Construção de um Sistema de SMS Seguro

Eduardo de Souza Cruz[1], **Geovandro C. F. Pereira**[1],
Rodrigo Rodrigues da Silva[1], **Paulo S. L. M. Barreto**[1]*.

[1] Departamento de Engenharia de Computação e Sistemas Digitais,
Escola Politécnica, Universidade de São Paulo, Brasil.
{eduardo.cruz,geovandro.pereira,rodrigo.silva1}@poli.usp.br, pbarreto@larc.usp.br

Abstract. *This paper presents the undergraduate work being developed by students of Computer Engineering at the Escola Politécnica of the University of Sao Paulo. The work consists of the implementation of a solution that guarantees security and integrity in the transmission of SMS messages, coupling conventional, certificateless and identity-based cryptography to achieve public key validation.*
Throughout the paper, we will present aspects of our solution, an innovative cryptographic scheme, metrics, results of performance tests, and considerations about the work in progress.

Resumo. *Este artigo visa a apresentar o projeto de formatura que vem sendo desenvolvido por alunos formandos em Engenharia de Computação pela Escola Politécnica da Universidade de São Paulo. O trabalho consiste na implementação de uma solução que garanta segurança e integridade no envio de mensagens SMS, utilizando criptografia convencional sem certificados e criptografia baseada em identidades para validação da chave pública.*
Ao longo do artigo, apresentaremos aspectos da nossa solução, um esquema criptográfico inovador, métricas, resultados de testes de desempenho, além de considerações sobre o andamento do trabalho.

1. Introdução

A motivação do projeto surgiu da ausência de soluções universalmente adotadas para garantir segurança em mensagens SMS. As mensagens trafegam pela rede celular de forma insegura, passando obrigatoriamente por pelo menos um intermediário não 100% confiável: a operadora do serviço de telefonia. As mensagens podem ficar armazenadas em texto plano no banco de dados da operadora [Ng 2006], de forma que pessoas mal intencionadas infiltradas no sistema podem ser capazes de visualizar, alterar e até enviar mensagens em nome de outra pessoa. Há também outros métodos para interceptar mensagens SMS. [Enck et al. 2005].

Como possíveis aplicações de nossa solução, podemos citar a realização de transações bancárias usando mensagens SMS, sistemas de comunicação que requeiram confidencialidade e integridade (órgãos militares e governamentais, executivos de grandes empresas) ou apenas usuários comuns em busca de maiores níveis de privacidade.

Este artigo compreende a descrição do cenário e desafios encontrados no levantamento dos requisitos de um sistema de troca de mensagens SMS seguro e na escolha

*Orientador do trabalho. Bolsista de Produtividade em Pesquisa CNPq, processo 312005/2006-7.

e aplicação de um esquema de criptografia que garantisse esses requisitos. Desse modo, são discutidos os conceitos dos algoritmos BLMQ (baseado em identidades) e BDCPS (CL-PKC), criado devido ao insucesso do primeiro em atender aos requisitos do projeto.

Os algoritmos citados anteriormente utilizam os conceitos de cifrassinatura e vericifração de mensagens. A cifrassinatura consiste em um método de criptografia de chave pública que garante infalsificabilidade e confidencialidade simultaneamente com um overhead menor do que o requerido pela assinatura digital seguida de encriptação de chave pública. Isto é alcançado assinando e encriptando uma mensagem em um único passo. A vericifração consiste da operação inversa, ou seja, a verificação da validade do autor da mensagem e sua decriptação de chave pública, simultaneamente. [Zheng 2005].

2. Objetivo

O objetivo principal de nosso projeto é criar uma aplicação capaz de prover confidencialidade, integridade e autenticidade a mensagens SMS (*Short Message Service*) sem extrapolar as limitações de recursos computacionais e de ocupação de banda típicas desse ambiente.

