina a necessidade de aluguel de equipamentos, já que esta tecnologia se encontra bastante difundida no mercado e um simples adaptador de baixo custo é o equipamento necessário para seu uso.

Duas das maiores bandeiras de cartões do mundo - Visa e Mastercard - já usam *smartphones* como meio de pagamento, substituindo os cartões. A Visa foi a pioneira ao lançar em 2009 (NIKOLAS, 2010), como parte da plataforma payWave, um aplicativo para celulares com tecnologia NFC - *Near Field Communication*. Em maio de 2011, a Mastercard e o Google lançaram um aplicativo denominado Google

Wallet utilizando também NFC. Porém, estas tecnologias modificam o POS, mas não o modelo de negócios e suas tarifas (NIKOLAS, 2010).

1.1.1 MERCADO

Pesquisa da ABECS em associação com a Datafolha de setembro de 2011 mostra que mais de 70% da população brasileira já possui algum meio eletrônico de pagamento. A evolução de posse ao longo dos últimos anos pode ser observada ma figura 1:

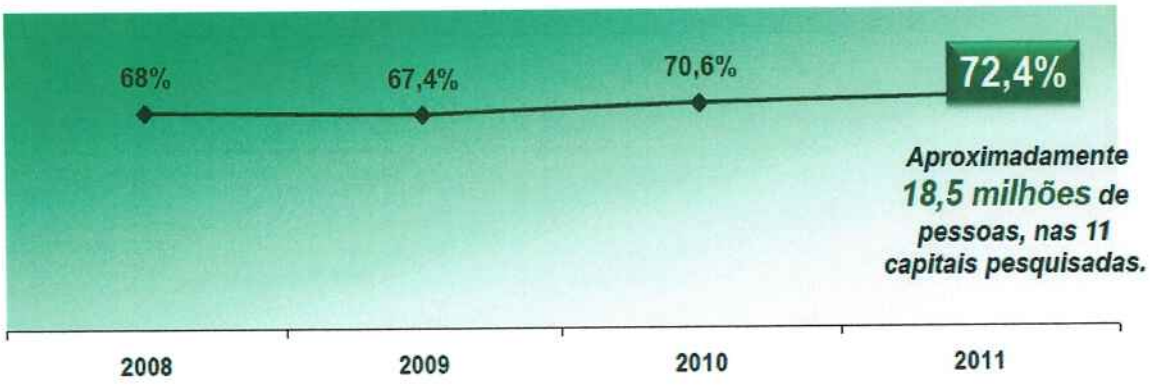


Figura 1: Evolução da Posse de Meios Eletrônicos de Pagamento. Fonte: (ABECS, 2011)

A pesquisa considera como meio eletrônico de pagamento cartões de crédito, débito ou os que possuem bandeiras de loja e oferecem opções de parcelamento. Os cartões de débito são os mais comuns no mercado, como mostra a figura 2:

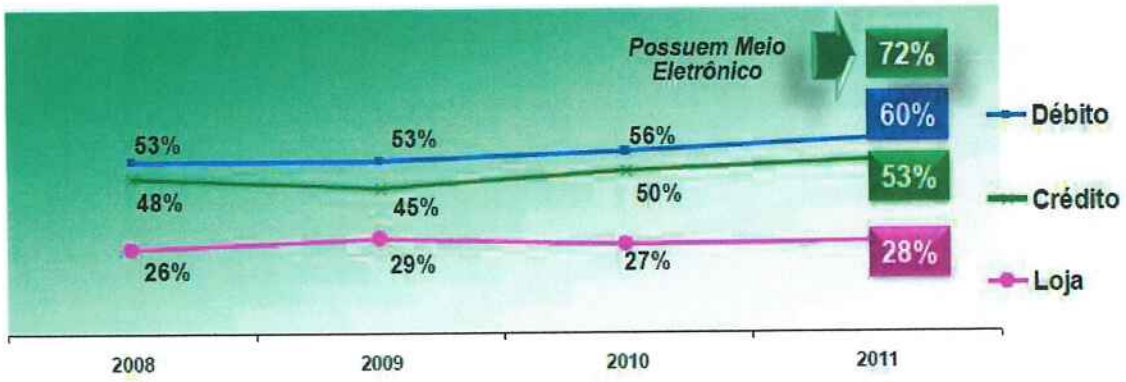


Figura 2: Evolução por meio de Pagamento Eletrônico. Fonte: (ABECS, 2011).

O uso habitual de meios eletrônicos de pagamento é alto: 95,83% dos que possuem um meio eletrônico a utilizam habitualmente, conforme pode ser observado na figura 3.

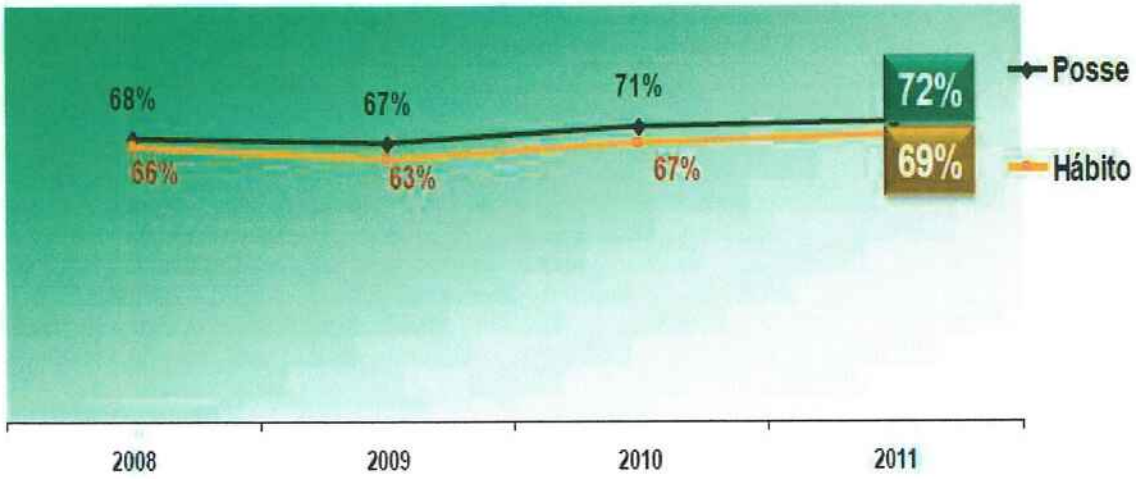


Figura 3: Evolução no Hábito de Uso de Pagamentos Eletrônicos. Fonte: (ABECS, 2011)

Para 40% da população é preferencial o uso de meios de pagamento eletrônicos. Mas o uso de dinheiro continua sendo o preferido dos consumidores, como demonstrado abaixo, como visto na figura 4.

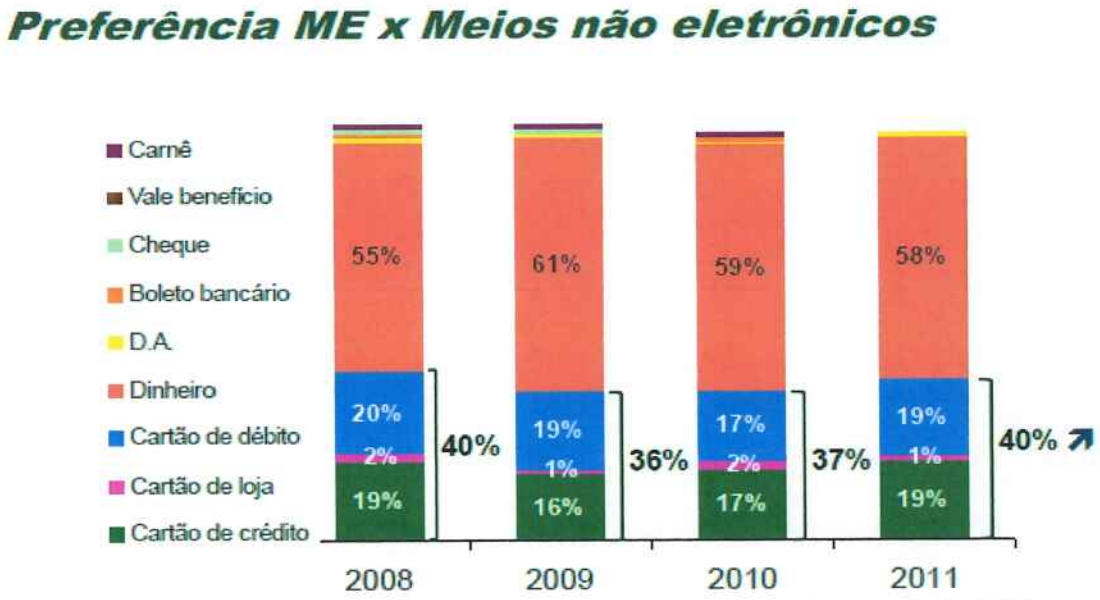


Figura 4: Preferência do Consumidor por Forma de Pagamento. Fonte: (ABECS, 2011)

Mas, como observamos na figura 5, os meios eletrônicos continuam tendo o maior volume pago mensalmente:

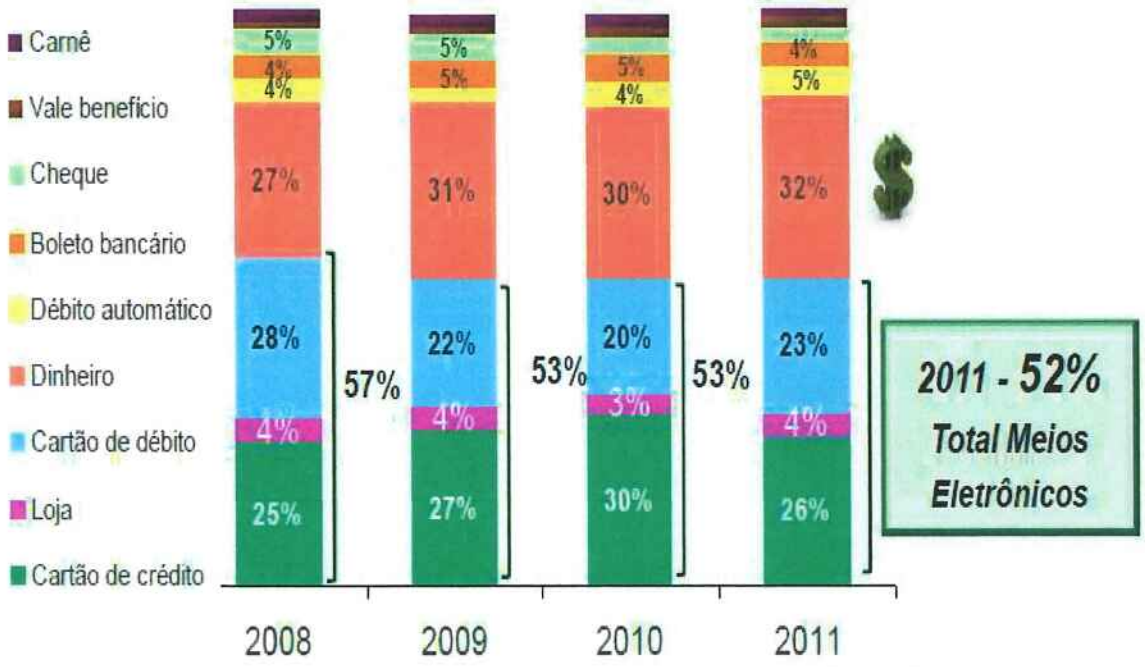


Figura 5: Volume Pago por Forma de Pagamento. Fonte: (ABECS, 2011)

A figura 6, o gráfico de gasto por faixa de valor, demonstra que os gastos estão concentrados nas faixas acima de R$50,00, o que reforça a necessidade de um sistema de micropagamentos.

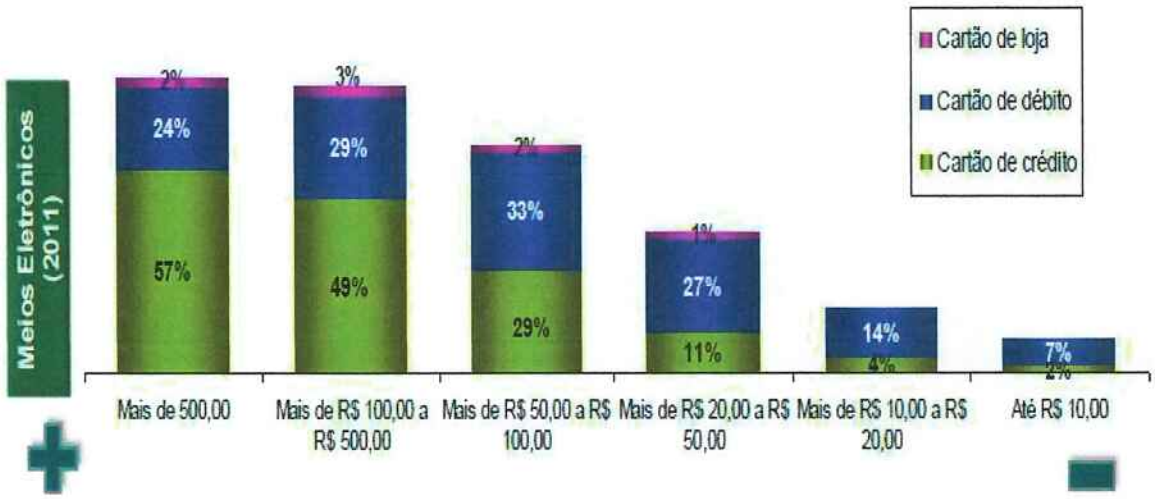


Figura 6: Gastos por Faixa no Pagamento Eletrônico. Fonte: (ABECS, 2011)

A mesma pesquisa avaliou as preferências dos comerciantes em relação ao meio de pagamento, conforme observado na figura 7. Quase três quartos preferem

receber em dinheiro, e a justificativa é simples: tarifas das operadoras de cartão. Há elevadas taxas de adiantamento do valor e um período de espera.

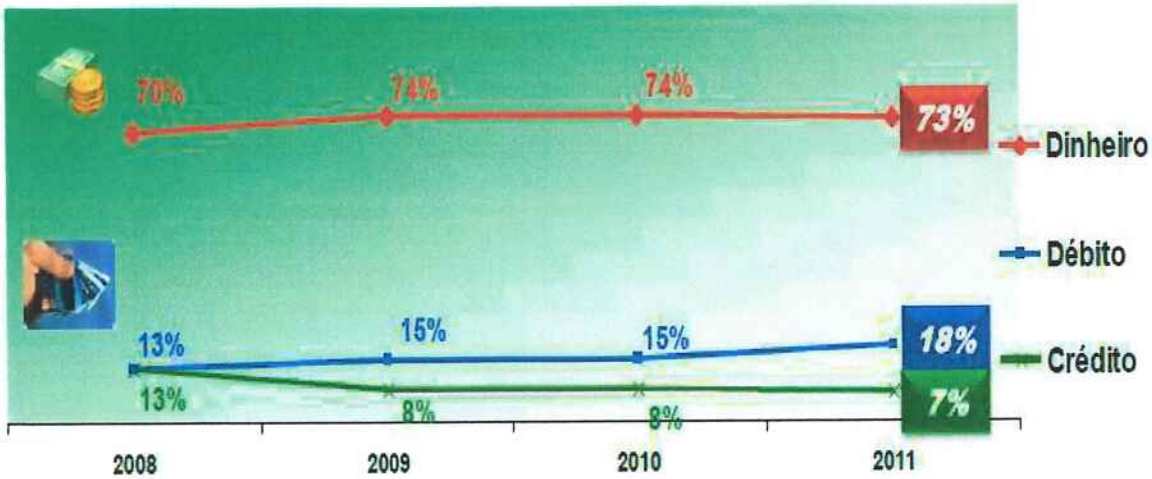


Figura 7: Preferência dos Comerciantes por Forma de Pagamento. Fonte: (ABECS, 2011)

A figura 8 mostra os Prós e Contras que os comerciantes identificam nestes dois meios de pagamento. Nota-se que os problemas estão claramente relacionados com as taxas elevadas de se utilizar meios eletrônicos de pagamento:

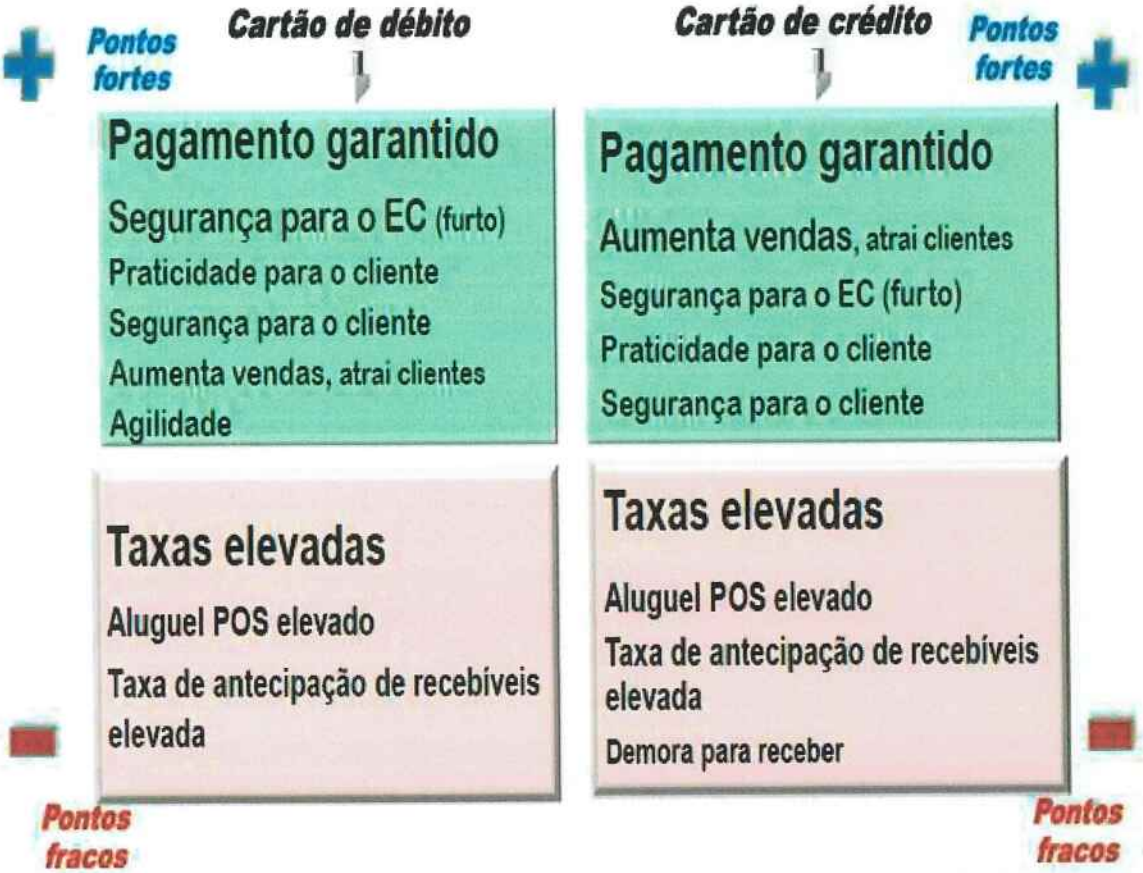


Figura 8: Pontos Fortes e Fracos dos Cartões segundo Comerciantes. Fonte: (ABECS, 2011)

A pesquisa demonstra que, apesar da posse maior de meios eletrônicos de pagamento pela população brasileira, os comerciantes preferem receber em dinheiro e os gastos se concentram em pagamentos acima dos R$50,00. Isto é explicado pelas taxas elevadas pelo uso do serviço.

1.1.2 TARIFAS

O principal problema dos meios eletrônicos de pagamentos, apontados pelos comerciantes, é a alta tarifa cobrada por bancos e administradoras de cartão de crédito. A figura 9, publicada na revista Veja em agosto de 2010, mostra que para cada R$100,00 recebidos pelos comerciantes, são cobrados R$1,50 para operações de débito e R$3,00 para operações de crédito. Estas tarifas incluem uma porcentagem do valor da venda e também uma tarifa fixa pelo uso do serviço a cada transação, mesmo que esta seja cancelada (CDL/BH, 2009).

Figura 9: Tarifas no Cartão de Crédito. Fonte: BETTI, Renata. A batalha feroz das maquininhas. Veja, São Paulo, ano 43, nº 33, ed. 2178, 18/08/2010, pág. 82;

O aluguel do POS também é um custo alto. Em 2010, a Cielo cobrava entre R$39,00 e R$135,00 pelo aluguel da máquina e a Redecard entre R$60,00 a R$120,00 (GAZETA DO POVO, 2010).

A combinação de tarifas fixas e variáveis por transação e uso do sistema leva muitos comerciantes a impor um valor mínimo de compra, não aceitando transações

no cartão para pagamentos abaixo dos R$5,00 ou R$10,00 (PEREIRA, 2011). A prática é considerada abusiva tanto pelo Ministério da Justiça quanto pelo Código de Defesa do Consumidor, e ainda há resolução específica do Conselho Nacional de Defesa do Consumidor que considera ilegal acréscimos de preço nas compras com cartão (GUIMARÃES, 2011). Transações de Débito ou Crédito em uma parcela são caracterizadas como compras à vista, portanto não podem sofrer taxação adicional (PEREIRA, 2011) (GUIMARÃES, 2011).Segundo o Procon, o comerciante que cobra valor mínimo está sujeito a multa de R$ 422,00 a R$ 6 milhões (PEREIRA, 2011), cometendo também infrações como a negação de venda a pronto pagamento e venda casada de produtos (GUIMARÃES, 2011).

1.1.3 NFC

Visa e Mastercard apresentaram sistemas de pagamentos eletrônicos em smartphones utilizando a tecnologia NFC - *Near Field Communication* - que permite (ORTIZ, 2006): transações simplificadas, troca de dados e conexão wireless entre dois dispositivos a uma distância de poucos centímetros.

A Visa (figura 10) atua neste segmento com seu programa payWave, que engloba cartões e aplicativos para smartphones. Os cartões permitem uma forma mais rápida de efetuar pagamentos, bastando aproximar o cartão do leitor. Não há introdução de senhas, e seu foco são pagamentos pequenos, limitados a US$ 25,00 nos Estados Unidos, £15,00 na Inglaterra ou 15,00 euros na Europa (NIKOLAS, 2010). Em 2009, a Visa lançou seu aplicativo payWave para celulares com a tecnologia NFC (SARAH, 2009), que se comunicam com os mesmos terminais que aceitam os cartões payWave e associam o aparelho a um cartão de crédito/débito.Este aplicativo pode utilizar um chip especial colocado no slot de memória para aprimorar a segurança (BUTCHER, 2011) (THE OFFICIAL GOOGLE BLOG, 2011).

Figura 10: Tecnologia NFC. Fontes:(BUTCHER, 2011) (E) e http://usa.visa.com/personal/cards/paywave/index.html (D)

O Google Wallet (figura 11) foi anunciado em maio de 2011, em parceria com a Mastercard (THE OFFICIAL GOOGLE BLOG, 2011). É um aplicativo para Android que permite a importação de cartões físicos para uma carteira virtual e fazer pagamentos utilizando NFC. Utiliza leitores da payPass, uma iniciativa da Mastercard semelhante ao payWave da Visa.

Figura 11: Funcionamento do Google Wallet. Fonte: http://www.google.com/wallet/how-it-works.html

Ambas as iniciativas buscam efetuar pagamentos utilizando smartphones, mas não atendem ao segmento de micropagamentos. As tarifas cobradas neste sistema muitas vezes superam as taxas utilizando cartões normais (NIKOLAS, 2010).

1.1.4 PRIVACIDADE

O sigilo bancário é um direito garantindo pela Lei Complementar 105/2001, com a quebra eventual somente com autorização judicial. A quebra do sigilo é considerado crime com punição de até quatro anos de prisão.

O sigilo dos gastos com moeda eletrônica é uma característica intrínseca do sistema. Ao se retirar uma moeda eletrônica, esta é assinada cegamente pelo banco e isto lhe confere validade. O comerciante que receber a moeda consegue determinar a validade da moeda inequivocamente, recebendo a chave pública do banco. No depósito, nenhuma informação sobre quem gerou e utilizou a moeda é repassada ao banco. O comerciante só consegue ter registro das compras feitas em seu estabelecimento, o que torna inviável tentar rastrear todas as compras de um consumidor.

O sistema é seguro enquanto for bem utilizado; ao se gastar a mesma moeda duas vezes é possível extrair a chave privada e pública de quem gerou a moeda, revelando a identidade do autor da fraude.

1.2 OBJETIVOS

O objetivo deste projeto é desenvolver uma prova de conceito do sistema de micropagamentos do artigo Lightweight Micro-Cash for the Internet, de Wenbo Mao, com pequenas alterações nos protocolos. As principais alterações são: a modificação do protocolo de retirada de moedas, trocando a assinatura cega RSA com fator de Chaum (CHAUM, 1982) para Assinatura de Schnorr Cega (POINTCHEVAL, 2000); e o uso de Curvas Elípticas para implementar os algoritmos de criptografia, ao invés do algoritmo original baseado em operações com dois grandes números primos.

O sistema deve ser seguro e de baixa complexidade computacional, uma vez que o aplicativo cliente deverá funcionar em smartphones com sistema operacional

Android. Este objetivo será alcançando através do uso de curvas elípticas, que garantem a mesma segurança do RSA, mas com chaves menores, o que implica em um processamento mais rápido do algoritmo de criptografia, menor consumo de energia, de memória e de banda de comunicação (GUPTA, 2002).

Para realização da prova de conceito, será desenvolvido também um sistema Caixa, para receber pagamentos, e um sistema Banco para simular a retirada/saque de moedas eletrônicas.

1.3 JUSTIFICATIVA

Apesar da grande penetração de meios eletrônicos de pagamentos, é comum a prática no mercado de não aceitar pagamentos com cartão, crédito ou débito, para valores abaixo de R$5,00 ou R$10,00. Porém, isto é uma prática ilegal e pode render multa de até R$ 6 milhões (PEREIRA, 2011).

A justificativa é o custo da transação, que supera o valor do produto vendido. E não há perspectiva de mudanças neste cenário, uma vez que as tarifas cobradas pelo uso de NFC são mais altas que as normais.

Nosso enfoque é em micropagamentos, criando um protótipo de sistema de pagamento que torne viável o pagamento de pequenas quantias eletronicamente. A viabilidade ocorre do fato dos protocolos de moeda eletrônica não precisarem da participação de intermediários como as administradoras de cartão; o próprio algoritmo consegue detectar a validade de moedas. Basta o envolvimento do banco na retirada/depósito das moedas e o uso de equipamentos comuns no mercado como smartphones com Android e PCs com Bluetooth.

