Proposta de framework para autenticação de remetente

Pedro Junior Ashidani
Uniaraxá
Faculdade de Computação
Universidade Federal de Uberlândia
Email: pedroja@araxa.com.br

Prof Dr. Jamil Salem Barbar
Faculdade de Computação
Universidade Federal de Uberlândia
Email: jamil@facom.ufu.br.br

Prof. Marcelo Rodrigues Sousa
Faculdade de Computação
Universidade Federal de Uberlândia
Email: marcelo@facom.ufu.br.br

Prof. Dr. Luís Fernando Faina
Faculdade de Computação
Universidade Federal de Uberlândia
Email: faina@facom.ufu.br.br

Italo Tiago da Cunha
Faculdade de Computação
Universidade Federal de Uberlândia
Email: italo@pos.facom.ufu.br

***Resumo*—This article introduces the concept of SAF(Sender Authenticated Framework). In contrast with the current mechanisms used to prevent the sender's address forging, which guarantees only the server's authenticity, the SAF offers a mechanism where it is simple to verify the sender's authenticity. The SAF is based on the use of IBS(Identity Based Signature) to allow the server, which receives a message, to veriry the sender's authenticity.**

I. Introdução

Desde a criação da *Internet*, o e-mail(correio eletrônico) permanece como uma das aplicações mais populares. Devido ao baixo custo de utilização em relação ao correio convencional e a facilidade de coletar-se endereços eletrônicos pela *Internet*, muitos oponentes passaram a usar o serviço de mensagens eletrônicas de maneira indevida. As ameaças podem ser simples propagandas comerciais, pirâmides, cavalos de tróia, etc. Programas especializados em coletar endereços pela *Internet* foram criados e agora tem-se caixas postais entulhadas de mensagens indesejadas provenientes de *spammers*, *fraudsters*, *vírus worms* e *phishers*. Mais de 84% do volume dos e-mails são Spam (mensagens não solicitadas)[15]. Criou-se inicialmente filtros baseados no campo From:, usados para proteger as caixas postais dos usuários. Listas Negras, Listas Brancas e Listas Cinzas são listas usadas para identificar mensagens de remetentes indesejados Atualmente a maior parte dos Spam possuem o remetente forjado, o que dificulta o uso de listas.

Devido à simplicidade dos protocolos envolvidos nas transmissão de mensagens eletrônicas, os mecanismos de identificação de spam podem ser dificultados ou até mesmo burlados[20]. Apesar de exigir um conhecimento técnico mais avançado, é possível forjar endereços IP(*Internet Protocol*) burlando as listas negras de servidores e as listas negra de redes. Forjar o endereço eletrônico do remetente é uma tarefa bem mais simples, o que torna a lista de remetentes inútil. Em um cenário de tantas mensagens inválidas, a credibilidade do sistema perante o usuário fica comprometida. É necessário, portanto, a certeza da identidade do remetente[6].

Este artigo propõe um framework para autenticação do endereço do remetente com IBS(Assinatura Baseada em Identidade). Nesta proposta, o próprio endereço eletrônico do remetente constitui sua chave pública, simplificando sua verificação de autenticidade. A verificação da autenticidade do remetente será feita pelo MTA (Mail Transfer Agent).

Na seção II apresenta-se algumas técnicas usadas para prevenir Spam.

Na Seção III apresenta-se o conceito de autenticação do remetente e os frameworks que utilizam tal conceito.

Na Seção IV apresenta-se o sistema criptográfico baseado em identidade.

Na Seção V mostra-se a proposta deste artigo, o SAF (*Sender Authenticated Framework*).

Na Seção VI apresenta-se a conclusão deste artigo.

II. Filtros de Mensagens

Os *spammers* possuem duas linhas de abordagem para enviar mensagens não solicitadas. No início, os atacantes possuíam seus próprios servidores para enviar as mensagens e aplicavam diversas técnicas para esconder a identidade real de sua origem. Atualmente os *spammers*, através de hackers e alguns vírus, utilizam-se de vários servidores para esconder a origem do remetente.

Uma vez que não se pode confiar na origem das mensagens usa-se os filtro de conteúdo como por exemplo o *Spamassassin*[10] que baseiam-se na procura de padrões no cabeçalho e no corpo do e-mail que permitam classificar se a mensagem é um spam ou não. Tem-se então um ciclo de caça e fuga, onde os *spammers* trocam as características das mensagens para escapar dos filtros, enquanto os desenvolvedores dos filtros adaptam sua ferramente para reconhecer os novos padrões de spam.

