ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Departamento de Engenharia de Computação e Sistemas Digitais

PCS 2040 - PROJETO DE FORMATURA I

4º Módulo Acadêmico, 2008.
Turma – Cooperativo 08

Aplicativo Seguro de SMS

RELATÓRIO FINAL DA ESPECIFICAÇÃO

Equipe:	Eduardo de Souza Cruz, eduardo.cruz@poli.usp.br , tel: (11) 8584-1768
	Geovandro Carlos F. Pereira, geovandro.pereira@poli.usp.br, tel: (11) 7474-9679
	Rodrigo Rodrigues da Silva, rodrigo.silva1@poli.usp.br , tel: (11)8241-0277
Orientador:	Prof. Dr. Paulo S. L. M. Barreto, email: pbarreto@larc.usp.br

Índice:

1. OBJETIVO

O software a ser desenvolvido será designado por "Aplicativo Seguro de SMS", ou, em uma forma abreviada e internacionalizada mais adequada ao mercado, "Secure-SMS".

O objetivo do software é prover uma camada de segurança a nível de aplicação para mensagens SMS em redes de telefonia móvel. O software deverá assinar, cifrar, decifrar e verificar mensagens enviadas por SMS, de forma a garantir a identidade do remetente, e garantir que ela somente poderá ser lida pelo destinatário real.

O software será aplicável em áreas que requerem segurança da informação que trafega nas redes de telefonia móvel. Alguns exemplos são aplicações militares, bancárias, comunicação pessoal sigilosa e comércio eletrônico.

2. JUSTIFICATIVA

A motivação para o projeto reside na ausência de implementações totalmente seguras e facilmente usáveis envolvendo transações em SMS. Nos dias atuais não é seguro enviar informações confidenciais por SMS, visto que a informação pode ser visualizada por intermediários, por exemplo, um funcionário malicioso trabalhando na operadora celular.

Implementando a camada de segurança em mensagens SMS, poderíamos inclusive realizar transações bancárias por esse meio, ou receber informações confidenciais sem o risco de serem interceptadas.

Como novidade no mercado, seremos pioneiros no uso da encriptação e assinatura baseada em identidades, ou “Identity-Based Encryption”. Essa técnica permite a escolha da chave pública a ser usada. No caso de SMS, pode-se utilizar o próprio telefone do usuário como chave pública.

Com essa técnica, introduz-se também a vantagem de não haver necessidade de um "catálogo" das chaves públicas dos usuários, bastando um usuário saber o número de telefone para o qual quer enviar a mensagem.

A rede GSM (Global System for Mobile Comunnication), sobre a qual as mensagens SMS trafegam, usa o mecanismo “store-and-forward” que é similar ao serviço SMTP de emails. Em vez de servidores de email, são usados centros de SMS (SMSC) que armazenam as mensagens SMS antes de serem enviadas para o fornecedor de serviços (operadora) ou para outro SMSC.

Embora as conexões entre um SMSC e os nós da rede GSM sejam protegidas por túneis VPN, as mensagens SMS ficam armazenadas em aberto no SMSC. Isto significa que os operadores ou alguém que invada o sistema pode visualizar todas as mensagens SMS que estão no SMSC. Muitos SMSC também retêm cópias das mensagens SMS para auditoria, fatura e discussão de propostas.