1.4 ESTRUTURA DESTE DOCUMENTO

Na seção 2 apresentamos um breve resumo do artigo original e os algoritmos criptográficos utilizados.

Na seção 3 expomos as modificações que propomos no artigo, como a modificação dos algoritmos para uso de Curvas Elípticas e a exclusão do algoritmo de Assinatura Cega RSA com fator de Chaum (CHAUM, 1982), substituído por Assinatura Cega Schnorr (POINTCHEVAL, 2000).

Na seção 4 descrevemos a metodologia de desenvolvimento e os documentos gerados, além de descrever as plataformas utilizadas.

Na seção 5 são discutidos os resultados do projeto, fazendo análises de viabilidade, desempenho, possíveis ataques e versões futuras.

Na seção 6 apresentamos a conclusão do projeto e as considerações finais.

2 LIGHTWEIGHT MICRO-CASH FOR THE INTERNET

O artigo Lightweight Micro-Cash for the Internet (MAO, 1996) é o artigo base para o sistema de micropagamentos. O autor propõe uma técnica de micropagamento baseado em assinatura única de uma mensagem; se houver duplicação, a chave privada é descoberta. Isto permite a identificação e a revogação de gastos de fraudadores que utilizam a mesma moeda eletrônica diversas vezes. Porém, enquanto o usuário não cometer fraudes, sua anonimidade e não rastreabilidade são assegurados pelo método proposto, que também possui métodos simples para os protocolos de retirada, pagamento e depósito de moedas eletrônicas.

As moedas eletrônicas não podem ser repassadas; uma vez gasta, é necessário que o comerciante vá até o banco depositar a moeda e retirar o valor correspondente.

O artigo utiliza Assinatura Digital de Schnorr (SCHNORR, 1991) para manter a segurança do sistema, um tipo de algoritmo assimétrico.

2.1 ASSINATURA DIGITAL

Em criptografia, a Assinatura Digital é um mecanismo de autenticação de informações digitais. Com base nela é possível que o autor da mensagem adicione a ela um código que funciona como sua assinatura (STALLINGS, 2011).

Assinaturas digitais são válidas legalmente e garantem (BARRETO, 2011):

1. Integridade: qualquer alteração na mensagem faz com que a assinatura não corresponda ao documento;
2. Autenticidade: a identidade alegada por um usuário é verificável e intransferível;
3. Irretratabilidade: nem o remetente nem o destinatário das informações pode negar a sua transmissão, recepção ou posse.

As assinaturas digitais são aplicadas sobre resumos criptográficos de mensagens (hash). Este resumo garante a integridade, uma vez que a alteração de um bit na mensagem deve alterar o valor do hash. O hash é depois criptografado por meio de um algoritmo assimétrico, também conhecido como algoritmo de chave pública, garantindo a autenticidade e a irretratabilidade. O autor da mensagem utiliza sua chave privada para assinar o documento (BARRETO, 2011).

Para verificar a validade da assinatura, calcula-se novamente o hash da mensagem recebida e decifra-se o hash criptografado usando a chave pública de quem assinou. Se ambos forem iguais, a mensagem é válida (STALLINGS, 2011).

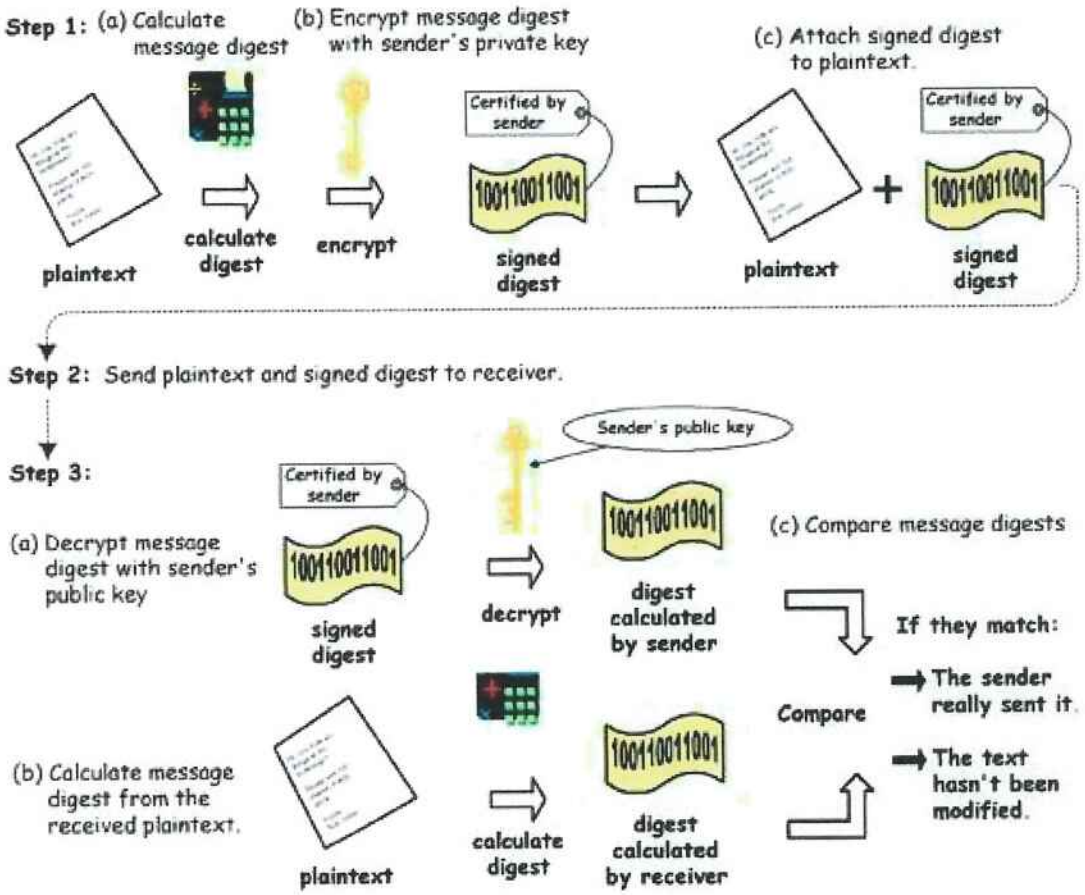


Figura 12: Esquema da Assinatura Digital. Fonte: http://www.uic.edu/depts/accc/newsletter/adn26/figure3.html

2.1.1 LEGISLAÇÃO BRASILEIRA

A medida provisória 2.200-2, de 24 de agosto de 2001, determina que o documento digital tem validade se for certificado pela ICP-Brasil. A ICP (Infraestrutura de Chaves Públicas) é uma entidade que tem como papel emitir chaves públicas, atuando como um intermediário que assegura a validade de chaves por meio de certificados digitais (INSTITUTO NACIONAL DA TECNOLOGIA DA INFORMAÇÃO, 2011a).

A Autoridade Certificadora Raiz da ICP-Brasil é a primeira autoridade na cadeia de certificação. É representada pelo Instituto Nacional de Tecnologia da Informação (ITI), e cabe a ela expedir, distribuir, revogar e gerenciar os certificados das autoridades certificadoras de nível imediatamente subsequente ao seu. No nível subsequente, temos Autoridades Certificadoras como a Serpro, a Receita Federal do Brasil e a Caixa (INSTITUTO NACIONAL DA TECNOLOGIA DA INFORMAÇÃO, 2011c).

2.2 FUNÇÃO DE HASH

Funções de hash, também conhecidas como resumos criptográficos, criam redundâncias anexadas a mensagens com o propósito de detectar alterações. Elas dependem exclusivamente da mensagem, sem a necessidade de nenhuma chave (BARRETO, 2011).

As Assinaturas Digitais são aplicadas em cima dos resumos criptográficos e não no conteúdo integral de documentos eletrônicos por um motivo simples: as funções de hash são muito mais rápidas do que os algoritmos de assinatura (BARRETO, 2011).

Toda função de hash deve respeitar três regras fundamentais (BARRETO, 2011), descritas abaixo e demonstradas na figura 13:

1. Resistência à primeira inversão: Dado um resumo R, é inviável encontrar uma mensagem M tal que R = H(M).

2. Resistência à segunda inversão: Dado um resumo R e uma mensagem M1 tal que R = H(M1), é inviável encontrar outra mensagem M2 ≠ M1 tal que R = H(M2).
3. Resistência à colisões: É inviável encontrar duas mensagens M1 e M2 tais que H(M1) = H(M2).

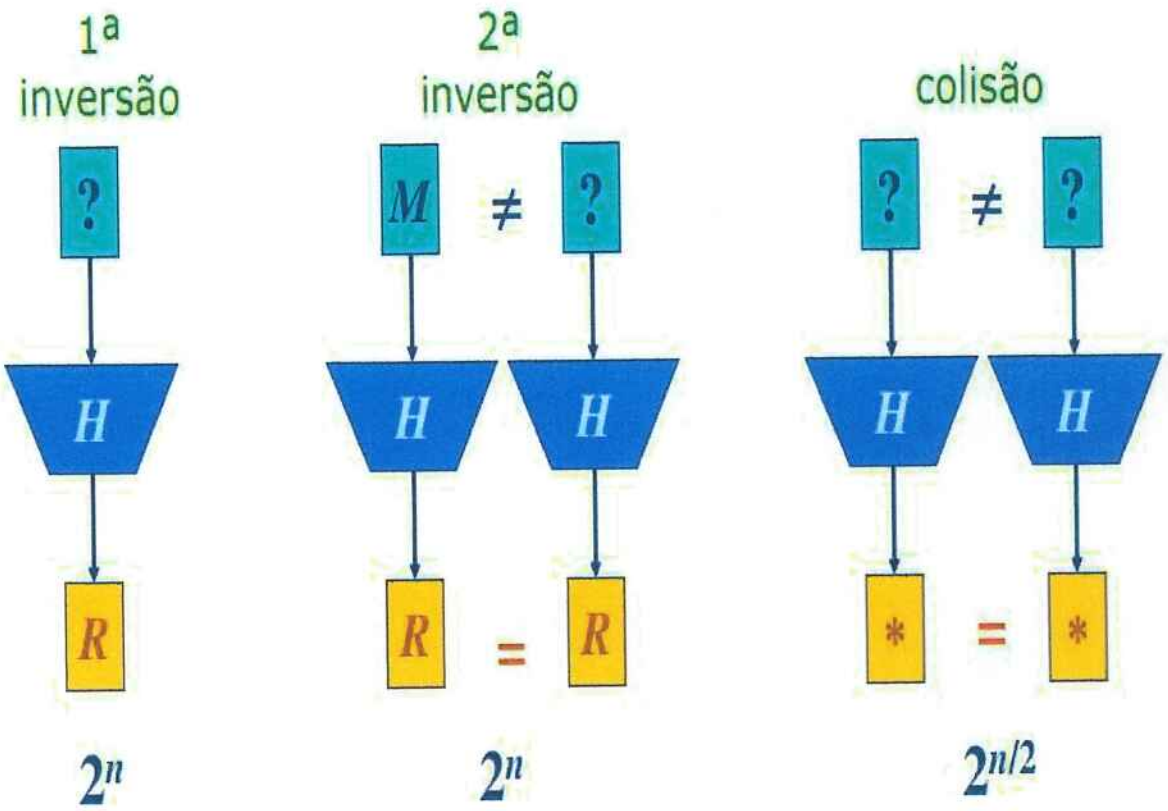


Figura 13: Regras Fundamentais dos Algoritmos de Hash. Fonte: (BARRETO, 2011)

Outras propriedades incluem (BARRETO, 2011):

1. Avalanche Completa: a alteração de um bit de entrada causa a alteração de cerca de metade dos bits de saída.
2. Balanço Perfeito: é inviável prever o valor de qualquer bit de saída.
3. É inviável inverter a função parcialmente (recuperar apenas alguns bits da entrada).
4. As propriedades fundamentais valem para qualquer sub-cadeia da saída.

A função de hash selecionada para a prova de conceito é o Keccak (BERTONI, 2011c).

2.2.1 KECCAK

A função de hash selecionada para o projeto é o Keccak, que é uma família de funções criptográficas de hash (BERTONI, 2011d), e um dos cinco finalistas do concurso da *NationalInstituteof Standard and Technology* (NIST) para determinar o algoritmo da função de hash SHA-3, que substituirá em 2012 os algoritmos SHA-1 e SHA-2 (NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, 2011). É baseado na construção esponja (BERTONI, 2011a), que é uma construção iterativa simples para criar uma função f com tamanho de dados de entrada variável e um tamanho de dados de saída arbitrário baseado numa transformação de tamanho fixo (ou permutação) f operando em um número fixo de b bits (BERTONI, 2011a). O número b é chamado de *width* (largura).

A construção da esponja opera em um estado de b = r + c bits. O valor r é chamado de *bitrate* e o valor c de capacidade. Inicialmente, todos os bits de estado são zerados. A mensagem de entrada sofre um processo de *padding* (enchimento) e é cortado em blocos de r bits. Em seguida, ocorre a fase de **absorbing** e em sequência a fase de **squeezing** (BERTONI, 2011a).

Na fase de **absorbing**, os r-bits dos blocos da mensagem de entrada sofrem XOR com os primeiros r bits do estado, intercaladas com a aplicação da função f. Quando todos os blocos da mensagem são processados, a função esponja muda para o modo **squeezing**.

Na fase de **squeezing**, os primeiros r bits do estado são retornados como blocos de saída, intercalados com aplicações da função f. O número de blocos de saída é escolhido pelo usuário.

Os últimos c bits do estado nunca são diretamente afetados pelos blocos de entrada e nunca são usados como saída durante a fase de **squeezing**.

Tais fases podem ser observadas na figura 14.

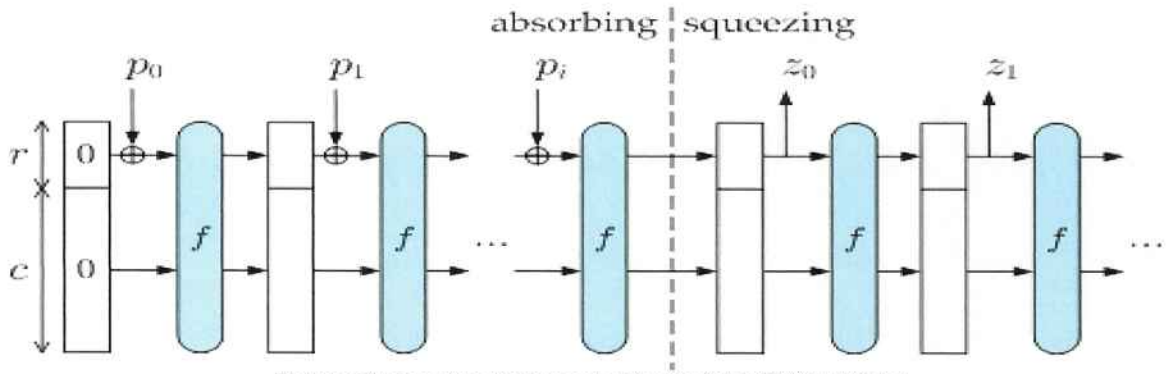


Figura 14: Construção Esponja. Fonte: (BERTONI, 2011a)

Para maiores informações sobre funções de esponja, consulte o documento Cryptographic sponge functions (BERTONI, 2011b). A especificação do Keccak na versão 3.0 pode ser encontrada em The Keccak reference (BERTONI, 2011c).

2.3 ASSINATURA SCHNORR

A assinatura de Schnorr (SCHNORR, 1991) tem a sua segurança baseada na intratabilidade dos problemas de logaritmos discretos. Seu algoritmo original está abaixo descrito:

1. São escolhidos dois número primos grandes p e q, com $q \geq 2^k$;
2. k é o parâmetro do nível de segurança;
3. q | p - 1 (q deve dividir p - 1);
4. É escolhido um elemento g de Z_p^* de ordem q;
5. Com base nestes parâmetros são criados os pares de chaves: $x \in Z_q^*$, que é a chave privada, e $y = g^{-x} \bmod p$, a chave pública;
6. Para assinar uma mensagem, é calculado um fator $r = g^K \bmod p$, $K \in Z_q^*$ é um número gerado aleatoriamente;
7. A seguir calcula-se e = H(m || r) mod q, onde H() é uma função de Hash;
8. Calcula-se s = (K + ex) mod q;
9. A assinatura é o par {e, s};
10. Na verificação, calcula-se e' = H(m || $g^s y^e \bmod p$). A assinatura é válida se e' = e;

A mensagem m, sua assinatura {e, s} e a chave pública y de quem assinou são os parâmetros necessários para validar a assinatura digital. Os parâmetros p, q e g são compartilhados por todos os usuários do sistema.

Uma etapa inicial do processo de validação é a verificação se a chave pública possui um certificado de legitimidade emitido por uma Autoridade Certificadora válida. Por simplificação, esta prova de conceito não realiza as funções de verificação junto a uma Autoridade Certificadora.

2.4 PROTOCOLOS

O artigo de Wenbo Mao define três protocolos para o funcionamento da moeda eletrônica: Retirada, Pagamento e Depósito. Nas seções 2.4.1 a 2.4.3são descritos estes protocolos originais.

Este projeto propõe algumas alterações nos protocolos de Wenbo Mao, como o uso de curvas elípticas e a substituição da Assinatura Cega RSA (CHAUM, 1982) por Assinatura Cega Schnorr (POINTCHEVAL, 2000). Os protocolos modificados foram utilizados na implementação do sistema, e são descritos na seção 3 deste documento.

2.4.1 PROTOCOLO DE RETIRADA

Seja Alice uma cliente que deseja retirar uma moeda eletrônica do Banco. O Banco possui uma chave pública RSA (RIVEST, 1978) representada pelo par (K_B, N_B) e ambos compartilham uma função de hash f(.) segura, enquanto Alice possui sua chave pública v no esquema de Schnorr (SCHNORR, 1991).

Os passos do protocolo descritos por Wenbo Mao são:

1. Alice inicia o processo, pedindo a retirada de uma moeda;
2. O Banco verifica se há saldo disponível. Se sim, continua o processo;

3. Alice calcula um valor x = (a^r mod p), denominado de *Public Signature Generator* (PSG), e escolhe um fator de cegamento b no esquema de Chaum;
4. Alice envia ao banco o valor:

$$(b^{K_B} * f(x \,||v))\ mod\ N_B$$

Fórmula 1: Mensagem Original

5. O banco assina cegamente a mensagem, retornando o valor e um certificado digital de sua chave pública:

$$(b^{K_B} * f(x \,||v))^{K_B^{-1}} mod\ N_B, Cert_B$$

Fórmula 2: Mensagem Assinada Cegamente

6. Alice desofusca a mensagem dividindo a assinatura cega do Banco por b; este novo valor é uma moeda eletrônica assinada cegamente.

$$Coin = f(x \,||v))^{K_B^{-1}} mod\ N_B,$$

Fórmula 3: Moeda Eletrônica Originalh.

Note que a assinatura cega do Banco torna a moeda não rastreável, uma vez que no depósito dela o Banco não sabe quem a gerou. Na assinatura pelo Banco, não são realizadas verificações no conteúdo enviado por Alice, uma vez que o valor correto do PSG e sua chave pública y serão validados pelo comerciante na fase de pagamento. O uso de valores incorretos fará Alice possuir uma moeda que não pode ser gasta sem fraudar os protocolos do sistema.

Em caso de conluio com um comerciante, que permita usar moedas assinadas mas criados com PSG ou chave pública inválidos, haverá detecção caso Alice tente gastar a moeda duas vezes. Neste caso, será possível identificar que o comerciante fraudou o protocolo. Caso Alice gaste apenas uma vez, não há maiores consequências para o sistema porque ela efetivamente pagou por esta moeda.

2.4.2 PROTOCOLO DE PAGAMENTO

O protocolo de pagamento é constituído de uma assinatura de pagamento por Alice e por um processo de verificação de validade pelo comerciante.

No processo de assinatura são incluídos a identidade M do comerciante, que é sua chave pública no esquema de Schnorr, além do DateTime que é um *timestamp*, *i.e.* um identificador do horário. A mensagem a ser assinada é constituída de "Coin, M, DateTime". A assinatura é o par (e, y):

$$e = h(Coin \,||\, M \,||\, DateTime \,||\, x)$$

$$y = ((\mathrm{r} + \mathrm{se}) \bmod \mathrm{q})$$

Fórmula 4: Assinatura do Pagamento

Alice envia ao comerciante Coin, M, DateTime, e, y, v, $Cert_A$, $Cert_B$.

O termo x é o PSG utilizado na geração da moeda. Alice é obrigada a utilizar este mesmo valor durante o processo de assinatura do pagamento, uma vez que isto permite a sua identificação caso haja fraude no uso duplicado de uma moeda eletrônica. Isto será discutido mais adiante na Detecção de Fraudes.

O comerciante, ao receber o pagamento, deve realizar os seguintes passos:

1. Verificar a validade da chave pública de Alice, através do certificado digital $Cert_A$ emitido por uma autoridade certificadora reconhecida;
2. Verificar a validade da assinatura cega em Coin;
3. A seguir verifica a assinatura do pagamento, calculando z = ($g^y v^e$mod p) e verificando se H(Coin || M || DateTime || z) é igual a e;
4. Por fim, o comerciante verifica se Alice assinou com o PSG correto, comparando se H(z || v) = H(x || v) em Coin.

Se todos os passos forem bem-sucedidos,o comerciante aceita o pagamento da moeda.

2.4.3 PROTOCOLO DE DEPÓSITO

O protocolo de depósito é muito semelhante ao protocolo de pagamento: o comerciante assina uma requisição de depósito e envia esta assinatura e seu certificado digital para o Banco:

SignM (Coin; M; DateTime; e; y; EM(z));

$Cert_M$, Coin, M, DateTime, e, y, $E_M(z)$, e_M, y_M.

O banco apenas valida a moeda eletrônica e a assinatura do comerciante. Isto porque o protocolo considera que ao assinar o depósito, o comerciante assume a responsabilidade que não houve fraudes no processo de pagamento.

O termo EM(z) é a encriptação do termo z por um algoritmo escolhido pelo comerciante. O termo z é calculado na verificação do pagamento, e caso depositado sem criptografia poderia levar a descoberta de quem utilizou a moeda. O termo z só é decriptado caso seja detectada uma fraude, fazendo com que o Banco chame o vendedor para ele decriptar o dado depositado.

O processo de verificação pelo Banco é simples:

1. O Banco valida o certificado digital do comerciante;
2. A seguir é validada a assinatura cega da moeda eletrônica;
3. A assinatura digital do comerciante é verificada: testa-se se H(Coin || M || DateTime || e || y || $E_M(z)$ || z') = e_M, com z' = ($g^{ym}v^{em}$mod p);
4. O Banco verifica se a moeda já foi utilizada; se foi, inicia o processo de investigação de fraudes; senão, efetua a troca da moeda eletrônica pelo seu valor real.

2.4.4 DETECÇÃO DE FRAUDES

A detecção de fraudes inicia quando o Banco percebe a tentativa de depósito duplicado de uma moeda eletrônica. Os seguintes casos podem ocorrer:

1. Alice age sozinha e gasta a mesma moeda eletrônica mais de uma vez: Haverá uma diferença em M (identificação única do comerciante), ou em

DateTime. Como Alice é obrigada a assinar os pagamentos utilizando o PSG, isto significa que ela assinou duas ou mais mensagens utilizando este valor, do que decorre:

$$s = \left(\frac{y - y'}{e - e'} \; mod \; q\right)$$

Fórmula 5: Chave Privada Descoberta com Moeda Duplicada

Com isto, conseguimos descobrir a chave privada de Alice e consequentemente sua chave pública, o que permite a sua identificação. Para cada pagamento realizado por Alice é verificado se os comerciantes seguiram os protocolos de pagamento corretamente. Se ambos seguiram, Alice agiu sozinha. Senão, há um conluio entre um ou ambos os comerciantes com Alice.

Note que caso Alice não gaste mais de uma vez a moeda, é inviável descobrir sua identidade, já que o termo r - r' não se anula:

$$y - y' \equiv r - r' + \; s(e - e')(mod \; q)$$

Fórmula 6: Chave Privada não é Descoberta sem Moeda Duplicada

2. O comerciante deposita a mesma moeda eletrônica diversas vezes: Como a origem dos depósitos é a mesma, basta ignorar os pedidos de depósito duplicados e advertir o comerciante.

   O comerciante poderia também tentar modificar o DateTime do pagamento ou repassar a moeda para outro comparsa. Neste caso, as assinaturas de depósito seriam diferentes. Mas basta verificar a assinatura do pagamento, que se tornam inválidas porque M ou DateTime foi alterada. Não é possível que um deles assine por Alice.

3. Alice e Comerciante fazem conluio: Neste caso Alice pode assinar o pagamento utilizando um par de chaves inválido ou não utilizar o PSG adequado. A tentativa de descobrir a sua chave privada e pública apontaria para uma chave inválida ou a de um inocente. Neste caso, ao se

detectar uma moeda duplicada, os comerciantes são chamados para decriptar o valor z. Este valor permite identificar a chave pública utilizada na assinatura do pagamento. Note os seguintes casos:

a. O Comerciante permite a Alice utilizar uma chave pública não certificada: determina-se a chave privada e pública. A Autoridade Certificadora informará que ninguém possui este par; identifica-se fraude do comerciante.
b. O Comerciante permite que Alice assine com dois pares de chaves diferentes, com chaves privadas s1 e s2: Não será possível identificar uma chave privada correta, porque temos a relação:

$$s' = \left(\frac{s_1 e_1 - s_2 e_2}{e_1 - e_2} \bmod q\right)$$

Fórmula 7: Cálculo da Chave Privada Falho (Chaves Falsas)

c. O mesmo ocorre se o Comerciante permitir que Alice assine o pagamento usando um PSG diferente do usado para gerar a moeda, gerando r diferentes (válidos ou não). A relação vira:

$$s' = \left(\frac{s(e_1 - e_2) \pm (r - r')}{e_1 - e_2} \bmod q\right)$$

Fórmula 8: Cálculo da Chave Privada Falho (PSGs Falsos)

Nestes casos calcula-se a chave privada v' a partir de s'. Haverá inconsistências: s' pode ser inválido, ambas as assinaturas de pagamento testadas com v' são inválidas ou o teste da validade da moeda falhará. Pede-se aos dois comerciantes decifrarem o termo z, o que permite a identificação das chaves públicas utilizadas em cada pagamento. Refaz-se todo o processo de pagamento com estas chaves; se houver falha há fraude do comerciante.

Assim obtemos o algoritmo de detecção de fraudes, sempre realizado pelo Banco na detecção de moedas duplas:

1. Verificar se assinaturas e moedas são idênticas. Se forem, ignorar depósito duplicado e advertir o comerciante. Senão, ir para passo 2.