Outro mecanismo para lutar contra o *spam* é lista branca. A lista branca consiste de uma relação de endereços que são aceitos pelo filtro. O uso de uma lista branca, dificulta

a atuação dos *spammers* pois para passar pelo filtro ele deve descobrir algum endereço que pertença a esta lista. O problema da lista branca é que alguns usuários desejam receber mensagens de outros usuários sem que haja um contato prévio.

A lista negra[16] utiliza uma abordagem oposta à da lista branca. A lista negra contém uma lista de remetentes ou servidores que deverão ser bloqueados pelo filtro. Um servidor que tenha seu endereço IP incluído em uma lista negra, terá problemas em se comunicar com outros servidores que adotem a técnica de consulta a lista negras, obrigando os administradores do servidor listado a tomar providências para impedir o *spammer* de usar o servidor. Esta técnica perde a eficiência, pois é fácil forjar ou trocar seu endereço.

Existem diversas outras propostas para prevenir spam, como graylisting[13], esquemas de desafio-resposta[1], autenticação de e-mail baseadas em domínios[3], proteção do *envelope sender address*[21], entre outras.

III. Autenticação do Remetente

Grande parte das mensagens abusivas possuem o endereço eletrônico do remetente falso o que dificulta a sua filtragem e remoção. Caso o endereço de e-mail da mensagem falsa pertença a um usuário legítimo, este terá sérios problemas, pois seu e-mail será adicionado nas listas negras e as mensagens do usuário legítimo não serão mais aceitas em diversos servidores.

É bastante inconveniente para o usuário comum trocar seu endereço de e-mail. Avisar seus contatos, trocar cartões de visita, *website*, impressos, estão entre os inconvenientes causados pela troca de endereço. Para o atacante é muito simples, pois basta escolher um outro endereço. Esta facilidade de trocar o endereço do remetente prejudica a credibilidade dos usuário no sistema de e-mail.

Deseja-se, portanto, um esquema onde o MTA verifica a autenticidade do remetente, descartando ou entregando conforme o resultado desta verificação.

A seguir alguns esquemas utilizados para autenticação do remetente.

A. SPF - Sender Policy Framework

Uma das medidas criadas para diminuir o problema de remetentes forjados foi o SPF(Sender Policy Framework)[21] onde o dono do domínio publica no seu registro de domínio quais servidores são autorizados a enviar e-mail deste domínio. Conforme a Figura 1 ,quando uma mensagem chega no servidor do destinatário, ela pode ser aceita ou recusada de acordo com o endereço IP de origem do servidor da mensagem.

O problema do SPF é sua capacidade de apenas relacionar domínio com endereço do servidor. Um oponente pode usar um servidor autorizado e forjar o endereço de algum usuário legítimo, ou ainda forjar o endereço IP origem.

B. Framework SenderID

O SenderID[14] é um framework proposto pela Microsoft, fortemente baseado no SPF. Basicamente a diferença entre os dois frameworks é o mecanismo para se obter a identidade do remetente no cabeçalho da mensagem.

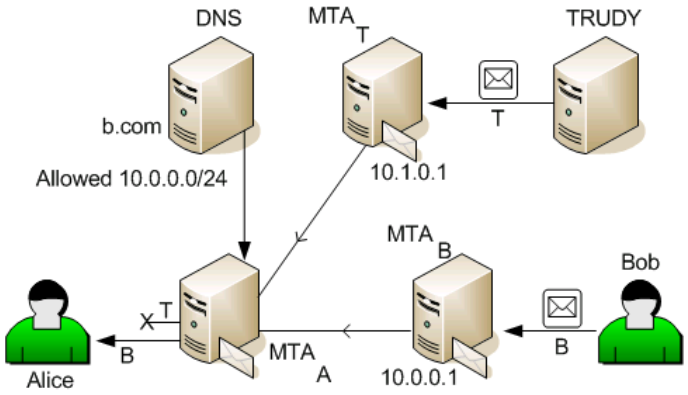

Figura 1. Esquema do SPF

O SPF verifica o MAIL FROM, enquanto o Sender ID verifica o PRA(*Purported Responsible Address*) que é uma avaliação de quatro campos do cabeçalho da mensagem: FROM, SENDER, RESENT-FROM e RESENT-SENDER. Esta avaliação resulta em um endereço *PRA* que é comparado ao endereço publicado pelo remetente no DNS.

