Aplicativo de SMS Seguro
Secure-SMS

Especificação de Requisitos do Software

Autores	Data de Emissão: 16/04/2008
Eduardo de Souza Cruz, eduardo.cruz@poli.usp.br Geovandro Carlos F. Pereira, geovandro.pereira@poli.usp.br Rodrigo Rodrigues da Silva, rodrigo.silva1@poli.usp.br	Versão: 1.0

Folha de controle de revisões

Versão	Data	Descrição	Responsáveis
0.1	31/03/2008	Criação do documento	Rodrigo
0.2	10/04/2008	Seção 1, início da seção 2	Rodrigo, Geovandro
0.3	12/04/2008	Início da seção 3	Rodrigo, Geovandro
0.4	13/04/2008	Requisitos funcionais	Rodrigo
0.5	14/04/2008	Requisitos não funcionais	Rodrigo
0.6	15/04/2008	Revisão	Rodrigo, Eduardo
1.0	15/04/2008	Publicação	-

1. Introdução

1.1 Objetivo

O objetivo deste documento de Especificação de Requisitos do Software é identificar de forma completa e clara todos os requisitos a serem atendidos pelo projeto de formatura desenvolvido pelo presente grupo.

Este documento é direcionado a pessoal técnico qualificado, porém deve ser de fácil entendimento por supostos clientes não especializados em engenharia. Este documento será apresentado ao orientador do projeto, aos docentes responsáveis pela disciplina PCS2040 - Projeto de Formatura I e, posteriormente, à Banca Examinadora.

Esta especificação procura atender às recomendações da norma IEEE830-1998.

1.2 Escopo

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. A criptografia das mensagens será realizada por algoritmos assimétricos baseados em identidades.

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. Os principais benefícios do sistema são a sua flexibilidade, podendo ser adaptado às necessidades dos clientes, e leveza, já que sua arquitetura é restrita à camada de aplicação: o software opera sobre a camada de aplicação do modelo OSI, sendo transparente à arquitetura interna da rede móvel. Essas duas qualidades tornam possível sua viabilidade em diversos cenários. Por outro lado, o software se utilizará de criptografia baseada

em identidades, o que dispensa a existência de um diretório de chaves públicas: a chave pública de determinado usuário será seu número de celular, obviamente conhecido pelos remetentes d das mensagens.

1.3 Definições, Acrônimos e Abreviações

SMS: Short Message Service

CLDC: Compact Limited Device Configuration

JSR: Java Specification Requests

WMA: Wireless Messaging API

1.4 Referências

Este documento complementa e é referenciado pelo documento “Aplicativo de SMS Seguro – Relatório Final da Especificação”. As referências bibliográficas desta especificação encontram-se no item 4.

1.5 Visão Geral do Documento

Esta especificação de requisitos contém mais duas seções. A seção 2 descreve os fatores gerais que afetam o software e seus requisitos. Esta seção não aborda requisitos específicos, mas apenas o contexto destes requisitos. Eles serão descritos em detalhes na seção 3 desta especificação, de modo que seja de fácil entendimento.

2. Descrição

2.1 Perspectiva do Produto

O software descrito nesta especificação pode funcionar de maneira independente, implementando apenas uma camada de segurança em mensagens trocadas por dois usuários comuns da rede. No entanto, em colaboração com outros sistemas, o software pode implementar uma arquitetura de suporte a aplicações que requeiram segurança da informação, como as de comércio eletrônico, financeiras e militares. O software poderá integrar-se a outros serviços do cliente. No entanto, seu escopo é apenas garantir a segurança da informação, e as funcionalidades devem ser implementadas separadamente.

2.1.1 Interfaces do Sistema

O sistema deverá interfacear com a rede GSM para a recepção e o envio de mensagens do tipo Short Message Service (SMS). Para isso, será utilizada a Sun Wireless Messaging API (WMA)[1], uma biblioteca implementada em Java que provê o acesso a essa camada para a transmissão e recepção de mensagens binárias.

2.1.2 Interfaces do Usuário

As interfaces com o usuário deverão ser simples e claras. O sistema não deve exceder 5 telas diferentes. As telas devem ser dimensionadas de modo a não exceder o espaço útil das telas dos telefones móveis comuns (não *smartphone*). As mensagens de erro devem conter no máximo 50 caracteres. Um usuário comum deverá estar apto a utilizar a interface básica do sistema após 5 minutos de treinamento acompanhado ou 10 minutos de treinamento desacompanhado (por tentativa e intuição).

2.1.3 Interfaces com Hardware

O software não possuirá nenhuma interface direta com hardware.

2.1.4 Interfaces com Software

O sistema será executado sobre a plataforma Java ME (Micro Edition) CLDC (Connected Limited Device Configuration), versão 1.1, em qualquer implementação que atenda à especificação JSR139 [2]. O software utilizará as bibliotecas BouncyCastle e SSPairing como ferramentas de apoio ao desenvolvimento dos algoritmos de segurança.

2.1.5 Limitações de Memória

O binário compilado do software deverá limitar-se a tamanho de 200kB.