2. O Banco tenta decifrar a chave privada do autor dos pagamentos. Com base na chave privada, é calculada a chave pública. Verifica se a o par de chaves é válido, consultada uma Autoridade Certificadora. Se for válida, vai para passo 3. Se for inválida, vai para passo 5.
3. A partir da chave pública, os pagamentos aos comerciantes são reverificados. Se forem válidos, vai para passo 4. Senão para passo 5.
4. O fraudador agiu sozinho e gastou a mesma moeda diversas vezes. Fim.
5. Cada comerciante deve decifrar o valor z encriptado no depósito. Desta forma se obtém a chave pública utilizada em cada pagamento. Os processos de verificação do pagamento são refeitos.
6. Se o processo de verificação de pagamento falhar, o comerciante ajudou na fraude do sistema.

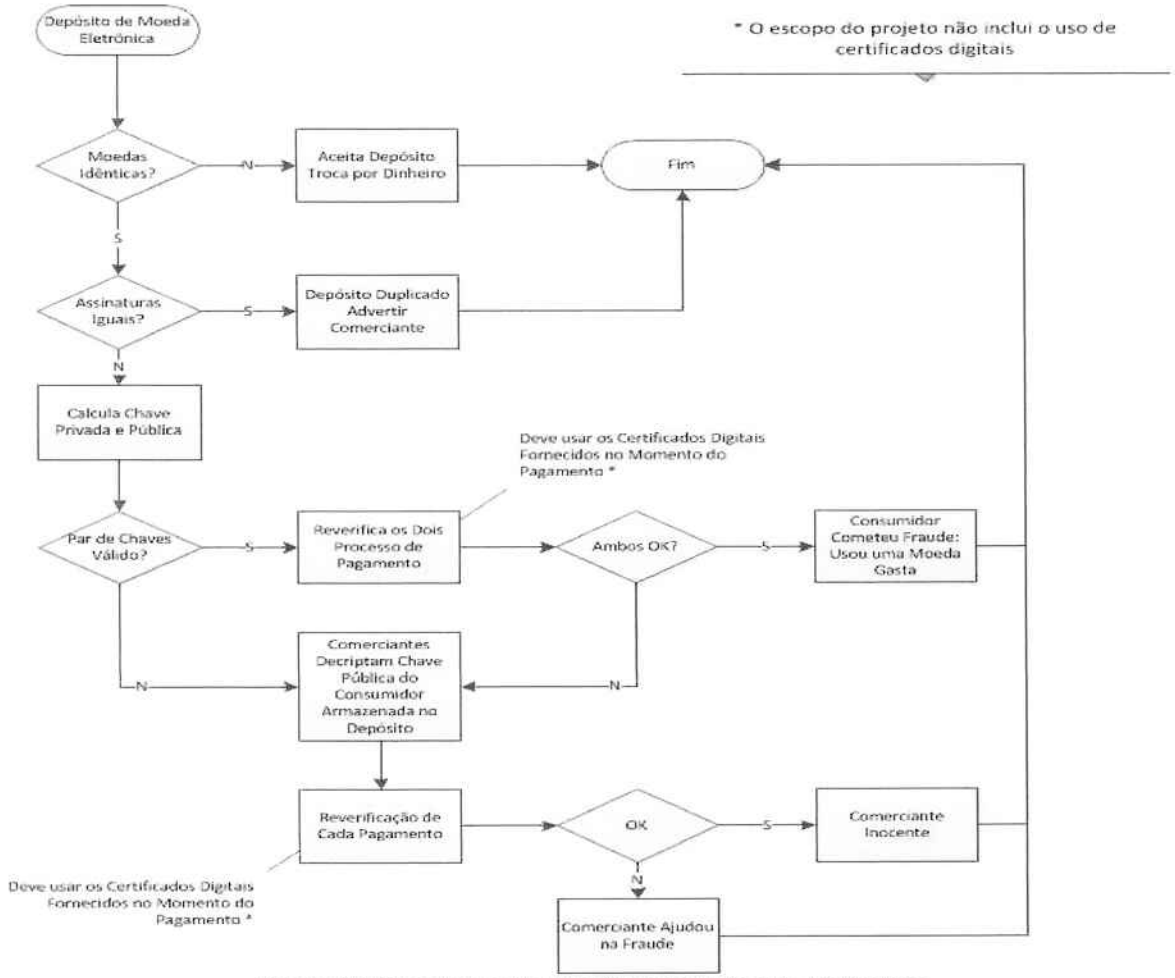


Figura 15: Fluxograma do Algoritmo de Detecção de Fraudes

Um caso interessante poderia ocorrer: Alice poderia escolher uma chave privada qualquer e calcular a chave pública correspondente. Suponha que Bob é o dono legítimo deste par.

O Comerciante permite a Alice gastar a mesma moeda duas vezes usando a chave pública de Bob, sem o certificado digital. Desta forma, o algoritmo de detecção de fraudes encontraria uma chave privada válida, identificaria que é Bob e o acusaria de fraude.

Para explicar que o sistema é seguro contra este tipo de problema, devemos lembrar que a noção de segurança computacional atual é que o sistema não deve ser inquebrável, apenas computacionalmente inviável de ser feito. Uma analogia interessante, feita pelo prof. Barreto, é que quando usamos chaves com tamanho comercial de 128 bits, temos 260 trilhões de mols de possibilidades de chave, e acertar uma chave por acaso equivale a acertar uma molécula escolhida ao acaso em toda a superfície da Terra. Portanto, mesmo que existam milhões de usuários no sistema, a chance de escolher ao acaso uma chave e ela ser correspondente a um usuário é extremamente baixa.

Outro fator inibe esta técnica: é necessário sofisticação para tal ataque, o que não é justificável em micropagamentos porque o custo se torna maior do que o provável lucro. Além disso, a localização geográfica se torna importante: uma investigação pode provar que a pessoa "clonada" nunca esteve no estabelecimento do comerciante, permitindo demonstrar que ele fraudou o sistema.

Com base nestes dados, podemos afirmar que tal fraude é inviável.

3 MODIFICAÇÕES NO SISTEMA ORIGINAL

O sistema foi implementado com algumas modificações em relação ao sistema original:

- Os algoritmos de criptografia foram implementados em curvas elípticas;
- Durante o protocolo de retirada, ao invés de assinatura cega RSA com fator de Chaum foi utilizada Assinatura de Schnorr Cega;
- No protocolo de depósito, ao invés de depositar o fator z encriptado, é depositado a chave pública do comprador encriptado.

3.1 CRIPTOGRAFIA DE CURVAS ELÍPTICAS

A Criptografia de Curvas Elípticas é uma variante da criptografia assimétrica, baseada em curvas elípticas. Foi proposta independentemente por Neal Koblitz (KOBLITZ, 1987) e Victor Miller (MILLER, 1986). Sua segurança se baseia no problema do logaritmo discreto num grupo formado pelos pontos de uma curva elíptica definida em torno de um corpo de Galois (PORTNOI, 2005). O melhor algoritmo conhecido para a resolução deste problema tem grau exponencial (PORTNOI, 2005), o que o torna adequado para aplicações criptográficas.

Curvas elípticas permitem o uso de chaves menores do que as utilizadas pelo RSA (GUPTA, 2002), garantindo o mesmo nível de segurança. Com chaves menores, o processamento necessário para os cálculos dos algoritmos também diminui, assim como o consumo de energia, de memória e de banda de comunicação (GUPTA, 2002), o que torna o uso das curvas elípticas em dispositivos embarcados especialmente atraente.

3.1.1 CURVAS ELÍPTICAS

Curvas Elípticas podem ser definidas em diversos campos, como o campo dos Números Reais, Corpos Finitos Primos e Corpos Finitos de Característica Dois (WOLSKI, ?).

A forma de Weierstrass de uma curva elíptica é representada por:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Fórmula 9: Equação de Weierstrass de Uma Curva Elíptica. Fonte: (WOLSKI)

Para o Corpo Finito Primo Zp, gerado por um grande número primo p, a curva elíptica E pode ser definida por:

$$y^2 = x^3 + a_4x + a_6$$

Fórmula 10: Equação de Weirstrass para Curva Elíptica em Zp. Fonte: (WOLSKI)

O conjunto E(Zp) é composto por todos os pontos (x,y), x ϵZp, y ϵZp que satisfazem a definição acima, juntamente com o ponto no infinito ***O*** (WOLSKI).

A adição de dois pontos da curva resulta em um terceiro ponto. A operação de adição e o conjunto de pontos E(Zp) formam um grupo, com ***O*** servindo como sua identidade. Com base nisto obtemos as regras de adição (WOLSKI):

1. P + O = O + P = P para todo P ϵ E(Zp)
2. Se P = (x, y) ϵ E(Zp), então (x, y) + (x, -y) = O. (O ponto (x, -y) é representado por –P e é chamado negativo de P. Observe que –P é, também, um ponto na curva).
3. Seja P = (x1, y1) ϵ E(Zp) e Q = (x2,y2) ϵ E(Zp), onde P $\neq$ -Q. Então P + Q = (x3,y3), onde:

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, \text{se P} \neq \text{Q} \\ \frac{3x_1^2 + a}{2y_1} \text{ se P} = \text{Q} \end{cases}$$

Fórmula 11: Soma de Pontos em Curvas Elípticas em Zp. Fonte: (PORTNOI, 2005)

Já as Curvas Elípticas sobre o Corpo Finito de Característica Dois permite a implementação eficiente da aritmética de curvas elípticas. As constantes da equação de Weierstrass são números de base polinomial ou canônica, e a equação pode ser simplificada como (WOLSKI):

$$y^2 + xy = x^3 + a_2x^2 + a_6$$

Fórmula 12: Equação de Weirstrass para Curvas Elípticas sobre Corpo Finito de Característica 2. Fonte: (WOLSKI)

A soma de dois pontos da curva elíptica é dada por:

Se P ≠ Q:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2$$
$$y_3 = \lambda(x_1 + x_3) - y_1$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

Fórmula 13: Soma de Dois Pontos sobre Corpo Finito de Carac. 2(I). Fonte: (WOLSKI)

Se P = Q:

$$x_3 = \lambda^2 + \lambda + a_2$$
$$y_3 = {x_1}^2 + (\lambda + 1)x_3$$

$$\lambda = x_1 + \frac{y_1}{x_1}$$

Fórmula 14: Soma de Dois Pontos sobre Corpo Finito de Carac. 2(II). Fonte: (WOLSKI)

A operação de multiplicação em curvas elípticas é igual tanto para Corpos Finitos Primos e Corpos Finitos de Característica Dois. Ela é o produto de um escalar por um ponto da curva (PORTNOI, 2005).

$$Q = kP$$

Fórmula 15: Multiplicação em Curvas Elípticas. Fonte: (WOLSKI)

Q e P são pontos da curva e k é um inteiro menor do que a ordem do ponto P.

O número de pontos de uma curva elíptica sobre um corpo finito deve satisfazer o teorema de Hasse. Dado um campo, GF(q), a ordem da curva N deverá satisfazer a equação:

$$q + 1 - 2\sqrt{q} \leq N \leq q + 1 + x\sqrt{q}$$

Fórmula 16: Teorema de Hasse. Fonte: (WOLSKI)

Portanto, o número de pontos de uma curva elíptica é aproximadamente o tamanho do corpo. Isto define a Ordem da Curva.

3.1.2 O PROBLEMA DO LOGARITMO DISCRETO

As equações logarítmicas discretas envolvem encontrar o valor inteiro x tal que $a^x = b$. Em curvas elípticas, a exponenciação de um ponto E é uma adição repetida. Portanto, as equações logarítmicas discretas em curvas elípticas envolvem encontrar um inteiro x tal que xa = b (PORTNOI, 2005).

Nenhum algoritmo eficiente para a resolução deste problema é conhecido (STINSON, 2006). Um algoritmo de força-bruta envolve testar expoentes iterativamente até se encontrar um valor que satisfaça a equação, e tem tempo de execução exponencial. Existem algoritmos mais eficientes, porém nenhum deles é executado em tempo polinomial (LÓPEZ, 2000).

3.2 ALGORITMOS MODIFICADOS

Os algoritmos modificados serão apresentados abaixo. As regras para conversão do algoritmo de Schnorr clássico para algoritmos em curvas elípticas, explicadas pelo prof. Barreto, são:

1. Sejam p, q dois números primos. Em curva elíptica, o número p deixa de ser utilizado. q passa a ser a ordem da curva;
2. O parâmetro g do algoritmo de Schnorr passa a ser o ponto gerador da curva;
3. Expressões terminadas em (mod p) são pontos da curva;
4. Expressões terminadas em (mod q) são inteiros;
5. Operações de exponenciação se tornam multiplicações em curvas elípticas;
6. Operações de multiplicação se tornam somas em curvas elípticas.

3.2.1 ASSINATURA SCHNORR

A assinatura de Schnorr modificada se torna:

1. Uma curva elíptica E é gerada com o nível de segurança desejado. A curva possui os parâmetros: q = inteiro que representa a ordem da curva e g = ponto gerador da curva. Estes parâmetros são compartilhados por todos os usuários do sistema;
2. Com base nestes parâmetros, é criado o par de chaves: x ϵZq*, que é a chave privada; e o ponto y = g * (- x), a chave pública;
3. Para assinar uma mensagem, é calculado um ponto r = g * K, K ϵ Zq* é um número gerado aleatoriamente;
4. Calcula-se e = H(m || r) mod q, onde H() é uma função de Hash;
5. Calcula-se o inteiro s = K + ex mod q;
6. A assinatura é o par {e, s};
7. Na verificação, calcula-se e' = H (m || g*s + y* e). A assinatura é válida se e' = e;

3.2.2 ASSINATURA SCHNORR CEGA

Seja Banco a entidade que assinará cegamente uma mensagem e Alice a portadora desta mensagem a ser assinada. O algoritmo da Assinatura Schnorr Cega é igual à Assinatura de Schnorr clássica, descrita no item 3.2.1, até o passo 2.

3. Alice pede ao Banco que assine uma mensagem;
4. Banco seleciona um valor aleatório K ϵ Zq*, e envia o "commitment" r = g * K que é um ponto da curva;
5. Alice recebe o "commitment" r e seleciona dois números aleatórios, α, β ϵ Zq;
6. Alice ofusca r calculando o ponto r' = r + g * (- α) + y * (- β) e faz e' = H(m || r') mod q;
7. Alice envia o "challenge" e = e' + βmod q ao Banco.
8. O Banco calcula s = K + x * e, enviando para Alice.
9. Alice calcula s' = s - α mod q. O par {e', s'} é a assinatura cega da mensagem m.
10. Na verificação, a assinatura é válida se e' = H(m || g * s' + y * e').

3.2.3 PROTOCOLO DE RETIRADA

O protocolo de retirada exclui a Assinatura Cega RSA com fator de Chaum e passa a utilizar a Assinatura Cega de Schnorr:

1. Alice inicia o processo, pedindo a retirada de uma moeda;
2. O Banco verifica se há saldo disponível. Se sim, continua com o processo;
3. O Banco gera o "commitment" e o envia a Alice;
4. Alice escolhe um número aleatório SSG ϵ Zq* (*Secret Signature Generator*) e calcula o ponto PSG = g * SSG (*Public Signature Generator*). H(PSG || y) é o identificador da moeda e a mensagem a ser assinada pelo Banco cegamente.

5. Alice envia o "challenge" conforme o protocolo de Assinatura Schnorr Cega;
6. Banco calcula e devolve a Alice o valor s e o certificado digital da chave pública do banco;
7. Alice extrai o par {e', s'} e verifica se a Assinatura é válida. A mensagem H(PSG || y) e o par {e', s'} formam COIN.

3.2.4 PROTOCOLO DE PAGAMENTO

O protocolo de pagamento não teve alterações significativas.

3.2.5 PROTOCOLO DE DEPÓSITO

O protocolo de depósito só foi alterado em um ponto: ao invés de depositar o valor z = (g * y + v * e) encriptado, deposita-se a chave pública encriptada de quem fez o pagamento. Com isto, eliminam-se os cálculos para obter a chave pública a partir de z em casos de fraude, ao mesmo tempo em que o Banco não tem como descobrir a identidade de quem fez o pagamento caso não haja fraude.

A versão atual do sistema deposita o valor de z como um vetor de bytes zerado. A explicação é que, sem trabalhar com certificados digitais, o algoritmo de detecção de fraudes nunca chega ao ponto de pedir a decriptação do z pelo comerciante. O valor de z encriptado deverá ser implementado na mesma versão do sistema que iniciar o uso de certificados digitais, o que permitirá o funcionamento pleno do sistema de detecção de fraudes.

4 IMPLEMENTAÇÃO

A prova de conceito está dividida em três subsistemas, que simulam um dispositivo cliente, um caixa de comércio e um caixa de banco. O dispositivo cliente tem como plataforma destino *smartphones* com a tecnologia Android. Já os sistemas caixa e banco utilizam computadores com sistema operacional Windows e suporte a tecnologia Bluetooth.

A metodologia de desenvolvimento e as ferramentas utilizadas se encontram descritas nesta seção.

4.1 METODOLOGIA

A metodologia adotada para o desenvolvimento desta prova de conceito é o modelo em cascata, no qual há fases definidas e pontos de validação. Uma fase é considerada concluída somente quando há aceitação de seus produtos.

As fases de análise do projeto envolvem: especificação de requisitos, modelo de casos de usos, interface homem computador, diagrama de classes e diagrama de interações. Após a aprovação destes documentos, temos as fases de Especificação da Arquitetura, implementação, testes e aceitação.

4.1.1 ESPECIFICAÇÃO DE REQUISITOS

Define os requisitos funcionais e não funcionais do projeto. Descreve o escopo da prova de conceito do sistema de micropagamentos.

Este documento se encontra no APÊNDICE A - Documento de Especificação de Requisitos

4.1.2 MODELO DE CASOS DE USO

Define a modelagem dos casos de uso do sistema de micropagamentos, caracterizando atores e funcionalidades do sistema.

Este documento se encontra no APÊNDICE B – Modelo de Casos de Uso.

4.1.3 INTERFACE HOMEM MÁQUINA

Define as interfaces do sistema com os usuários.

Este documento se encontra no APÊNDICE C – Interface Homem Computador.

4.1.4 MODELO DE CLASSES

Define a modelagem de classes que compõem o sistema. Define atributos e métodos, numa visão estática.

Este documento se encontra no APÊNDICE D – Modelo de Classes.

4.1.5 MODELO DINÂMICO

Modelo que mostra a troca de mensagens entre os diversos objetos que compõem o sistema. Consolida os atributos e métodos do Modelo de Classes.

Este documento se encontra no APÊNDICE E – Modelo Dinâmico.

4.1.6 DESCRIÇÃO DAS INTERFACES DE COMUNICAÇÃO

Especifica o formato das mensagens trocadas pelos subsistemas para comunicação.

Este documento se encontra no APÊNDICE F – Interfaces de Comunicação.

4.1.7 ESPECIFICAÇÃO DA ARQUITETURA

Especifica a Arquitetura que será utilizada para a implementação do sistema, definindo ferramentas e hardware utilizados, os pacotes que compõem o sistema e também os nós de *hardware*.

Este documento se encontra no APÊNDICE G – Especificação da Arquitetura.

4.1.8 PLANO DE TESTES E ACEITAÇÃO

Plano de Testes e Aceitação especifica quais são os testes e os cenários de testes necessários para a validação e aceitação do sistema, passo-a-passo e com resultados esperados.

Este documento se encontra no APÊNDICE H – Plano de Testes e Aceitação.

4.2 AMBIENTE DE DESENVOLVIMENTO

Nesta seção descrevemos o ambiente de desenvolvimento utilizado.

4.2.1 JAVA

Java é uma linguagem de programação desenvolvida pela Sun Microsystems e lançada em 1996. Tinha como foco inicial dispositivos móveis digitais como os celulares, mas se popularizou em aplicações para a Internet.

O código fonte é escrito em arquivos de extensão .java, que se tornam arquivos .class ao serem compilados pelo javac. Ao invés do código nativo de um processador, o arquivo .class possui a linguagem de máquina do *Java Virtual Machine* (JVM), que são os *bytecodes*. Cada programa é executado em uma instância da JVM (THE JAVA TUTORIALS).

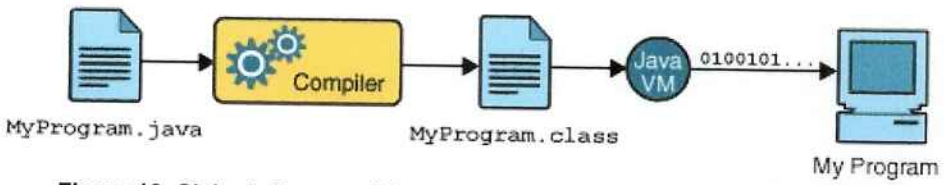


Figura 16: Ciclo de Desenvolvimento Java. Fonte: (THE JAVA TUTORIALS).

A existência da JVM para diversas plataformas permite que o mesmo arquivo .class seja executado em todas elas. Porém, esta característica torna o Java mais lento do que código nativo para cada plataforma, o que tenta se contornar com avanços nas áreas de compiladores e máquinas virtuais (idem).

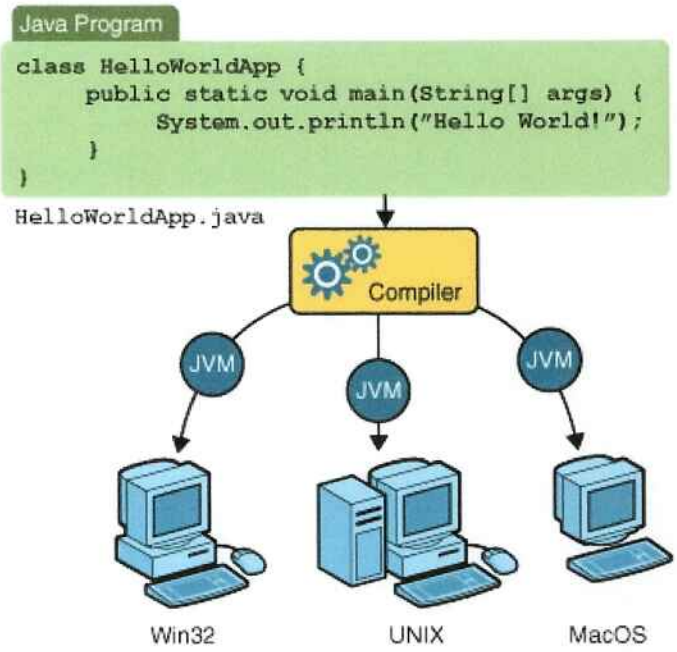


Figura 17: Java Virtual Machine para Diversas Plataformas. Fonte: (idem)

A plataforma do Java é composta por dois componentes: *Java Virtual Machine* e *Java Application Programming Interface* (API). O API é uma coleção de *softwares* prontos implementam as funções mais comuns, agrupados em bibliotecas de classes relacionadas e interfaces denominadas pacotes(idem).

4.2.2 ANDROID

O Android é um conjunto de aplicações para dispositivos móveis que inclui um sistema operacional, um *middleware* e algumas aplicações que constituem seu núcleo (ANDROID DEVELOPERS, 2011). É desenvolvido pela *Open Handset Alliance*, um grupo de 84 empresas fundada em novembro/2007 (OPEN HANDSET ALLIANCE, 2011), que busca estabelecer padrões abertos para a telefonia móvel, encabeçada pelo Google.

A empresa Android Inc., que é a responsável pelo desenvolvimento do Android, foi fundada em outubro/2003, sendo adquirida pelo Google em agosto de 2006 (BAUER, 2011). O primeiro celular com Android foi lançado em setembro/2008,

alcançando uma fatia de 43% do mercado americano de smartphones no terceiro quadrimestre de 2011 (YIN, 2011). A mesma pesquisa aponta que os celulares comuns ainda são a maioria dos dispositivos, mas há uma tendência de alta para o mercado de *smartphones*.

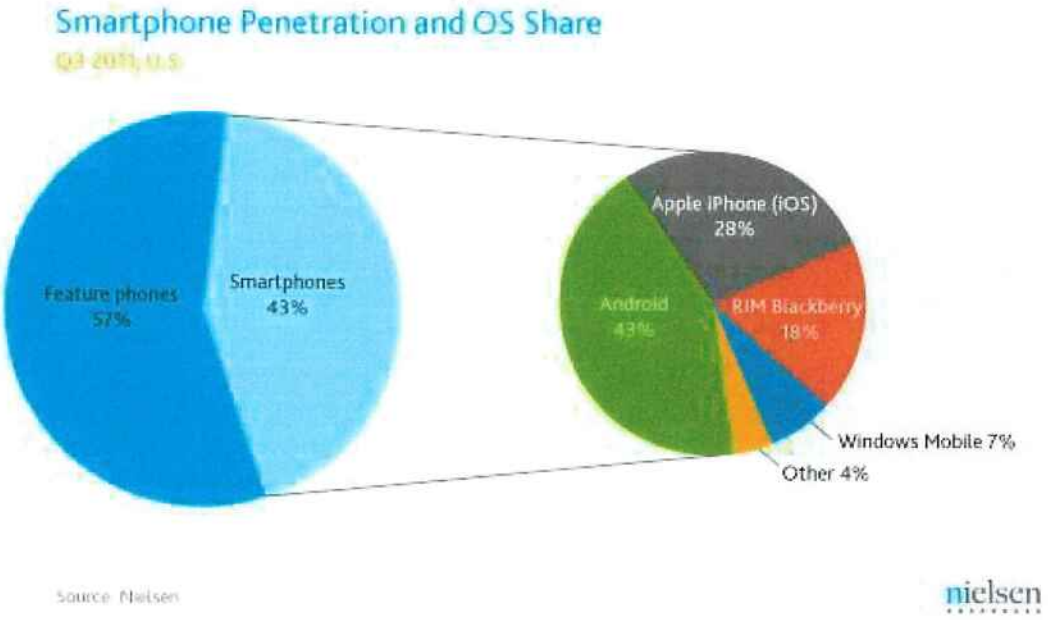


Figura 18: Mercado de Smartphones. Fonte: (YIN, 2011)

A arquitetura do Android é composta pelas camadas (ANDROID DEVELOPERS, 2011):

- Aplicações: as aplicações do Android são escritas em linguagem Java. O Android possui algumas aplicações que integram seu núcleo como cliente de email, programa de SMS, calendário, mapas, etc.
- *Framework* de Aplicações: o Android é uma plataforma aberta de desenvolvimento, permitindo aos desenvolvedores o acesso completo às APIs do *framework*. Estes APIs fornecem ferramentas para uso de elementos visuais (*grids*, *textboxes*), acesso ao conteúdo de outras aplicações como contatos, entre outros.
- Bibliotecas: O Android possui bibliotecas C/C++ que são utilizadas pelos diversos componentes do sistema. As principais bibliotecas são (ANDROID DEVELOPERS, 2011)
  - Biblioteca C do sistema: implementação da biblioteca padrão do C, otimizada para sistemas embarcados Linux

- Bibliotecas de Mídia: permite executar e gravar formatos de áudio, vídeo e imagens. Suporta MPEG4, H.264, MP3, AAC, AMR, JPG e PNG.
- SQLite: banco de dados relacional disponível para todas as aplicações.