Esse objetivo deve ser alcançado sem comprometer a usabilidade do serviço. Nossos principais desafios foram as limitações presentes no ambiente, tal como o pequeno poder de processamento dos aparelhos celulares e, principalmente, a pequena largura de banda e espaço disponíveis, já que, de acordo com a especificação do padrão GSM, as mensagens SMS não comportam mais do que 160 caracteres [Ng 2006].

3. Cenário e métricas

O processo de escolha do algoritmo mais adequado a ser integrado ao projeto foi iniciado apenas após termos definido e avaliado os requisitos do sistema e suas respectivas métricas - em especial para os requisitos de segurança de acesso e usabilidade.

A seguir, definimos as métricas e suas limitações.

- Tempo de espera: Consiste nos tempos para cifrassinar e vericifrar uma mensagem. Baseando-se em aplicações já existentes e satisfazendo os requisitos de usabilidade de nosso projeto, estimamos que um intervalo de espera para processamento de uma mensagem de no máximo 5 segundos seja tolerável pelo usuário.
- Tamanho máximo da mensagem: Consiste da soma dos bytes úteis da mensagem com os bytes de controle do algoritmo. Implementações SMS baseadas em *Sun Wireless Messaging API (WMA)* podem dividir uma única mensagem em, no máximo, 3 segmentos. Recomenda-se que as aplicações SMS utilizem mensagens com menos de 399 bytes binários de modo a não comprometer sua portabilidade [Ortiz 2002]. Desse modo, estabelecemos um tamanho máximo de 399 bytes para as mensagens transmitidas, sendo este espaço compartilhado entre os dados de controle do algoritmo criptográfico utilizado e a mensagem criptografada em si.
- Tamanho das chaves privada/pública e da assinatura: Devido às limitações de espaço de armazenamento das mensagens, estabeleceu-se que a assinatura de uma mensagem, bem como a chave privada do usuário, não deveriam exceder 200 bits. No entanto, essa restrição não deveria comprometer os requisitos de segurança do sistema.

Sabendo que um certificado digital típico ocupa entre 2KB e 4KB, nota-se aqui que uma solução baseada em infra-estrutura convencional de chaves públicas inviabilizaria completamente o sistema: antes de se enviar uma mensagem SMS segura para algum usuário, seria necessário receber o certificado desse usuário particionado em 15 a 30 mensagens SMS, além de enviar em resposta outro certificado em mais 15 a 30 mensagens SMS. Esse esforço precisaria ser efetuado novamente para cada novo destinatário a quem determinado usuário desejasse enviar mensagens, ou em cada caso de renovação ou revogação de certificado. Some-se a isto o espaço ocupado por uma única assinatura convencional, tipicamente de 128 bytes por estar baseada no algoritmo RSA com 1024 bits; este *overhead* seria duplicado com o requisito de cifrar e assinar a mensagem, isto é, tomaria 256 bytes do espaço disponível.

Por outro lado, a manutenção de um diretório confiável de chaves públicas, típico de sistemas de criptografia convencionais, seria impraticável em uma rede de telefonia celular. Uma solução tecnológica baseada em alternativas à criptografia convencional é, portanto, imprescindível.

Sendo assim, foi considerado o uso de criptografia com assinatura baseada em identidades, de acordo com o conceito proposto inicialmente por Shamir [Shamir 1984]. Aprofundando-se na especificação, percebeu-se ainda que a chave pública do usuário poderia ser estabelecida essencialmente a partir de sua identificação única no sistema, ou seja, seu próprio número de celular. Desse modo, a criptografia baseada em identidades com emparelhamentos bilineares parecia atender aos requisitos do nosso aplicativo e foi inicialmente adotada na solução do projeto.