C. DKIM - DomainKeys Identified Mail

Este método de autenticação de mensagem baseia-se na assinatura da mensagem pelo MTA do remetente e pela verificação da assinatura pelo MTA do destinatário.

O DKIM[3] adiciona ao cabeçalho da mensagem um assinatura digital que utiliza mecanismos como SHA-1[9] como *hash* criptográfico e o RSA[17] como sistema de chaves públicas. O DKIM utiliza o serviço DNS para divulgar as chaves públicas dos domínios.

O MTA do destinatário ao receber a mensagem assinada pelo MTA do remetente, consulta os registros DNS e obtém a chave pública do remetente. Uma vez de posse da chave pública, a assinatura da mensagem é verificada podendo ser aprovada ou rejeitada.

D. LES - Lightweight Email Signatures

Este esquema baseia-se no IBS proposto por Guillion e Quisquater[11] que utiliza como mecanismo de assinatura o RSA[17]. O LES trabalha basicamente com um esquema onde o usuário final tem papel importante tanto no envio como na recepção das mensagens e utiliza o próprio sistemas de mensagens como mecanismo de transmissão da chave privada.

IV. Criptografia baseada em identidade

O conceito do IBE(Identity Based Encription)/IBS, que foi proposto inicialmente por A. Shamir[18], o qual permite aos usuários verificar assinaturas digitais usando apenas informações públicas como a identidade do usuário. O sistema proposto por A. Shamir possui algumas restrições que dificultam seu uso prático. Um esquema proposto por C. Cocks[8] baseado em resíduos quadráticos não apresentou as restrições do esquema de A. Shamir. Porém, devido ao seu mecanismo de cifragem bit a bit e ao alto grau de expansão do texto cifrado, seu uso restringe-se a mensagens pequenas.

Sistemas mais eficientes foram propostos por Boneh-Franklin[7] e Baldwin[5] baseados em pareamentos bilineares em curvas elípticas. Estes sistemas superaram as limitações dos esquemas de A. Shamir e C. Cocks.

No esquema do IBE[4], conforme apresentado na Figura 2, Alice é o remetente da mensagem e Bob o destinatário. A remetente Alice pode usar uma identificação do destinatário Bob, por exemplo o e-mail do Bob, como chave pública e com ela cifrar a mensagem a ser enviada. Por sua vez, para poder decifrar a mensagem enviada por Alice, Bob deve obter de uma entidade confiável chamada PKG(*Public Key Generator*) a chave privada associada a sua chave pública.

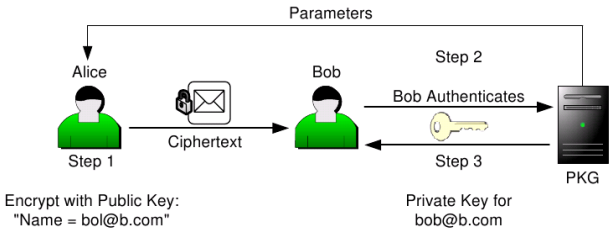

Figura 2. Esquema de criptografia baseada em identidade

Pode-se resumir o esquema utilizado pelo IBE com as seguintes etapas:

- Setup - O PKG cria inicialmente sua senha secreta *Master* e sua chave pública correspondente. Os parâmetros serão distribuidos livremente para todos os interessados e serão mantidos por um longo período;
- Private Key Extraction - O destinatário autentica-se perante o PKG e recebe sua chave privada associada a bob@b.com;
- Encryption - Usando a identidade de Bob e o *parâmetro* fornecido pelo PKG, Alice cifra a mensagem obtendo um texto cifrado;
- Decryption - Ao receber o texto cifrado de Alice, Bob decifra a mensagem com a sua chave privada.

Ao reverter este esquema temos um IBS, conforme observado na Figura 3. Alice iniciamente obtém uma chave privada associada ao seu endereço público alice@a.com. Alice assina a mensagem usando sua chave privada. Bob usa a identidade de Alice para verificar a assinatura.