2.1.6 Requisitos de Adaptação

O software deverá adaptar-se a cenários de implantação que requeiram características específicas: interface com o usuário especializada ou customizada, método de inserção de dados, mecanismo de geração e distribuição de chaves, interface com serviços web.

2.2 Funções do Produto

A principal funcionalidade do sistema pode ser resumida em enviar e receber mensagens de forma segura, através da assinatura e encriptação do seu conteúdo. As funções específicas estão listadas e detalhadas abaixo:

- Assinatura de mensagens: a assinatura de uma mensagem consiste em calcular um hash da mensagem com base na chave privada do usuário que a envia. Este hash deve ser concatenado no final da mensagem.

- Encriptação de mensagens: permite a cifrassinatura (assinatura e encriptação) de uma mensagem com a chave privada do remetente e a chave pública do destinatário.

- Decriptação de mensagens: permite a decriptação e verificação de uma mensagem com a chave pública do remetente e a chave privada do destinatário.
- Verificação de mensagens: permite a verificação da integridade e da identidade do remetente através da chave privada do destinatário.
- Geração de chaves privadas: uma autoridade de confiança deverá gerar as chaves

privadas dos usuários com base em suas chaves públicas.

- Troca de chaves privadas: quando solicitado, a autoridade de confiança deverá gerar uma nova chave privada para um usuário.

2.3 Características do Usuário

Os requisitos mínimos de hardware e software do sistema requerem aparelhos ligeiramente mais sofisticados que os mais baratos do mercado. No entanto, como simplificação, assumiremos que os usuários do software correspondem ao perfil médio dos usuários de telefone celular no Brasil.

2.4 Restrições

2.4.1 Métricas

2.4.1.1 Tempo de espera

O tempo de espera pelo usuário para encriptação e decriptação de uma mensagem não deverá exceder 5 segundos.

2.4.1.2 Tamanho da chave privada

De modo a maximizar a segurança sem comprometer a usabilidade, a chave privada não deverá exceder 128 bits.

2.4.2 Limitações de Hardware

Devido ao relativo baixo desempenho dos processadores para celular, o software deverá implementar rotinas de maneira eficiente. Por outro lado, o software carregado na memória não deverá exceder o apontado no item 2.1.e) desta especificação.

2.4.3 Requisitos de Linguagem

O software deverá ser implementado na linguagem Java, de maneira compatível com o ambiente de execução Java ME CLDC.

2.5 Hipóteses e Dependências

O funcionamento do software é garantido para a plataforma Java ME CLDC 1.1. Atualizações na versão dessa plataforma podem exigir modificações nesta especificação.

3. Requisitos Específicos

3.1 Requisitos Não-Funcionais

3.1.1 Usabilidade

O tamanho da chave privada não deve prejudicar a usabilidade do software. O usuário deve ser capaz de digitá-la em menos de 60 segundos. O método de entrada das mensagens deve ser semelhante ao dos aparelhos celulares convencionais.

3.1.2 Desempenho

A aplicação deve ser capaz de cifrar ou decifar uma mensagem em menos de 5 segundos.

3.1.3 Confiabilidade

O software deverá apresentar MTTF de 1 ano. Entende-se como falha a parada do software pela subida de uma exceção não tratada. Essa medida ignora falhas de componentes externos ao software (hardware do celular, plataforma Java).

3.1.4 Disponibilidade

O software deverá apresentar disponibilidade de 99%. Como não se trata de um sistema on-line, entende-se como disponibilidade a razão entre as tentativas bem sucedidas de acessar o software e o total de tentativas. Ou seja, a cada 100 tentativas de acessar o software apenas uma não terá sucesso.

3.1.5 Compatibilidade

O software deverá ser compatível com todos os dispositivos celulares equipados com a plataforma Java ME (Micro Edition) e com a configuração CLDC.

3.1.6 Portabilidade

O software deverá ser portável para a plataforma Java SE (Standard Edition) tendo em vista o uso da aplicação como interface com web services.

3.1.7 Segurança

O software deve garantir que o destinatário da mensagem e apenas ele, além do remetente, tenha acesso ao seu conteúdo em tempo viável. O software também deve garantir a integridade da

mensagem em relação a corrupções maliciosas ou acidentais durante o tráfego.

3.2 Requisitos Funcionais

Os requisitos funcionais foram divididos de acordo com os dois módulos previstos na arquitetura do software:

- Software embarcado – aplicação que irá rodar no celular, capaz de cifrar e decifrar, assinar e verificar mensagens SMS.
- Autoridade de Confiança (PKG – Private Key Generator) – responsável pela geração das chaves privadas dos usuários com base em seu par de chaves mestras.

3.2.1 Autoridade de Confiança

3.2.1.1 Geração do par de chaves mestras da autoridade

Introdução/Propósito: a PKG deverá ser capaz de gerar um par de chaves mestras (privada e pública). A chave mestra pública será conhecida em toda a rede.