4. Proposta

A primeira tentativa de solução adotava o esquema de cifrassinatura baseada em identidades BLMQ [Barreto et al. 2005]. O esquema foi escolhido por, aparentemente, atender aos requisitos estabelecidos inicialmente. O BLMQ era notadamente mais eficiente que esquemas de criptografia baseada em identidades anteriores, como o de Boneh-Franklin [Boneh and Franklin 2001], o que poderia tornar o uso desse tipo de criptografia viável em ambientes produtivos. Além disso, uma assinatura de 160 bits garantiria um nível de segurança de aproximadamente 80 bits equivalente ao do RSA de 1024 bits[Kaliski 2003].

O esquema foi implementado na linguagem de programação Java, e testes foram realizados em um aparelho celular Nokia 6275.

O desempenho observado inicialmente foi insatisfatório, não atendendo aos requisitos de usabilidade estabelecidos na especificação. Foram feitas tentativas de melhoria do desempenho, como variação do tamanho das chaves, uso de diferentes funções de emparelhamento (Ate, Eta)[Freeman et al. 2006], e implementações com diferentes bibliotecas que fornecessem a classe *BigInteger* - a implementação da Sun se mostrou mais eficiente do que a implementação da Bouncy Castle. Algumas adaptações no esquema em si foram feitas, como inversão da ordem das curvas utilizadas.

Os melhores resultados obtidos são apresentados na tabela 1.

Tabela 1. Testes com BLMQ

Operação	Tempo (s)
Inicialização das classes	12.9
Emparelhamento Eta	4.2
Emparelhamento Ate	3.9

Como estes tempos não atendiam aos requisitos de usabilidade do projeto, fez-se necessário buscar alternativas. Estas dificuldades serviram como motivação para a criação de um esquema inovador. Como resultado de pesquisas realizadas, foi idealizado um novo esquema, apresentado em [Barreto et al. 2008] e brevemente descrito a seguir.

4.1. Esquema de Cifrassinatura proposto - BDCPS

Na criação do novo esquema, em vez de utilizarmos exclusivamente criptografia e assinaturas baseadas em identidades, estendemos um esquema de criptografia sem certificados com um esquema de assinatura convencional, mas utilizando técnicas baseadas em identidades para validar a chave pública, evitando o uso de certificados [Barreto et al. 2008]. A nova técnica mescla esses dois paradigmas, garantindo baixo tempo de cifrassinatura e de vericifração, tamanhos de chaves dentro dos limites adotados e níveis de segurança satisfatórios.

O esquema proposto integra esquemas preexistentes como as assinaturas BLMQ e Schnorr [Schnorr 1991] e o esquema isento de certificados de Zheng[Zheng 1997]. Neste esquema, a geração das chaves dos usuários dispensa a necessidade de uma autoridade certificadora e a utilização de certificados para validar sua chave pública. Estas características implicaram em importantes, mas não únicas, melhorias em relação ao esquema anteriormente implementado e serão discutidas na seção "Análise da Proposta". Por criptografia convencional entende-se o fato de o usuário poder escolher seu par de chaves não certificado, ou seja, ele escolhe apropriadamente uma chave privada e gera sua chave pública a partir dela. Desse modo, somente o usuário gerador de seu par de chaves convencional conhece sua chave privada, e este fato elimina a possibilidade de "Key Escrow", que é o comprometimento da chave e conseqüentemente das mensagens com ela encriptadas. O par de chaves não certificado é combinado com a solução de criptografia baseada em identidades para que seja posteriormente validado por outro usuário do sistema.

O novo esquema consiste dos seguintes algoritmos:

- **Setup:** Algoritmo gerador do conjunto dos parâmetros públicos necessários. O algoritmo escolhe um parâmetro de segurança k e define:
  n : Um inteiro primo de k bits.
  $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ Grupos de mapeamento bilinear de ordem n
  $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$: Emparelhamento eficientemente computável e não-degradado.
  $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$: Os pontos geradores dos grupos $\mathbb{G}_1$ e $\mathbb{G}_2$ respectivamente.
  Resumos criptográficos *(hashes)*
  $h_0 : \mathbb{G}_T^2 \times \{0,1\}^* \rightarrow \mathbb{Z}_n^*$,
  $h_1 : \mathbb{G}_T \times \{0,1\}^* \rightarrow \mathbb{Z}_n^*$,
  $h_2 : \mathbb{G}_T \rightarrow \{0,1\}^*$,

$h_3 : (\mathbb{G}_T \times \{0,1\}^*)^3 \rightarrow \mathbb{Z}_n^*$.