As bibliotecas são expostas pelas APIs do framework.

- AndroidRuntime: o Android implementa um conjunto de bibliotecas centrais que implementam a maioria das funções do Java. Os aplicativos do Android são escritos em Java, porém são executados em uma máquina virtual própria denominada Dalvik. Assim como no Java, cada aplicação executa seu processo em uma instância do Dalvik, executando um código compilado com extensão .dex. A ferramenta dx é a responsável pela transformação do código Java para o código da Dalvik. O Dalvik utiliza o *kernel* do Linux para gerenciamento de memória e de *threads*
- O Android utiliza o Linux versão 2.6 como uma camada de abstração entre o *hardware* e as outras camadas de *software*. O Linux é responsável pela segurança, pelo gerenciamento de memória e de processos, entre outros.

A arquitetura do Android pode ser observada na figura 17:

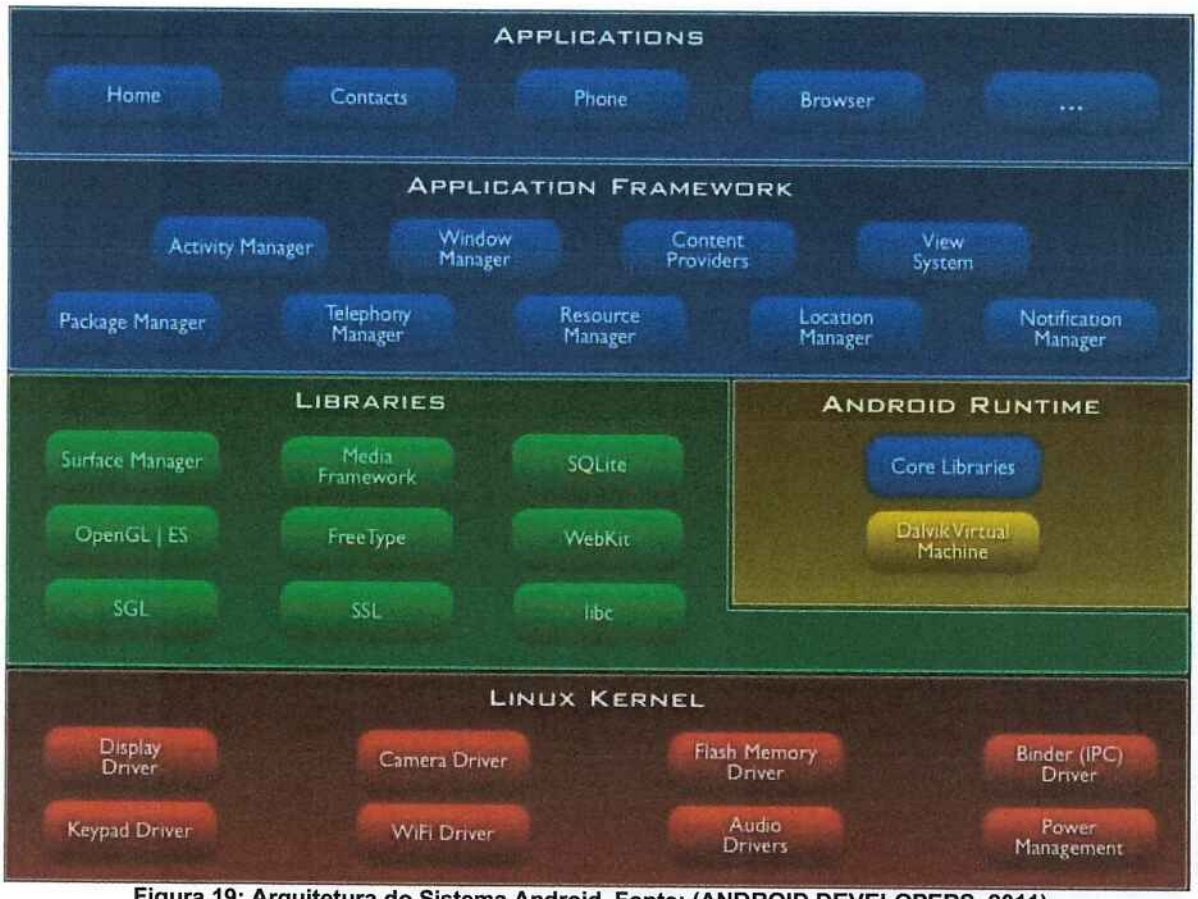


Figura 19: Arquitetura do Sistema Android. Fonte: (ANDROID DEVELOPERS, 2011)

4.2.3 BLUETOOTH

O Bluetooth é uma tecnologia sem fio e de baixo alcance para a transmissão de dados e voz, constituindo uma *Wireless Personal Área Network* (WPAN), especificado pelo IEEE. 802.15.1 (MITCHELL, 2011). A comunicação é feita usando ondas de rádio unidirecionais na frequência de 2.4 GHz, a mesma do Wi-Fi (IEEE 802.11), mas com um método diferente de transmissão, baseada em F*requency Hopping Spread Spectrum* (FHSS). Este método seleciona aleatoriamente um dos 79 canais de transmissão, repetindo este processo 1600 vezes por segundo (PC MAGAZINE, 2011). O Bluetooth é

Foi desenvolvida para criar redes wireless simples entre dispositivos e periféricos de uso pessoal (MITCHELL, 2011), como telefones celulares, computadores e periféricos como mouse e teclados. As tabelas de velocidade de transferência (tabela 1) e consumo de energia (tabela 2) são demonstradas a seguir:

Tabela 1: Velocidade de Transferência por Versão do Bluetooth

Versão	Taxa de Transferência (Mbps)
1.2	1
2.0 + EDR	3
2.0 + HS	24
4.0 + HS	24

Fonte: (PC MAGAZINE, 2011)

Tabela 2: Potência e Alcance por Versão do Bluetooth

Classe	*Potência Máxima*	*Alcance Operacional*
Classe 1	100 mW (20 dBm)	100 metros
Classe 2	2.5 mW (4 dBm)	10 metros
Classe 3	1 mW (0 dBm)	1 metro

Fonte: (BLUETOOTH INSIGHT, 2008)

Existem no mercado diversas implementações da pilha do protocolo Bluetooth. Este projeto utiliza o Microsoft Windows Stack do Windows 7, que suporta a versão 2.1+EDR.

O Android utiliza a pilha de protocolo Bluetooth BlueZ versão 3.36, suportando a versão 2.0 + EDR do Bluetooth. A sua arquitetura pode ser vista na figura 18 (ANDROID OPEN SOURCE PROJECT, 2011):

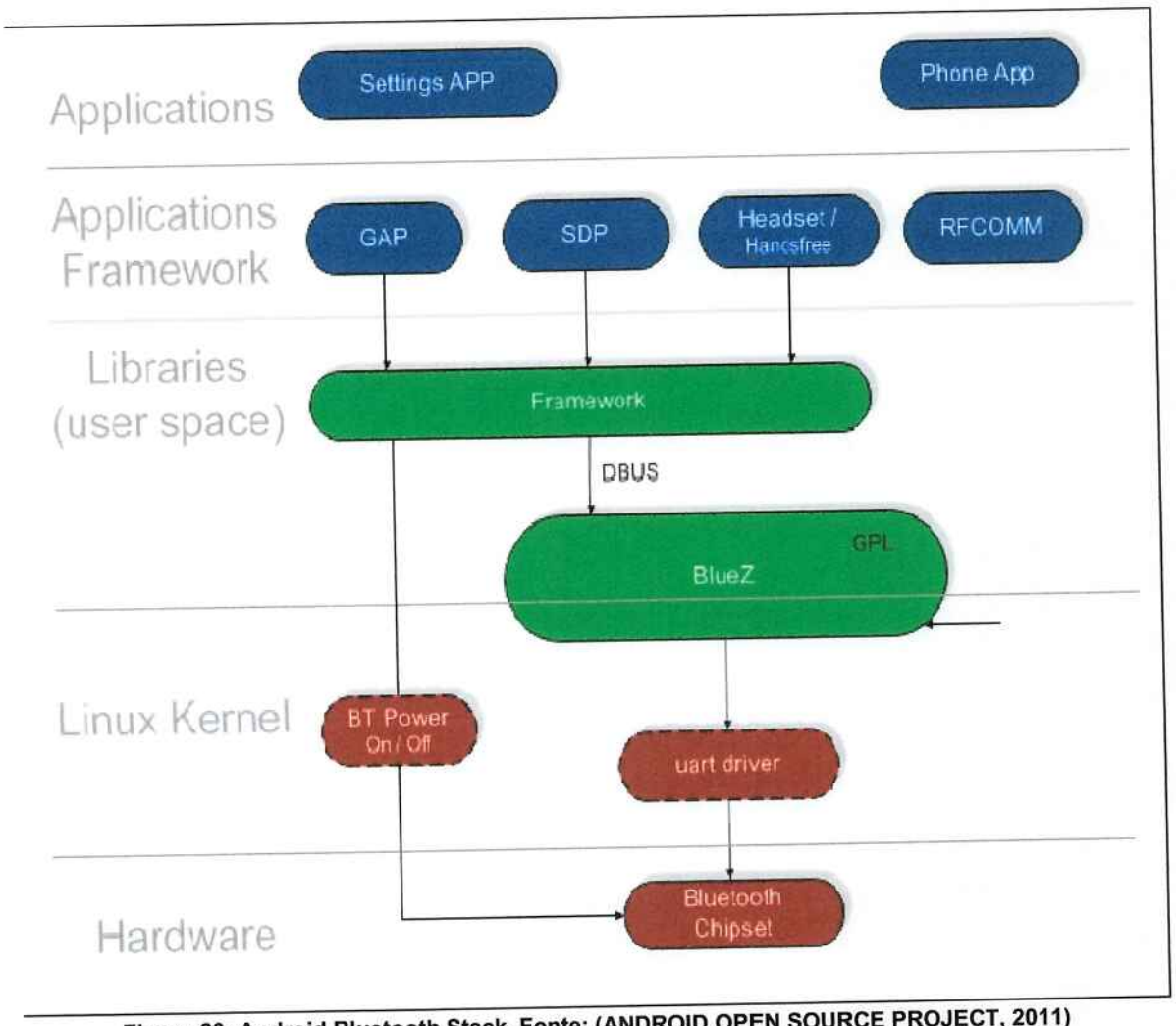


Figura 20: Android Bluetooth Stack. Fonte: (ANDROID OPEN SOURCE PROJECT, 2011)

4.2.4 BNPAIRINGS

A biblioteca BNPairings, de autoria do Prof. Doutor Paulo S. L. M. Barreto e de Geovandro C. C. F. Pereira, é uma implementação Java para pareamento bilinear e operações em curvas elípticas (PEREIRA, G. C, 2011).

Esta biblioteca é utilizada para implementar os algoritmos dos protocolos de moeda eletrônica. É altamente eficiente para calcular operações em curvas elípticas (PEREIRA, G. C, 2011), o que possibilita o seu uso em *smartphones* Android. Além disso, é bem documentada e concisa, retirando grande parte da complexidade de se trabalhar com curvas elípticas.

É um software livre, sob licença GNU *General Public Licence version* 2. Pode ser obtida em http://code.google.com/p/bnpairings/.

4.2.5 OUTRAS FERRAMENTAS

Para o controle de versões e acesso ao código foi utilizado o programa TortoiseSVN, que é um cliente do *Subversion* para Windows, disponível em http://tortoisesvn.tigris.org/. O servidor SVN utilizado foi o Assembla (www.assembla.com).

A documentação do sistema utilizou o auxílio de ferramentas como o Microsoft Visio, ArgoUML (http://argouml.tigris.org/) e AstahCommunity (http://astah.net/editions/community)

5 RESULTADOS E ANÁLISE

Nesta seção analisamos a prova de conceito desenvolvida.

5.1 RESULTADOS

Nossa prova de conceitos conseguiu implementar com sucesso os protocolos de moeda eletrônica propostos por Wenbo Mao, com as adaptações como uso de Curvas Elípticas e Assinatura de Schnorr Cega (POINTCHEVAL, 2000) em substituição à Assinatura RSA Cega com fator de Chaum (CHAUM, 1982).

O ciclo de Retirada de Moedas Eletrônicas, Pagamento e Depósito foi concluído, assim como a comunicação entre os subsistemas e os equipamentos.
Em comum acordo com orientador, Prof. Dr. Paulo Barreto, o uso de Certificados Digitais ficou fora do escopo do projeto, o que não permite a aplicação integral do algoritmo de detecção de fraude. Em uma implementação completa do sistema é obrigatório a implementação do uso destes certificados.

A seguir discutimos a viabilidade do Sistema de Moedas Eletrônicas, seu desempenho em *smartphones*, considerando que eles possuem poder de processamento mais limitado; e também futuras versões e a contribuição deste projeto aos protocolos propostos por Wenbo Mao.

5.2 ANÁLISE DE VIABILIDADE

A prova de conceito desenvolvida não utiliza certificados digitais para adequar o escopo a um trabalho de conclusão de curso. Os certificados digitais são essenciais para o funcionamento real do sistema, porque permitem autenticar a posse da chave pública (BARRETO, 2011b). Sem eles, é possível se passar por outra pessoa e os algoritmos de detecção de fraude não consegue provar o autor da

fraude. É essencial que estas funções sejam implementadas em uma versão completa do sistema.

Outro ponto que precisa ser resolvido é a comunicação por Bluetooth. Apesar de ser comum em dispositivos celulares, a necessidade de emparelhamento dos aparelhos, característica de WPAN, e o alcance de 10 metros, que permite localizar muitos dispositivos podem se tornar empecilhos para o uso em larga escala do sistema. Uma solução é utilizar o NFC, que para o usuário significa autorizar o pagamento e aproximar o aparelho do leitor, sem ter que percorrer toda uma lista de dispositivos para encontrar o aparelho Bluetooth correto.

Como qualquer troca de comunicação, mensagens podem ser corrompidas ou perdidas. É necessário estabelecer um protocolo adequado, com *checksum* das mensagens trocadas e algoritmos que permitam o reenvio de uma mensagem específica, com pontos de sincronização do fluxo de mensagens. No modelo atual, caso ocorra um erro todo o processo é descartado.

Para cada moeda eletrônica gerada há os processos de assinatura. Para diminuir este processamento, o autor do artigo Wenbo Mao especifica um protocolo levemente alterado que permite que uma moeda seja subdividida para valores menores. Isto permite a diminuição do processamento, uma vez que pode se retirar uma moeda com valor alto e subdividi-la no momento do pagamento. O protocolo modificado pode ser encontrado em (MAO, 1996) e torna o sistema mais robusto.

A unidade Moeda não possui nenhuma relação com uma unidade monetária real. Essa relação só pode ser estabelecida em uma aplicação comercial do sistema completo. Também é importante salientar que o protocolo de moeda eletrônica não especifica troco, é sempre pago exatamente a quantia desejada.

O custo do sistema é relacionado com a taxa cobrada pelo banco para retirada/depósito de moedas eletrônicas e uma possível taxa pelo uso comercial do sistema. Porém, os custos de manutenção do sistema são baixos, o que permite que uma taxa muito menor seja cobrada em relação às tradicionais administradoras de cartão de crédito. Isto ocorre porque não é necessária utilizar uma rede para aprovar a transação, o próprio algoritmo é capaz de identificar se o processo de pagamento é válido.

5.3 ANÁLISE DE DESEMPENHO

Nesta seção é feita a análise de desempenho do sistema desenvolvido.

5.3.1 TEMPO DE COMUNICAÇÃO

Foram executados testes de desempenho com a versão final do sistema.

Foi simulada a operação de recarga diversas vezes, para um número variado de moedas, e a partir dos resultados, verificamos o tempo médio de transação do protocolo inteiro para a recarga daquela quantidade. O mesmo método foi utilizado para averiguar o tempo médio necessário para realizar a operação de pagamento.

Os testes foram realizados utilizando valores de 1, 2, 3, 5, 10, 20 e 50 moedas, tanto para o processo de recarga quanto para o processo de pagamento. Foi utilizado um celular Motorola Milestone 2 (modelo A953), com procesador OMAP 3640 1.2 GHz, 512 MB de memória RAM e sistema operacional Android 2.2 Froyo para rodar o aplicativo, e um computador MacBook Pro modelo mc374bz/a, com processador Intel Core 2 Duo 2.4GHz e 4 GB de memória RAM para rodar os módulos Banco e Caixa.

A medição de tempo foi feita utilizando os módulos Banco e Caixa. Assim que o processo de recarga ou pagamento fosse iniciado, era iniciado a contagem do tempo. Quando a mensagem de fim de operação fosse enviada ou recebida, o contador era interrompido. Vale ressaltar que tal medida de tempo desconsidera o tempo necessário para iniciar a comunicação *bluetooth* e a confirmação do usuário. Para cada valor de moeda, foram realizadas 15 medidas de tempo.

Tabela 3: Tempo Médio de Operação - Recarga

Moedas	Recarga (ms)	Desvio Padrão (ms)	Valor Mínimo (ms)	Valor Máximo (ms)
1	755,7	27,04338	713	801
2	1449,8	30,34908	1395	1489
3	2134,1	20,13538	2097	2156
5	3261,7	17,15971	3234	3280
10	7280,5	31,05282	7236	7314
20	12705	53,44571	12635	12798
50	30787,4	94,88847	30645	30926

Tabela 4: Tempo Médio de Operação - Pagamento

Moedas	Pagamento (ms)	Desvio Padrão (ms)	Valor Mínimo (ms)	Valor Máximo (ms)
1	494,9	36,00	438	562
2	751,4	30,92	710	793
3	1010	36,23	962	1066
5	1639,4	37,44	1583	1684
10	2664,4	24,28	2624	2703
20	5247,3	30,72	5198	5297
50	12364,7	27,66	12316	12406

É possível observar que as operações de recarga e pagamento tem tempo de execução com comportamento linear. Isso ocorre devido ao fato que o processo de troca de mensagens para recarga ou pagamento de uma moeda precisa ser repetido um número de vezes equivalente à quantidade de moedas que foram adquiridas ou que estão sendo cobradas.

A velocidade maior da operação de pagamento se explica pela quantidade menor de mensagens trocadas durante o processo de pagamento de uma moeda. O protocolo também permite que haja uma redução do número de passos quando é efetuado o pagamento de várias moedas, visto que a mensagem que contém a

chave pública do Comerciante só precisa ser enviada uma única vez por pagamento, independente do número de moedas que serão pagas.

Foi proposto um limite de tempo de espera de 8 segundos, após o qual o uso do aplicativo seria considerado incômodo pelo usuário. Os testes mostram que para operações envolvendo 10 moedas, o tempo de recarga fica ligeiramente abaixo do limite, enquanto que o tempo de pagamento tem uma margem confortável. Considerando que o uso proposto do sistema é a realização de micropagamentos, *i.e.* operações de baixo valor, e que mesmo para operações envolvendo 10 moedas o tempo de operação se encontra dentro do limite considerado como confortável ao usuário, o sistema se mostra aceitável sob esse critério.

5.3.2 TAMANHO DA MOEDA

Com o intuito de averiguar o desempenho em utilização de espaço armazenamento e banda de transmissão, foi comparado o tamanho de uma moeda eletrônica gerada utilizando o modelo proposto por Wenbo Mao e uma gerada pelo modelo revisto.

As moedas tem informação idênticas:

- uma identificação;
- uma assinatura;
- a chave pública de quem assinou a moeda;
- o PSG;
- e o SSG;

A diferença entre os métodos se dá no tamanho da chave pública e no PSG. No modelo clássico, o número de bits recomendado para uma chave é de 1024 bits, ou seja, 128 bytes. No modelo revisado, utilizando curvas elípticas, é possível oferecer nível de segurança semelhante utilizando uma chave de 158 bits (GUPTA, 2002). Considerando que a substituição ocorre em dois campos, temos uma redução de 2048 bits para 316 bits, o que significa uma economia de 1732 bits, ou 216,5 bytes.

Ainda que tal valor pareça pequeno, frente à capacidade de armazenamento disponível atualmente, a economia pode ser significativa caso o sistema seja expandido para contemplar modelos mais simples de celulares. Também é digno de nota a economia no consumo de banda nas operações proporcionado pela redução de tamanho da moeda.

5.4 ANÁLISE DE ALGUNS ATAQUES

A prova de conceitos não trabalha com certificados digitais, o que impede a comprovação da identidade do dono da chave (BARRETO, 2011b). Isto permite que uma pessoa se passe por outra, caso que não ocorre no protocolo completo, seguro neste quesito.

O caso discutido na seção 2.2.4 Detecção de Fraudes, no qual um atacante seleciona uma chave privada qualquer que coincide com um usuário real do sistema, e comete fraudes com esta chave, é um ponto teoricamente vulnerável, porém inviável na realidade. O grande número de chaves torna muito difícil encontrar uma chave que sirva a este golpe, e como o sistema é presencial, em muitos casos é possível provar que o verdadeiro dono da chave não esteve naquele local, permitindo a identificação do comerciante como fraudador. E como o sistema é de micropagamentos, a sofisticação necessária ao ataque torna os custos maiores do que os possíveis benefícios.

Mesmo sem trabalhar com certificados digitais, não é possível gastar uma moeda sem o conhecimento da chave privada, o que implica que, mesmo escutando a comunicação, um agressor não consegue obter todas as informações para fingir que é outra pessoa. É o próprio algoritmo de Assinatura de Schnorr que garante esta característica, o que torna desnecessário o uso de canais seguros de comunicação.

A implementação errada de um algoritmo seguro pode resultar num sistema falho. Portanto, parâmetros como a chave privada devem ser armazenados seguramente nos sistemas, até envolvendo processos de criptografia na armazenagem da chave privada em banco ou arquivos.

As mensagens trocadas pelo protocolo também devem ser seguras, utilizando mecanismos para detectar alterações e mensagens corrompidas. A alteração de algum parâmetro da Assinatura de Schnorr seria detectada na verificação da assinatura, mas se as mensagens envolvessem, por exemplo, um número de conta, o agressor poderia alterar a conta destino.

Por fim, é importante salientar que o sistema é destinado a micropagamentos. Fraudes, como gasto duplo de uma moeda, são detectados apenas no depósito da segunda moeda eletrônica. E como o sistema não é online na parte de depósitos, há um intervalo de tempo considerável para detecção. Meios para impedir tais ocorrências envolvem tornar as fraudes não atrativas, cobrando uma taxa para emissão de chave pública suficientemente alta para não compensar os micropagamentos fraudados, já que os envolvidos têm suas chaves públicas banidas. Outra solução é tornar o protocolo de depósitos automático, comunicando com o banco em curtos intervalos de tempo e torná-lo quase online. Desta forma, fraudes são detectadas rapidamente e chaves públicas podem ser suspensas para investigação, evitando novos gastos.

5.5 CONTRIBUIÇÕES

Esta prova de conceito tem duas contribuições:

1. Adaptação do protocolo para uso de Curvas Elípticas. Por utilizar chaves menores do que as utilizadas pelo RSA, a Criptografia em Curvas Elípticas é mais eficiente e consome menos energia, memória e banda de comunicação (GUPTA, 2002).
2. Substituição do Algoritmo de Assinatura Cega RSA com fator de cegamento de Chaum (CHAUM, 1982) por Assinatura Cega Schnorr (POINTCHEVAL, 2000). A uniformização do algoritmo diminui a chance de *bugs* que surgem na integração de algoritmos.

5.6 FUTURAS VERSÕES

Futuras versões devem implementar:

- Certificados Digitais: para garantir o pleno funcionamento do algoritmo de detecção de fraude e garantir a Autenticidade nos protocolos de moeda eletrônica.
- Encriptação da chave pública do cliente no depósito: Atualmente este valor não é depositado, já que, sem certificados digitais, o algoritmo de detecção de fraudes nunca alcança o ponto de pedir a decriptação do valor da chave pública depositada.
- Aperfeiçoar o ID da moeda: atualmente utilizamos a mensagem a ser assinada cegamente pelo Banco, H(PSG || v), como ID que identifica a moeda nos processos de detecção de fraude. Apesar de ser extremamente baixa, podem ocorrer casos em que o PSG sorteado já foi utilizado, levando a uma moeda duplicada. É necessário adicionar um termo que não se repita. Um *Timestamp* poderia ser uma solução, mas abriria uma brecha para identificar o cliente baseado na hora de retirada da moeda.
- Versão do módulo Caixa como um aplicativo do Android, oferecendo mais liberdade ao Comerciante, e expandindo o uso do sistema.

6 CONCLUSÃO

O projeto de formatura apresentado teve como objetivo mostrar a viabilidade de um sistema de micropagamento, com custo computacional baixo o suficiente para ser utilizável em aparelhos celulares sem que seu desempenho afetasse negativamente a experiência do usuário.

Foi observado que em situações de estresse do sistema, em especial em recargas com quantidades elevadas de moedas eletrônicas, a troca de mensagens torna o processo lento. Porém, em situações normais, e considerando o uso para micropagamentos, o tempo de recarga fica dentro do limite proposto como aceitável.

Foi adotada uma simplificação na prova de conceito, a não utilização de um certificado digital, o que impede que seja garantida a autenticidade dos usuários, abrindo, assim, a possibilidade de fraudes envolvendo o uso de identidades falsas. No entanto, uma aplicação real do sistema envolveria o uso de certificados digitais, prevenindo tais fraudes.

O protocolo desenvolvido, utilizando Criptografia de Curvas Elípticas, dá ao sistema características importantes, como o reaproveitamento de código, decorrente da homogeneidade do algoritmo, além de ser mais eficiente, fornecendo um mesmo nível de segurança, mas utilizando menos recursos, sendo também uma solução mais moderna, de acordo com as normas mais recentes adotadas no campo de Criptografia.

Visto que em condições normais de execução o sistema se mostrou aceitável aos limites propostos, que os requisitos funcionais exigidos foram cumpridos, e que o protocolo utilizado se mostrou seguro, rápido e confiável para a aplicação proposta, o sistema se mostra viável como uma alternativa para a realização de micropagamentos.

7 REFERÊNCIAS

ANDROID DEVELOPERS.**What is Android?** 2011.
Disponível em: <http://developer.android.com/guide/basics/what-is-android.html>. Acesso em: 13 de novembro de 2011.