O esquema IBS pode ser descrito em quatro etapas:

- Setup - O PKG cria inicialmente sua senha secreta *Master* e sua chave pública correspondente;
- Private Key Extraction - O destinatário autentica-se perante o PKG e recebe sua chave privada associada a alice@a.com;
- Signature Generation - Usando sua chave privada, Alice assina sua mensagem *M*. A assinatura e a mensagem são enviadas a Bob;
- Signature Verification - Ao receber a mensagem assinada de Alice, Bob utiliza a chave pública da Alice juntamente com o parâmetro do PKG para verificar a autenticidade da assinatura.

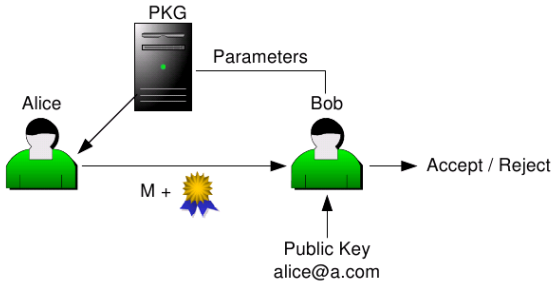

Figura 3. Esquema de assinatura digital baseada em identidade

V. SAF - *Sender Autenticated Framework*

Um esquema onde o servidor verifica a identidade do remetente é preferível sobre um esquema onde a verificação é feita pelo cliente, pois com a verificação no cliente[2], as mensagens indesejadas ficam armazenadas no servidor e depois devem ser transferidas ao cliente para então serem descartadas. Fica evidente o desperdício de recursos de armazenamento e de recursos de rede.

Ao enviar uma mensagem, o programa cliente do remetente deve incluir uma assinatura no cabeçalho da mensagem que deve ser verificada pelo MTA. Para o MTA verificar a autenticidade da assinatura ele deve encontrar a chave pública do remetente. O remetente pode enviar sua chave pública junto com sua certificação, mas além de aumentar o *overhead* da mensagem, o servidor ainda necessita consultar uma lista de certificados revogados, o que atrasaria consideravelmente o serviço.

Pode-se simplificar o processo usando uma chave pública que seja determinada com facilidade pelo MTA, que sua autenticação seja simples e que tenha um mecanismo de revogação do certificado que evite consultas a listas. O IBS atende estes requisitos.

Em um cenário onde Alice, cujo e-mail é alice@a.com, deseja enviar uma mensagem para Bob, cujo e-mail é bob@b.com, um oponente que possui acesso ao MTA do domínio de Alice, por exemplo trudy@a.com, pode forjar o endereço de Alice e ainda obter sucesso no ataque caso verifique-se apenas a autenticidade do domínio do remetente. Para obter-se maior proteção contra remetentes forjados, deve-se usar assinatura digital com certificação do remetente e não do servidor.

Os mecanismos tradicionais de assinatura digital com certificação exigem do servidor do destinatário algumas tarefas:

- Encontrar a chave pública do remetente em um repositório de chaves públicas;
- Consultar em uma lista de certificados revogados, a validade da chave encontrada;
- Verificar a autenticidade da mensagem.

Estas tarefas introduzem um atraso indesejado no processamento das mensagens pelo servidor.

Com o IBS pode-se prover certificação sem a necessidade do servidor encontrar a chave pública do remetente. A vantagem do uso do IBS é que a chave está implícita na mensagem, sendo fácil ao MTA obter a chave pública, sem a necessidade de dados adicionais na mensagem ou consulta a um repositório de chaves.

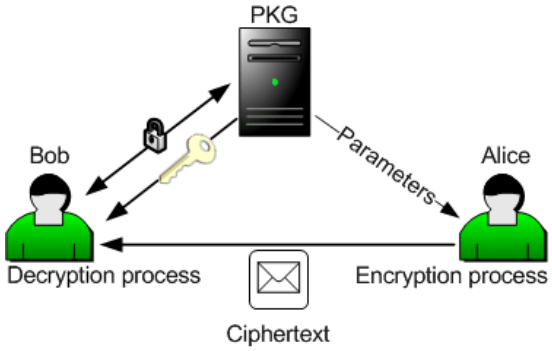

Figura 4. Esquema SAF

O SAF tem como objetivos:

- Autenticar o remetente ao nível do usuário;
- Usar um mecanismo de assinatura de certificação simples;
- A verificação da assinatura seja feita pelo MTA.