Uma chave mestra $s \overset{R}{\leftarrow} \mathbb{Z}_n^*$ também é escolhida, à qual a chave pública $P_{pub} = sP \in \mathbb{G}_1$ é associada.

O gerador $g = e(P,Q) \in \mathbb{G}_T$ também é incluso entre os parâmetros públicos do sistema, $\texttt{params} = (k, n, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, Q, g, P_{pub}, h_0, h_1, h_2, h_3)$.

- **Set-Secret-Value:** dados `params`, o algoritmo toma $x_A \overset{R}{\leftarrow} \mathbb{Z}_n^*$ como o valor secreto da identidade A. O usuário A pode definir x_A, sua chave parcial privada, independente do algoritmo e, neste caso, será usada como uma senha comum.
- **Set-Public-Value:** dado o valor secreto $x_A \in \mathbb{Z}_n^*$ da identidade A, computa $y_A \leftarrow g^{x_A} \in \mathbb{G}_T$ como o valor público de A.
- **Private-Key-Extract:** Obtém $\mathsf{ID}_A \in \{0,1\}^*$, o identificador de A e o valor público $y_A \in \mathbb{G}_T$, e calcula a chave privada baseada em identidade de A, $Q_A \leftarrow (h_1(y_A, \mathsf{ID}_A) + s)^{-1} Q \in \mathbb{G}_2$. A entidade A consegue verificar a consistência desta chave checando se $e(h_1(y_A, \mathsf{ID}_A)P + P_{pub}, Q_A) = g$. Esta configuração é denominada estilo de chave Sakai-Kasahara [Sakai and Kasahara 2003].
- **Set-Private-Key:** dada a chave privada parcial da entidade A, $Q_A \in \mathbb{G}_2$ e o valor secreto $x_A \in \mathbb{Z}_n^*$, este algoritmo estabelece o par $(x_A, Q_A) \in \mathbb{Z}_n^* \times \mathbb{G}_2$ como o par completo da chave privada de A.
- **Set-Public-Key:** dada a chave privada parcial de A, $Q_A \in \mathbb{G}_2$, o valor secreto $x_A \in \mathbb{Z}_n^*$, e o correspondente valor público $y_A \in \mathbb{G}_T$, o assinante toma $u_A \overset{R}{\leftarrow} \mathbb{Z}_n^*$ e calcula
  1. $r_A \leftarrow g^{u_A}$
  2. $h_A \leftarrow h_0(r_A, y_A, \mathsf{ID}_A)$
  3. $T_A \leftarrow (u_A - x_A h_A) Q_A$

  A chave pública completa da entidade A é a tripla $(y_A, h_A, T_A) \in \mathbb{G}_T \times \mathbb{Z}_n^* \times \mathbb{G}_2$. Este configuração é uma combinação da assinatura de Schnorr (sob a chave x_A) com a assinatura BLMQ (sob a chave Q_A) no valor público y_A e a identidade ID_A.
- **Public-Key-Validate:** dada a chave pública completa da entidade A, (y_A, h_A, T_A), este algoritmo verifica que y_A tem ordem n (i.e. que $y_A \neq 1$ mas $y_A^n = 1$) e calcula
  1. $r_A \leftarrow e(h_1(y_A, \mathsf{ID}_A)P + P_{pub}, T_A) y_A^{h_A}$
  2. $v_A \leftarrow h_0(r_A, y_A, \mathsf{ID}_A)$