ANDROID OPEN SOURCE PROJECT.**Bluetooth.** 2011.
Disponível em: <http://www.kandroid.org/online-pdk/guide/bluetooth.html>. Acesso em: 13 de novembro de 2011.

BARRETO, P. S. L. M. **Infra-estrutura de Chaves Públicas ICP – Brasil.** São Paulo: EPUSP – PCS – LARC, 2011b. 77 slides, color. Acompanha texto.

BARRETO, P. S. L. M. **Resumos criptográficos (funções de hash).** São Paulo: EPUSP – PCS – LARC, 2011. 49 slides, color. Acompanhatexto.

BAUER, J. **The AndroidStory – History Infographic**. 2011.
Disponível em: <http://technorati.com/technology/android/article/the-android-story-history-infographic/>. Acesso em: 13 de novembro de 2011.

BERTONI, G. et al. **Cryptographic Sponges.**2011a.
Disponível em: <http://sponge.noekeon.org/.>. Acesso em: 03 de novembro de 2011.

BERTONI, G. et al. **Cryptographic Sponge Functions**. 2011b.
Disponível em: <http://sponge.noekeon.org/CSF-0.1.pdf>. Acesso em: 04 de novembro de 2011.

BERTONI, G. et al. **The Keccak reference.**2011c.
Disponível em: <http://keccak.noekeon.org/Keccak-reference-3.0.pdf>. Acesso em: 04 de novembro de 2011.

BERTONI, G. et al. **The KeccakSponge Function Family.** 2011d.
Disponível em: <http://keccak.noekeon.org/>. Acesso em: 03 de novembro de 2011.

BLUETOOTH INSIGHT. **Bluetooth Power Classes**. 2008.
Disponível em: <http://bluetoothinsight.blogspot.com/2008/01/bluetooth-power-classes.html>. Acesso em: 13 de novembro de 2011.

BRASIL. Associação Brasileira de Empresas de Cartões de Crédito e Serviços. **Pesquisa de Mercado de Meios Eletrônicos de Pagamento – População e Comércio.**
Disponível em: <http://www.abecs.org.br/site/indicadores/pesquisas.aspx>. Acesso em 26 de outubro de 2011.

BRASIL. Instituto nacional da tecnologia da informação. **ICP-Brasil**. 2011b. Disponível em:
<http://www.iti.gov.br/twiki/bin/view/Certificacao/PerguntasFrequentes>. Acesso em: 03 de novembro de 2011.

BRASIL. Instituto nacional da tecnologia da informação. **Estrutura da ICP-Brasil**. 2011c. Disponível em:
<http://www.iti.gov.br/twiki/bin/view/Certificacao/PerguntasFrequentes>. Acesso em: 03 de novembro de 2011.

BRASIL. Instituto nacional da tecnologia da informação. **Perguntas Frequentes**. 2011a. Disponível em:
<http://www.iti.gov.br/twiki/bin/view/Certificacao/PerguntasFrequentes>. Acesso em: 03 de novembro de 2011.

BRASIL | (PAÍS). Lei Complementar nº 105 de 10 de janeiro de 2001. Dispõe sobre o sigilo das operações de instituições financeiras e dá outras providências. Diário Oficial da União, Brasília, 11/01/01 p. 1-3(E).
Disponível em: <http://www.bcb.gov.br/pre/leisedecretos/Port/Lei_Compl105.pdf>. Acesso em: 31 de outubro de 2011.

BRASIL | (PAÍS). Medida Provisória nº 2.200-2 de 24 de agosto de 2001. Institui a Infra-Estrutura de Chaves Públicas Brasileira – ICP-Brasil, transforma o Instituto Nacional de Tecnologia da Informação em autarquia, e dá outras providências. Brasília, 27/08/01.
Disponível em: <http://www.planalto.gov.br/ccivil_03/mpv/Antigas_2001/2200-2.htm>. Acesso em: 03 de novembro de 2011.

BUTCHER, D. Visa rolls out NFC contactless mobile paymentscommercially.**Mobile Marketer,** 13 de abril de 2011.
Disponível em: <http://www.mobilemarketer.com/cms/news/banking-payments/3028.html>. Acesso em: 26 de outubro de 2011.

CHAUM, D. **Blind signatures for untraceable payments.** Advances in Cryptology – Crypto '82, Springer-Verlag (1983), 199-203. Disponível em: <http://blog.koehntopp.de/uploads/Chaum.BlindSigForPayment.1982.PDF>. Acesso em: 03 de dezembro de 2011.

CDL/BH. **Mudanças nos Contratos da Redecard e Visanet**. 2009. Disponível em: <http://www.cdlbh.com.br/materia.aspx?or=Comunicados&fo=39>. Acesso em 31 de outubro de 2011.

ESTADOS UNIDOS. Nationalinstituteofstandardsandtechnology.**Cryptographic Hash Algorithm Competition**. 2011. Disponível em: <http://www.nist.gov/itl/csd/ct/hash_competition.cfm>. Acesso em: 03 de novembro de 2011.

GAZETA DO POVO. **Cielo fecha Acordo com a Mastercard.** 2010.Acesso em: 26 de outubro de 2011. Disponível em: <http://www.gazetadopovo.com.br/economia/conteudo.phtml?tl=1&id=1011924&tit=Cielo-fecha-acordo-com-a-MasterCard>

GUIMARÃES, F. **Valor mínimo para pagamento no cartão**. 2011. Disponível em: <http://diariodeconsumoporfernanda.blogspot.com/2011/02/valor-minimo-para-pagamento-no-cartao.html>. Acesso em: 03 de novembro de 2011.

GUPTA, V.; GUPTA, S.; CHANG S. **Performance Analysis of Elliptic Curve Cryptography for SSL.** In: WiSe'02, Atlanta, Georgia, USA, September 28, 2002. Disponível em: <https://labs.oracle.com/projects/crypto/performance.pdf>. Acessoem 20 de novembro de 2011.

KOBLITZ, N. **Elliptic curve cryptosystems.** Mathematics of Computation v. 48, n. 177. pp 203–209, 1987

MAO, W. **Lightweight Micro-Cash for the Internet**. Computer Security - ESORICS 96, 4th European Symposium on Research in Computer Security, Rome, Italy, September 25-27, 1996, Proceedings.Volume 1146 ofLectures Notes in Computer Science, pages 15-32, Springer, 1996.

LÓPEZ, J., DAHAB, R. **An Overview of Elliptic Curve Crytpography**. 2000. Disponível em: < http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.2771>. Acesso em 16 de dezembro de 2011.

MITCHELL, B. **Bluetooth.** 2011. Disponível em: <http://compnetworking.about.com/cs/bluetooth/g/bldef_bluetooth.htm>. Acesso em: 13 de novembro de 2011.

MILLER, V. **Use of elliptic curves in cryptography.** Advances in Cryptology - CRYPTO '85 Proceedings. v. 218, pp 417–426, 1986.

NIKOLAS, K. **Whatis Visa payWave?**. 2010.
Disponível em: <http://www.helium.com/items/1978665-what-is-visa-paywave>. Acesso em 26 de outubro de 2011.

OPEN HANDSET ALLIANCE. **Android**. 2011.
Disponível em: <http://www.openhandsetalliance.com/android_overview.html>. Acesso em: 13 de novembro de 2011.

ORTIZ, C. E. **An Introduction to Near-Field Communication and the Contactless Communication API.** 2006. Disponível em: < http://java.sun.com/developer/technicalArticles/javame/nfc/>
PC MAGAZINE. **Encyclopedia – Bluetooth Definition.** 2011. Disponível em: <http://www.pcmag.com/encyclopedia_term/0,2542,t=Bluetooth&i=38794,00.asp#fbid=bm34-PhJaoY>. Acesso em: 13 de novembro de 2011.

PEREIRA, G. Impor valor mínimo em cartão dá multa de até R$ 6 mi. **R7 Notícias,** São Paulo, de 16 de nov. 2011.
Disponível em: <http://noticias.r7.com/economia/noticias/impor-valor-minimo-em-cartao-da-multa-de-ate-r-6-mi-20110916.html?question=0>. Acessado em: 03 de novembro de 2011.

PEREIRA, G. C. C. F. **Parametrização e otimização de criptografia de curvas elípticas amigáveis a emparelhamentos**. 2011. Dissertação (Mestrado em Sistemas Digitais) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2011. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-13062011-144903/>. Acesso em: 16 de novembro de 2011.

POINTCHEVAL, D.; STERN, J. **Security Arguments for Digital Signatures and Blind Signatures**. JournalofCryptology, v.13, n. 3, p.361-396, 2000. Disponível em: <ftp://ftp.di.ens.fr/pub/users/pointche/Papers/2000_joc.pdf>. Acesso em: 22 de setembro de 2011.

PORTNOI, M. **Criptografia com Curvas Elípticas**. 2005. 7f. Dissertação (Mestrado em Redes de Computadores) - Núcleo de Pesquisas Interdepartamental em Redes de Computadores, Universidade Salvador, Bahia. 2005. Disponível em <http://www.eecis.udel.edu/~portnoi/publications/criptografia_com_curvas_elipticas.pdf. Acesso em 02/11/11.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. **A method for obtaining digital signatures and public-key cryptosystems.** Commun. ACM, ACM, New York, NY (USA),. V.21, p.120-126, 1978. Disponível em: <http://dl.acm.org/citation.cfm?doid=359340.359342>. Acesso em: 04 de dezembro de 2011.

SARAH, P. **Visa to Launch Contactless Mobile Payments for IPhone**. 2010. Disponível em: <http://www.readwriteweb.com/archives/visa_to_launch_contactless_mobile_payments_for_iphone.php>. Acesso em: 26 de outubro de 2011.

SCHNORR, C. P. **Efficient Signature Generation for SmartCards.** Journal of Cryptology, Springer New York, v. 4, n. 3, p.161-174, 1991.

STALLINGS, W. **Cryptography and Network Security: Principles and Practice**.5a ed. NJ: Prentice Hall, 2011, 719p.

STINSON, D. R. **Cryptography: Theory and Practice.** 3a ed. NW: Chapman & Hall/CRC, 2006, 593p.

THE JAVA TUTORIALS. **About the Java Technology**. Disponível em: <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>. Acesso em: 10 de novembro de 2011.

THE OFFICIAL GOOGLE BLOG.**Launching Google Wallet on Sprint and working with Visa, American Express and Discover.**2011. Disponível em: <http://googleblog.blogspot.com/2011/09/launching-google-wallet-on-sprint-and.html>. Acesso em: 26 de outubro de 2011.

WOLSKI, E. **Sistemas Criptográficos baseados em Curvas Elípticas.** Disponível em: <http://www.redes.unb.br/security/criptografia/curvaselipticas/curvas.html#dois>. Acesso em: 03 de novembro de 2011.

YIN, S. **Android Continues toDominate, Gain Mobile U.S. Market Share**. 2011. Disponível em: <http://www.pcmag.com/article2/0,2817,2395804,00.asp#fbid=bm34-PhJaoY>. Acesso em: 13 de novembro de 2011.

APÊNDICE A – ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

1. OBJETIVO DO DOCUMENTO

Este documento tem por objetivo especificar os requisitos do Sistema de Micropagamentos com Assinatura Cega em Aparelhos Celulares, doravante denominado Sistema de Micropagamentos.

Nas seções seguintes há a definição do escopo do sistema, dos outros documentos que irão compor a documentação completa do sistema e seus critérios de aceitação.

2. OBJETIVO DO SISTEMA

2.1. NOME DO SISTEMA

O nome completo é Sistema de Micropagamentos com Assinatura Cega em Aparelhos Celulares, mas na documentação adotaremos o nome de Sistema de Micropagamentos.

O aplicativo para os usuários do sistema de moeda eletrônica, em plataforma Android, tem o nome de Micropag.

2.2. ESCOPO

O objetivo deste trabalho de conclusão de curso é implementar uma prova de conceito para o sistema de micropagamentos descrito pelo artigo Lightweight Micro-Cash for the Internet, de Wenbo Mao, que define três protocolos para uso de

moedas eletrônicas: Retirada de Crédito, Pagamento e Depósito. Estes protocolos poderão sofrer algumas alterações por decisão de projeto, como algumas alterações nos algoritmos de criptografia, mas estas alterações não devem descaracterizar a ideia original do autor.

Este sistema terá os seguintes componentes: um aplicativo para celulares Android, com a capacidade de comunicação e de executar os algoritmos de criptografia do protocolo, inclusive assinatura digital cega; um aplicativo Windows para simular um caixa de um comerciante; outro aplicativo Windows para simular um caixa de um banco.

O modelo de negócio é simples: o usuário possui um aplicativo em seu celular, que possui um saldo de moedas eletrônicas. Estas moedas são retiradas de um caixa de banco. Ao realizar um pagamento, o valor é debitado do saldo do usuário, e os dados desta transação são armazenados no sistema de caixa do comerciante. Para trocar a moeda eletrônica por dinheiro real, o comerciante deve depositá-la no banco. Moedas eletrônicas podem ser utilizadas apenas uma vez e somente pela pessoa que retirou a moeda. Nesta prova de conceito, a moeda eletrônica tem apenas valor unitário.

Moedas são retiradas do banco anonimamente, utilizando algoritmos de assinatura digital cega. O comerciante, ao receber uma moeda eletrônica, verifica a sua validade ao conferir a assinatura do banco na moeda. No depósito, ele não fornece informações sobre quem gastou a moeda, garantindo assim a anonimidade do usuário do sistema. Porém, ao se detectar moedas duplicadas os mecanismos de detecção de fraude permitem identificar o(s) autor(es) da fraude. Seguindo-se os protocolos de moeda eletrônica, outros casos de fraude além de moeda duplicada são infactíveis (MAO, 1996).

O sistema na plataforma Android deve possuir uma interface simples, que permita ao usuário autorizar um micropagamento com o toque de um botão. O celular se comunica com a estação Caixa através de uma rede sem fio, sendo estabelecidos os protocolos para comunicação por Bluetooth. A seguir, as mensagens do protocolo de moeda eletrônica são trocadas.

Os sistemas que simulam o caixa de um comerciante e o caixa de um banco também devem implementar os protocolos de comunicação e criptografia. Suas interfaces devem ser funcionais, permitindo realizar os passos dos protocolos de moeda eletrônica de forma simples.

É importante observar que, apesar dos protocolos de moeda eletrônica necessitarem de Certificados Digitais (MAO, 1996), está fora do escopo deste projeto implementar as funcionalidades para a validação destes certificados.

2.3. DEFINIÇÕES, SIGLAS E ABREVIATURAS

- Android: Sistema Operacional de smartphones.
- Assinatura Cega: algoritmo criptográfico que permite que uma mensagem seja assinada digitalmente sem o conhecimento de seu conteúdo.
- Bluetooth: Tecnologia usada para comunicação entre o aparelho (celular) e a Estação Caixa;
- Micropagamentos: uma transação financeira de uma pequena quantia monetária, no formato eletrônico.
- MicroPag: aplicativo Android que é utilizado pelos clientes do sistema de micropagamentos. Clientes são os indivíduos que retiram moedas eletrônicas de um banco e as gastam em comerciantes na compra de produtos ou serviços.
- Moeda eletrônica: quantia monetária unitária em formato eletrônico;
- Sistema Caixa: sistema que simula um caixa de comerciante, com capacidade de se comunicar por Bluetooth com o dispositivo Android, para receber micropagamentos, e se comunicar com o Sistema Banco para depósitos. Armazena as moedas eletrônicas usadas e é responsável por validar o pagamento.
- Sistema Banco: sistema que simula um caixa de banco, com capacidade de se comunicar por Bluetooth com o dispositivo Android, para permitir a retirada de moedas eletrônicas, e se comunicar com o Sistema Caixa para depósitos.

3. DESCRIÇÃO GERAL

3.1. PERSPECTIVASDO PRODUTO

O Sistema de Micropagamentos será composto por três subsistemas independentes:

- A aplicação Android (MicroPag), onde o usuário pode autorizar débitos de seus fundos;
- Sistema Caixa, que representa o caixa de um comerciante. Recebe moedas eletrônicas de clientes e faz o seu depósito em um banco.
- Sistema Banco: representa o caixa de um banco. Permite a retirada de moedas eletrônicas e seu depósito, trocando por dinheiro real. Ativa protocolos de detecção de fraude nos casos de moeda duplicada.

3.1.1. INTERFACE NO SISTEMA

Cada um dos três subsistemas propostos deve trocar mensagens entre si para o pleno funcionamento do sistema de micropagamentos. Não há interfaces com sistemas externos, uma vez que está fora do escopo o uso de Certificados Digitais.

3.1.2. INTERFACES DE USUÁRIO

O Aplicativo MicroPag possuirá interfaces gráficas para o estabelecimentos das seguintes funções:

- Autorizar Micropagamento
- Verificação de créditos disponíveis
- Recebimento de crédito

O Sistema Caixa possuirá as seguintes interfaces:

- Cobrança de Micropagamento
- Depósito de Moeda Eletrônica

O Banco terá as interfaces:

- Efetuar recarga de crédito;
- Pagamento de depósito

3.1.3. INTERFACES DE HARDWARE

Os hardware necessários são um celular com sistema operacional Android e um computador com sistema operacional Windows 7. Ambos deverão ter comunicação por Bluetooth.

3.1.4. INTERFACES DE COMUNICAÇÃO

Para a comunicação entre o aplicativo MicroPag e os Sistemas Comerciante e Banco, devem ser estabelecidas interfaces de comunicação para que o sistema seja consistente, levando em consideração as limitações impostas pelos meio físico de comunicação, o Bluetooth.

O Sistema Caixa e Sistema Banco também se comunicam entre si, e podem utilizar tecnologias comuns para a comunicação entre dois aplicativos Windows: socket ou TCP/IP.

Estas interfaces serão definidas no documento Descrição de Interfaces.

3.1.5. OPERAÇÃO

O cliente deve adquirir créditos com o Sistema Banco, efetuando uma carga inicial ou uma recarga. São geradas moedas eletrônicas, assinadas cegamente pelo banco.

Ao realizar uma compra, o Sistema Caixa informará o valor e iniciará a comunicação com o celular pela rede Bluetooth. A distância máxima entre os dois deverá ser de 10 metros, limite de alcance do Bluetooth. O cliente deverá então autorizar o micropagamento daquele valor.

Uma vez estabelecida a operação de pagamento e com saldo suficiente, o aplicativo enviará moedas eletrônicas até totalizar o valor da compra. Cada moeda eletrônica tem a assinatura do banco e do pagamento verificadas pelo Comerciante. Se o processo foi bem sucedido, as moedas são deduzidas do saldo do cliente no aplicativo MicroPag e estas moedas são armazenadas no Sistema Caixa.

Para resgatar o valor da moeda eletrônica, o Sistema Caixa se comunica com o sistema de um banco. Este banco verifica a assinatura da moeda eletrônica e também a assinatura de depósito. Se a validação for bem-sucedida, o banco troca a moeda eletrônica pelo valor correspondente.

3.2. FUNÇÕES DO SOFTWARE

As funções de software serão divididas nos três subsistemas:

1. MicroPag
   - Saldo disponível
   - Autorizar Micropagamento
   - Recebimento de crédito
2. Sistema Caixa
   - Cobrança de Micropagamento
   - Recebimento de Compensação
3. Sistema Banco
   - Efetuar recarga de crédito
   - Pagamento de depósito

3.3. CARACTERÍSTICAS DOS USUÁRIOS

Os clientes do aplicativo MicroPag devem ter conhecimento a respeito do uso do sistema operacional Android e de seus aplicativos.

Os comerciantes, ou os usuários do Sistema Caixa devem ter treinamento de informática, já que irão trabalhar com uma interface computadorizada. Seu trabalho será iniciar a operação de cobrança, e participar do processo de compensação de Moeda.

Os funcionários do banco devem ter treinamento em informática e conhecimentos financeiros, sendo responsáveis por gerar crédito, e realizar a troca das moedas eletrônicas por dinheiro.

Os protocolos de comunicação e criptografia não serão acessíveis aos usuários de qualquer subsistema. Sua complexidade não precisa ser entendida pelos clientes do MicroPag, comerciantes, ou funcionários do Banco.

3.4. RESTRIÇÕES

Devido à limitação na capacidade computacional de smartphones, o tempo de execução dos protocolos de moeda eletrônica será um fator limitante.

Além disso, a segurança é um requisito fundamental em qualquer sistema que lide com valores monetários. O sistema deve detectar o uso de moedas duplicadas e tornar pouco atrativo a tentativa de fraude, uma vez que é destinado para micropagamentos.

3.5. HIPÓTESES E DEPENDÊNCIAS

Na hipótese de não funcionamento da comunicação entre os módulos, será desenvolvido um simulador que mostre as operações do protocolo.

3.6. VERSÕES FUTURAS

O artigo de WenboMaotraz ao seu final mudanças nos protocolos para criar moedas divisíveis, inclusive para frações abaixo dos centavos. Os algoritmos são modificados para permitir estas operações, sofrendo assinaturas recursivas. Em futuras versões, tal capacidade pode ser implementada.

Conforme combinado com o orientador, o uso de Certificados Digitais estão fora do escopo por ser de grande complexidade. Isto possibilita algumas brechas de segurança na implementação, porque não é possível autenticar as chaves públicas utilizadas nas assinaturas. Uma futura versão deve implementar seu uso, se comunicando com Autoridades Certificadoras e utilizando os seus protocolos de comunicação e segurança.

4. REQUISITOS ESPECIFICOS

Esta documentação utiliza a análise clássica da orientação à objetos, em cascata. Os seguintes documentos serão elaborados:

4.1. MODELO DE CASOS DE USO

O Documento de Especificação de Casos de Uso contém o Diagrama de Casos de Uso, a descrição dos atores e a descrição dos casos.

4.2. MODELO DE CLASSES

A Documentação do Modelo de Classes contém o Diagrama de Classes e a especificação dos seus atributos e seus métodos, conforme o procedimento de elaboração do modelo de classes.

4.3. MODELO DE INTERAÇÃO

O Documento de Iteração contém os Diagramas de Interação dos principais casos de uso. Contém também uma descrição sucinta do que os objetos realizam nos seus focos de controle. Concentrado no Documento Modelo Dinâmico.

4.4. MODELO DE ESTADOS

O Documento de Estados contém os Diagramas de Estados das classes e dos elementos considerados relevantes. Concentrado no Documento Modelo Dinâmico.

4.5. DESCRIÇÃO DAS INTERFACES

O Documento de Descrição das Interfaces descreve as interfaces para a troca de mensagens.

5. REQUISITOS NÃO FUNCIONAIS

O tempo de resposta é um requisito não funcional crucial: os sistemas embarcados possuem menor capacidade de processamento do que computadores, por isto o processo de criptografia deve ser otimizado para este caso, não consumindo mais do que alguns poucos segundos. Estudos para sites apontam que um usuário frustra-se após 8 a 10 segundos de espera, sendo que ⅓ dos usuários de banda larga desiste após 4 segundos e metade dos usuários de banda normal desiste após 6 segundos (IN USABILITY WE TRUST, 2008).

O tempo máximo de espera proposto é de 8 segundos, e para sua verificação utilizaremos ferramentas de análise de desempenho que analisam códigos para ver quais segmentos demoram mais, além de efetuar testes estatísticos para verificar se o limite está sendo respeitado.

Outro requisito é a segurança. Analisaremos o sistema de criptografia para tentar achar falhas teóricas de segurança.

6. CRITÉRIOS DE ACEITACÃO

A primeira fase da aceitação se constituirá na aprovação da documentação pelo professor orientador, com avaliação individual dos documentos e em seu conjunto.

A segunda fase se dará com ao longo do desenvolvimento, estabelecendo-se pontos de validação onde se verifica se os requisitos estão sendo atingidos.

A aceitação final ocorrerá com testes planejados no Documento de Aceitação, a ser elaborado.

7. REFERÊNCIAS

IN USABILITY WE TRUST. **Page Load Times vs. Conversion Rates**.2008.Disponível em: <http://www.svennerberg.com/2008/12/page-load-times-vs-conversion-rates/>. Acesso em 15 de abril de 2011.

MAO, Wenbo. **Lightweight Micro-Cash for the Internet**. In: Computer Security - ESORICS 96, 4th European Symposium on Research in Computer Security, Rome, Italy, September 25-27, 1996. Proceedings. Volume 1146 of Lectures Notes in Computer Science, pages 15-32, Springer, 1996.

APÊNDICE B – MODELO DE CASOS DE USO

1. INTRODUÇÃO

Este documento contém o Modelo de Casos de Uso do Sistema de Micropagamentos com Assinatura Cega para Aparelhos Celulares.

A especificação do sistema pode ser encontrada no Documento de Especificação de Requisitos de Software.

2. DESCRIÇÃO DE ATORES

Os atores do sistema são:

- Cliente: utiliza o aplicativo para AndroidMicroPag. É o cliente do sistema, carregando em seu celular moedas eletrônicas para efetuar a compra de produtos ou serviços nos comerciantes.
- Comerciante: recebe pagamentos com moedas eletrônicas e valida se o processo de pagamento é válido. Faz depósitos das moedas no Banco para trocar pelo valor real
- Funcionário do Banco: o funcionário do Banco interage com o Sistema Banco para permitir a retirada de moedas eletrônicas, mediante verificação de saldo. Também recebe deposito de moedas eletrônicas e as valida para trocar por dinheiro real.

3. DIAGRAMA DE CASOS DE USO

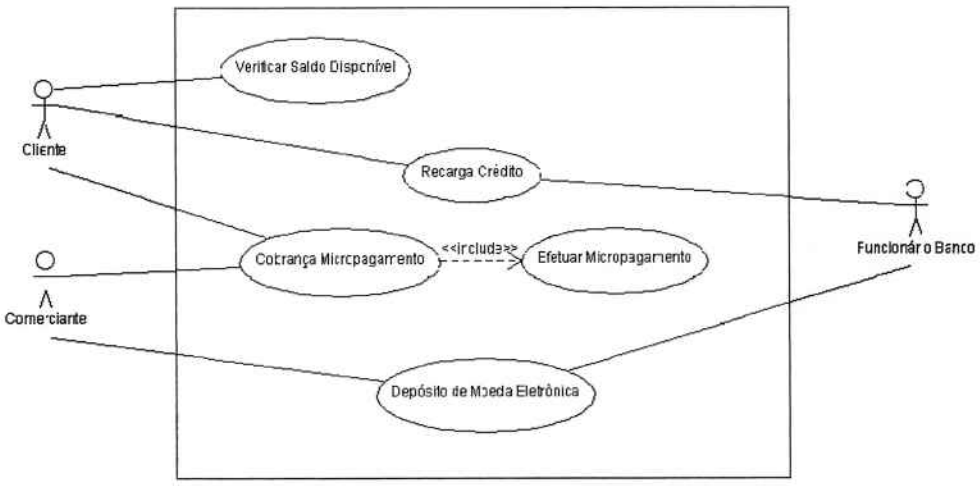


Diagrama 1: Casos de Uso

4. DESCRICÃO DE CASOS DE USO

Caso de uso 1: Recarga de Crédito

Descrição: Este caso de uso descreve o processo de recarga de moedas eletrônicas no celular Android, através do aplicativo MicroPag.