O SAF esquematizado na Figura V pode ser dividido nas seguintes fases:

Na fase 1, o dono do domínio escolhe uma senha secreta sk_{PKG} e calcula um parâmetro público pk_{PKG}. A senha secreta será usada para gerar as chaves privadas associadas a cada e-mail, enquanto o parâmetro público será divulgado a todos os interessados;

Na fase 2, exemplificado na Figura 5 o remetente requisita uma chave secreta associada ao seu e-mail e assina a mensagem a ser enviada ao destinatário. O servidor gera a chave privada e transmite através de um meio seguro(SSL,TSL) para o cliente durante a autenticação do cliente de e-mail perante o servidor. Com a chave privada assina-se a mensagem (M). A assinatura obtida é então inserida no cabeçalho da mensagem;

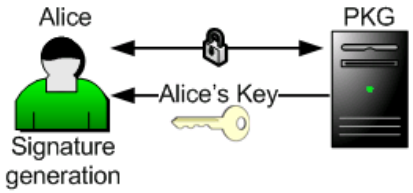

Figura 5. Fase 2

Na fase 3, o MTA do destinatário obtém o parâmetro geral do domínio através de consulta ao PKG e armazena para usar futuramente com outras mensagens provenientes do mesmo domínio;

Na fase 4, como mostrado na Figura 6, o servidor de posse do parâmetro geral, juntamente com a chave pública do usuário remetente, verifica a identidade do remetente, descartando/devolvendo a mensagem em caso de falha na certificação. Em caso de sucesso na verificação da identidade, a mensagem é entregue à caixa postal do destinatário.

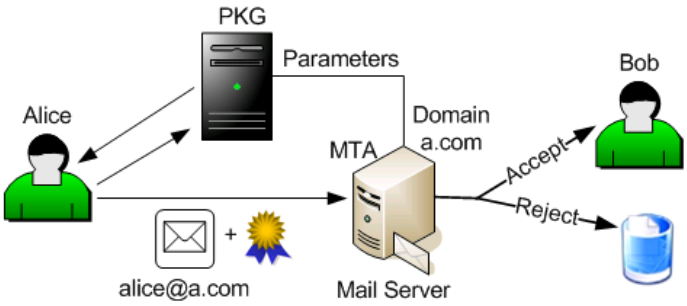

Figura 6. Fase 4

A chave pública neste caso não necessita de uma CA(Certificate Authority) para garantir sua autenticidade, pois a certificação é implícita no conhecimento de dados públicos do remetente. Pelo fato da chave pública estar no próprio cabeçalho da mensagem, facilita ao MTA sua obtenção.

A. *Chave Pública*

Para construir a chave pública, propõ-se o uso dos seguintes campos:

- email;
- data de expiração.

B. *Assinatura*

Para compor a mensagem(M) a ser assinada usa-se:

- Campo **FROM:**
- Campo **TO:**
- Campo **SUBJECT:**
- **TIMESTAMP**
- **HASH(BODY)**

Do campo **FROM** retira-se parte da chave pública. O campo **TO** faz-se necessário para evitar que a assinatura seja usada para enviar a mensagem para outros destinatários. O **TIMESTAMP** garante que a assinatura não será usada posteriormente pelo oponente e o resumo do corpo da mensagem evitará o ataque por repetição.

VI. Conclusão

Neste artigo, apresentou-se uma proposta para um framework de autenticação de remetente. A vantagem do SAF em relação a esquemas onde autentica-se o servidor é que na autenticação do servidor ainda é possível forjar o remetente. O uso do IBS permite ao SAF simplificar o processo de obtenção da chave pública/certificação que viabiliza a autenticação ao nível de usuário.

Tanto o SPF como o SenderID, que não trabalham com sistemas criptográficos, apresentam um problema quando a mensagem é redirecionada por algum MTA, como por exemplo no caso de uma lista de discussão, ocasionando numa falha de autenticação. Assim o SPF e o SenderID autenticam somente o

último MTA pelo qual a mensagem percorreu. Este problema não afeta os sistemas DKIM e SAF, porém no caso de listas de discussão que alteram o conteúdo da mensagem original, como por exemplo, ao incluir uma assinatura ou uma mensagem do patrocinador da lista na mensagem original, o DKIM apresentará uma falha de autenticação.