  O verificador aceita a mensagem se, e somente se $v_A = h_A$. O processo de validação combina a verificação da assinatura Schnorr com a verificação da assinatura BLMQ.
- **Signcrypt:** Para encriptar $m \in \{0,1\}^*$ sob a chave pública do receptor $y_B \in \mathbb{G}_T$ previamente validade para a identidade ID_B e P_{pub}, e a chave privada do emissor $x_A \in \mathbb{Z}_n^*$, chave pública $y_A \in \mathbb{G}_T$ e a identidade ID_A, o emissor toma $u \overset{R}{\leftarrow} \mathbb{Z}_n^*$ e calcula
  1. $r \leftarrow y_B^u$
  2. $c \leftarrow h_2(r) \oplus m$
  3. $h \leftarrow h_3(r, m, y_A, \mathsf{ID}_A, y_B, \mathsf{ID}_B)$
  4. $z \leftarrow u - x_A h$

  O criptograma de assinatura é a tripla $(c, h, z) \in \{0,1\}^* \times \mathbb{Z}_n^2$. Comparado ao método de cifrassinatura de Zheng, as identidades de ambos o emissor e o destinatário são inclusas na equação de autenticação 3, e a equação de assinatura 4 segue o estilo Schnorr em vez do dedicado, porém levemente mais complicado(devido à presença da inversão de campos), estilo Zheng, similar ao DSA [NIST 2000].

- **Unsigncrypt:** dada a chave pública do emissor $y_A \in \mathbb{G}_T$ previamente validade para a identidade ID_A e P_{pub}, e a chave privada do receptor $x_B \in \mathbb{Z}_n^*$, a chave pública $y_B \in \mathbb{G}_T$ e a identidade ID_B, sob a recepção da tripla (c, h, z) o receptor verifica se $h, z \in \mathbb{Z}_n^*$ e calcula
  1. $r \leftarrow y_A^{hx_B} y_B^z$
  2. $m \leftarrow h_2(r) \oplus c$
  3. $v \leftarrow h_3(r, m, y_A, \mathsf{ID}_A, y_B, \mathsf{ID}_B)$

  O receptor aceita a mensagem se, e somente se, $v = h$. A equação 1 é levemente mais simples que seu correlato em Zheng devido ao estilo Schnorr adotado para a cifrassinatura.

5. Análise da Proposta

5.1. Implementação

A linguagem de programação Java e a plataforma JME, adotadas na implementação da aplicação, permitem sua implantação na maioria dos dispositivos móveis atuais e futuros, em virtude de a máquina virtual Java já ser instalada por padrão pelos principais fabricantes. O acesso à infra-estrutura de mensagens se dá-se através da *Sun Wireless Messaging API (WMA)*, uma biblioteca que permite a transmissão e recepção de mensagens binárias. Além disso, é possível trocar mensagens SMS de forma segura também com a *WEB* via conexões HTTP.

5.2. Utilização

O funcionamento da solução SMS, do ponto de vista do usuário, pode ser descrito pelas seguintes etapas:

- Implantação: O usuário recebe a aplicação de SMS seguro por meio confiável, instalando-a em seu aparelho.
- Cadastro: Um novo usuário que deve cadastrar sua senha, denominada x_A que é usada toda vez que um SMS seguro for enviado ou recebido. A aplicação realiza, então, as operações necessárias para a geração de suas chaves pública e privada. Nesta etapa um SMS é para que o valor y_A seja enviado ao provedor de serviço e este calcule e retorne o valor de Q_A (encriptado sob y_A), parte de sua chave privada, para o usuário. Observe que no cálculo de Q_A utilizam-se ambos y_A e ID_A (o número do celular). Esse fato possibilita a reutilização do mesmo número de celular por outro usuário caso o primeiro desligue-se de seu número e este seja reutilizado pela operadora de telefonia celular.
- Validação de chave pública: Para enviar mensagens seguras a um destinatário o usuário envia, inicialmente, uma mensagem de validação de chave pública. Esta operação de validação demanda tempo computacional em torno de quinze vezes maior que o tempo gasto pelas operações de cifrassinatura e vericifração. Contudo, o impacto ao usuário pode ser considerado pequeno já que esta operação é realizada uma única vez para cada destinatário que determinado usuário deseje contatar.
- Troca de mensagens: Após validada a chave pública por ambos os usuários em comunicação, a aplicação é transparente ao esquema de criptografia, permitindo que um usuário escreva sua mensagem normalmente, sendo requisitado apenas a inserir sua senha x_A para concluir a operação.