Estímulo/Resposta: A chave mestra privada da autoridade de confiança / O ponto público da autoridade

Detalhamento: As chaves deverão ser geradas sobre curvas elíptcas BN conforme [3]

3.2.1.2 Geração de chaves privadas de usuários baseadas em ID

Introdução/Propósito: este requisito tem como objetivo a geração de uma chave privada a partir da chave pública de um usuário. O sistema de criptografia por ID, porposto por Shamir, 1984 [4], permite que um identificador arbitrário possa ser usado como chave pública de um usuário, dispensando o uso de um diretório de chave pública. Os nós da rede (usuários de celular) terão como chave pública seu próprio número de celular. Deve ser considerado o número completo, com código do país e código de área (ex: +551199990000).

Estímulo/Resposta: o software da PKG deverá receber uma string contendo o número completo do usuário. O resultado será a chave privada do usuário (uma seqüência de bits), respeitando o tamanho máximo da chave.

Detalhamento: seja (s, T=s.Q) o par de chaves mestras da autoridade de confiança. A chave privada de um usuário identificado publicamente por ID será $U_{ID} = ((H(ID)+s)^{-1}).P$, onde H(ID) é o hash criptográfico de ID e P é o ponto público da curva E'.

3.2.1.3 Revogação de chaves privadas

Introdução/Propósito: este requisito tem o propósito de permitir que o usuário revogue uma chave privada quando roubada ou perdida. Este esquema deve permitir a geração de uma nova chave privada sem que a chave pública seja alterada (Hanaoka et al, 2005).

Estímulo/Resposta: A chave pública do usuário / A nova chave privada do usuário.

Detalhamento: a implementação deverá seguir o esquema proposto por [5]

3.2.2 Software Embarcado

3.2.2.1 Assinatura e encriptação de mensagens (cifrassinatura)

Introdução/Propósito: permitir a crifrassinatura de uma mensagem m através do algoritmo BLMQ.

Estímulo/Resposta: A mensagem m, a chave privada do remetente Ua e a chave pública B do destinatário / a mensagem cifrada c, R e S (parâmetros do algoritmo)

Detalhamento:

$r \leftarrow$ random mod n

$k \leftarrow g^r$

$c \leftarrow E_k(m)$, $h \leftarrow M_k(m)$

$R \leftarrow r \cdot (H(B) \cdot P + T)$

$S \leftarrow (h + r) \cdot U_A$

Ek e Mk são algoritmos de encriptação e de assinatura, respectivamente. E pode ser um algoritmo nulo, ou seja, Ek(m)=m (a mensagem será somente assinada).

B é a identidade do usuário B (destinatário). U_A é a identidade do usuário A (remetente)

c, R e S serão enviados na mensagem SMS.

3.2.2.2 Verificação e decriptação de mensagens (vericifração)

Introdução/Propósito: permitir a vericifração de uma mensagem cifrada c através do algoritmo BLMQ.

Estímulo/Resposta: a mensagem cifrada c, a chave privada do destinatário, a chave pública do remetente, R e S (parâmetros do algoritmo) / a mensagem original m, e a variável booleana que indica a validade da mensagem ou não.

Detalhamento:

$k \leftarrow e(R, U_B)$

$m \leftarrow E^{-1}_k(c)$, $h \leftarrow M_k(m)$

$v \leftarrow e(H(A) \cdot P + T, S) \cdot g^{-h}$

accept $\Leftrightarrow v = k$

3.2.2.3 Envio de mensagens pelo protocolo SMS

Introdução/Propósito: gerar o corpo da mensagem SMS e enviá-la pela rede GSM através da Wireless Messaging API.

Estímulo/Resposta: a mensagem cifrada c (como um array de bytes) e os parâmetros R e S do algoritmo, também em forma binária / um array de bytes que deverá ser o corpo da mensagem binária.

Detalhamento

O formato da mensagem: a mensagem deverá conter, respeitando o tamanho máximo permitido, na seguinte ordem os parâmetros para envio, em formato binário:

- Um cabeçalho de 6 bytes contendo, nos primeiros 4 bytes, o texto "SSMS" (em codificação ASCII de 8 bits);
- no quinto byte o número de versão do aplicativo SSMS que a gerou;
- no sexto byte o número da subversão;
- os parâmetros R e S, consecutivamente, ocupando 16 bytes cada um;

- um byte que indica o algoritmo de encriptação de c; e
- a mensagem cifrada c nos 360 bytes seguintes e, eventualmente, um byte contador de controle concatenado ao final da mensagem.

Segmentação: a WMA permite que a mensagem tenha um tamanho máximo de 399 bytes, divididos em até 3 segmentos de 133 bytes.

4. Referências

[1] http://developers.sun.com/mobility/midp/articles/wma/index.html

[2] http://jcp.org/aboutJava/communityprocess/final/jsr139/

[3] Paulo S. L. M. Barreto, Benoît Libert, Noel McCullagh, Jean-Jacques Quisquater, "Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps," *Advances in Cryptology -- Asiacrypt'2005*, Lecture Notes in Computer Science **3788**, Springer (2005), pp. 515--532.

[4]Adi Shamir, Identity-Based Cryptosystems and Signature Schemes. *Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science*, 7:47--53, 1984

[5]Hanaoka, Y. et al in Advances in cryptology : ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, December 4-8, 2005 : proceedings / Bimal Roy (ed.)