Eventoiniciador: Cliente solicita recarga de crédito no MicroPag.

Atores: Cliente, Funcionário do Banco

Pré-condição: Aplicativo no estado inicial, Sistema Banco no modo recarga.

Sequência de eventos:

1. Funcionário do Banco seleciona o modo recarga no Sistema Banco.
2. Cliente seleciona botão de recarga.
3. Sistema MicroPag exibe tela para entrada de valor de recarga.
4. Cliente escolhe valor e confirma.
5. Banco recebe valor de recarga.
6. Banco verifica saldo.

7. Banco assina e envia moedas eletrônicas para o MicroPag.
8. Aplicativo MicroPag valida moedas.
9. Micropag confirma recebimento.
10. Aplicativo MicroPag exibe novo saldo.

Pós-condição: Recarga de créditos efetuada e sistema no estado inicial.

Extensões: Falha de Comunicação (passos 5, 7, 9): exibe erro. Todo o processo é desfeito.

Sem saldo disponível (passo 6): exibe erro.

Moedas Inválidas (passo 8): exibe erro.

Inclusão: Não há.

Caso de uso 2: Verificar Saldo Disponível

Descrição: Este caso de uso descreve o processo para verificar o saldo no MicroPag.

Evento iniciador: Cliente solicita saldo disponível.

Ator: Cliente.

Pré-condição: Aplicativo no estado inicial.

Sequência de eventos:

1. Cliente seleciona botão de verificação de saldo.
2. Sistema MicroPag exibe na tela saldo disponível.

Pós-condição: Sistema no estado inicial.

Extensões: Não há.

Inclusão: Não há.

Caso de uso 3: Efetuar Micropagamento

Descrição: Este caso de uso descreve o processo para pagar uma compra utilizando moedas eletrônicas.

Evento iniciador: Valor do pagamento na tela do MicroPag.

Atores: Cliente, Comerciante.

Pré-condição: MicroPag no modo pagamento.

Sequência de eventos:

1. Sistema Caixa envia ao Cliente MicroPag o valor da compra.

2. Sistema MicroPag recebe o valor e exibe na tela.
3. Cliente confirma pagamento.
4. Saldo do Cliente é verificado.
5. Moedas eletrônicas são transferidas do Cliente para o Comerciante.
6. Comerciante valida as moedas eletrônicas.
7. Comerciante confirma operação.
8. Saldo do Cliente é debitado.

Pós-condição: Pagamento efetuado.

Extensões: Falha de Comunicação (passos 1,5,7): exibe erro. Todo o processo é desfeito.

Sem saldo (passo 5).

Inclusão: Não há.

Caso de uso 4: Cobrança de Micropagamento

Descrição: Este caso de uso descreve o processo para cobrar um pagamento

Evento iniciador: Seleção da opção de receber pagamento no Sistema Caixa.

Atores: Cliente, Comerciante.

Pré-condição: Sistema Caixa e MicroPag no estado inicial.

Sequência de eventos:

1. Comerciante seleciona o valor a ser pago no Sistema Caixa.
2. Sistema Caixa envia ao MicroPag pedido para autorizar pagamento.
3. MicroPag autoriza inicio do processo de pagamento.
4. Sistema Caixa e MicroPag no modo Pagamento.
5. Processo de recebimento do pagamento.
6. Moedas eletrônicas recebidas são armazenadas no Sistema Caixa.

Pós-condição: Sistema Caixa no estado inicial.

Extensões: Falha de Comunicação (passos 2, 3): exibe erro. Todo o processo é desfeito.

Cliente não autoriza (passo 3): exibe erro.

Inclusão: Efetuar Micropagamento (passo 5)

Caso de uso 5: Depósito de Moeda Eletrônica

Descrição: Este caso de uso descreve o processo para depositar uma moeda eletrônica

Evento iniciador: Comerciante escolhe opção de depositar moedas eletrônicas.

Atores: Funcionário do Banco, Comerciante.

Pré-condição: Sistema Caixa e Sistema Banco no estado inicial.

Sequência de eventos:

1. Funcionário do Banco seleciona modo Depósito.
2. Comerciante seleciona a opção depositar moedas eletrônicas.
3. Comerciante escolhe no Sistema Caixa o valor a ser depositado.
4. Sistema Caixa envia moedas eletrônicas ao Banco.
5. Sistema Banco verifica validade das moedas eletrônicas.
6. Sistema Banco confirma procedimento com sucesso.
7. Sistema Banco troca moedas eletrônicas pelo valor real.
8. Comerciante retira moedas eletrônicas depositadas de seu sistema.

Pós-condição: Moeda depositada, Sistemas Caixa e Banco no estado inicial.

Extensões: Falha de Comunicação (passos 4, 5): exibe erro. Todo o processo é desfeito.

Sem saldo (passo 3): exibe erro.

Moedas inválidas (passo 5): exibe erro

Moeda Duplicada (passo 5): exibe erro. Sistema analisa fraude.

Inclusão: Não há.

5. REFERÊNCIAS

MELNIKOFF, Selma S. S. **Modelagem de Casos de Uso.** São Paulo: EPUSP – PCS, 2011. 59 slides, color. Acompanha texto.

APÊNDICE C – INTERFACE HOMEM COMPUTADOR

1. OBJETIVO DO DOCUMENTO

O objetivo deste documento é apresentar a interface homem-computador (IHC).

Esta versão final do documento apresenta a IHC definitiva.

2. CARACTERIZAÇÃO DO USUÁRIO

Nesta seção são apresentados os usuários do sistema.

Tabela 5: Funcionalidades por Usuário

N	Papel do Usuário	Funçõespermitidas	Frequência de uso	Conhecimento da tarefa
1	Cliente	Recarregar créditos Verificar saldo disponível Autorizar cobrança Efetuar micropagamento	Ocasional	Operacional
2	Comerciante	Cobrança de micropagamento Depósito de moeda eletrônica	Diária	Operacional
3	Funcionário do Banco	Retirada de moedas eletrônicas Receber depósitos de moedas eletrônicas	Diária	Operacional

3. ESTILO DA INTERFACE

O Sistema de Micropagamentos é composto por três subsistemas, dois deles para plataforma Windows e um para plataforma Android.

Para Android, será desenvolvido um aplicativo que é constituído por telas e elementos como botões e *textbox*. Sua interface será *touch*.

Os dois subsistemas para Windows serão *web* baseados em Java.

4. IDENTIFICAÇÃO DOS PRINCIPAIS OBJETOS

4.1 ANDROID

Os principais objetos da interface são menus, botões e texto.

O botão de confirmação de operação tem cor verde, enquanto que o botão de negação tem cor vermelha. O botão que dá continuidade ao funcionamento em caso erro tem cor azul.

Figura 21: Tela de Recarga

Figura 22: Tela de Pagamento

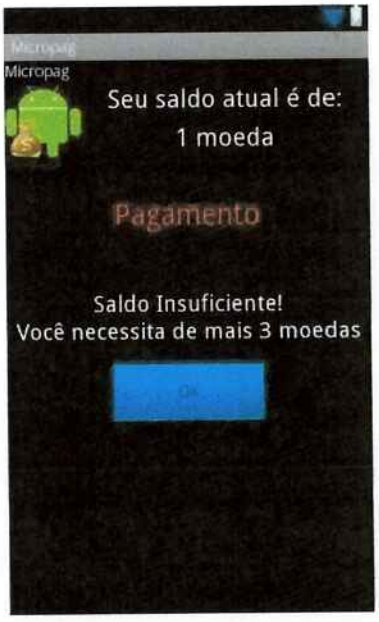


Figura 23: Tela de Saldo Insuficiente

Figura 24: Tela de Erro

4.2 BANCO

Os principais objetos da interface são botões, caixas de texto e *links*.

Os botões tem coloração roxa, para aumentar o contraste com o fundo salmão adotado.

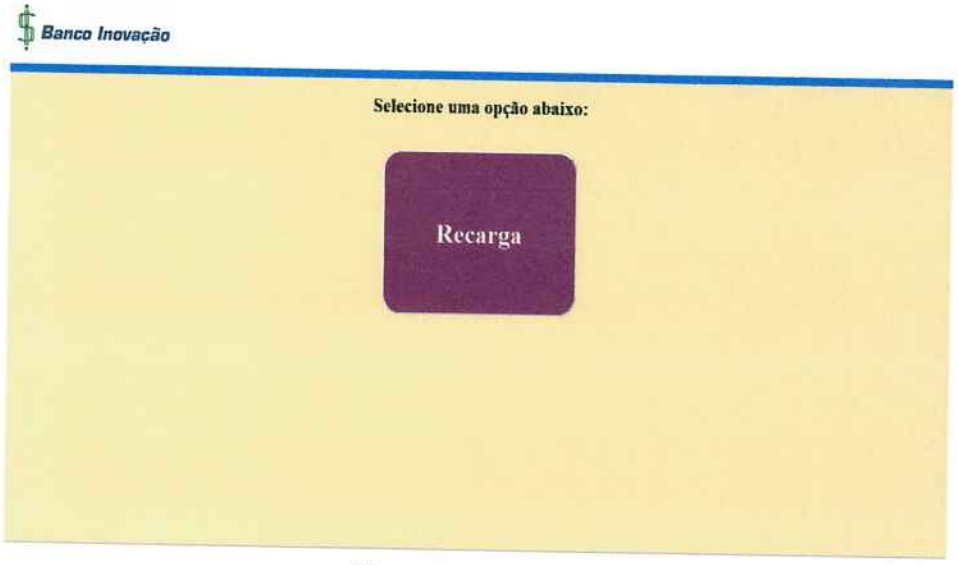


Figura 25: Tela Inicial

Figura 26: Tela de Recarga

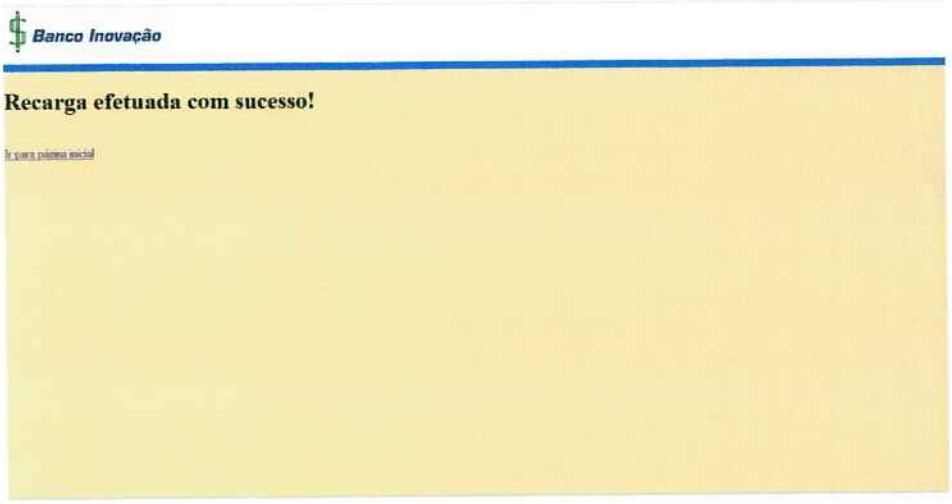


Figura 27: Tela Sucesso Recarga

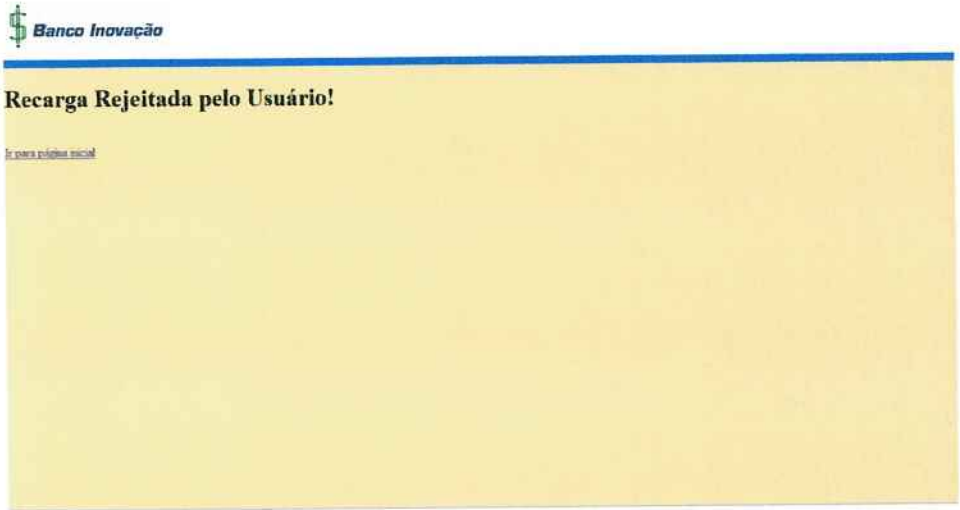


Figura 28: Tela Recarga Rejeitada

Figura 29: Tela Falha Recarga

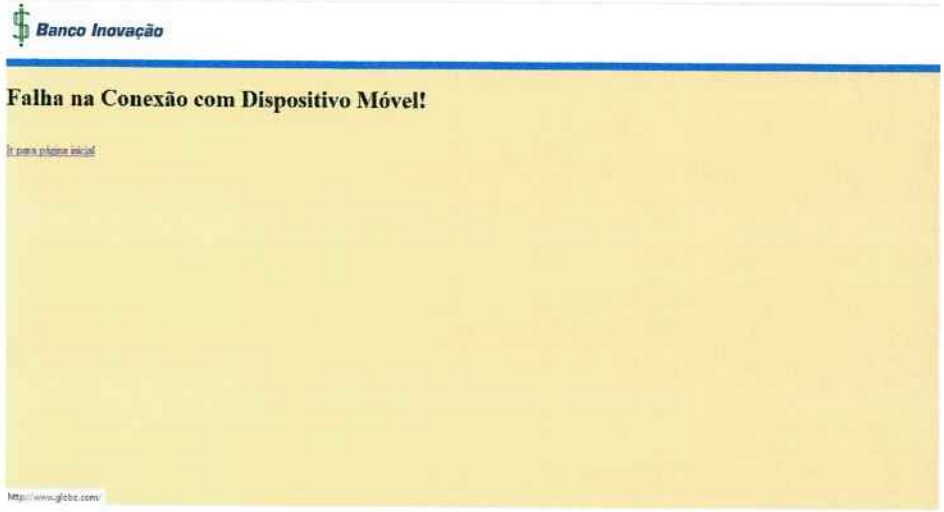


Figura 30: Tela Falha de Conexão

4.3CAIXA

Os principais objetos da interface são botões, caixas de texto e *links*.

Os botões tem coloração roxa, para aumentar o contraste com o fundo salmão adotado.