Além disso, existem dois modos de envio seguro: simples assinatura da mensagem (o esquema permite o uso de um algoritmo de encriptação nulo), ou cifrassinatura.

6. Resultados

O novo esquema também foi implementado na plataforma JME (*Java Platform Micro Edition*), e testes para validar a viabilidade foram feitos em diversos modelos de aparelhos celulares, além dos emuladores dos ambientes de desenvolvimento *Eclipse* e *NetBeans*.

Os resultados dos testes são apresentados nas tabelas 2 e 3. Foram feitos testes com chaves de 127 e 160 bits, para dois modelos distintos de celulares, Nokia 6275 e Sony Ericsson W200i.

Tabela 2. Testes com o novo esquema (chaves de 127 bits) e comparação com o RSA

Operação	Tempo Nokia 6275 (s)	Tempo Sony Ericsson W200i (s)
Emparelhamento Eta	7,30	2,37
Emparelhamento Ate	7,43	2,38
Private-Key-Extract	2,63	0,93
Check-Private-Key	9,31	2,92
Set-Public-Value	0,66	0,22
Set-Public-Key	3,40	1,15
Public-Key-Validate	10,50	3,35
Signcrypt	0,57	0,21
Unsigncrypt	0,80	0,29
Private RSA-508	1,05	0,39
Public RSA-508	0,03	0,02

Tabela 3. Testes com o novo esquema (chaves de 160 bits) e comparação com o RSA

Operação	Tempo Nokia 6275 (s)	Tempo Sony Ericsson W200i (s)
Emparelhamento Eta	10,53	3,59
Emparelhamento Ate	10,54	3,64
Private-Key-Extract	3,72	1,32
Check-Private-Key	12,70	4,46
Set-Public-Value	0,96	0,33
Set-Public-Key	4,96	1,63
Public-Key-Validate	14,94	5,12
Signcrypt	0,77	0,31
Unsigncrypt	1,22	0,45
Private RSA-640	1,85	0,74
Public RSA-640	0,16	0,03

Os resultados foram satisfatórios, já que os tempos de cifrassinatura e vericifração estão de acordo com as métricas estabelecidas e bem mais eficientes em relação ao esquema inicialmente adotado.

O tempo necessário para validar uma chave pública é um pouco maior do que para as demais operações. Porém, conforme observado anteriormente, esta é uma operação

que será executada apenas uma vez para cada nova identidade que se deseje validar. A chave validada fica armazenada na memória do aplicativo, não sendo necessário validá-la novamente em uma comunicação futura com o mesmo par.

Pode-se verificar a partir das tabelas 2 e 3 que os tempos de assinatura e verificação no algoritmo proposto são menores do que os tempos do RSA, para o mesmo nível de segurança.

Dado que o tempo de uso do RSA de 1024 bits está no fim, uma nova versão será necessária [Kaliski 2003]. Contudo, para um aumento no nível de segurança do RSA, é preciso aumentar o tamanho das chaves, o que será um impacto razoável nos tempos de assinatura e verificação, uma vez que o aumento é relativamente grande. Em paralelo, um aumento equivalente no nível de segurança do BDCPS, acarreta menor aumento no tamanho das chaves e os impactos nos tempos das operações são menores. Este fato ocorre devido à relação entre o nível de segurança do BDCPS e o tamanho das chaves, que é uma relação diretamente proporcional.