3. ESPECIFICAÇÃO FUNCIONAL

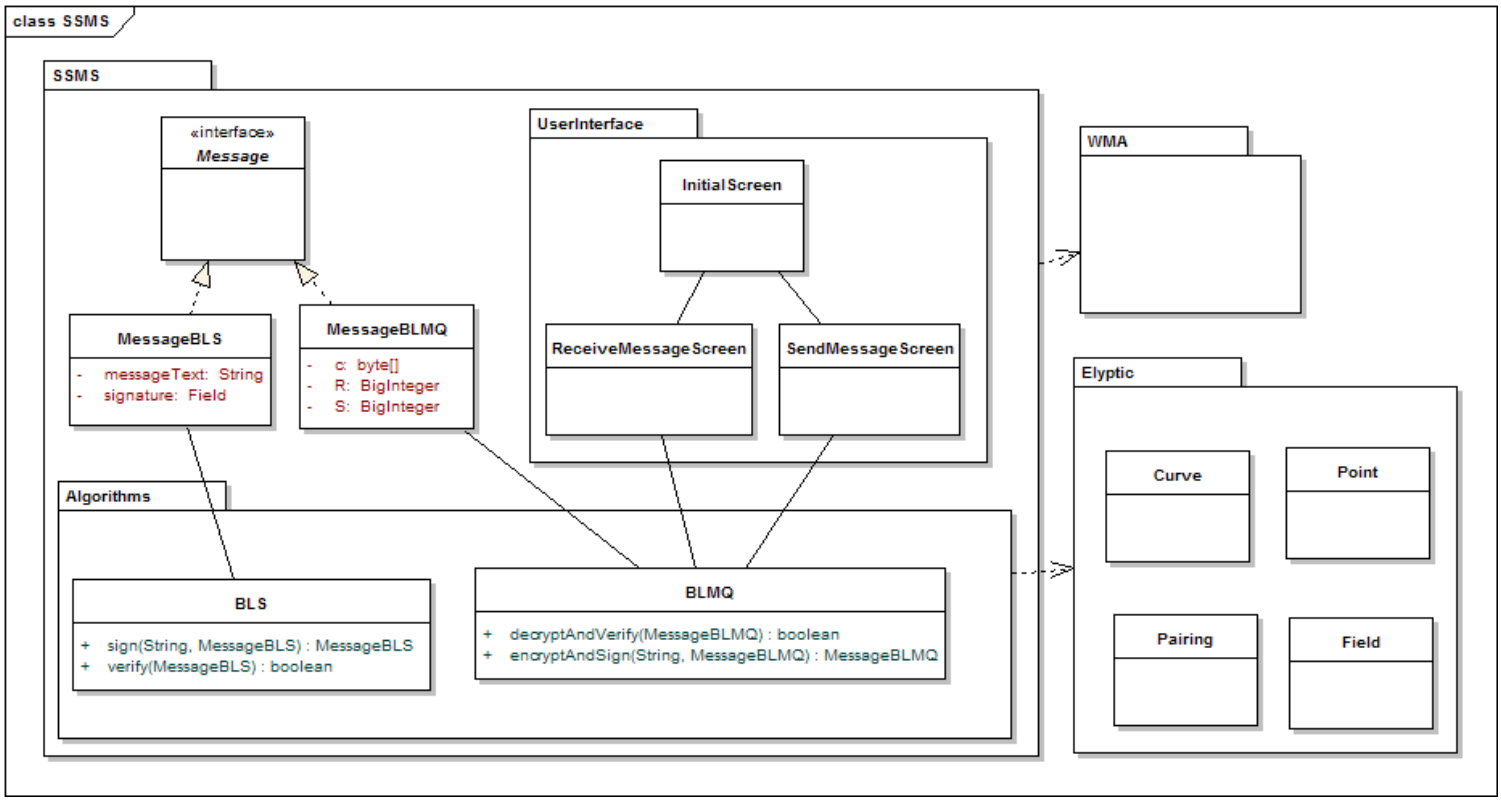
A especificação de requisitos está contida no documento "Aplicativo Seguro de SMS - Especificação de Requisitos do Software", anexo a este, de forma a atender às recomendações da norma IEEE830-1998.

4. DETALHAMENTO

Será utilizado o algoritmo BLMQ para cifrar e assinar as mensagens. Nosso orientador é um de seus idealizadores. Este algoritmo é baseado em identidades, ou seja, a chave pública pode ser a identidade do usuário. No nosso caso, usaremos o telefone do usuário como chave pública.

Iremos primeiramente implementar envio e recebimento usando BLS, por ser mais simples que o BLMQ. Posteriormente implementaremos com o algoritmo BLMQ, que é baseado no BLS. O produto final só suportará envio e recebimento usando o algoritmo BLMQ, ou seja, a implementação do BLS será apenas para fins de aprendizado, testes e provas de conceito.

Montamos um esboço do diagrama de classes do aplicativo SSMS. Ele será melhor detalhado quando estivermos implementando, a medida que surgirem necessidades específicas para as classes.



O pacote WMA é o "Wireless Messaging API", uma biblioteca do J2ME para manipular envios e recebimentos de SMS.

O pacote Elyptic contém classes para manipulação de curvas elípticas, desenvolvidas pelo nosso orientador.

5. PLANEJAMENTO E METODOLOGIA

Estamos trabalhando em conjunto para desenvolver o código. Criamos um repositório de código hospedado no Google Code. Fazemos reuniões semanais para dividir as tarefas e discutir sobre o andamento.

Nossa metodologia consiste em estudar os algoritmos a serem utilizados, fazer testes de viabilidade, e depois implementar os algoritmos.

Fizemos testes de benchmark em processadores de celulares para verificar a viabilidade do projeto.

Segundo nosso orientador, precisávamos verificar se a operação de potência para números inteiros grandes podia ser feita rapidamente no celular, utilizando a classe BigInteger da biblioteca “BouncyCastle”.