	SPF	SenderID	DKIM	LES	SAF
Método de Validação	Endereço do remetente, IP, DNS	PRA, IP, DNS	Assinatura digital do servidor, DNS	Assinatura digital do remetente, DNS	Assinatura digital do remetente, DNS
Vantagens	Fácil implementação, Verificação feita antes da chegada dos dados, Proteção contra o problema de *fishing*	Fácil implementação, verificação feita antes da chegada dos dados, proteção contra o problema de *fishing*	Proteção contra o problema de *fishing*, não é afetado por múltiplos saltos SMTP(*Simple Mail Transfer Protocol*.	Proteção contra o problema de *fishing*, não é afetado por múltiplos saltos SMTP	Proteção contra o problema de *fishing*, não é afetado por múltiplos saltos SMTP, verificação feita antes da chegada dos dados.
Desvantagens	Usuários do MTA podem forjar identidades de outros usuários, validação apenas do último salto	Usuários do MTA podem forjar identidades de outros usuários, validação apenas do último salto	Problemas com reenvio de mensagens, difícil de implementar, problemas de validação de listas de discussão	Dificuldade de implementação, custo, verificação fim-a-fim, participação do usuário final.	Dificuldade de implementação, custo, participação do usuário final.

Tabela I
COMPARATIVO ENTRE SPF, SENDERID, DKIM, LES E SAF

Um resumo comparativo entre os frameworks SPF, SenderID, DKIM, LES e SAF é mostrado na tabela I.

Uma vez implantado o SAF, apesar do maior custo causado pela introdução do cálculo da assinatura e do resumo[12], os ganhos na proteção da identidade dos remetentes, justificarão o esforço de implantação.

Futuramente, pretende-se a implementação do SAF em um servidor de mensages e o estudo do desempenho do esquema. Também pretende-se estudar o impacto de ataques DOS(*Deny of Service)* e DDOS(*Distributed Deny of Service)*[19] ao SAF.

REFERÊNCIAS

[1] Spamarrest. http://www.spamarrest.com/.

[2] Ben Adida, David Chau, Susan Hohenberger, and Ronald L. Rivest. Lightweight email signatures (extended abstract). In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2006.

[3] E. Allman et al. Domainkeys identified mail (dkim) signatures. http://www.rfc-editor.org/rfc/rfc4871.txt, 2007.

[4] Joonsang Baek, Jan Newmarch, Reihaneh Safavi-Naini, and Willy Susilo. A survey of identity-based cryptography. *AUUG*, 2004.

[5] Matthew Baldwin. Identity based encryption for the tate pairing to secure email comunications. Master's thesis, University of Bristol - Department of Computer Science, 2002.

[6] Steven M. Bellovin. Spamming, phishing, authentication, and privacy. *Commun. ACM*, 47(12):144, 2004.

[7] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *Lecture Notes in Computer Science*, 2139:213–??, 2001.

[8] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, London, UK, 2001. Springer-Verlag.

[9] D. Eastlake and P. Jones. Us secure hash algorithm 1 (sha1). http://www.rfc-editor.org/rfc/rfc3329.txt, 2001.

[10] The Apache Software Foundation. The apache spamassassin project. http://spamassassin.apache.org/.

[11] L. C. Guillou and J. J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 216–231, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[12] Darrel Hankerson, Alfred Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer, 1st edition, 2004.

[13] E. Harris. The next step in the spam control war:graylisting. http://projects.puremagic.com/greylisting/whitepaper.html, 2003.

[14] J. Lyon and M. Wong. Sender id: Authenticating e-mail. http://www.rfc-editor.org/rfc/rfc4406.txt, 2006.

[15] MessageLabs. Annual email security report. http://www.messagelabs.com, December 2007. Message Labs.

[16] The Spamhaus Project. The spamhaus project. http://www.spamhaus.org/, 2008.

[17] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Technical Report 2, 1978.

[18] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, http://link.springer.de/link/service/series/0558/bibs/0196/01960047.htm, 1984.

[19] Christian Veigner and Chunming Rong. On email spamming under the shadow of large scale use of identity-based encryption. In *ATC*, pages 521–530, 2006.

[20] Brett Watson. Beyond identity: Addressing problems that persist in an electronic mail system with reliable sender identification. In *CEAS*, 2004.

[21] M. Wong and W. Schlitt. Sender policy framework (spf) for authorizing use of domains in e-mail, version 1. http://www.rfc-editor.org/rfc/rfc4408.txt, 2006.