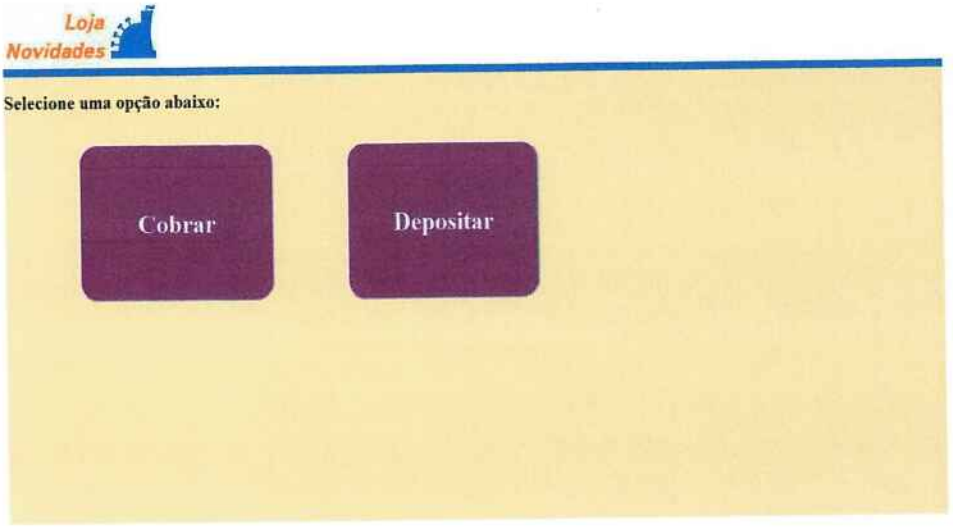


Figura 31: Tela Inicial

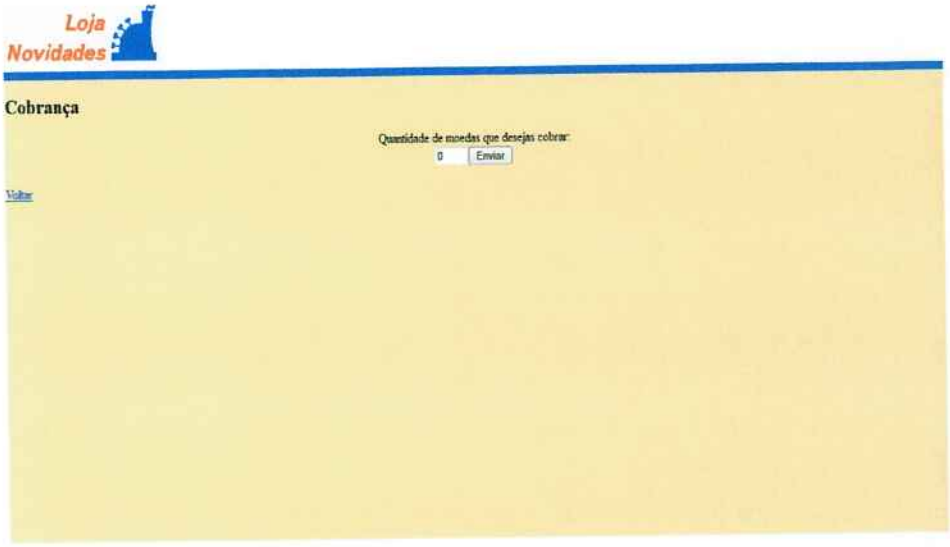


Figura 32: Tela de Cobrança

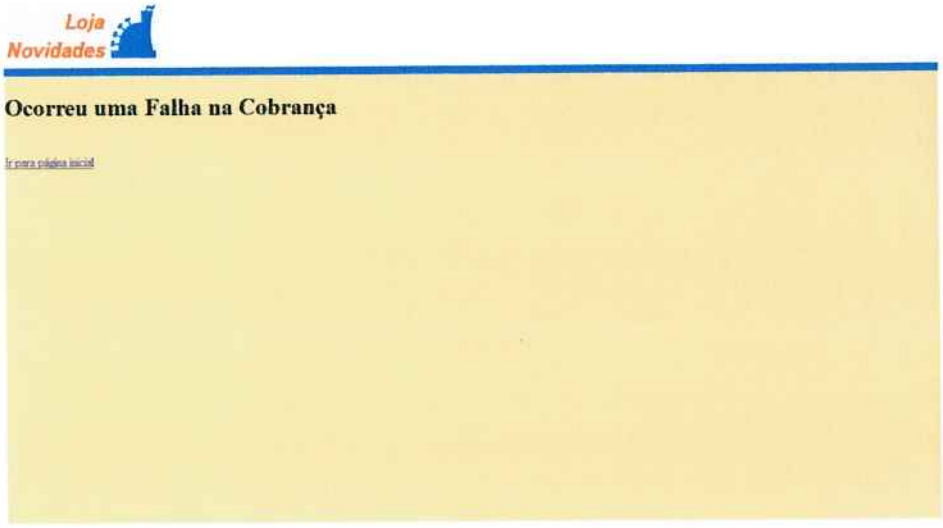


Figura 33:Tela de Falha na Cobrança

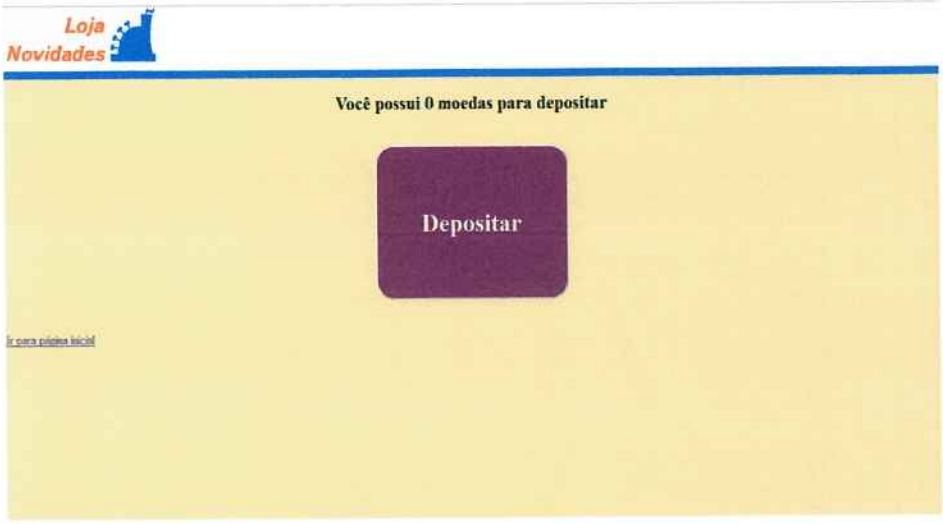


Figura 34: Tela de Depósito

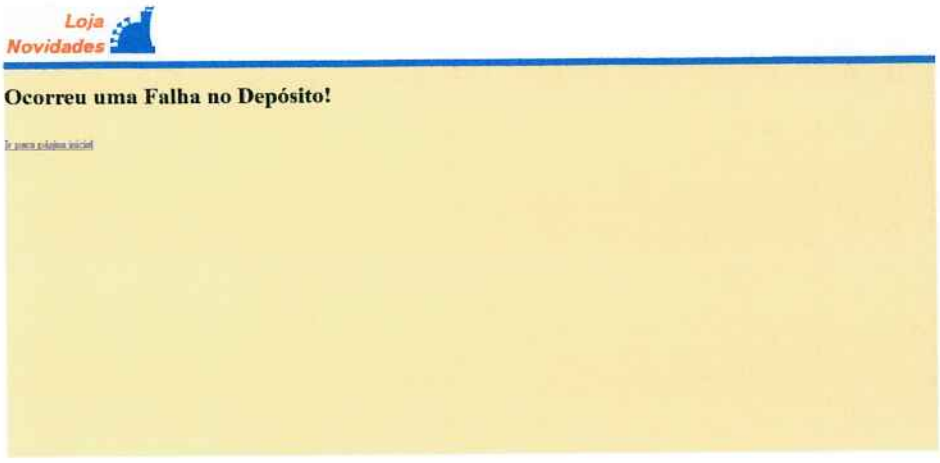


Figura 35: Tela Falha no Depósito

5. PROTÓTIPO DE NAVEGAÇÃO

5.1 ANDROID

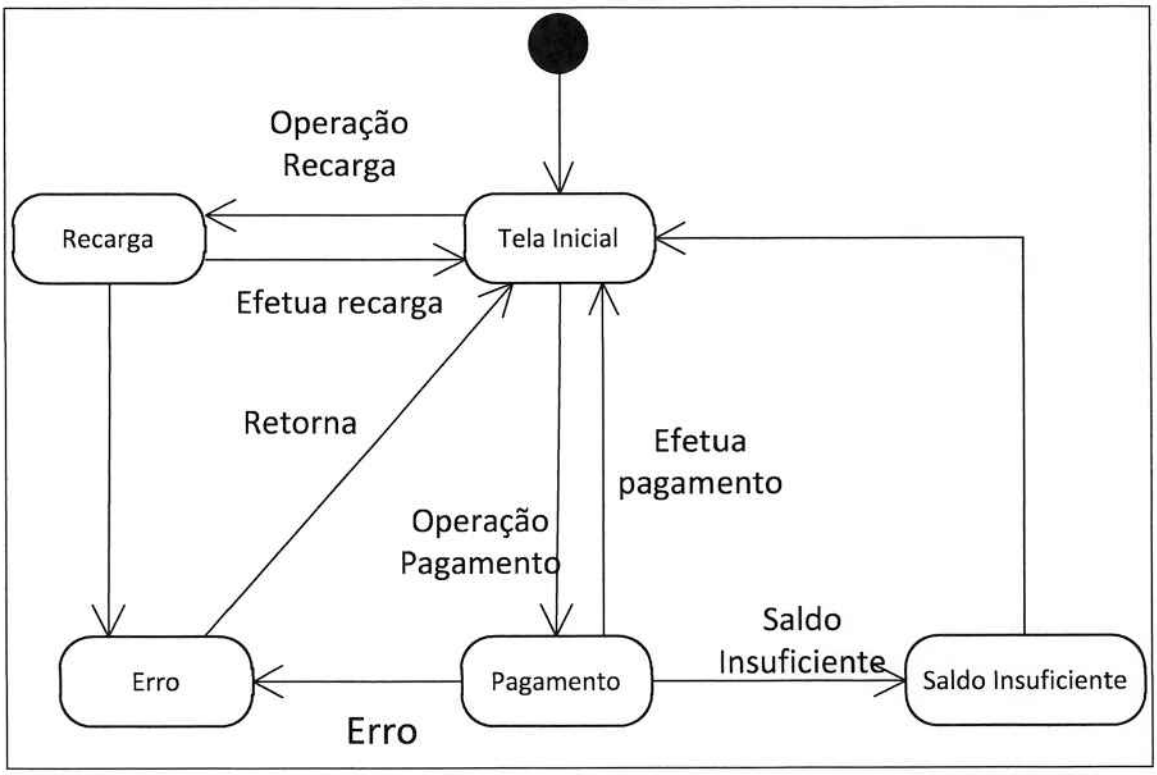


Figura 36: Navegação do Aplicativo Android

5.2 BANCO

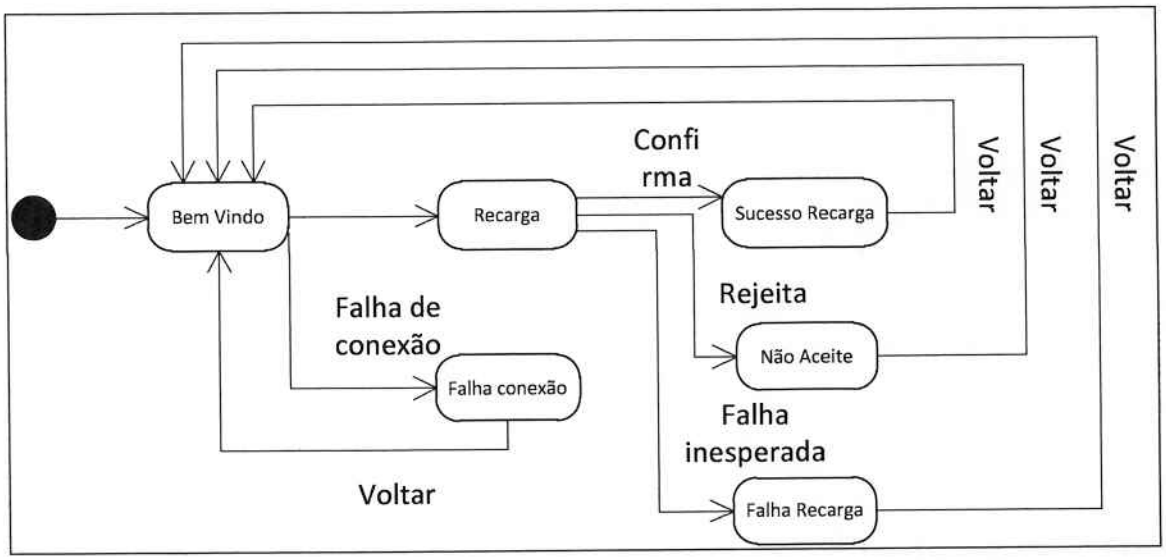


Figura 37: Navegação do Sistema Banco

5.3 CAIXA

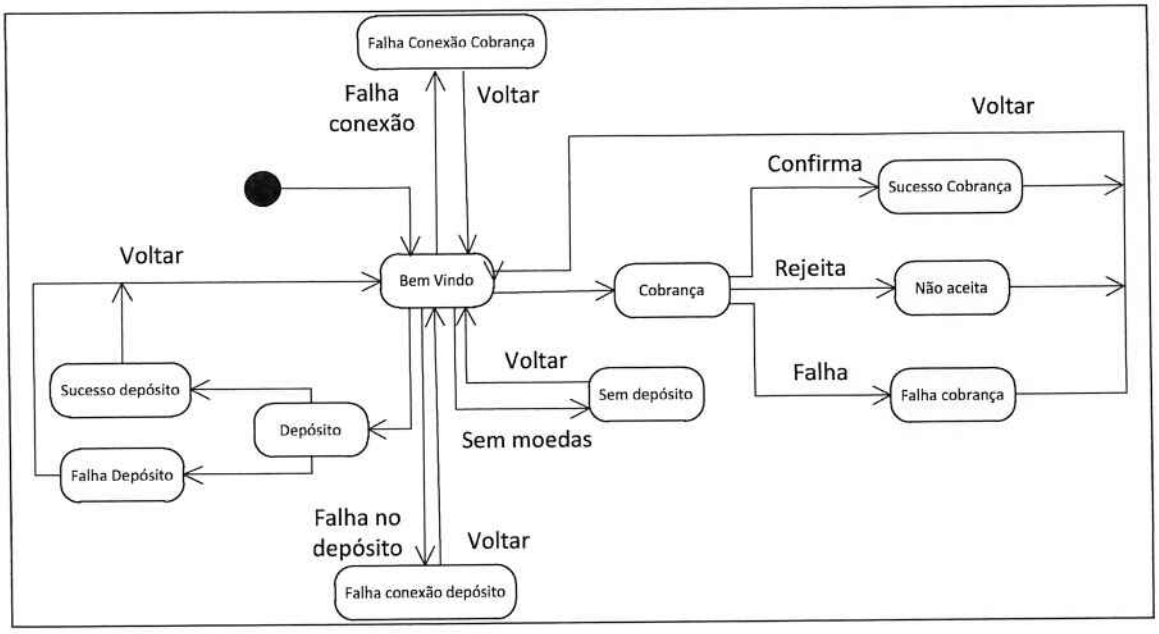


Figura 38: Navegação do Sistema Caixa

6. MENSAGENS DE ERRO

Tabela 6: Mensagens de Erro

Mensagem	Situação
Android: "Saldo insuficiente! Você necessita de mais _ moedas."	Quando é feito uma tentativa de pagamento, e o saldo atual é inferior ao valor da transação.
Android: "Um erro ocorreu. Operação não foi concluída."	Quando ocorre algum erro na operação de recarga ou pagamento.
Android: "Conexão com dispositivo perdida."	Quando a conexão bluetooth com o dispositivo (modulo Caixa ou Banco) é perdida.
Android: "Não é possível se conectar."	Quando o Android não consegue se conectar a um dispositivo.
Banco: "Ocorreu uma Falha na Conexão com Dispositivo Móvel!"	Quando há alguma falha na hora de se estabelecer uma conexão com o dispositivo móvel.
Banco: "Ocorreu uma Falha na Recarga!"	Quando há alguma falha durante a troca de mensagens entre o Sistema Banco e o dispositivo móvel.
Caixa: "Ocorreu uma Falha na Conexão com Dispositivo Móvel!"	Quando há alguma falha na hora de se estabelecer uma conexão com o dispositivo móvel.
Caixa: "Payment Recebido Inválido!"	Quando o Sistema Caixa verifica que a mensagem contendo um payment não possui um valor válido.
Caixa: "A Operação Terminou Sem Ter Recebido Todas as Moedas!"	Quando o Sistema Caixa recebe uma mensagem do dispositivo móvel informando que terminou de enviar o pagamento sem ter enviado todas as moedas necessárias.
Caixa: "Ocorreu uma Falha na Cobrança!"	Quando há alguma falha durante a troca de mensagens entre o Sistema

	Caixa e o dispositivo móvel.
Caixa: "Ocorreu uma Falha ao Recuperar Quantidade de Moedas Disponíveis para Depósito!"	Quando o Sistema Caixa não consegue recuperar a quantidade de moedas que ainda não foram depositadas.
Caixa: "Ocorreu uma Falha no Depósito!"	Quando há alguma falha durante a troca de mensagens entre o Sistema Caixa e o Sistema Banco.

APÊNDICE D – MODELO DE CLASSES

1. OBJETIVO DO DOCUMENTO

Este documento descreve o Modele de Classes do Sistema Micropagamentos com Assinatura Cega para Celulares.

A seção 2 apresenta uma breve descrição do sistema.

A seção 3 apresenta o Diagrama de Classes

A seção 4 descreve as classes, seus atributos e métodos.

2. DESCRIÇÃO DO SISTEMA

O Sistema de Micropagamentos com Assinatura Cega para Celulares é dividido em três subsistemas: MicroPag (aplicativo Android), Sistema Caixa e Sistema Banco. Este sistema permite a retirada de moedas eletrônicas, seu uso em comércios e o depósito em bancos para trocar por dinheiro real. Busca garantir a anonimidade e a segurança.

Maiores informações podem ser encontradas no documento de Especificação de Requisitos.

3. DIAGRAMA DE CLASSES

O Diagrama de Classes se encontra em anexo.

4. DESCRIÇÃO DOS ELEMENTOS

Pacote Assinatura Schnorr: Classes que implementam o algoritmo de assinatura digital de Schnorr, tanto a clássica quanto a assinatura cega. Seus atributos e métodos estão relacionados com os algoritmos de criptografia.

Pacote Android: Pacote com as classes necessárias para o MicroPag. Implementa classes que permitem a retirada de moedas eletrônicas e classes de comunicação. Seus atributos são Moedas eletrônicas armazenadas e a chave privada do usuário, e os métodos são para receber e gastar moedas eletrônicas, utilizando assinaturas e verificações de assinaturas.

Pacote Comerciante: implementa as classes necessárias para simular o caixa de um comerciante. Seus atributos são a chave privada do comerciante e também moedas eletrônicas usadas. Métodos permitem receber e depositar moedas eletrônicas.

Pacote Banco: classes que implementam o sistema banco. Possui moedas depositadas e a chave privada do banco. Métodos para assinar cegamente moedas e para verificar validade de depósitos, detectando fraudes.

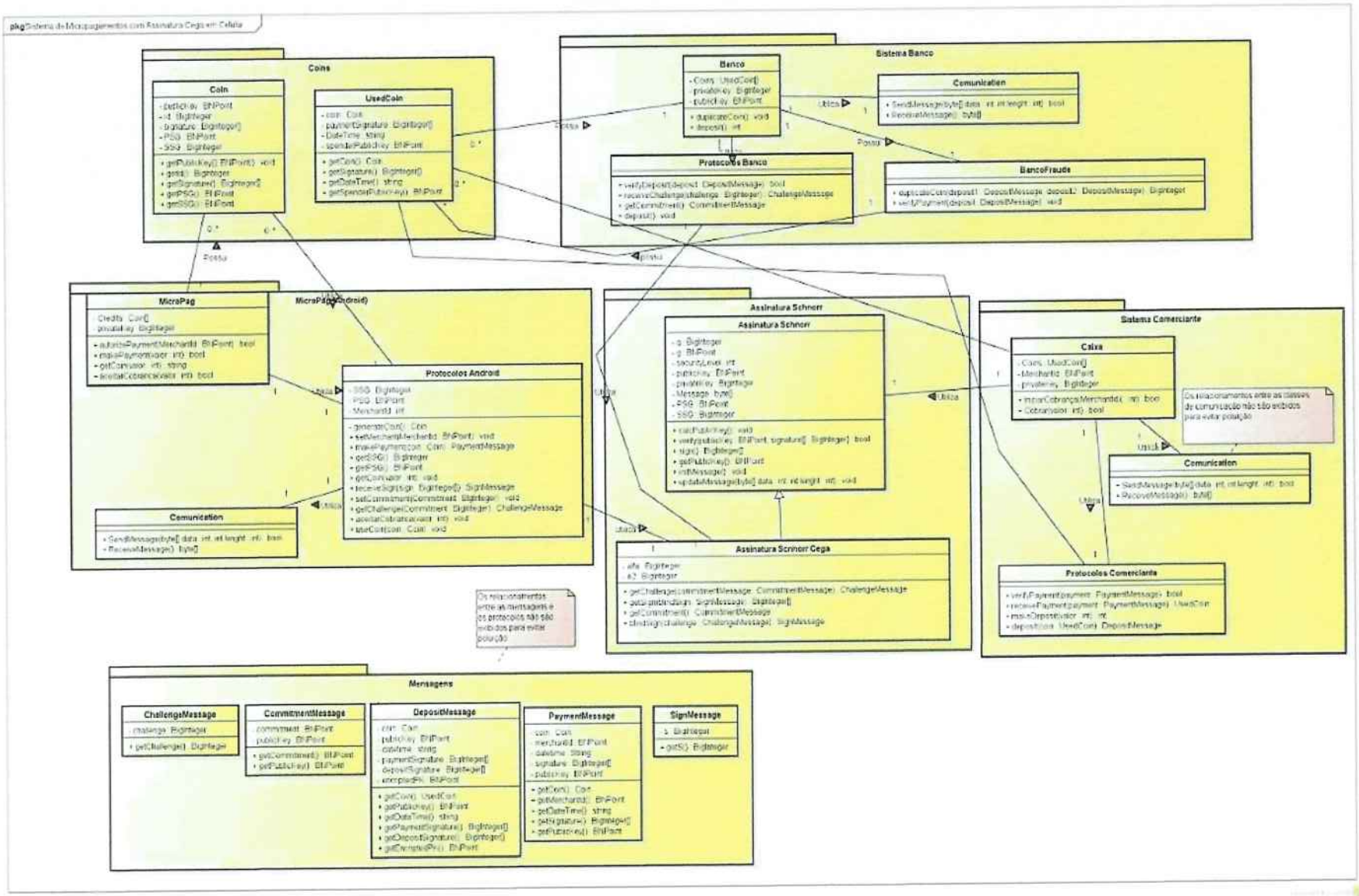
Os pacotes dos subsistemas utilizam o pacote Assinatura Schnorr.

O pacote Mensagens contém as mensagens trocadas pelos subsistemas de acordo com o protocolo de moeda eletrônica.

Os pacotes Android API e BNPairings estão omitidos no diagrama. O pacote BNPairings, de autoria do Prof. Doutor Paulo S. L. M. Barreto e de Geovandro C. C. F. Pereira, implementa as operações de curvas elípticas utilizadas.

5. REFERÊNCIAS

MELNIKOFF, Selma S. S. **Modelagem de Classes.** São Paulo: EPUSP – PCS, 2009. 68 slides, color. Acompanha texto.


pkg Sistema de Micropagamentos com Assinatura Cega em Celular
Coins
Coin
UsedCoin
Sistema Banco
Banco
Comunication
Protocolos Banco
BancoFraude
MicroPag
MicroPag (Android)
Protocolos Android
Comunication
Assinatura Schnorr
Assinatura Schnorr
Assinatura Schnorr Cega
Sistema Comerciante
Caixa
Comunication
Protocolos Comerciante
Os relacionamentos entre as classes de comunicação não são exibidos para evitar poluição
Os relacionamentos entre as mensagens e os protocolos não são exibidos para evitar poluição
Mensagens
ChallengeMessage
CommitmentMessage
DepositMessage
PaymentMessage
SignMessage

APÊNDICE E – MODELO DINÂMICO

1. OBJETIVO DO DOCUMENTO

Este documento descreve o Modelo dinâmico do Sistema de Micropagamentos com Assinatura Cega para Celulares.

A seção 2 descreve suscintamente o sistema.

A seção 3 relaciona os Diagramas de Colaboração ou de Sequência do Sistema.

A seção 4 apresenta os Diagramas de Colaboração ou de Sequência do Sistema.

A seção 5 apresenta os Diagramas de Estado do sistema.

2. DESCRIÇÃO DO SISTEMA

A descrição do sistema pode ser encontrada no documento Especificação de Requisitos do Sistema.

3. RELAÇÃO DOS DIAGRAMAS DO MODELO DINÂMICO

Os principais diagramas de sequência do Sistema são: Retirada de Dinheiro, Pagamento e Depósito. Os modelos omitem a classe de comunicação por considerá-la transparente a estes processos.

4. DIAGRAMAS DE SEQUÊNCIA

Diagrama de Sequencia de Retirada de Moedas Eletrônicas:

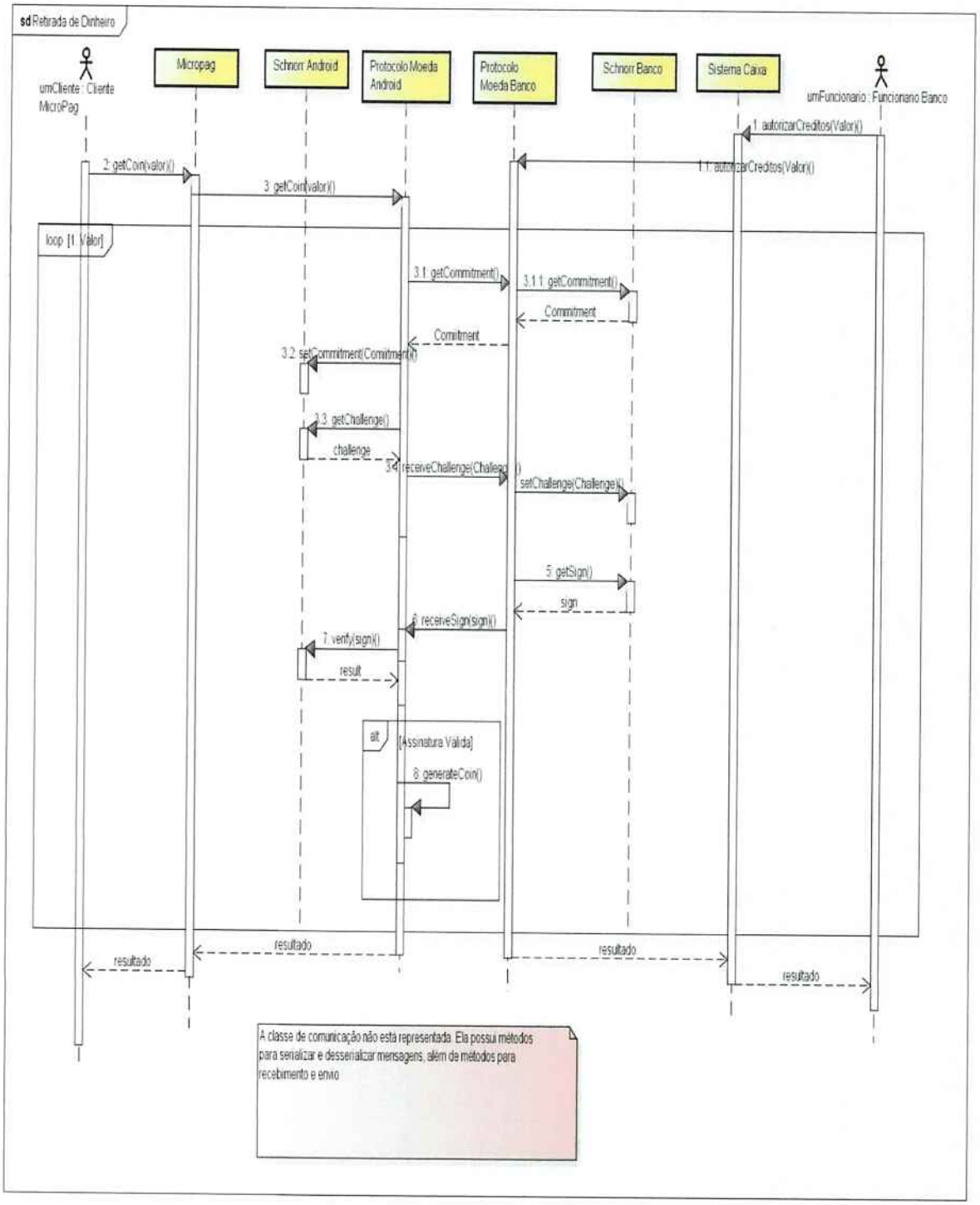


Diagrama 2: Retirada de Moeda

Diagrama de Sequência de Pagamentos com Moeda Eletrônica:

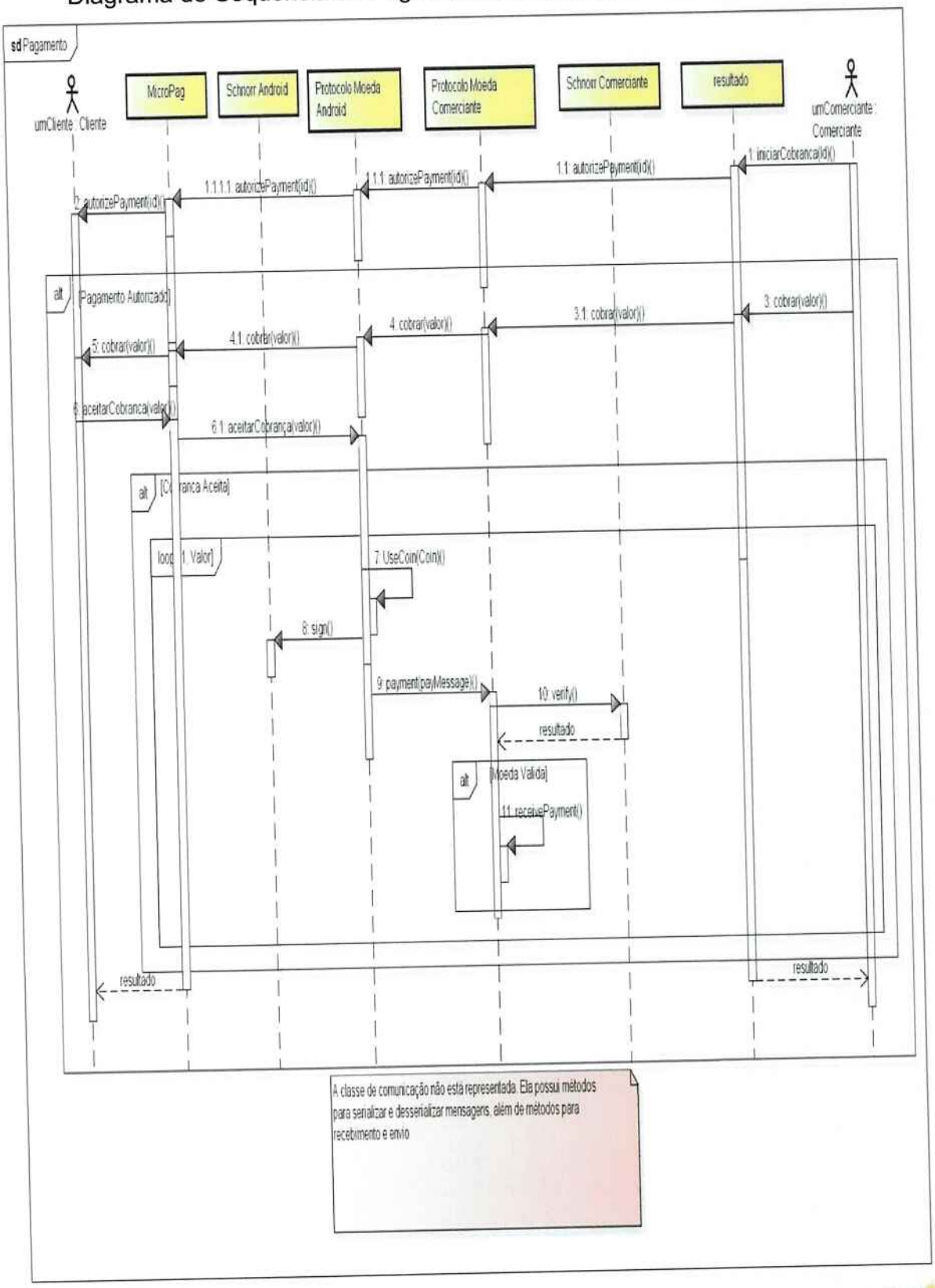


Diagrama 3: Pagamento de uma Moeda

Diagrama de Sequência de Depósito de uma Moeda Eletrônica:

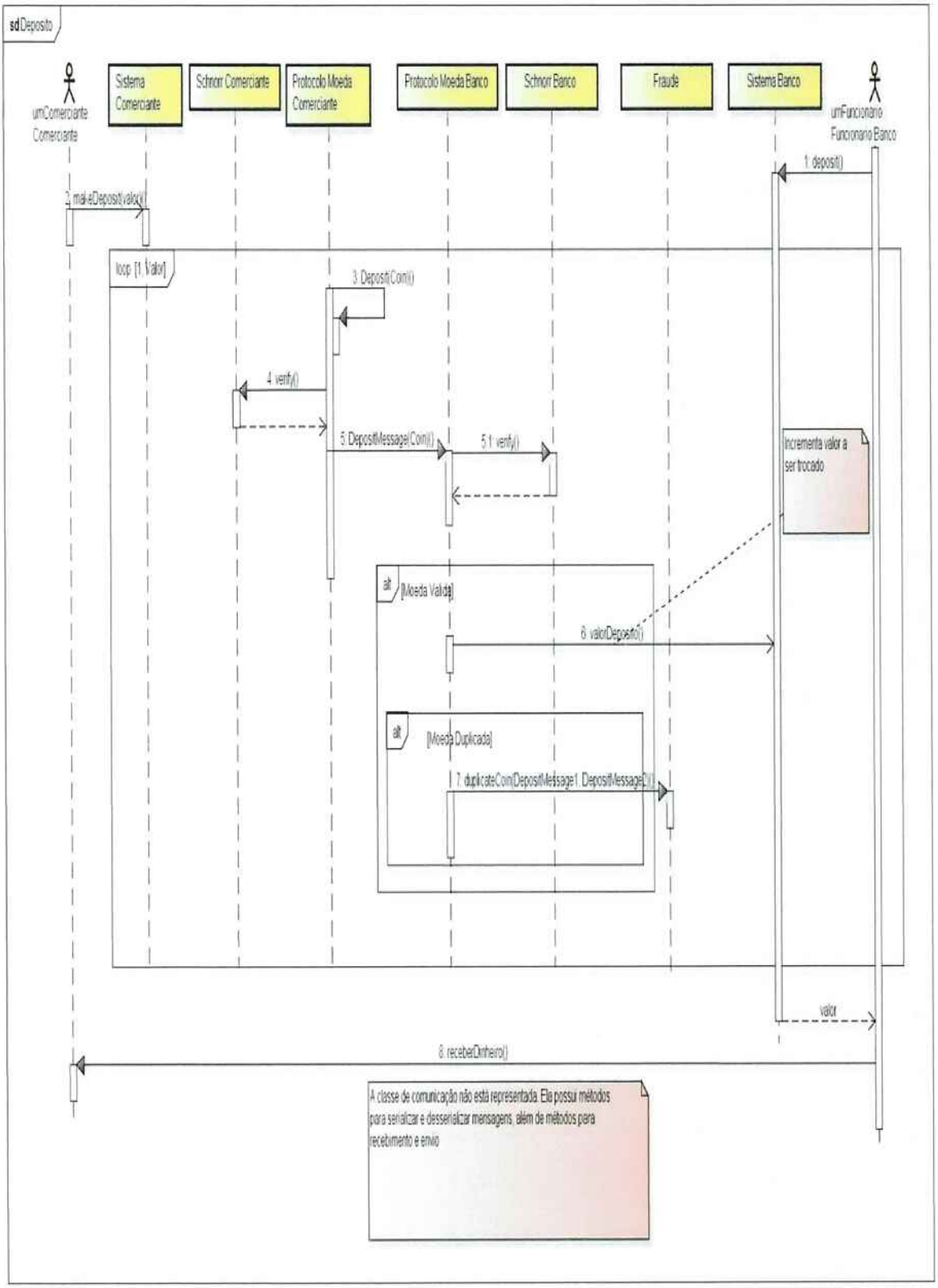


Diagrama 4: Depósito de uma Moeda

5. DIAGRAMAS DE ESTADO

O diagrama de estado mais complexo são os da Assinatura de Schnorr Cega para os subsistemas Android e Sistema Banco. Os seus estados se interligam, conforme pode ser observado nos diagramas abaixo:

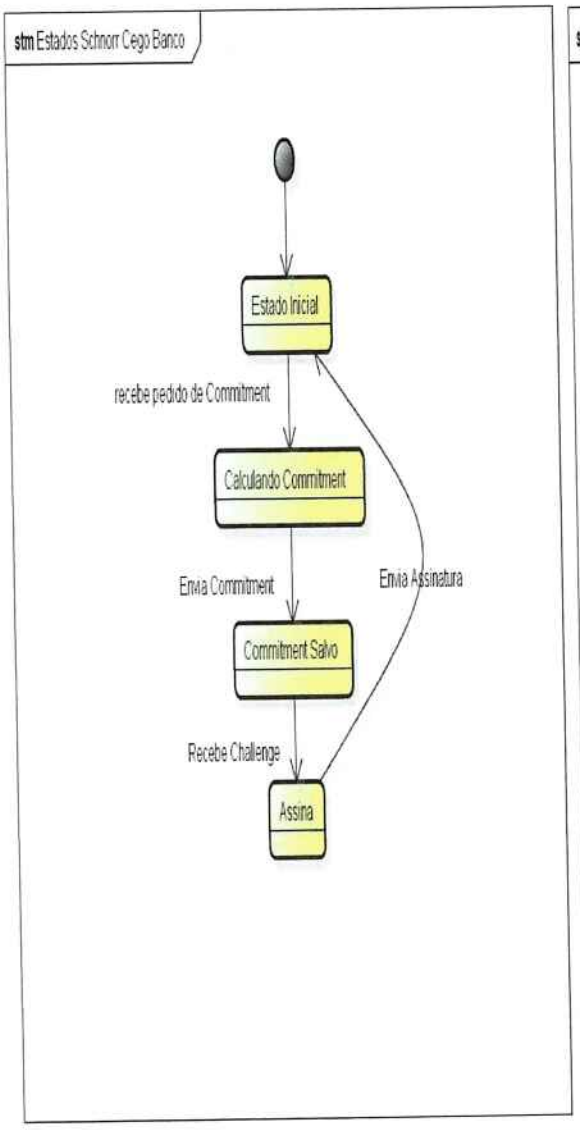


Diagrama 5: Estados do Objeto Schnorr Cego Banco

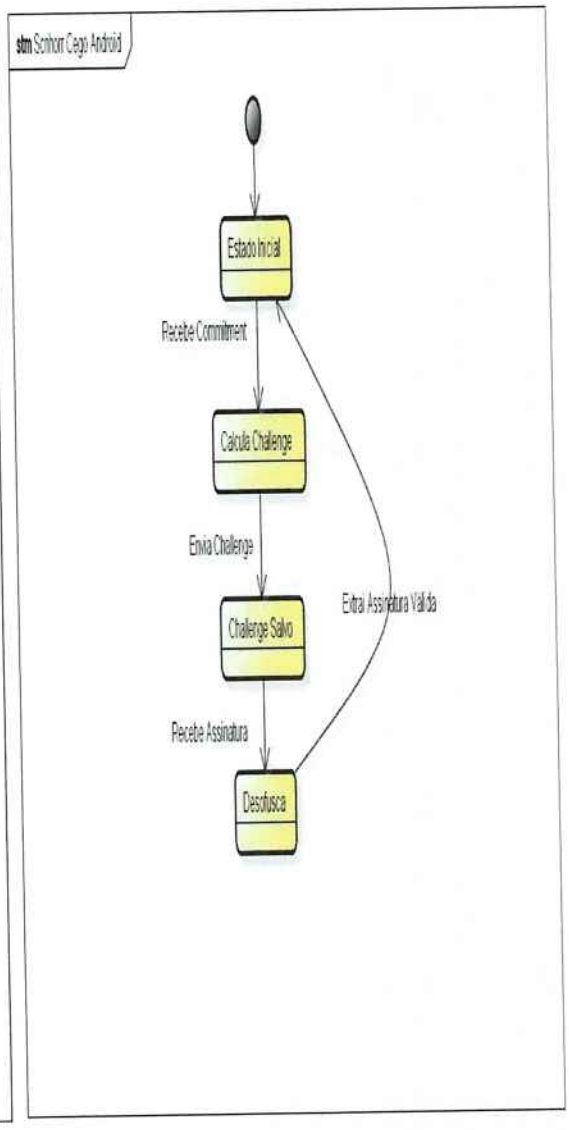


Diagrama 6: Estados do Objeto Schnorr Cego Android

6.REFERÊNCIAS

MELNIKOFF, Selma S. S. **Modelagem Dinâmica.** São Paulo: EPUSP – PCS, 2009. 76 slides, color. Acompanha texto.