Foi escrita uma rotina de benchmark que executava 100 operações deste tipo para números inteiros de 160 bits. Em um celular Samsung SGH-E570, as 100 operações demoraram no total 5958 ms, ou seja, 60 ms por operação, o que é satisfatório.

Terminada a fase de especificação e estudos, estaremos dando início à etapa de implementação, iremos desenvolver os módulos principais do sistema: o aplicativo SSMS para o celular e a autoridade de confiança. Posteriormente, pretendemos desenvolver uma aplicação que simula o funcionamento de um banco, que suportará o uso de SMS seguro para fazer transações.

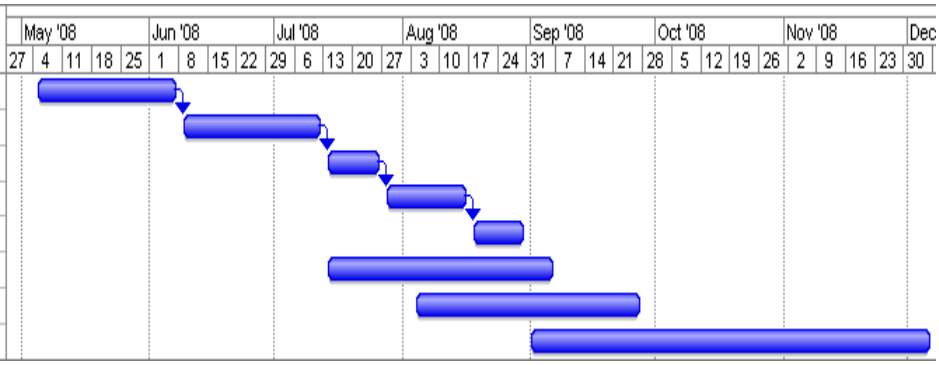
6. CRONOGRAMA

Abaixo temos o cronograma do nosso projeto, a partir deste momento.

Iniciaremos nossa etapa de implementação em 05/05/2008, e pretendemos estar com o aplicativo pronto em 19/09/2008.

Implementaremos o envio e recebimento com o algoritmo BLS apenas para fins conceituais e de testes. Na nossa aplicação final, será usado o algoritmo BLMQ, que se baseia no BLS.

ID	Task Name
1	Implementação de send/receive BLS
2	Implementação de send/receive BLMQ
3	Melhorias de performance no celular
4	Implementação da autoridade de confiança
5	Implementação de um simulador de um banco que usa nosso sistema
6	Testes
7	Correção de problemas
8	Elaboração da monografia



7. RECURSOS E INFRA-ESTRUTURA REQUERIDOS

Para o desenvolvimento, será utilizada a plataforma Java, que é gratuita. Serão também utilizadas algumas bibliotecas, como por exemplo o "Bouncy Castle" uma biblioteca gratuita fornecendo alguns utilitários leves auxiliando aplicações de criptografia, capaz de rodar na plataforma J2ME.

Também serão utilizadas classes que manipulam curvas elípticas, desenvolvidas por nosso orientador.

Serão necessários ao menos dois aparelhos celulares capazes de rodar aplicações J2ME para testarmos a aplicação.

8. ESTRUTURA DO RELATÓRIO FINAL

Usaremos o MiKTeX com alterações fornecidas pelo nosso orientador para gerar a monografia na forma padronizada aceita pela EPUSP, que é ligeiramente diferente da formato previsto na norma ABNT.

Na monografia, explicaremos o algoritmo BLMQ utilizado, apresentaremos discussões sobre a implementação do sistema, resultados de testes, e perspectivas de uso do projeto como um produto, aplicado a um negócio real.

9. COMENTÁRIOS FINAIS DO GRUPO

Reservamos um horário de cada semana para nos dedicarmos ao projeto. As reuniões com o orientador ocorreram quase todas as semanas deste módulo, onde discutíamos sobre os aspectos teóricos dos algoritmos e sobre detalhes da implementação.

A cada reunião, levávamos o resultado do andamento do projeto na semana.

Pesquisamos bastante sobre os algoritmos a serem utilizados, e temos uma boa idéia de como iremos implementar o sistema. Acreditamos que estamos em dia com nossas metas atualmente.

10. BIBLIOGRAFIA

[1] R. Rivest. Selected Topics in Cryptography. Lecture 26. 2004

[2] Securing your J2ME/MIDP apps. Disponível em <http://www.ibm.com/developerworks/library/j-midpds.html>. Acesso em 25 fev. 2008.

[3] Terence Spies on Identity-Based Encryption. Disponível em <http://www.onlamp.com/pub/a/onlamp/2003/07/17/ibe.html>. Acesso em 25 fev. 2008.