A tabela 4 ilustra, para um mesmo nível de segurança, os tamanhos correspondentes de chaves para o RSA e para os algoritmos baseados em curvas elípticas.

Tabela 4. Comparação dos tamanhos das chaves em bits

Nível de segurança	RSA	Curvas elípticas
1	512	128
2	704	131
3	1024	163
4	1536	193

O código já implementado contempla apenas as operações efetuadas pelo protocolo. No momento, estamos em fase de aperfeiçoamento da especificação de requisitos do software, de modo a descrever melhor as interfaces com o sistema de controle de mensagens do aparelho celular e com usuário final. No entanto, testes com um maior número de modelos de aparelhos celulares ainda são necessários, uma vez que o desempenho pode variar muito entre fabricantes.

7. Conclusão

Apresentamos uma aplicação prática de um novo esquema de criptografia viável em ambientes com recursos computacionais tipicamente limitados. Se, por um lado, nossa abordagem inicial não nos conduziu a resultados satisfatórios, a necessidade de um protocolo que atendesse aos requisitos inicialmente propostos estimulou o desenvolvimento do esquema BDCPS, inovando no campo de criptografia em aplicações móveis.

Os resultados dos testes com o novo esquema mostram que este é uma alternativa viável ao uso de certificados, e que o projeto pode ser facilmente implantado para uso prático em um curto espaço de tempo.

Os próximos passos no desenvolvimento do projeto envolvem, ainda, uma especificação mais clara do formato da mensagem SMS cifrassinada, permitindo a acomodação dos parâmetros do esquema BDCPS sem comprometer o espaço útil disponível para a mensagem em si e de modo que esta possa ser reconhecida e processada

por versões futuras do software e por outras implementações, permitindo que esta solução possa ser universalmente adotada.

Referências

Barreto, P. S. L. M., Deusajute, A., Cruz, E., Pereira, G., and Silva, R. (2008). Toward efficient certificateless signcryption from (and without) bilinear pairings. Preprint.

Barreto, P. S. L. M., Libert, B., McCullagh, N., and Quisquater, J.-J. (2005). Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advanced in Cryptology – Asiacrypt'2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 515–532. Springer.

Boneh, D. and Franklin, M. (2001). Identity-based encryption from the Weil pairing. In *Advanced in Cryptology – Crypto'2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer.

Enck, W., Traynor, P., McDaniel, P., and Porta, T. L. (2005). Exploiting open functionality in sms-capable cellular networks. In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 393–404, New York, NY, USA. ACM Press.

Freeman, D., Scott, M., and Teske, E. (2006). A taxonomy of pairing-friendly elliptic curves. IACR ePrint Archive, report 2006/372. `http://eprint.iacr.org/2006/372`.

Kaliski, B. (2003). Twirl and rsa key size. `http://www.rsa.com/rsalabs/node.asp?id=2004`.

Ng, Y. L. (2006). Short message service (sms) security solution for mobile devices.

NIST (2000). *Federal Information Processing Standard (FIPS 186-2) – Digital Signature Standard (DSS)*. National Institute of Standards and Technology – NIST.

Ortiz, C. (2002). The wireless messaging api. Sun Developer Network (SDN) article. `http://developers.sun.com/mobility/midp/articles/wma/index.html`.

Sakai, R. and Kasahara, M. (2003). ID based cryptosystems with pairing on elliptic curve. In *SCIS'2003*, Hamamatsu, Japan.

Schnorr, C. P. (1991). Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174.

Shamir, A. (1984). Identity based cryptosystems and signature schemes. In *Advances in Cryptology – Crypto'84*, volume 0196 of *Lecture Notes in Computer Science*, pages 47–53. Springer.

Zheng, Y. (1997). Digital signcryption or how to achieve cost(signature & encryption) ¡¡ cost(signature) + cost(encryption). In *Advanced in Cryptology – Crypto'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer.

Zheng, Y. (2005). Signcryption central. `http://www.signcryption.net`.