BEZERRA, Eduardo. Princípios de Análise e Projetos de Sistemas com UML. 2ª ed. Rio de Janeiro: Ed. Campus, 200. 365p.

APÊNDICE F – DESCRIÇÃO DAS INTERFACES DE COMUNICAÇÃO

1. OBJETIVO DO DOCUMENTO

Este documento descreve as mensagens trocadas entre os módulo do sistema.

A seção 2 contém descrições sucintas das mensagens que são trocadas pelo protocolo.

2. MENSAGENS

- ChallengeMessage: mensagem que contém um *challenge* – um desafio ao Banco;
- SignMessage: mensagem contendo a assinatura do banco, que será utilizada para criar a assinatura cega da moeda;
- DepositMessage: mensagem de depósito, contendo a moeda sendo depositada, a chave pública do Comerciante, a assinatura do pagamento, a assinatura do depósito e a chave pública do cliente;
- PaymentMessage: mensagem de pagamento, contém a moeda que está sendo utilizada no pagamento, o ponto da curva utilizada pelo Comerciante para criptografar os dados, um *timestamp* e a chave pública de quem faz o pagamento;
- CommitmentMessage: mensagem que contém o ponto da Curva Elíptica utilizada no processo de criptografia;
- GenericMessage: mensagem genérica, que contém um *header*, identificando seu tipo, e um conteúdo, que pode ser variável. É utilizada para transmitir informações como confirmação, negação, fim de transmissão e erro;

APÊNDICE G – ESPECIFICAÇÂO DA ARQUITETURA

1. OBJETIVO DO DOCUMENTO

O objetivo deste documento é descrever a arquitetura do sistema, através da arquitetura de software, arquitetura de hardware e a alocação dos elementos de software na arquitetura de hardware.

A arquitetura de software é representada através de *packages* e as dependências entre eles. A arquitetura de hardware é representada através de Diagramas de Implantação. A alocação dos elementos de software no hardware é representada através da distribuição dos *packages* nos Diagramas de Implantação.

2. VISÃO GERAL DO DOCUMENTO

A seção 3 apresenta a descrição do sistema.
A seção 4 especifica a plataforma de desenvolvimento.
A seção 5 descreve a arquitetura de software.
A seção 6 descreve a arquitetura de hardware.
A seção 7 descreve a alocação do software em hardware.
A seção 8 descreve o plano de integração.

3. DESCRIÇÃO DO PRODUTO

A descrição do produto pode ser encontrada no documento de Especificação de Requisitos de Software.

4. ESPECIFICAÇÃO DA PLATAFORMA DE DESENVOLVIMENTO

O Sistema de Micropagamentos com Assinatura Cega para Aparelhos Celulares é divido em três subsistemas, sendo dois deles destinados a computadores com Sistema Operacional Windows e um para smartphones Android.

Para os subsistemas, as ferramentas e o ambiente de desenvolvimento adotado são:

- Linguagem de Programação Java, a partir da versão jdk-6u21.
- Biblioteca BNPairings, rev. 13 (19/set/2011).
- Biblioteca para Java que implementa Bluetooth: BlueCove ver. 2.1.0
- Ambiente de desenvolvimento Eclipse, versão Indigo (3.7)
- Gerenciador de versões de código fonte TortoiseSVN, versão 1.6.12.20536.
- AndroidSdk release 12.
- Sistema Operacional Android 2.2 (Froyo)
- Windows 7 Pro SP1 64 bits.
- MySQL versão 5.5

Para comunicação por Bluetooth são utilizadas bibliotecas do Java e do Android.

5. DESCRIÇÃO DA ARQUITETURA DE SOFTWARE

Esta seção contém o diagrama com a arquitetura de software e a descrição dos seus elementos constituintes.

5.1 ARQUITETURA DE SOFTWARE

O Diagrama de Classes com Pacotes pode ser observado no documento Modelo de Classes.

Nesta seção apresentamos a interação entre os diversos pacotes do sistema, inclusive alguns componentes de comunicação e de sistema operacional omitidos no Modelo de Classes.

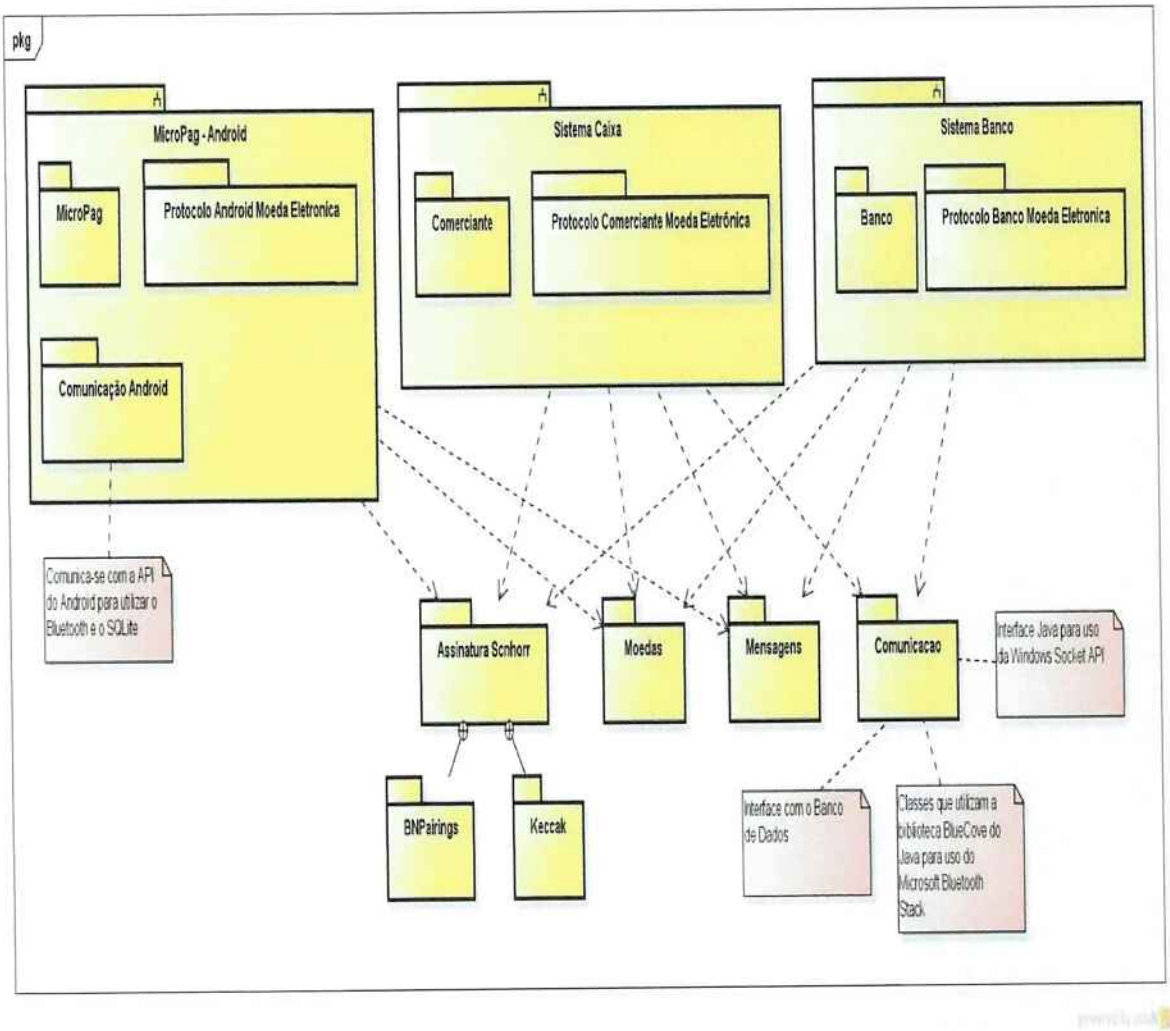


Diagrama 7: Diagrama de Pacotes

5.2 DESCRIÇÃO DO CONTEÚDO DOS PACKAGES

Subsistema MicroPag: aplicativo Android, contém os pacotes:

1. MicroPag: Contém classes que controlam telas e recebem dados do usuário.
2. Protocolo Android Moeda Eletrônica: Classes que implementam as funções do protocolo de moeda eletrônica necessárias ao cliente. Controlam o fluxo de mensagens do protocolo de moeda eletrônica e mantém o saldo disponível. Depende dos pacotes Assinatura Schnorr, Moedas e Mensagens para realizar sua função.
3. Pacote Comunicação Android: Classes que se comunicam com a API do Android para comunicação Bluetooth ou acesso ao SQLite.

Subsistema Caixa:

1. Comerciante: Contém classes que controlam telas e recebem dados do usuário.
2. Protocolo Comerciante Moeda Eletrônica: Classes que implementam as funções do protocolo de moeda eletrônica necessárias ao cliente. Controlam o fluxo de mensagens do protocolo de moeda eletrônica e mantém o saldo disponível. Depende dos pacotes Assinatura Schnorr, Moedas e Mensagens para realizar sua função.

Subsistema Banco:

1. Banco: Contém classes que controlam telas e recebem dados do usuário.
2. Protocolo Banco Moeda Eletrônica: Classes que implementam as funções do protocolo de moeda eletrônica necessárias ao cliente. Controlam o fluxo de mensagens do protocolo de moeda eletrônica e mantém o saldo disponível. Depende dos pacotes Assinatura Schnorr, Moedas e Mensagens para realizar sua função.

Pacote Comunicação: Classe que utiliza a biblioteca Java BlueCove para comunicação Bluetooth pelo Windows Bluetooth Stack. Classe que utiliza o Java para usar o Windows Socket API para comunicação por socket. Não é utilizado pelo Sistema MicroPag - Android.

Pacote Assinatura de Schnorr: Classes que implementam o algoritmo de Assinatura de Schnorr e Assinatura de Schnorr Cega em curvas elípticas. Os

algoritmos utilizam o pacote BNPairings para a criptografia em curvas elípticas e a função de hashKeccak, necessária na assinatura, está no pacote Keccak.

Pacote BNPairings: bibliotecas para pareamento bilinear e operações em curvas elípticas. Desenvolvida pelo Prof. Doutor Paulo S. L. M. Barreto e Geovandro C. C. F. Pereira.

Pacote Keccak: implementação do algoritmo de hashKeccak, cedida pelo Prof. Doutor Paulo S. L. M. Barreto e Geovandro C. C. F. Pereira.

Pacote Moedas: Cria a classe Moeda, que representa uma moeda eletrônica devidamente assinada por um banco. Outra classe é MoedaUsada, criada para representar uma moeda usada e que possui alguns atributos a mais como identidade do comerciante que vendeu o produto e o *datetime* da operação.

Pacote Mensagens: Classes que representam as mensagens que devem ser trocadas pelo protocolo de moeda eletrônica.

Seguiremos o modelo de 3 camadas para implementação:

1. Camada de Interface: contém a interface gráfica e o controle para entrada e saída de dados para o usuário. São as classes dos pacotes MicroPag, Comerciante e Banco que manipulam telas e interagem com o usuário.
2. Camada de Negócios: Controlam o modelo de Negócios. Há uma hierarquia de classes:
   - As classes de controle de negócios dos pacotes de Protocolos gerenciam o saldo de moedas, os protocolos de moeda eletrônica e a utilização das classes de comunicação.
   - As classes de protocolos de moeda eletrônica criam as mensagens necessárias, com as assinaturas digitais e verificações, e as retornam para a camada de controle de negócios.
   - As classes de comunicação são as interfaces para APIs (API do Android, API de Sockets, BlueCove para Windows Bluetooth Stack). São ativadas pelas classes de controle de negócios.

3. Camada de Dados: corresponde ao Banco de Dados MySQL ou as rawmessages recebidas pelo Bluetooth.

6. DESCRIÇÃO DA ARQUITETURA DE HARDWARE

Esta seção contém o diagrama com a arquitetura de hardware e a descrição dos seus elementos constituintes. O diagrama também apresenta a alocação dos componentes nos nós, em camadas.

6.1 DIAGRAMA DE IMPLANTAÇÃO

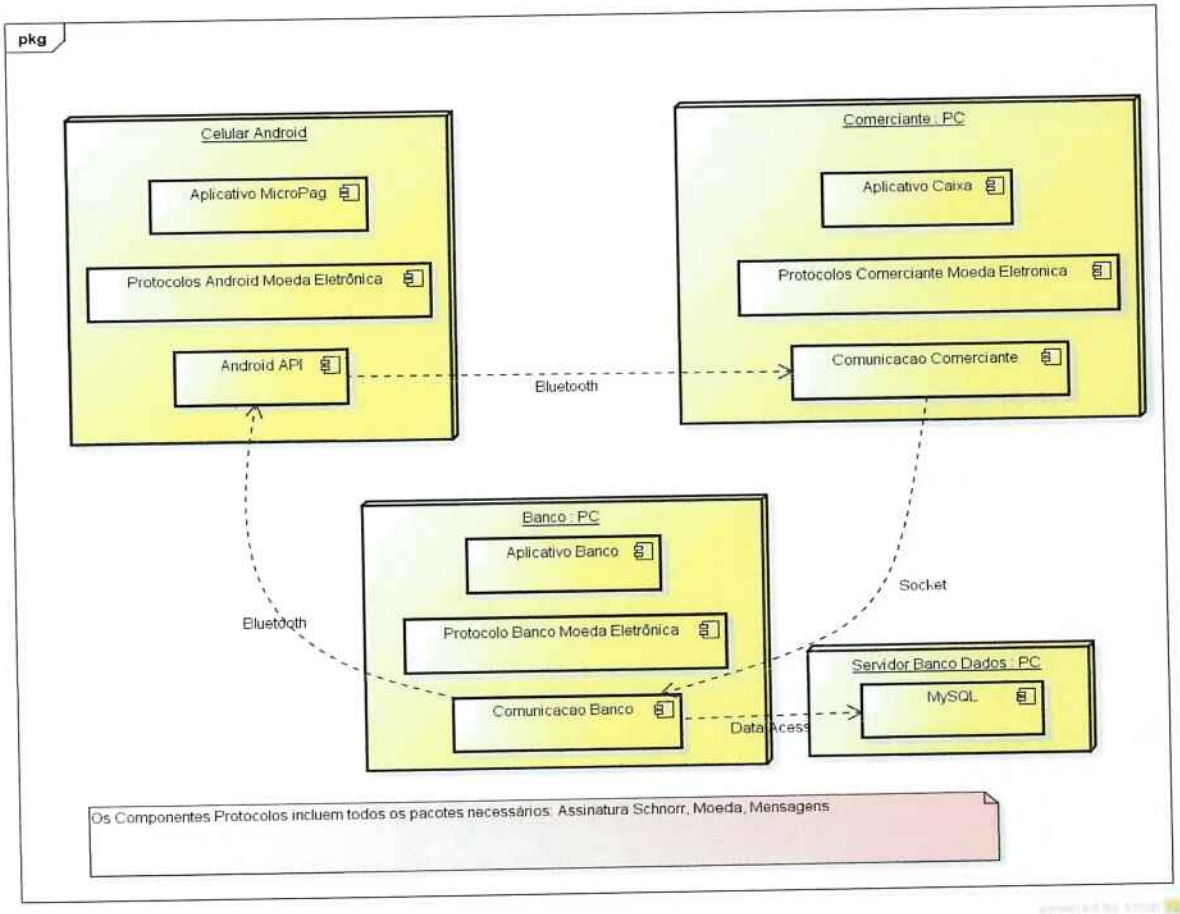


Diagrama 8: Diagrama de Implantação

7. PLANO DE INTEGRAÇÃO

7.1 ESTRATÉGIA DE IMPLEMENTAÇÃO, INTEGRAÇÃO E TESTE

Os protocolos de moeda eletrônica serão implementados inicialmente. São os casos de uso: Recarga de créditos, Pagamento e Depósito.

Os módulo de comunicação e negócios serão os próximos. Uma interface simples mas funcional permitirá a integração e testes de todo o sistema.

Integram-se: protocolos e controle de negócios, comunicação e por fim interface.

Cada fase é testada em sua unidade e as integrações também são testadas.

7.2 AMBIENTE DE TESTE

O ambiente de testes inicial será um simulador em Java que faz uma retirada de moeda, um pagamento e um depósito.

Após o sucesso desta fase, cada parte do protocolo será testada no equipamento final para testes de comunicação.

APÊNDICE H – PLANO DE TESTES E ACEITAÇÃO

1. OBJETIVO DO DOCUMENTO

Este documento visa estabelecer a sequência de testes que demonstram que o sistema funciona conforme especificado. Para cada teste são definidos os casos de testes que consistem de dados de entrada, saída esperada e observações necessárias para a execução do teste.

2. RESUMO DO TESTE DE ACEITAÇÃO

Produto avaliado: **Sistema de Micropagamentos com Assinatura Cega para Aparelhos Celulares**		
Data	*Hora de início*	*Hora de término*

Participantes
Apresentador
Avaliadores
1.
2. .
3. .

Resumo dos testes	
Número total de testes: 4	*Número de cenários :* 1
Número de testes aceitos: _____	*Número de cenários aceitos:* _____
Número de testes não aceitos: _____	*Número de cenários não aceitos:* _____

Resultado do teste de aceitação	
ACEITO	*NÃO ACEITO (Nova aceitação é necessária)*
_____ *Como está*	_____ *Revisões maiores*
_____ *Com revisões menores*	_____ *Reconstruir*
	_____ *Aceitação incompleta*

3. INFRA-ESTRUTURA PARA A ACEITAÇÃO

Configuração dos Equipamentos	*Software necessário*
Celular Motorola Milestone2 Notebook com adaptador Bluetooth com suporte a Microsoft Windows Bluetooth Stack	Android 2.2 Windows 7 Pro 64 bits MySQL Eclipse Java jdk-6u21 BNPairings BlueCove

4. LISTA DE TESTES

Teste no.	**Tipo**	**Requisito testado**
1	F	Retirada de Moeda
2	F	Consulta de Saldo
3	F	Pagamento de Moeda
4	F	Depósito de Moeda

5. TESTES

Teste no. **01**	*Requisito testado:* Retirada de Moeda	
Funcional: ☒ *Não funcional:* ☐	*Observações:*Caso de Uso para a retirada de uma moeda eletrônica	
Sequência para verificação do caso comum	*Comportamento esperado do software*	*Testado*
1. Funcionário do Banco Autoriza retirada de uma moeda eletrônica	*Sistema Banco no modo Recarga de Créditos*	
2. Cliente Android inicia a retirada uma moeda	*Android no modo Recarga de Créditos*	
3.Mensagem de Sucesso é exibida no cliente Android	*Android recebeu os créditos*	
4.		
5.		
6.		
7		
Sequência para verificação de exceções	*Comportamento esperado do software*	
1. Valor de moedas a ser retirado incorreto entre banco e Android	*É exibido erro nos sistemas*	
2. Durante a comunicação para a retirada de moeda o sinal é interrompido	*Erro é exibido.*	
3.		
4.		
5.		
6.		
7.		
☐ *Aprovado*	*Comentários*	
☐ *Reprovado*		
Cientes: *Avaliadores*	*Apresentador*	

Teste no. **02**	*Requisito testado:* Consulta de Saldo
Funcional: ☒ *Não funcional:* ☐	*Observações:*Caso de Uso para a consulta de saldo de moedas eletrônicas no Cliente MicroPag (Android)

Sequência para verificação do caso comum	*Comportamento esperado do software*	*Testado*
1. Cliente seleciona opção Saldo na tela principal do Aplicativo Micropag	*MicroPag exibe o saldo atual*	
2.		
3.		
4.		
5.		
6.		
7		

Sequência para verificação de exceções	*Comportamento esperado do software*	
1.		
2.		
3.		
4.		
5.		
6.		
7.		

☐ *Aprovado*	*Comentários*
☐ *Reprovado*	

Cientes:

Avaliadores *Apresentador*

Teste no. **03**	*Requisito testado:* Pagamento de uma moeda eletrônica
Funcional: ☒ *Não funcional:* ☐	*Observações:*Caso de Uso para o pagamento de uma moeda eletrônica de um Cliente MicroPag para o Sistema Comerciante.

Sequência para verificação do caso comum	*Comportamento esperado do software*	*Testado*
1. Comerciante pede o início de operação de pagamento de um cliente	*Sistema exibe a informação aguardando autorização*	
2. Cliente MicroPag aceita início do processo de pagamento.	*MicroPag e Sistema Comerciante no modo Pagamento*	
3.Comerciante insere o valor 1 e confirma	*Valor é exibido na tela do Comerciante, que tenta espera autorização do Cliente MicroPag.*	
4. Cliente recebe o valor e confirma	*MicroPag e Cliente estão no modo Trocando Moedas*	
5. Procedimento com Sucesso	*Mensagem de Sucesso é exibido nos dois sistemas.*	
6.		
7		
Sequência para verificação de exceções	*Comportamento esperado do software*	
1. Cliente não aceita pagamento ou valor	*Sistema exibe mensagem de não aceitação. Volta ao estado inicial.*	
2. Erro de comunicação durante a troca de moeda.	*Exibe erro.*	
3.		
4.		
5.		
6.		
7.		

☐ *Aprovado*	*Comentários*
☐ *Reprovado*	

Cientes:

Avaliadores *Apresentador*

Teste no. **04**	*Requisito testado:* Depósito de uma moeda eletrônica
Funcional: ☒ *Não funcional:* ☐	*Observações:*Caso de Uso para o depósito de uma moedas eletrônicas do Sistema Caixa para o Sistema Banco

Sequência para verificação do caso comum	*Comportamento esperado do software*	*Testado*
1. Funcionário do Banco ativa modo depósito	*Sistema Banco no modo depósito*	
2. Comerciante seleciona a opção de depósito	*Sistema Comerciante pergunta valor*	
3. Comerciante seleciona 1 moeda e confirma	*Sistema Comerciante se comunica com Sistema Banco para depositar valor*	
4. Sucesso	*Mensagens são exibidas nos dois sistemas*	
5.		
6.		
7		
Sequência para verificação de exceções	*Comportamento esperado do software*	
1. Erro de comunicação	*Exibe erro*	
2.		
3.		
4.		
5.		
6.		
7.		

☐ *Aprovado*	*Comentários*
☐ *Reprovado*	

Cientes:

Avaliadores *Apresentador*

5. CENÁRIO DE TESTE GLOBAL

Teste completo com retiradas de moedas, pagamentos e depósitos.

1. Retirar um número de moedas eletrônicas maior do que um.
2. Fazer uma compra dentro do saldo disponível, com valor de pagamento maior do que um.
3. Depositar um número de moedas maior do que um.

Obs: fazer testes desligando o Bluetooth no meio da comunicação. Erros de comunicação no processo devem resultar no cancelamento de toda a operação.

6. REFERÊNCIAS

MELNIKOFF, Selma S. S. **Teste de Software.** São Paulo: EPUSP – PCS, 2011. 39 slides, color. Acompanha texto.